Internet of Things

Introduction to the Internet of Things

History of IoT

• The first telemetry system was rolled out in Chicago way back in 1912. It is said to have used telephone lines to monitor data from power plants.

• Telemetry expanded to weather monitoring in the 1930s, when a device known as a radiosonde became widely used to monitor weather conditions from balloons.

• In 1957 the Soviet Union launched Sputnik, and with it the Space Race. This has been the entry of aerospace telemetry that created the basis of our global satellite communications today.

• Broad adoption of M2M technology began in the 1980s with wired connections for SCADA (supervisory control and data acquisition) on the factory floor and in home and business security systems.

• In the 1990s, M2M began moving toward wireless technologies. ADEMCO built their own private radio network to address intrusion and smoke detection because budding cellular connectivity was too expensive.

• In 1995, Siemens introduced the first cellular module built for M2M

Why IoT now?

• Ubiquitous Connectivity

• Widespread Adoption of IP

• Computing Economics

• Miniaturization

• Advances in Data Analytics

• Rise of Cloud Computing

IoT Definition

The **Internet of things** (**IoT**) is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

There are 7 crucial IoT characteristics:

1. **Connectivity.** This doesn't need too much further explanation. With everything going on in IoT devices and hardware, with sensors and other electronics and connected hardware and control systems there needs to be a connection between various levels.
2. **Things**. Anything that can be tagged or connected as such as it's designed to be connected. From sensors and household appliances to tagged livestock. Devices can contain sensors or sensing materials can be attached to devices and items.
3. **Data**. Data is the glue of the Internet of Things, the first step towards action and intelligence.
4. **Communication**. Devices get connected so they can communicate data and this data can be analyzed. Communication can occur over short distances or over a long range to very long range. Examples: Wi-Fi, LPWA network technologies such as LoRa or NB-IoT.
5. **Intelligence**. The aspect of intelligence as in the sensing capabilities in IoT devices and the intelligence gathered from big data analytics (also artificial intelligence).
6. **Action**. The consequence of intelligence. This can be manual action, action based upon debates regarding phenomena (for instance in smart factory decisions) and automation, often the most important piece.
7. **Ecosystem**. The place of the Internet of Things from a perspective of other technologies, communities, goals and

the picture in which the Internet of Things fits. The Internet of Everything dimension, the platform dimension and the need for solid partnerships.

Physical Design of IoT

- Things in IoT

- IoT Protocols

Things in IoT

Refers to IoT devices which have unique identities that can perform sensing, actuating and monitoring capabilities.

•IoT devices can exchange data with other connected devices or collect data from other devices and process the data either locally or send the data to centralized servers or cloud – based application back-ends for processing the data.

Generic Block Diagram of an IoT Device

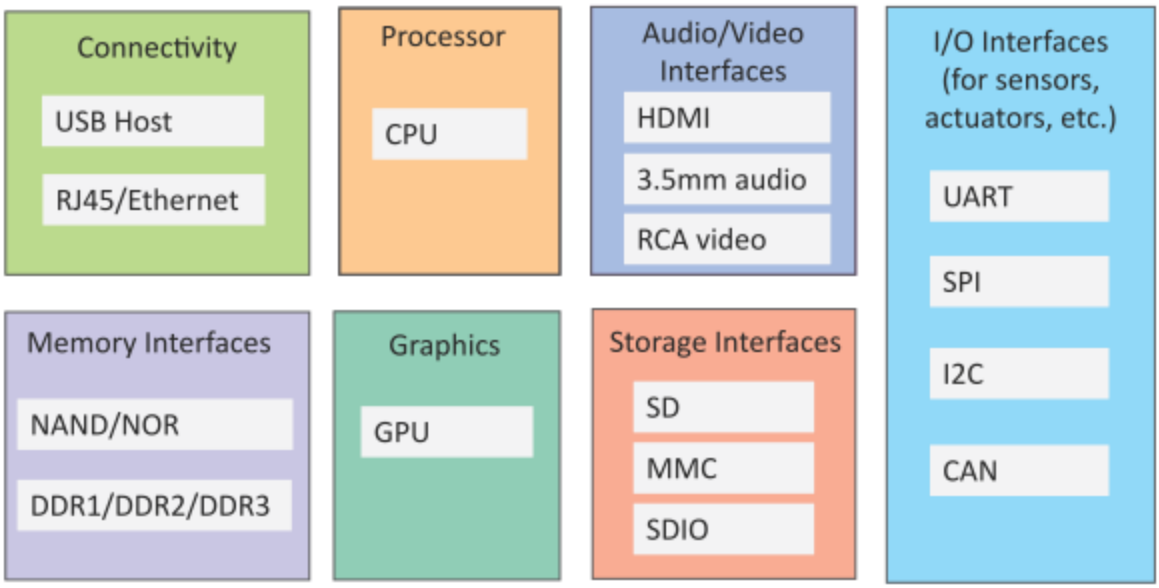
•An IoT device may consist of several interfaces for connections to other devices, both wired and wireless.

•I/O interfaces for sensors

•Interfaces for internet connectivity

•Memory and storage interfaces

•Audio/video interfaces



IoT Protocols

•Link Layer

•802.3 – Ethernet

•802.11 – WiFi

•802.16 – WiMax

•802.15.4 – LR-WPAN

•2G/3G/4G

•Network/Internet Layer

•IPv4

•IPv6

•6LoWPAN

•Transport Layer

•TCP

•UDP

•Application Layer

•HTTP

•CoAP

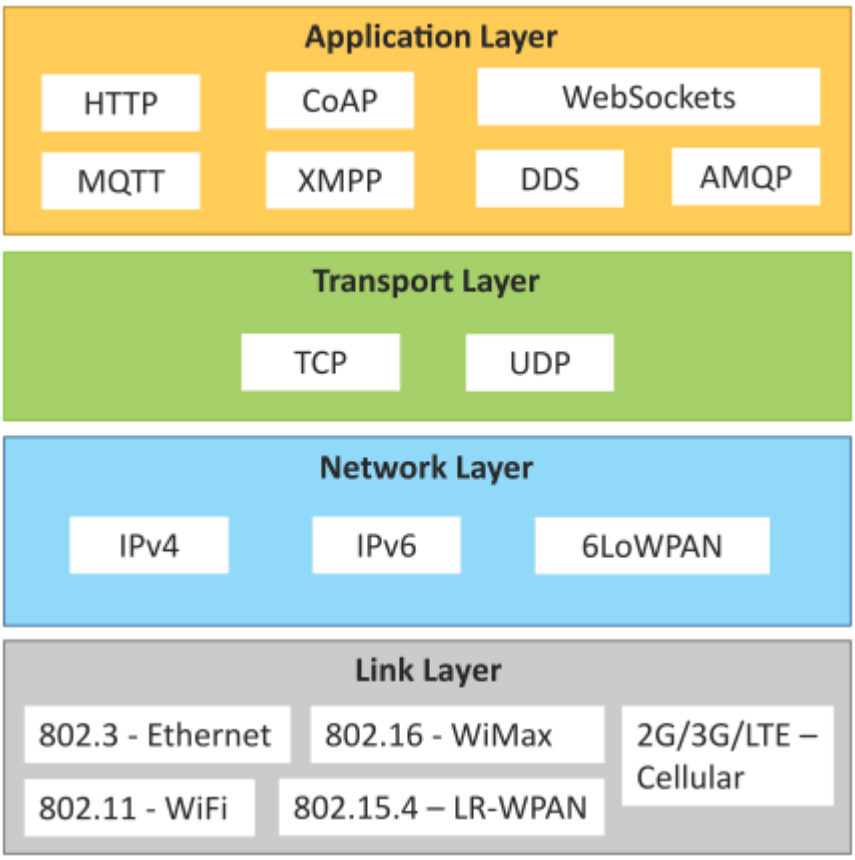
•WebSocket

•MQTT

•XMPP

•DDS

•AMQP



IoT Protocols…Link Layer…Ethernet

Sr.No	Standard	Shared medium
1	802.3	Coaxial Cable…10BASE5
2	802.3.i	Copper Twisted pair …..10BASE-T
3	802.3.j	Fiber

4	802.3.ae	Optic……10BASE-F Fiber…..10Gbits/s

Data Rates are provided from 10Gbit/s to 40Gb/s and higher

IoT Protocols…Link Layer…WiFi

Sr.No	**Standard**	**Operates in**
1	802.11a	5 GHz band
2	802.11b and 802.11g	2.4GHz band
3	802.11.n	2.4/5 GHz bands
4	802.11.ac	5GHz band
5	802.11.ad	60Hz band

- Collection of Wireless LAN
- Data Rates from 1Mb/s to 6.75 Gb/s

IoT Protocols…Link Layer…WiMax

Sr.No	Standard	Data Rate
1	802.16m	100Mb/s for mobile stations 1Gb/s for fixed stations

•Collection of Wireless Broadband standards

•Data Rates from 1.5Mb/s to 1 Gb/s

IoT Protocols…Link Layer…LR-WPAN

•Collection of standards for low-rate wireless personal area networks

•Basis for high level communication protocols such as Zigbee

•Data Rates from 40Kb/s to 250Kb/s

•Provide low-cost and low-speed communication for power constrained devices

IoT Protocols…Link Layer…2G/3G/4G –Mobile Communication

Sr.No	Standard	Operates in
1	2G	GSM-CDMA
2	3G	UMTS and

CDMA 2000

3 4G LTE

•Data Rates from 9.6Kb/s (for 2G) to up to 100Mb/s (for 4G)

IoT Protocols…Network/Internet Layer

•Responsible for sending of IP datagrams from source to destination network

•Performs the host addressing and packet routing

•Host identification is done using hierarchical IP addressing schemes such as IPV4 or IPV6

•IPV4

•Used to identify the devices on a network using hierarchical addressing scheme

•Uses 32-bit address scheme

•IPV6

•Uses 128-bit address scheme

•6LoWPAN (IPV6 over Low power Wireless Personal Area Network)

•Used for devices with limited processing capacity

•Operates in 2.4 Ghz

•Data Rates of 250Kb/s

•Provide end-to-end message transfer capability independent of the underlying network

•It provides functions such as error control, segmentation, flow-control and congestion control

IoT Protocols…TCP

•Transmission Control Protocol

•Connection Oriented

•Ensures Reliable transmission

•Provides Error Detection Capability to ensure no duplicacy of packets and retransmit lost packets

•Flow Control capability to ensure the sending data rate is not too high for the receiver process

•Congestion control capability helps in avoiding congestion which leads to degradation of n/w performance

IoT Protocols…UDP

•User Datagram Protocol

•Connectionless

•Does not ensures Reliable transmission

•Does not do connection before transmitting

•Does not provide proper ordering of messages

•Transaction oriented and stateless

Logical Design of IoT:

In this article we discuss Logical design of Internet of things. Logical design of IoT system refers to an abstract representation of the entities & processes without going into the low-level specifies of the implementation. For understanding Logical Design of IoT, we describes given below terms.

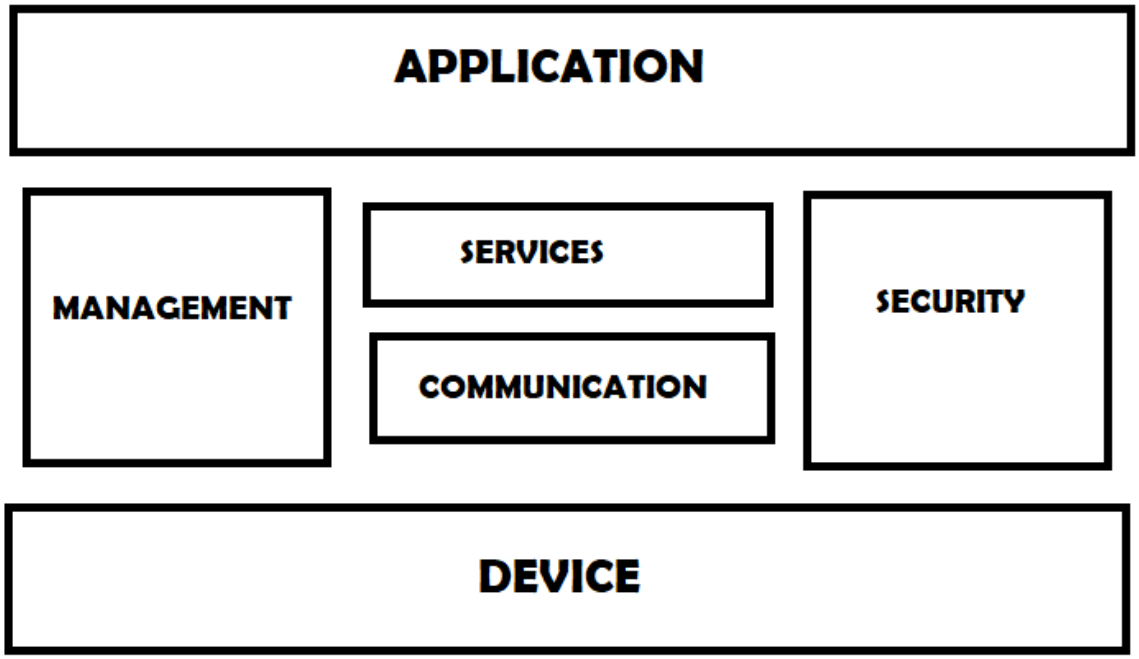
- IoT Functional Blocks
- IoT Communication Models
- IoT Communication APIs

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 blocks are:

Device: An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.
Communication: Handles the communication for the IoT system.
Services: services for device monitoring, device control service, data publishing services and services for device discovery.
Management: this blocks provides various functions to govern the IoT system.
Security: this block secures the IoT system and by providing functions such as authentication , authorization, message and content integrity, and data security.
Application: This is an interface that the users can use to control and monitor various aspects of the IoT system. Application also allow users to view the system status and view or analyze the processed data.



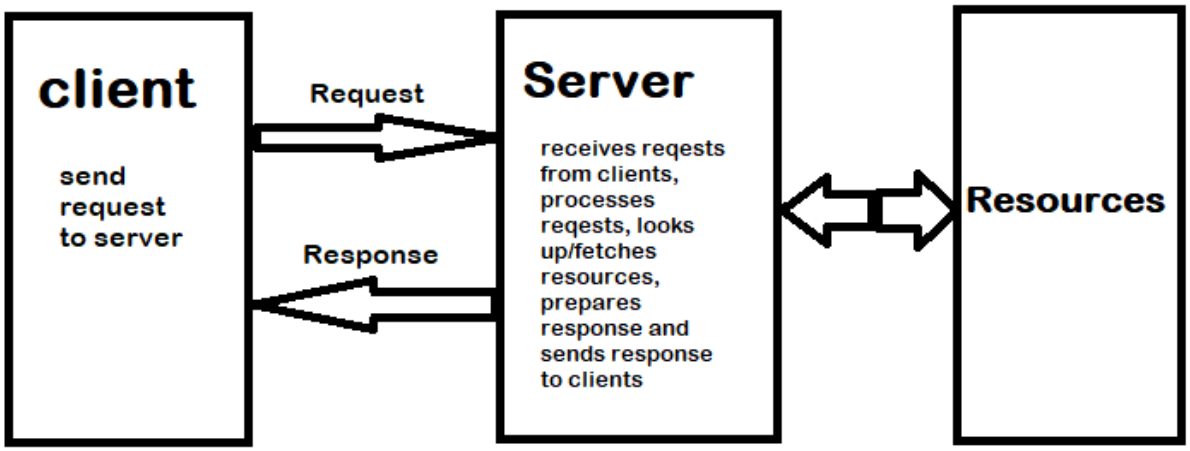
IoT Communication Models

Request-Response Model:

Request-response model is communication model in which the client sends requests to the server and the server responds to the requests. When the server receives a request, it decides how to respond, fetches the data, retrieves resource representation, prepares the response, and then sends the response to the client. Request-response is a stateless communication model and each request-response pair is independent of others.

HTTP works as a request-response protocol between a client and server. A web browser may be the client, and an application on a computer that hosts a web site may be the server.

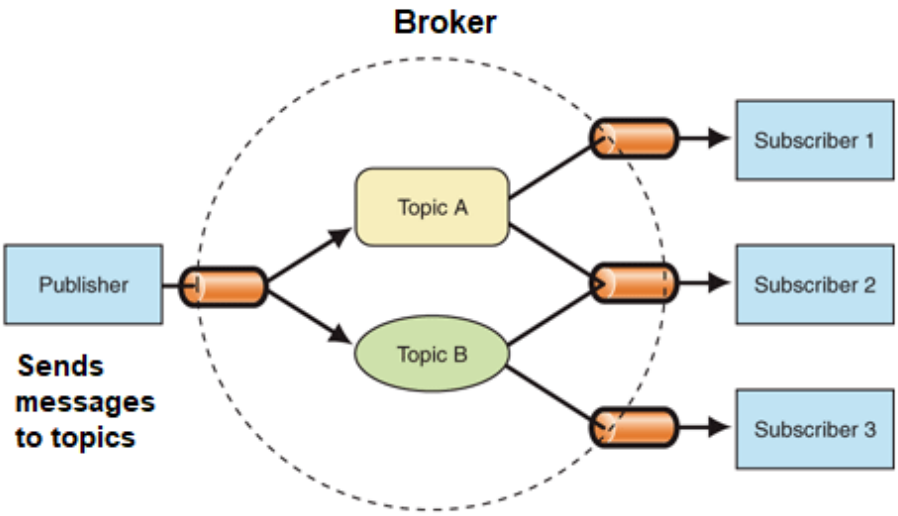
Example: A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.



Request-Response Communication Model

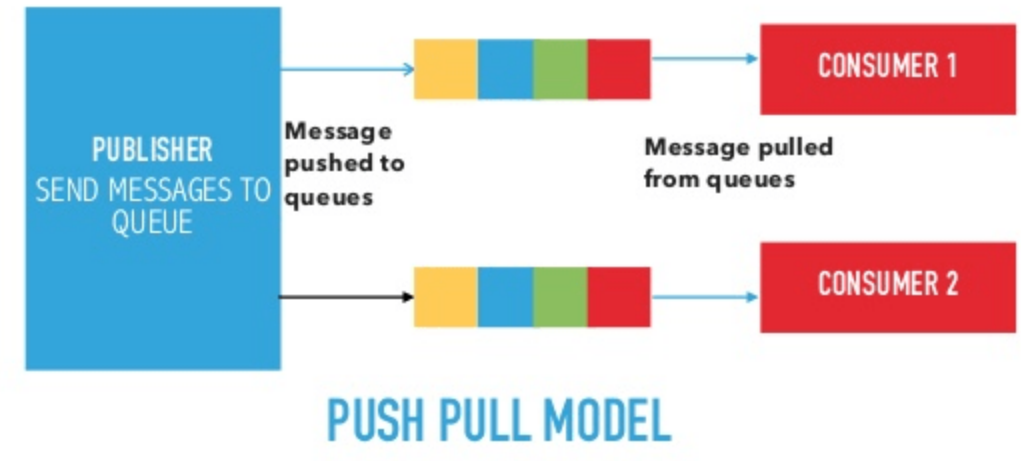
Publish-Subscribe Model

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. When the broker receive data for a topic from the publisher, it sends the data to all the subscribed consumers.



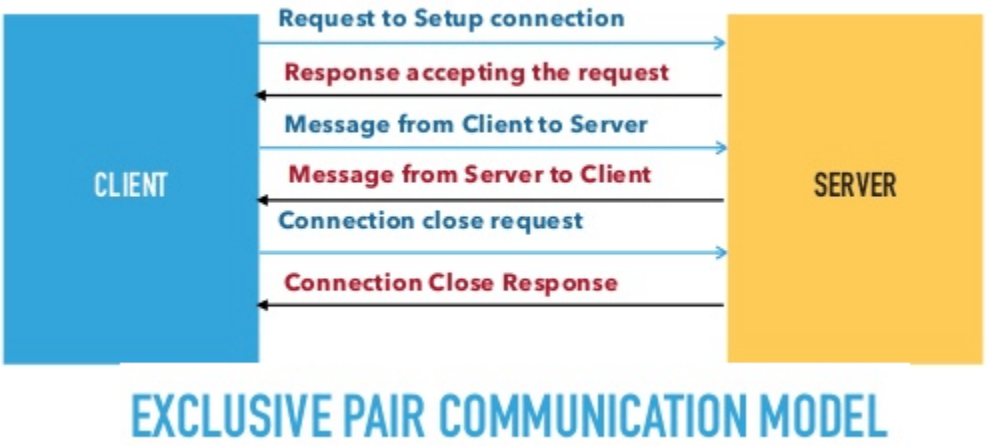
Push-Pull Model

Push-Pull is a communication model in which the data producers push the data to queues and the consumers Pull the data from the Queues. 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 consumer pull data.



Exclusive Pair Model

Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and server. 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 stateful communication model and the server is aware of all the open connections.



EXCLUSIVE PAIR COMMUNICATION MODEL

IoT Communication APIs

Generally we used Two APIs For IoT Communication. These IoT Communication APIs are:

- REST-based Communication APIs
- WebSocket-based Communication APIs

REST-based Communication APIs

Representational state transfer (REST) is a set of architectural principles by which you can design Web services the Web APIs that focus on systems's resources and how resource states are addressed and transferred. REST APIs that follow the request response communication model, the rest architectural constraint apply to the components, connector and data elements, within a distributed hypermedia system. The rest architectural constraint are as follows:

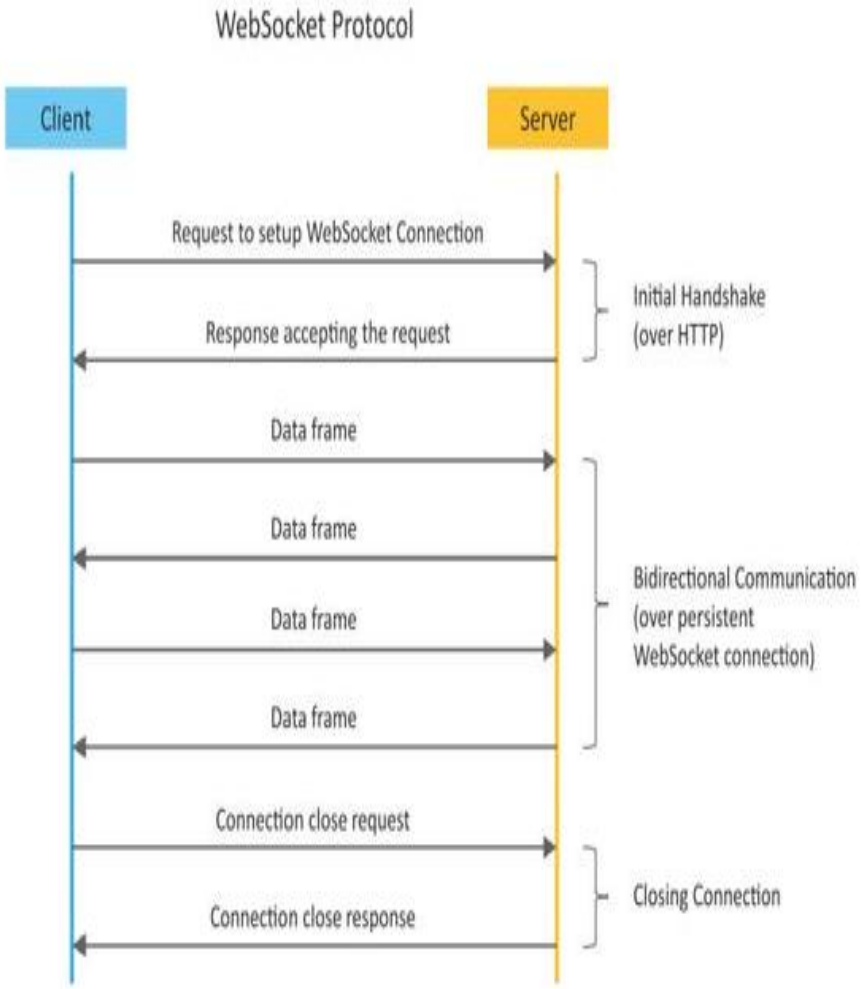
Client-server – The principle behind the client-server constraint is the separation of concerns. for example clients should not be concerned with the storage of data which

is concern of the serve. Similarly the server should not be concerned about the user interface, which is concern of the clien. Separation allows client and server to be independently developed and updated.
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.
Cache-able – Cache constraints requires that the data within a response to a request be implicitly or explicitly leveled as cache-able or non cache-able. If a response is cache-able, then a client cache is given the right to reuse that repsonse data for later, equivalent requests. caching can partially or completely eliminate some instructions and improve efficiency and scalability.
Layered system – layered system constraints, constrains the behavior of components such that each component cannot see beyond the immediate layer with they are interacting. For example, the client cannot tell whether it is connected directly to the end server or two an intermediaryalong 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.
Uniform interface – uniform interface constraints requires that the method of communication between client and server must be uniform. Resources are identified in the requests (by URIsin web based systems) and are themselves is separate from the representations of the resources data returned to the client. When a client holds a representation of resources it has all the information required to update or delete the resource you (provided the client has required permissions). Each message includes enough information to describe how to process the message.
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. REST is most popular IoT Communication APIs.

WebSocket based communication API

Websocket APIs allow bi-directional, full duplex communication between clients and servers. Websocket APIs follow the exclusive pair communication model. Unlike request-response model such as REST, the WebSocket APIs allow full duplex communication and do not require new coonection to be setup for each message to be sent. Websocket communication begins with a connection setup request sent by the client to the server. The request (called websocket handshake) is sent over HTTP and the server interprets it is an upgrade request. If the server supports websocket protocol, the server responds to the websocket handshake response. After the connection setup client and server can send data/mesages to each other in full duplex

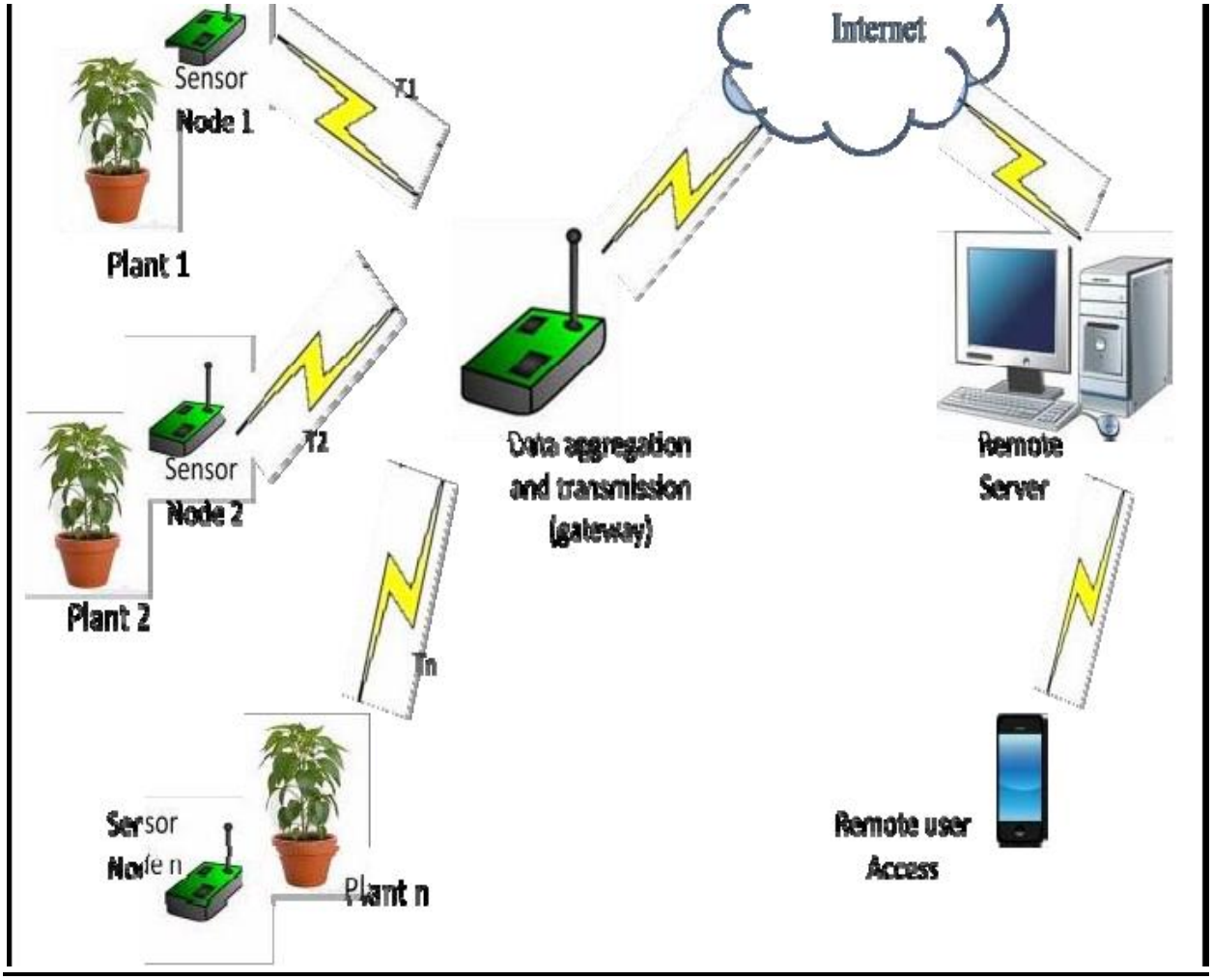
mode. Websocket API reduce the network traffic and letency as there is no overhead for connection setup and termination requests for each message. Websocket suitable for IoT applications that have low latency or high throughput requirements. So Web socket is most suitable IoT Communication APIs for IoT System.



IoT Enabling Technologies

- Wireless Sensor Network
- Cloud Computing
- Big Data Analytics
- Embedded Systems

WSN



Wireless Sensor Network Components:

Distributed Devices with sensors

end-nodes

Coordinators

Routers

•**Distributed Devices with sensors** used to monitor the environmental and physical conditions

•Consists of several **end-nodes acting as routers or coordinators too**

•**Coordinators collects data** from all nodes / **acts as gateway** that connects WSN to internet

•**Routers route the data packets** from end nodes to coordinators.

Example of WSNs in IoT & Protocols used

Example

•Weather monitoring system

•Indoor Air quality monitoring system

•Soil moisture monitoring system

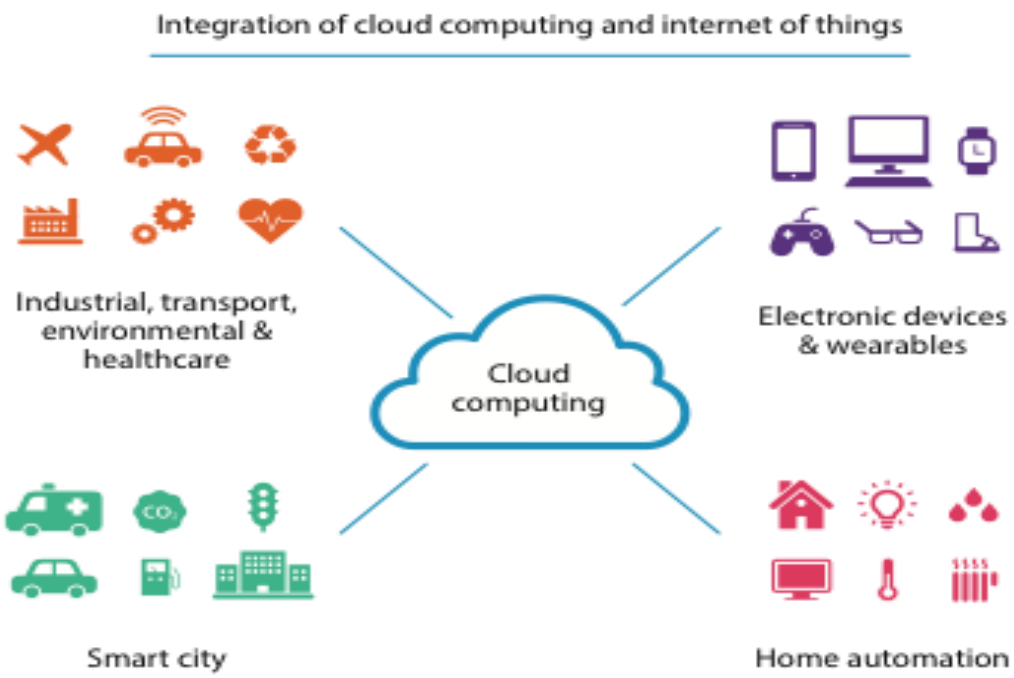
•Survelliance systems

•Health monitoring systems

Protocols

•Zigbee

Cloud Computing



•**Deliver applications and services over internet**

•Provides computing, networking and storage resources on demand

•Cloud computing performs services such as Iaas, Paas and Saas

•Iaas : Rent Infrastructure

•Paas : supply an on-demand environment for developing, testing, delivering and managing software applications.

•Saas : method for delivering software applications over the Internet, on demand and typically on a subscription basis.

Big Data Analytics

•Collection of data whose volume, velocity or variety is too large and difficult to store, manage, process and analyze the data using traditional databases.

•It involves data cleansing, processing and visualization

•Lots of data is being collected and warehoused

•Web data, e-commerce

•purchases at department/ grocery stores

•Bank/Credit Card transactions

•Social Network

Big Data Analytics

Variety Includes different types of data

•Structured

•Unstructured

•SemiStructured

•All of above

Velocity Refers to speed at which data is processed

•Batch

•Real-time

•STreams

Volume refers to the amount of data

•Terabyte

•Records

•Transactions

•Files

•Tables

IoT Levels and Deployment Templates

An IoT system comprises the following components:

•**Device**: An IoT device allows identification, remote sensing, actuating and remote monitoring capabilities.

•**Resource**: Resources are software components on the IoT device for accessing, processing and storing sensor information, or for controlling actuators connected to the device. Resources also include the software components that enable network access for the device.

•**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.

•**Database**: Database can be either local or in the cloud and stores the data generated by the IoT device.

•**Web Service**: Web services serve as a link between the IoT device, application, database and analysis components. Web service can be implemented using HTTP and REST principles (REST service) or using the WebSocket protocol (WebSocket service).

•**Analysis Component**: This is responsible for analyzing the IoT data and generating results in a form that is easy for the user to understand.

•**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 the processed data.

IoT Level-1

•A level-1 IoT system **has a single node/device** that performs sensing and/or actuation, stores data, performs analysis and hosts the application.

•Level-1 IoT systems are suitable for **modelling low-cost and low-complexity solutions** where the data involved is not big and the **analysis requirements are not computationally intensive.**

IoT Level-2

•A level-2 IoT system has a **single node that performs sensing and/or actuation and local analysis.**

•**Data is stored in the cloud** and the application is usually cloud-based.

•Level-2 IoT systems are **suitable for solutions where the data involved is big;** however, the primary **analysis requirement is not computationally intensive** and can be done locally.

IoT Level-3

•A level-3 IoT system has a **single node. Data is stored and analyzed in the cloud** and the application is cloud-based.

•Level-3 IoT systems are suitable for solutions **where the data involved is big and the analysis requirements are computationally intensive.**

IoT Level-4

•A level-4 IoT system has multiple nodes that perform local analysis. Data is stored in the cloud and the application is cloud-based.

•Level-4 contains local and cloud-based observer nodes which can subscribe to and receive information collected in the cloud from IoT devices.

•Level-4 IoT systems are suitable for solutions where multiple nodes are required, the data involved is big and the analysis requirements are computationally intensive.

IoT Level-5

•A level-5 IoT system has multiple end nodes and one coordinator node.

•The end nodes perform sensing and/or actuation.

•The coordinator node collects data from the end nodes and sends it to the cloud.

•Data is stored and analyzed in the cloud and the application is cloud-based.

•Level-5 IoT systems are suitable for solutions based on wireless sensor networks, in which the data involved is big and the analysis requirements are computationally intensive.

IoT Level-6

•A level-6 IoT system has multiple independent end nodes that perform sensing and/or actuation and send data to the cloud.

•Data is stored in the cloud and the application is cloud-based.

•The analytics component analyzes the data and stores the results in the cloud database.

•The results are visualized with the cloud-based application.

•The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes.

DOMAIN SPECIFIC IoTs

1) Home Automation:

a) **Smart Lighting:** helps in saving energy by adapting the lighting to the ambient conditions and switching on/off or diming the light when needed.

b) **Smart Appliances:** make the management easier and also provide status information to the users remotely.

c) **Intrusion Detection:** use security cameras and sensors(PIR sensors and door sensors) to detect intrusion and raise alerts. Alerts can be in the form of SMS or email sent to theuser.

d) **Smoke/Gas Detectors:** Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of fire. 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 CO, LPGetc.,

2) Cities:

a) **Smart Parking:** make the search for parking space easier and convenient for drivers. Smart parking are powered by IoT systems that detect the no. of empty parking slots and send information over internet to smart application back ends.

b) **Smart Lighting:** for roads, parks and buildings can help in saving energy.

c) **Smart Roads:** Equipped with sensors can provide information on driving condition, travel time estimating and alert in case of poor driving conditions, traffic condition and accidents.

d) **Structural Health Monitoring:** uses a network of sensors to monitor the vibration levels in the structures such as bridges and buildings.

e) **Surveillance:** The video feeds from surveillance cameras can be aggregated in cloud based scalable storage solution.

f) **Emergency Response:** IoT systems for fire detection, gas and water leakage detection can help in generating alerts and minimizing their effects on the critical infrastructures.

3) Environment:

a) **Weather Monitoring:** Systems collect data from a no. of sensors attached and send the data to cloud based applications and storage back ends. The data collected in cloud can then be analyzed and visualized by cloud based applications.

b) **Air Pollution Monitoring:** System can monitor emission of harmful gases(CO2, CO, NO, NO2 etc.,) by factories and automobiles using gaseous and meteorological sensors. The collected data can be analyzed to make informed decisions on pollutions control approaches.

c) **Noise Pollution Monitoring:** Due to growing urban development, noise levels in cities have increased and even become alarmingly high in some cities. IoT based noise pollution monitoring systems use a no. of noise monitoring systems that are deployed at different places in a city. The data on noise levels from the station is collected on servers or in the cloud. The collected data is then aggregated to generate noise maps.

d) **Forest Fire Detection:** Forest fire can cause damage to natural resources, property and human life. Early detection of forest fire can help in minimizing damage.

e) **River Flood Detection:** River floods can cause damage to natural and human resources and human life. Early warnings of floods can be given by monitoring the water level and flow rate. IoT based river flood monitoring system uses a no. of sensor nodes that monitor the water level and flow rate sensors.

4) Energy:

a) **Smart Grids:** is a data communication network integrated with the electrical grids that collects and analyze 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. By using IoT based sensing and measurement technologies, the health of equipment and integrity of the grid can be evaluated.

b) **Renewable Energy Systems:** IoT based systems integrated with the transformers at the point of interconnection measure the electrical variables and how much power is fed into the grid. For wind energy systems, closed-loop controls can be used to regulate the voltage at point of interconnection which coordinate wind turbine outputs and provides power support.

c) **Prognostics:** In systems such as power grids, real-time information is collected using specialized electrical sensors called Phasor Measurment Units(PMUs) 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.

5) Retail:

a) **Inventory Management:** IoT systems enable remote monitoring of inventory using data collected by RFID readers.

b) **Smart Payments:** Solutions such as contact-less payments powered by technologies such as Near Field Communication(NFC) and Bluetooth.

c) **Smart Vending Machines:** Sensors in a smart vending machines monitors its operations and send the data to cloud which can be used for predictive maintenance.

6) Logistics:

a) **Route generation & scheduling:** IoT based system backed by cloud can provide first response to the route generation queries and can be scaled up to serve a large transportation network.

b) **Fleet Tracking:** Use GPS to track locations of vehicles in real-time.

c) **Shipment Monitoring:** IoT based shipment monitoring systems use sensors such as temp, humidity, to monitor the conditions and send data to cloud, where it can be analyzed to detect food spoilage.

d) **Remote Vehicle Diagnostics:** Systems use on-board IoT devices for collecting data on Vehicle operaions (speed, RPMetc.,) and status of various vehicle subsystems.

7) Agriculture:

a) **Smart Irrigation:** to determine moisture amount in soil.

b) **Green House Control:** to improve productivity.

8) Industry:

a) Machine diagnosis and prognosis

b) Indoor Air Quality Monitoring

9) Health and LifeStyle:

a) Health & Fitness Monitoring

b) Wearable Electronics