RGPVNOTES.IN

Please do not share these notes on apps like WhatsApp or Telegram.

The revenue we generate from the ads we show on our website and app funds our services. The generated revenue **helps us prepare new notes and improve the quality of existing study materials**, which are available on our website and mobile app.

If you don't use our website and app directly, it will hurt our revenue, and we might not be able to run the services and **have to close them**. So, it is a humble request for all to **stop sharing the study material** we provide on various apps. Please **share the website's URL instead.**

Subject Notes

Unit 1

Syllabus: IoT definition, Characteristics, IoT conceptual and architectural framework, Components of IoT ecosystems, Physical and logical design of IoT, IoT enablers, Modern day IoT applications, M2M communications, IoT vs M2M, IoT vs WoT, IoT reference architecture, IoT Network configurations, IoT LAN, IoT WAN, IoT Node, IoT Gateway, Review of Basic Microcontrollers and interfacing.

INTERNET OF THINGS (IoT):

IoT Vision: The vision behind IoT is to have plug-n-play smart objects that can be deployed in any environment with an interoperable interconnection backbone that allows them to blend with other smart objects around them. Standardization of frequency bands and protocols plays a pivotal role in accomplishing this goal.

IoT Definition: The Internet of Things refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems.

CHARACTERISTICS of IoT

The fundamental characteristics of the IoT are as follows:

1. **Interconnectivity:** With regard to the IoT, anything can be interconnected with the global information and communication infrastructure.
2. **Things-related services:** The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in physical world and information world will change.
3. **Heterogeneity:** The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.
4. **Dynamic changes:** The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.
5. **Enormous scale:** The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet.
6. **Safety:** As we gain benefits from the IoT, we must not forget about safety. As both the creators and recipients of the IoT, we must design for safety. This includes the safety of our personal data and the safety of our physical well-being
7. **Connectivity:** Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data.

IOT Conceptual Framework

The main tasks of this framework are to analyze and determine the smart activities of these intelligent devices through maintaining a dynamic interconnection among those devices. The proposed framework will help to

standardize IoT infrastructure so that it can receive e-services based on context information leaving the current infrastructure unchanged.

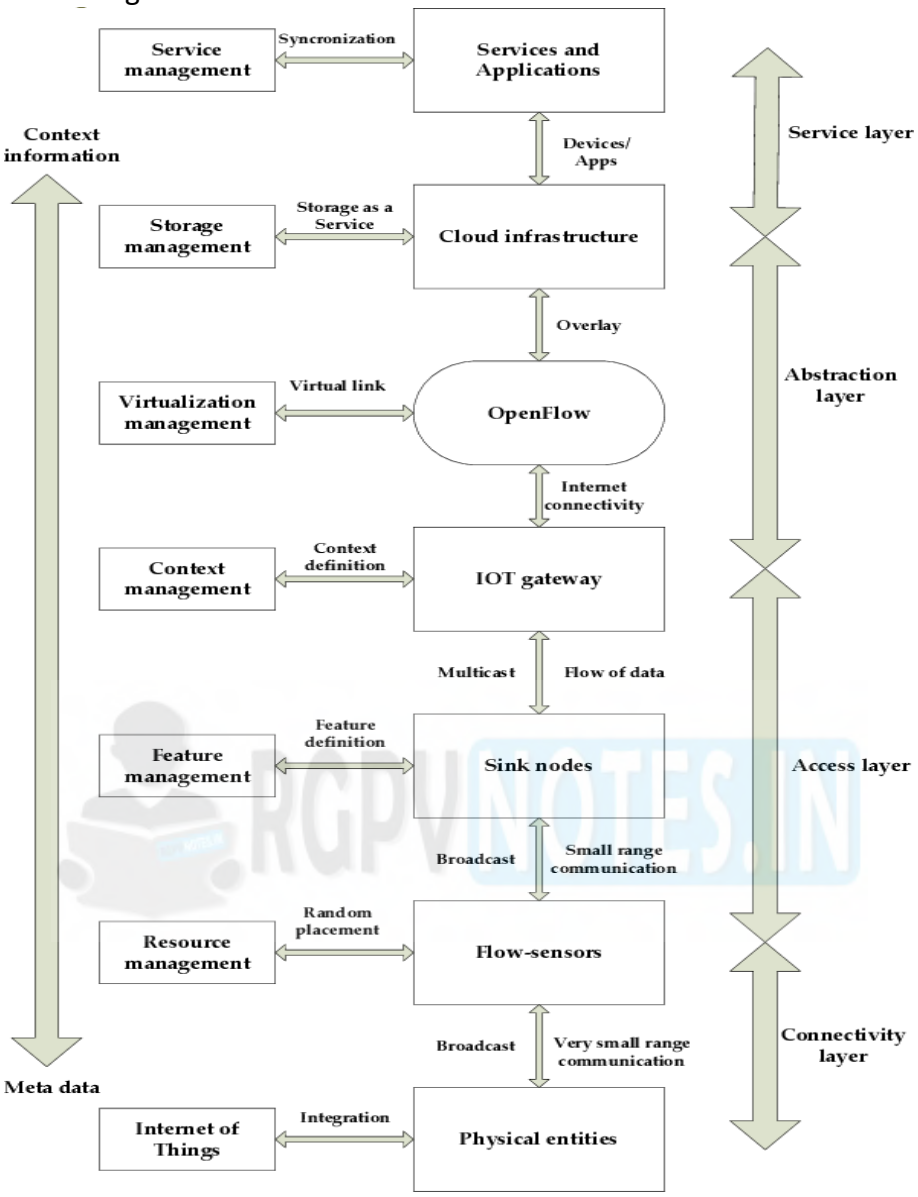

Figure 1.1: Conceptual Framework

1. Connectivity Layer

This layer includes all the physical devices involved in the framework and the interconnection among them. Future internet largely depends on the unification of these common objects found everywhere near us and these should be distinctly identifiable and controllable.

2. Access Layer

Context Data will be reached to internet via IoT Gateway as captured by short range devices in form of raw data. Access layer comprises topology definition, network initiation, creation of domains etc. This layer also includes connection setup, intra-inter domain communication, scheduling, packet transmissions between flow-sensors and IoT gateway.

3. Abstraction Layer

One of the most important characteristics of Open Flow is to add virtual layers with the preset layers, leaving the established infrastructure unchanged.

4. Service Layer

Storage management bears the idea about all sorts of unfamiliar and/or important technologies and information which can turn the system scalable and efficient. It is not only responsible for storing data but also to provide security along with it.

IOT Architectural Framework

.

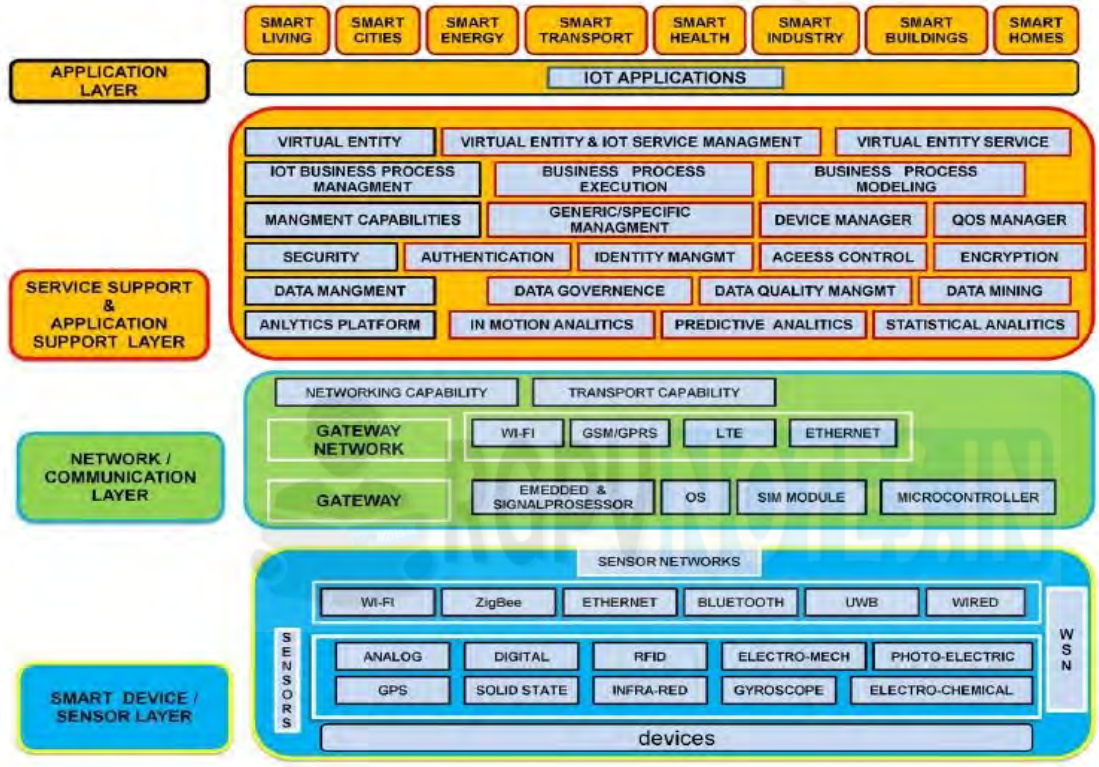

Figure 1.2: IoT Architectural Framework

IOT architecture consists of different layers of technologies supporting IOT. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IOT deployments in different scenarios

The functionality of each layer is described below:

- **Smart Device / Sensor Layer:** The lowest layer is made up of smart objects integrated with sensors. The sensors enable the interconnection of the physical and digital worlds allowing real-time information to be collected and processed. There are various types of sensors for different purposes.
- **Gateways and Networks-**Massive volume of data will be produced by these tiny sensors and this requires a robust and high performance wired or wireless network infrastructure as a transport medium.
- **Management Service Layer-**The management service renders the processing of information possible through analytics, security controls, process modeling and management of devices. One of the important features of the management service layer is the business and process rule engines

- **Application Layer-**The IoT application covers "smart" environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy.

Components of the IoT ecosystem

The IoT ecosystem consists of multiple components that allow businesses, governments, and consumers to connect to their IoT devices. These components include:

- **Sensors and actuators –** sensors and actuators are at the centre of the entire IoT network. Sensors are connected to assets in the form of a physical micro appliance, embedded into an IoT device. These sensors are responsible for collecting and gathering data in order to send signals or commands to the actuator. The actuator then responds to the signal or command and "acts" or makes something happen based on this signal. As an example, your office may make use of a smart air conditioning system that is set to a specific temperature. Sensors are used to monitor any changes in temperature in the office environment. If a change is detected, they send a signal to the actuators, which will then automatically adjust the airflow.
- **Connectivity** – This is largely referred to as the network layer and talks to how data is transferred and processed to ensure seamless communication between connected devices, sensors, the cloud, and actuators. For this to work efficiently, these elements need to be interconnected in order to understand the data and respond with the appropriate action. This is where IoT protocols and IoT gateways come in. IoT protocols provide a medium of transport for data collected from sensors. Data then goes through an IoT gateway that collects and translates the data being received via the protocols.
- **IoT Cloud** – Once the data has traveled through the IoT protocols and gateway, it moves to the cloud. The cloud is a high performance compute and storage ecosystem that is used for processing and data storage and brings all the different components of IoT together. In the cloud, data is filtered, managed, and stored. The data is then used to provide real-time analytics for fast decision making about what action should be taken in response to the data collected and signals received.
- **IoT analytics and data management** – This is used to make sense of the large amounts of data being processed. IoT technology can compute all raw data, being collected and transported, into data analytics which provides actionable insights and real-time solutions that can be used for effective decision making.
- **Devices and interface** – This is the visible component that an IoT user can use to control the system and set their preferences. This interaction is usually conducted on the device itself or remotely via smartphones, tablets, and laptops.

Physical and Logical Design of IoT

Physical Design of IoT

Physical Design of IoT refers to IoT Devices and IoT Protocols. Things are Node device which have unique identities and can perform remote sensing, actuating and monitoring capabilities. Communication established between things and cloud-based server over the Internet by various IoT protocols.

Basically, Things refers to IoT Devices which have unique identities and can perform remote sensing, actuating and monitoring capabilities. Things are is main part of IoT Application. IoT Devices can be various type, Sensing Devices, Smart Watches, Smart Electronics appliances, Wearable Sensors, Automobiles, and industrial

machines. These devices generate data in some forms or the other which when processed by data analytics systems leads to useful information to guide further actions locally or remotely.

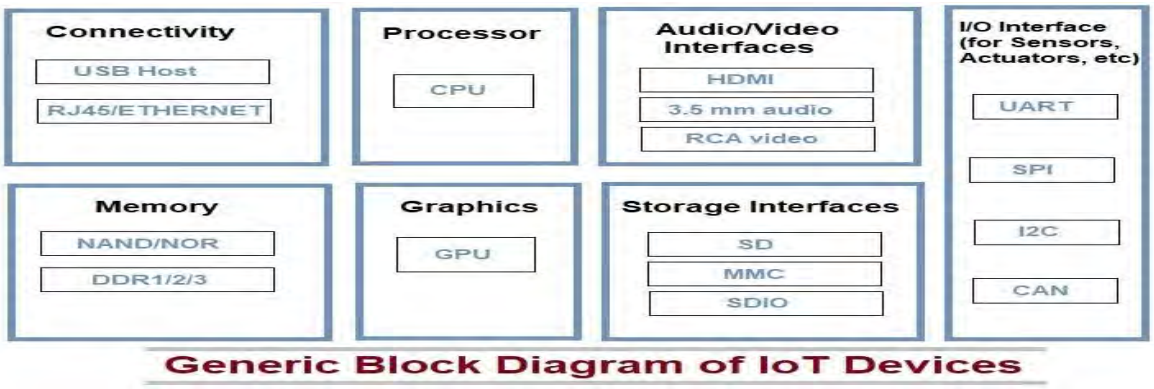

Figure 1.3: Block Diagram of IoT Devices (**Physical Design**)

For example, Temperature data generated by a Temperature Sensor in Home or other place, when processed can help in determining temperature and act according to users. Above picture, shows a generic block diagram of IoT device. It may consist of several interfaces for connections to other devices. IoT Device has I/O interface for Sensors, Similarly for Internet connectivity, Storage and Audio/Video. IoT Device collect data from on-board or attached Sensors and Sensed data communicated either to other device or Cloud based sever. Today many cloud servers available for especially IoT System. These Platform known as IoT Platform. These cloud especially design for IoT purpose. So here we can analysis and processed data easily.

Logical Design of IoT

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 describe 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, Services, Management, Security, and Application.

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.

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.

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.

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

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.
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 labelled 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 eliminate some instructions and improve efficiency and scalability. A RESTful web service is a “Web API” implemented using HTTP and REST principles. REST is most popular IoT Communication APIs.

IoT enablers

Internet of things (IoT) is a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies.
With the Internet of Things, the communication is extended via Internet to all the things that surround us. The Internet of Things is much more than machine to machine communication, wireless sensor networks, sensor networks, 2G/3G/4G, GSM, GPRS, RFID, WI-FI, GPS, microcontroller, microprocessor etc. These are considered as being the enabling technologies that make "Internet of Things" applications possible.

Sensors Used by IoT

IoT platforms function and deliver various kinds of intelligence and data using a variety of sensors. They serve to collect data, pushing it and sharing it with a whole network of connected devices. All this collected data makes it possible for devices to autonomously function, and the whole ecosystem is becoming "smarter" every day. By combining a set of sensors and a communication network, devices share information with one another and are improving their effectiveness and functionality. There many sensors also used by IoT system are as Temperature sensors, Proximity sensor, Pressure sensor, Water quality sensor, Chemical sensor, Gas sensor, Smoke sensor, IR sensors, Level sensors, Image sensors, Motion detection sensors, Accelerometer sensors, Gyroscope sensors, Humidity sensors, Optical sensors.

Modern day IoT applications

Potential applications of the IoT are numerous and diverse, permeating into practically all areas of every-day life of individuals, enterprises, and society. The IoT application covers "smart" environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, User interaction, Culture and tourism, Environment and Energy. Below are some of the IOT applications.

A. IOsL (Internet of smart living):

Remote Control Appliances: Switching on and off remotely appliances to avoid accidents and save energy, Weather, Smart Home Appliances, Safety Monitoring, and Intrusion Detection Systems: Detection of window and door openings and violations to prevent intruders.

B. IOsC (Internet of smart cities):

Structural Health: Monitoring of vibrations and material conditions in buildings, bridges and historical monuments, **Lightning**, **Safety**, **Transportation**, **Smart Parking**, **and Waste Management**: Detection of rubbish levels in containers to optimize the trash collection routes. Garbage cans and recycle bins with RFID tags allow the sanitation staff to see when garbage has been put out.

C. IOsE (Internet of smart environment):

Air Pollution monitoring, Forest Fire Detection, **Weather monitoring**, **Water Quality**, **River Floods**, **Protecting wildlife:** Tracking collars utilizing GPS/GSM modules to locate and track wild animals and communicate their coordinates via SMS.

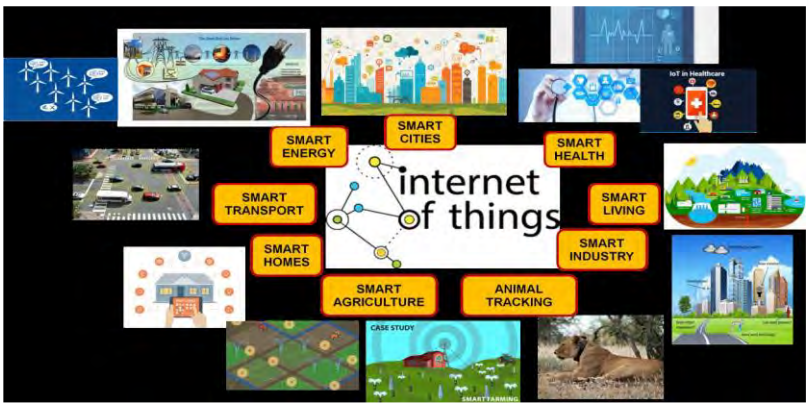

Figure 1.4: IoT Applications

MACHINE-TO-MACHINE COMMUNICATION

M2M means two machines "communicating," or exchanging data, without human interfacing or interaction. This includes serial connection, power line connection (PLC), or wireless communications in the industrial Internet of Things (IoT). Switching over to wireless has made M2M communication much easier and enabled more applications to be connected.

M2M allows virtually any sensor to communicate, which opens the possibility of systems monitoring themselves and automatically responding to changes in the environment, with a much-reduced need for human involvement. M2M can refer to any two machines—wired or wireless—communicating with one another.

M2M Working

The machine-to-machine communication makes the Internet of Things possible. M2M technologies can connect millions of devices within a single network. The range of connected devices includes anything from vending machines to medical equipment to vehicles to buildings. Virtually anything that houses sensor or control technology can be connected to some sort of wireless network.

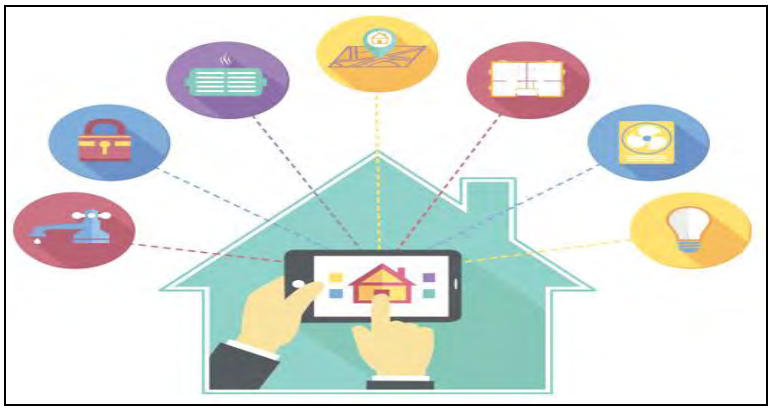

Figure 1.5: M2M Communication

M2M networks are very similar to LAN or WAN networks, but are exclusively used to allow machines, sensors, and controls, to communicate. These devices feed information they collect back to other devices in the network.

IoT Vs M2M

Internet Engineering Task Force (IETF) is responsible for creating the design and standardization of IOT abs M2M. IETF suggests some IOT specifications like:

Basis of	IoT (Internet of Things)	M2M (Machine to Machine)
Intelligence	Devices have objects that are responsible for decision making	Some degree of intelligence is observed in this
Connection type used	The connection is via Network and using various communication types.	The connection is a point to point
Communication protocol used	Internet protocols are used such as HTTP, FTP, and Telnet.	Traditional protocols and communication technology techniques are used
Data Sharing	Data is shared between other applications that are used to improve the end-user experience.	Data is shared with only the communicating parties.
Internet	Internet connection is required for communication	Devices are not dependent on the Internet.
Scope	A large number of devices yet scope is large.	Limited Scope for devices.
Business Type used	Business 2 Business(B2B) and Business 2 Consumer(B2C)	Business 2 Business (B2B)
Open API support	Supports Open API integrations.	There is no support for Open Api's
Examples	Smart wearable's, Big Data and Cloud, etc.	Sensors, Data and Information, etc.

Table 1.1: Compression between IoT &M2M

IoT vs WoT

- From the developer's perspective, the WoT enables access and control over IoT resources and applications using mainstream web technologies (such as HTML 5.0, JavaScript, Ajax, PHP, Ruby n Rails, etc).
- The approach to building WoT is therefore based on RESTful principles and REST API s, which enable s both developers and deployers to benefit from the popularity and maturity of web technologies.
- Still, building the WoT has various scalability security etc challenges especially as part of a roadmap towards a global WoT.

- The WoT is very similar to the IoT in some ways and in others it is drastically different. WoT was inspired by the IoT as in common everyday devices are connected to the Web and can communicate through various systems.
- However, where both begin to differentiate is the WoT is focused on reusing the already established Web system to help these everyday connected devices connect to one single application – in this case, the Web.

Reference Architecture of IoT

The reference architecture consists of a set of components. Layers can be realized by means of specific technologies, and we will discuss options for realizing each component. There are also some cross-cutting/vertical layers such as access/identity management.

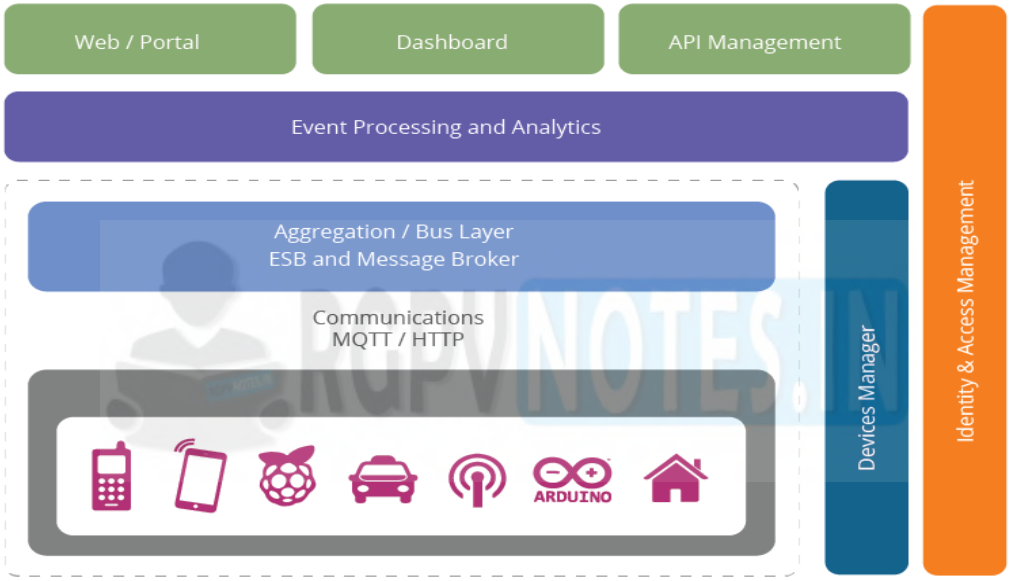

Figure 1.6: Reference architecture for IoT

Reference architecture layers are:

- Client/external communications - Web/Portal, Dashboard, APIs
- Event processing and analytics (including data storage)
- Aggregation/bus layer – ESB and message broker
- Relevant transports - MQTT/HTTP/XMPP/CoAP/AMQP, etc.
- Devices

The cross-cutting layers are:

- Device manager
- Identity and access management

The Device Layer

The bottom layer of the architecture is the device layer. Devices can be of various types, but in order to be considered as IoT devices, they must have some communications that either indirectly or directly attaches to the Internet. Examples of direct connections are

- Arduino with Arduino Ethernet connection

- Arduino Yun with a Wi-Fi connection
- Raspberry Pi connected via Ethernet or Wi-Fi
- Intel Galileo connected via Ethernet or Wi-Fi
- ZigBee devices connected via a ZigBee gateway
- Bluetooth or Bluetooth Low Energy devices connecting via a mobile phone
- Devices communicating via low power radios to a Raspberry Pi

Each device typically needs an identity. The identity may be one of the following:

- A unique identifier (UUID) burnt into the device (typically part of the System-on-Chip, or provided by a secondary chip)
- A UUID provided by the radio subsystem (e.g. Bluetooth identifier, Wi-Fi MAC address)
- An OAuth2 Refresh/Bearer Token (this may be in addition to one of the above)
- An identifier stored in nonvolatile memory such as EEPROM

The Communications Layer

The communication layer supports the connectivity of the devices. There are multiple potential protocols for communication between the devices and the cloud. The most well-known three potential protocols are

- HTTP/HTTPS (and RESTful approaches on those)
- MQTT 3.1/3.1.1
- Constrained application protocol (CoAP)

The Aggregation/Bus Layer

An important layer of the architecture is the layer that aggregates and brokers communications. This is an important layer for three reasons:

1. The ability to support an HTTP server and/or an MQTT broker to talk to the devices;
2. The ability to aggregate and combine communications from different devices and to route communications to a specific device (possibly via a gateway)
3. The ability to bridge and transform between different protocols, e.g. to offer HTTP-based APIs that are mediated into an MQTT message going to the device.

The aggregation/bus layer provides these capabilities as well as adapting into legacy protocols. The bus layer may also provide some simple correlation and mapping from different correlation models (e.g. mapping a device ID into an owner's ID or vice-versa).

The Event Processing and Analytics Layer

This layer takes the events from the bus and provides the ability to process and act upon these events. A core capability here is the requirement to store the data into a database. This may happen in three forms. The traditional model here would be to write a server-side application, e.g. this could be a JAX-RS application backed by a database. However, there are many approaches where we can support more agile approaches.

Client/External Communications Layer

The reference architecture needs to provide a way for these devices to communicate outside of the device-oriented system. This includes three main approaches. Firstly, we need the ability to create web-based front ends and portals that interact with devices and with the event-processing layer. Secondly, we need the ability to create dashboards that offer views into analytics and event processing. Finally, we need to be able to interact with systems outside this network using machine-to-machine communications (APIs). These APIs need to be managed and controlled and this happens in an API management system.

IoT Network configurations

IoT Network configuration is the process of setting a network's controls, flow and operation to support the network communication of an organization and/or network owner. This broad term incorporates multiple configuration and setup processes on network hardware, software and other supporting devices and components. In simple terms, the 4 Stage IoT architecture consists of Network is

1. Sensors and actuators
2. Internet getaways and Data Acquisition Systems
3. Edge IT
4. Data center and cloud.

An IoT network refers to a collection of interconnected devices that communicate with other devices without the need for human involvement, such as autonomous cars, smart appliances, and wearable tech.
To configure wireless IoT device to use a Programmable Wireless SIM requires only a few small configuration settings. Need to set the Programmable Wireless Access Point Name (APN) and need to use the Twilio Commands phone number.
Depending on the device you are using, you may also be required to enter TCP and UDP network timer settings.
Configure the Programmable Wireless APN
Broadband IoT
The APN for the Programmable Wireless SIM is:
wireless.twilio.com
No authentication is required for this APN — leaves any username and password entries blank.
Narrowband IoT
The APN for the Narrowband SIM, provided in partnership with T-Mobile USA, is:
iot.nb
No authentication is required for the NB-IoT APN — leaves any username and password entries blank.

IoT LAN

Wireless personal and local area network technologies that are commonly incorporated into IoT connectivity solutions are WiFi and Bluetooth. WiFi can be used for applications that run in a local environment, or in a distributed setting if there are multiple access points integrated into a larger network.

Industrial IoT, where the local network is based on any one of many different technologies. The IoT device will typically transmit data over the global Internet. Commercial IoT, where local communication is typically either Bluetooth or Ethernet (wired or wireless).

IoT WAN

There are many low-power wide area (LPWA) radio technologies to choose from when deploying an Internet of Things (IoT) network. The final choice of radio technology, however, is only one of the many considerations when designing the low-power wide area network (LPWAN) where factors such as application type, network topology, total cost of ownership (TCO), reliability, security, and business model must also be considered.

The LPWA radio technologies choices can be broadly subdivided into those that use the unlicensed ISM (Industrial Scientific and Medical) frequency bands and those that operate in the licensed frequency bands. Each of the LPWA technologies offers a choice or trade-off between transmit range, data rate, frequency, channel bandwidth, and power consumption.

IoT Node

The IoT node as we know it today, in its most minimal use case, can be a sensor embedded in an object that is never serviced again across the life of the device. They can be wireless and operated on a coin cell battery for years. What seemed impossible just a few years ago is now quickly becoming standard. And that's thanks to incredible innovations in low-power operation of wireless modules. The most numerous types of device in the IoT can be referred to as the node. These are all the exciting devices that are providing sensor data, or devices that are being controlled from the cloud. This means things like door locks, security sensors, temperature sensors, and more.

Laird Connectivity's BL654, for example, is a product that comes from a long line of Bluetooth modules that support Bluetooth Low Energy (BLE). BLE, introduced in the Bluetooth v4.0 specification, enables infrequent status-type messaging between Bluetooth devices with long sleep cycles in between messages.

IoT Gateway

The IoT gateway is the central hub for sensors that collects their data, and they come in many forms. They interface directly with sensors and provide the path for that data to go to the cloud. Gateways can be designed to operate in so many ways that it can be hard to generalize.

In some cases, they may listen passively, and the sensor operates without even knowing the gateway is there. In some cases, they may establish bidirectional communication with the sensor, allowing the sensor to be controlled by the cloud through the gateway. Most IoT devices communicate over either Wi-Fi, LTE, Bluetooth, or LoRaWAN.

Architectural Overview

- The following gateway architecture diagram is the most common architectural design where the gateway itself is not equipped with sensors. The gateway software installed on the device is responsible for collecting data from the sensor, pre-processing that data, and sending the results to the data centre.
- Keep in mind that it is possible to have variations on this sensor architecture where some of the sensors are located at the gateway device, as illustrated in the following diagram.

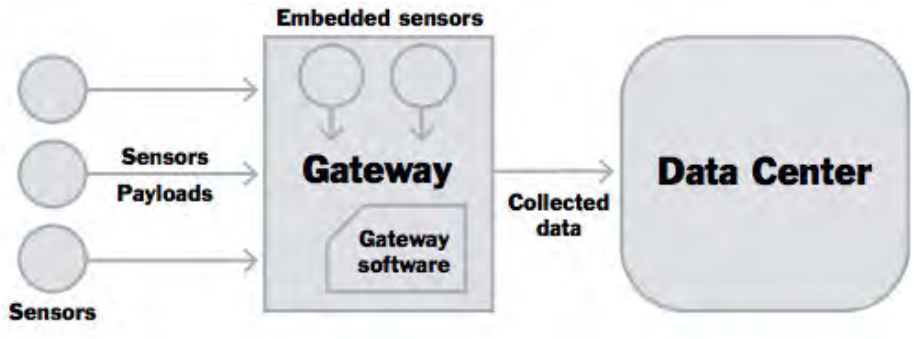

Figure 1.7: Gateway Architecture Diagram

- Embedded sensors that might be present at the gateway could include options like a GPS unit or a temperature sensor connected to the gateway using the GPIO interface.

IoT Proxy

- The IoT proxy is a server as well as a client between the IoT client and Work Space Protocol (WSP). The IoT proxy has the RD functionalities for registering information of resources which expose services in the network and discovering the information by IoT clients.
- A proxy server, also known as a "proxy" or "application-level gateway", is a computer that acts as a gateway between a local network (for example, all the computers at one company or in one building) and a larger-scale network such as the internet. Proxy servers provide increased performance and security.

Review of Basic Microcontrollers and interfacing

A microcontroller is a small, low-cost, and self-contained computer-on-a-chip that can be used as an embedded system. A few microcontrollers may utilize four-bit expressions and work at clock rate frequencies, which usually include:

- An 8 or 16-bit microprocessor.
- A little measure of RAM.
- Programmable ROM and flash memory.
- Parallel and serial I/O.
- Timers and signal generators.
- Analog to Digital and Digital to Analog conversion

Microcontrollers usually must have low-power requirements since many devices they control are battery-operated. Microcontrollers are used in many consumer electronics, car engines, computer peripherals, and test or measurement equipment and these are well suited for long-lasting battery applications.

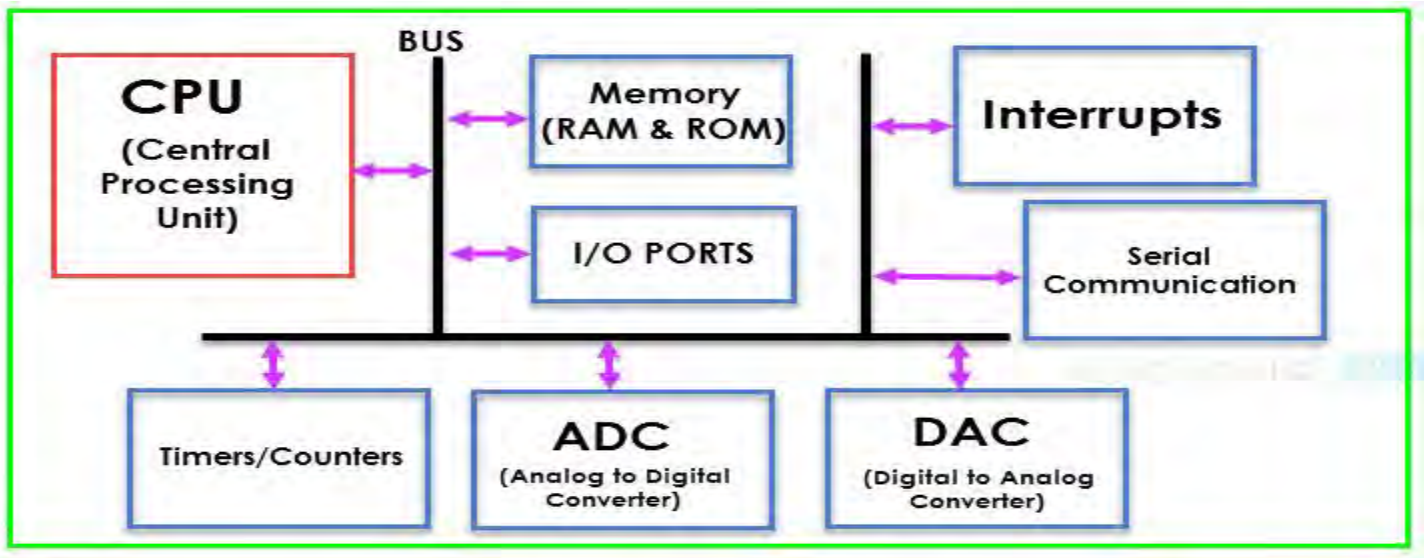

Figure 1.8: Basic Structure of Microcontroller

The Components of Microcontrollers are:

- **CPU**- The microcontroller is called a CPU device, used to carry & decode the data & finally completes the allocated task effectively. By using a central processing unit, all the microcontroller components are connected to a particular system. Instruction fetched through the programmable memory can be decoded through the CPU.
- **Memory-** In a microcontroller, the memory chip works like a microprocessor because it stores all the data as well as programs. Microcontrollers are designed with some amount of RAM/ROM/flash memory to store the program source code.
- **I/O Ports-** Basically, these ports are used to interface otherwise drive different appliances like LEDs, LCDs, printers, etc.
- **Serial Ports**- Serial ports are used to provide serial interfaces between microcontroller as well as a variety of other peripherals like parallel port.
- **Timers**- A microcontroller includes timers otherwise counters. These are used to manage all the operations of timing and counting in a microcontroller. The main function of the counter is to count outside pulses whereas the operations which are performed through timers are clock functions, pulse generations, modulations, measuring frequency, making oscillations, etc.
- **ADC (Analog to Digital Converter)**- ADC is the acronym of analog to digital converter. The main function of ADC is to change the signals from analog to digital. For ADC, the required input signals are analog and the production of a digital signal is used in different digital applications like measurement devices
- **DAC (Digital to Analog Converter)**- The acronym of DAC is digital to analog converter, used to perform reverse functions to ADC. Generally, this device is used to manage analog devices such as DC motors, etc.
- **Interpret Control**- This controller is employed to give delayed control to a running program & interpretation is either internal otherwise external.

RGPVNOTES.IN

Thank you for using our services. Please support us so that we can improve further and help more people.
https://www.rgpvnotes.in/support-us

If you have questions or doubts, contact us on
WhatsApp at +91-8989595022 or by email at hey@rgpvnotes.in.

For frequent updates, you can follow us on
Instagram: https://www.instagram.com/rgpvnotes.in/.