**4.1 Definition**

The Internet of things (IoT) is the extension of internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

The definition of the Internet of things has evolved due to convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded system. Traditional fields of embedded system wireless sensor networks, control system, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such smartphones and smart packets.

The Internet of Things (IoT) is the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure. The “Internet of Things” connects devices and vehicles using electronic sensors and the Internet.

**4.2 Introduction**

The Internet of Things (IoT) is the network of physical objects devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data. The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer based systems, and resulting in improved efficiency, accuracy and economic benefit, when IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure.

So, Internet of Things or IoT is an architecture that comprises specialized hardware boards, Software systems, web APIs, protocols which together creates a seamless environment which allows smart embedded devices to be connected to internet such that sensory data can be accessed and control system can be triggered over internet.

Also, devices could be connected to internet using various means like Wi-Fi, Ethernet and so on. Furthermore, devices may not need to be connected to internet independently. Rather a cluster of devices could be created (for example a sensor network) and the base station or the cluster head could be connected to internet. This leads to more abstract architecture for communication protocols which ranges from high level to low level.

Most interestingly, these devices must be uniquely discovered. For unique discovery of the devices in a Network, they need to have unique IP address. IoT devices essentially have IPv6 addressing scheme. All these devices have either fixed or Subnet masked IP addresses of type v6. Unique IP addresses makes IoT devices discoverable in the internet as independent node. This is the most important concept to have in mind to understand IoT.

Following figure.4.1 explain what IoT is all about.



Figure 4.1 Internet of Things (IoT) Basic Architecture

**4.3 What Devices Makes it to IoT**

Since IoT are essentially embedded systems and smart objects connected to internet with unique IP address which can be discovered and communicated over internet. We have also seen that the IoT devices may have external peripheral like Actuators and Sensors.

**4.3.1 Are Mobile Phones are IoT Devices**

One of the most common in day to day life are mobile phones. Mobile phone is essentially an embedded system with a processor at the core having display and keypad. They support wide variety of sensors like ambient light Sensors, Accelerometer, Gyroscope and so on.

They are connected to internet. Mobile phones get IP addresses, can access internet. In other words, it virtually fits every description of IoT. So, can we call mobile phones IoT devices? This doubt was clarified at a keynote event during Sept 2011's Mobile World Congress in Barcelona by Qualcomm Chairman and CEO Dr. Paul Jacobs.

Paul Jacobs talked about how mobile technology could be used to **connect non-phone,** **non-tablet devices called IoT devices** and objects to the Internet. In this future where everything is Web-connected, **mobile phones will serve as the hub, or the remote control**, for Internet of Things.

So IoT is internet connectivity of smart objects and embedded system other than mobile phones which can be connected with external hardware and Mobiles, Tablets, Laptops and PCs are remote control/access center of IoT.

**4.3.2 IoT Devices**

The most common and popular technologies in IoT will give an overview device. The IoT devices into two broad categories: The wearable ones and Microcontroller /Microprocessor driven embedded IoT devices. Some of the embedded devices like Arduino Lilly pad are Monique and it can further utilize them to make wearable solution. But wearable includes hardware which are pretty standard and IoT has only software scope for the developer. Some peripheral hardware is which might require are in IoT hardware in embedded level. Apps can be used with popular wearable platforms, Embedded IoT platform may include broader technologies like Raspberry Pi, Arduino or Galileo, etc.



Figure 4.2 Common IoT Devices and Technologies

**4.3.3 IoT Platforms**

IoT development can be divided into two parallel technologies: Wearable and Embedded. Developers can build apps for custom Wearable devices like Pebble, Samsung Gear or can often create their own platform using Embedded solution and then can develop app for that platform.

IoT platform is an essential component of a huge IoT ecosystem that supports and connects all components within the system. It helps to facilitate device management, handle hardware/software communication protocols, collect/analyze data, enhance data flow and functionality of smart applications.

The overall IoT system includes:

* hardware (devices and sensors)
* connectivity through a router, gateway, wi-fi, satellite, ethernet, etc.
* software
* user interface

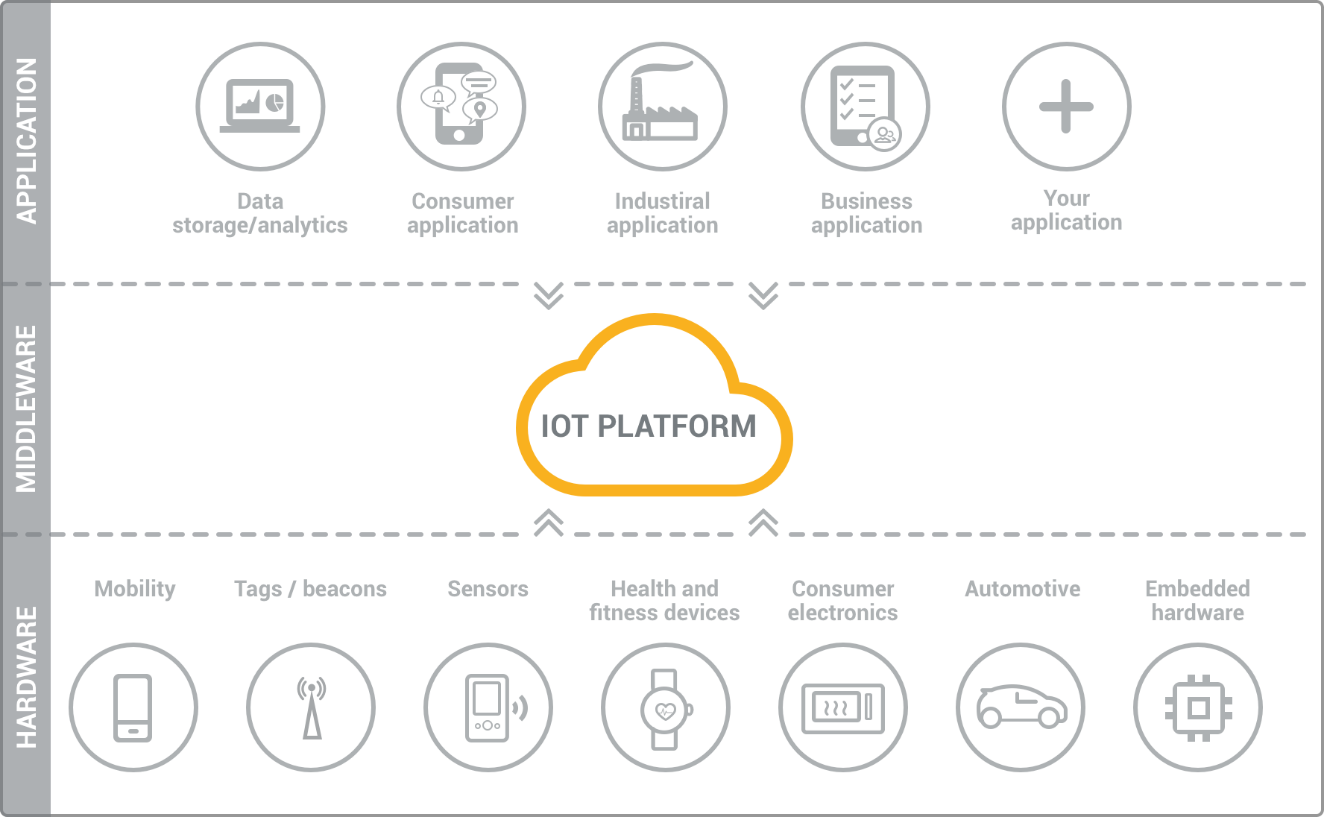


Figure 4.3 IoT platform

**4.3.3.1 Wearable Platform**

Tizen is fast becoming one of the most popular platforms for Mobile and wearable devices. Tizen SDK comes ported with wearable emulator which makes it easier to develop wearable solutions for Tizen platform.

As figure 4.2 suggests, a large Android Wear device are now being made and marketed. Smart watches are getting popular by every day. Android Wear apps can be developed and tested in Eclipse.

This Android Developer Guide helps you in setting up Android Wear development environment in Eclipse. Salesforce is another platform which is coming up with awesome development environment, APIs in wearable technologies. Their solution is extended from Pebble to Google glass.

**4.3.3.2 Embedded Platform**

**Arduino** is probably the best starting point for embedded based IoT. Basic Arduino boards don't come with Ethernet shield or Wi-Fi shield and for Arduino to be able to work as IoT device, their need to select Arduino with Ethernet shield or Wi-Fi shield. Arduino Yun on the other hand is a board that comes ported with Ethernet shield.

**Raspberry Pi** is probably one of the best things to happen in DIY IoT. A wide range of Data driven applications like Home Automation Server to Home Multimedia server, File Server can be developed with Pi. PI like Arduino has general purpose IO pins. But seamless working with sensors is bit tedious in Pi.

Another efficient IoT board is **Intel Edition** which has integrated BLE, Wi-Fi among host of other features. It supports wide range of Industry standard hardware (over 30) through 70-pin interface.

**Intel Galileo** is another good offering by Intel which supports the same shielding that of Arduino Uno. So, it can be said to be first Intel powered device which is Arduino compatible. It has among other thing a USB host controller like Raspberry Pi which makes this an attractive hardware. Galileo also has ethernet shield in built.

**4.3.3.3 Cloud Platform**

IoT really can bring several services (like online payment gateway), several hardware platforms (like embedded board of the vending machine) and smart objects and data like NFC, GPS into a seamless environment. Now it can integrate online payment into beverage vending machine, if one is using location service for beverage machine, then utilizing the location and payment service can be done.

One can get the data of a medical diagnosis like ECG (acquired through another

embedded board pertaining to medical electronics) into cloud such that several doctors can view it and form a comprehensive opinion about the patient's state.

Well, in fact all of them are possible. A little understanding of web and software design would take your mind towards cloud. Just like Web of Machines, in a Machine to Machine (M2M) or Machine to Objects (M2O) or any similar communication several modules will be common and several modules demands data to be available for sharing. Cloud APIs comes in handy in this regard.

For instance, to make a device discoverable in web, then assign a fixed IP address, maintain a router and follow several networking skills.

**Yaler** is a great example of what services and cloud can bring to table. This provides connection as a service such that your device is easily discoverable and communicable over the web without much hassle and take care of underneath security.

**Axeda** Provides infrastructure for M2M architecture.

**OpenIoT** is an open source IoT platform that provides out of other services a unique Sensing as a Service.

**Google** has already integrated location services with its cloud. Location extracted from your devices are silently put in your status updates in Facebook and twitter and are also used for more personalized searches.

So, cloud APIs has a great potential in IoT in all levels of architecture starting from firmware to hardware to more top-level architecture.

**4.4 Implementation using IoT**

This project uses concept of IoT for monitoring and controlling the system using a public server called **MQTT** server. It uses an android app called **MyMQTT**. In this app, one has to subscribe a topic and publish a message of specific function. The server will call-back to perform the function.

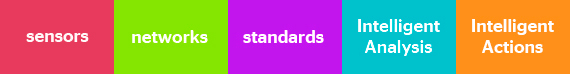


Figure 4.4 component of IoT implementation

**4.4.1 MQTT**

MQTT stands for Message Queue Telemetry Transport. It is a publish /subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimize network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium.

**4.4.2 MQTT Architecture**

MQTT has a client/server model, where every sensor is a client and connects to a server, known as a broker, over TCP.

MQTT is message oriented. Every message is a discrete chunk of data, opaque to the broker.

Every message is published to an address, known as a topic. Clients may subscribe to multiple topics. Every client subscribed to a topic receives every message published to the topic.

MQTT defines methods (sometimes referred to as *verbs*) to indicate the desired action to be performed on the identified resource. What this resource represents, whether pre-existing data or data that is generated dynamically, depends on the implementation of the server.

Often, the resource corresponds to a file or the output of an executable residing on the server.

**Connect:** Waits for a connection to be established with the server.

**Disconnect:** Waits for the MQTT client to finish any work it must do, and for

the TCP/IP session to disconnect.

**Subscribe:** Waits for completion of the Subscribe or Unsubscribe method.

**Unsubscribe:** Requests the server unsubscribe the client from one or more topics.

**Publish:** Returns immediately to the application thread after passing the request to the MQTT client

**4.4.3 MQTT Ports**

The server listens on the following ports:

* 1883: MQTT, unencrypted
* 8883: MQTT, encrypted
* 8884: MQTT, encrypted, client certificate required
* 8080: MQTT over WebSocket’s, unencrypted
* 8081: MQTT over WebSocket’s, encrypted

This project uses 1883 an unencrypted MQTT port.

**4.4.4 MQTT Example**

Imagine a simple network with three clients and a central broker. All three clients open TCP connections with the broker. Clients B and C subscribe to the topic temperature (Figure 4.5).



Figure 2.3 Client B and C Subscribing Topic temperature.

The publisher subscriber model allows MQTT clients to communicate one-to-one, one-to many and many-to-one.