International Journal on Future Revolution in Computer Science & Communication Engineering ISSN: 2454-4248
Volume: 4 Issue: 10 117 – 119

IoT – A Pathway to Smart India – Part 1 (Basic Anatomy)

AmitaPareek, MTech Scholar, SIET Sikar,
Mr Om Pal, Asst Professor, SIT Sikar

Abstract: In this exploratory paper, the basic anatomy is revealed. It also describes the Architecture, Technology, Protocol & Interface. Here the novice reader can easily go through the part 1 and specific part in next publication as per their interest.

*****

1. INTRODUCTION

The Internet of things (IoT) is the inter-networking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data. In 2013, the Global Standards Initiative on Internet of Things (IoT-GSI)(Global Standard Initiative) defined the IoT as "the infrastructure of the information society." The IoT allows objects to be sensed or 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 in addition to reduced human intervention. 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, virtual power plants, 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. Typically, IoT is expected to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine (M2M) communications and covers a variety of protocols, domains, and applications. The interconnection of these embedded devices (including smart objects), is expected to usher in automation in nearly all fields, while also enabling advanced applications like a smart grid, and expanding to areas such as smart cities.

2. HOW IOT WORKS

IoT devices on network enabled embedded systems. This device is always connected to a centralized server. IoT device collect the data retrieve from physical parameters using their sensors. This data is uploaded to service using TCP IP Protocol and various intermediate networking Technologies. Downloading data is stored in a database which is available on that centralized server. Plants are also connected to same server client give commands to the server which are then send to hardware devices for the execution of the hardware device are the IoT devices which are available on various geo-locations. After performing any action the device generates feedback. This feedback is again uploaded to the same centralized server on that Database, from that database the clients retrieve that feedback and on the basis of that feedback web based user interface provide the current status of the IoT device which is present on that specific geo-location.

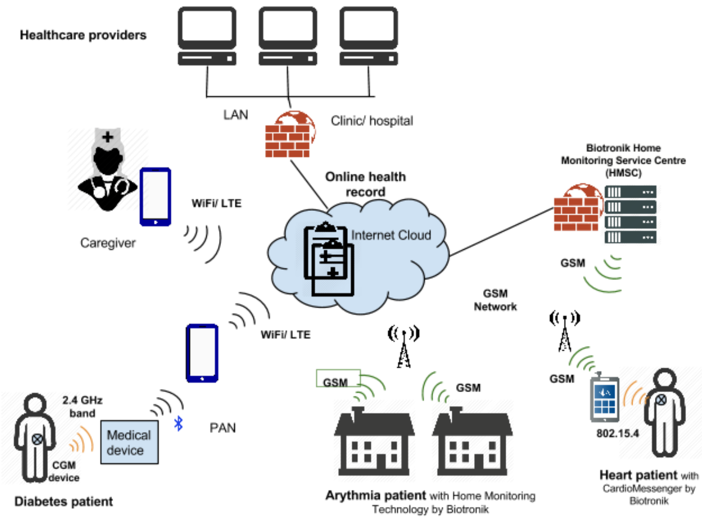

Figure 1. - Basic IoT Anatomy

3. TECHNOLOGIES AND PROTOCOLS REQUIRED FOR IoT

IoT primarily exploits standard protocols and networking technologies. However, the major enabling technologies and protocols of IoT are RFID, NFC, low-energy Bluetooth, low-energy wireless, low-energy radio protocols, LTE-A, and WiFi-Direct. These technologies support the specific networking functionality needed in an IoT system in contrast to a standard uniform network of common systems.

3.1 NFC and RFID

RFID (radio-frequency identification) and NFC (near-field communication) provide simple, low-energy, and versatile options for identity and access tokens, connection bootstrapping, and payments.

- RFID technology employs 2-way radio transmitter-receivers to identify and track tags associated with objects.
- NFC consists of communication protocols for electronic devices, typically a mobile device and a standard device.

3.2 Low-Energy Bluetooth
This technology supports the low-power, long-use need of IoT function while exploiting a standard technology with native support across systems.

3.3 Low-Energy Wireless
This technology replaces the most power hungry aspect of an IoT system. Though sensors and other elements can power down over long periods, communication links (i.e., wireless) must remain in listening mode. Low-energy wireless not only reduces consumption, but also extends the life of the device through less use.

3.4 Radio Protocols
ZigBee, Z-Wave, and Thread are radio protocols for creating low-rate private area networks. These technologies are low-power, but offer high throughput unlike many similar options. This increases the power of small local device networks without the typical costs.

3.5 LTE-A
LTE-A, or LTE Advanced, delivers an important upgrade to LTE (Long-Term Evolution) technology by increasing not only its coverage, but also reducing its latency and raising its throughput. It gives IoT a tremendous power through expanding its range, with its most significant applications being vehicle, UAV (Unmanned Aerial Vehicle), and similar communication.

3.6 WiFi-Direct
WiFi-Direct eliminates the need for an access point. It allows P2P (peer-to-peer) connections with the speed of WiFi, but with lower latency. WiFi-Direct eliminates an element of a network that often bogs it down, and it does not compromise on speed or throughput.

3.7 Web Development
Every person with a smart-phone has experienced IoT at some point or the other. For instance, the auto-notification system that sends messages based on a person's mobile phone location can be considered a most basic application of IoT. Another basic application is the auto-updating of a business database on the basis of movement of RFID-tagged assets within the company premises. In the near future, businesses will be seen using a combination of sensor and wireless technologies to track the customer's buying behavior. On the other hand, customers will use IoT technologies to automatically avail customized products and personalized services. Use of sensors and analytics will enable people to track their health as well as control their home appliances. The complex communication system will drastically change the process of web development. Out of the large number of things impacted by IoT, web designing will be one of the key areas that will be radically affected by the future developments.

3.8 User Interfaces and IoT
The successful implementation of any IoT system depends on two key elements: the user interface and the user experience. Since, the data acquired from the interaction between the devices will be accessible on a device screen, web designers will be required to build great user interfaces. Web designers will need to design user interfaces for a large number of networked devices with different screen sizes and resolutions. Moreover, these devices will be running on different platforms. It becomes imperative that web designers enhance the user experience by designing responsive user interface based on the user devices and environments. Designers are already developing responsive websites for mobile devices. In the future, they will need to design responsive interfaces for other types of devices that form the IoT.

3.9 Server and IoT
For many Internet of Things (IoT) applications, embedding a HTTP/WebSocket server on a device is enough to get the job done. Oftentimes, this solution can be faster, simpler and more cost effective. Choosing the right web server can be difficult, so let's run through the decision making factors.

3.9.1 Why a web server?
First, a step back. Can an embedded web server really be an alternative to building an entire IoT infrastructure?
Depending on what we are trying to achieve, absolutely.
A good web server can serve Web GUI on devices, implement RESTful services, RPC (e.g. JSON-RPC), asynchronous WebSocket communication and handle telemetry data exchange amongst other functionalities. It should be plug & play for developers: taking existing code and just by adding some additional lines of code be able to integrate it into your existing product.

It can solve remote access demand from consumers by creating a dashboard server, enable predictive maintenance and monitoring on a one to many level through RESTful API server application and offer real-time and server-side data push through WebSocket functionality.

ISSN: 2454-4248
117 – 119

For many applications an embedded web server makes sense; financially as well as from a development complexity perspective.

3.9.2 Choosing the right solution

There are plenty of solutions on the market. Wikipedia alone lists 30+ on their comparison page. Let this be a first stop and check which web servers have the technical capacities needed. Then measure them against these 8 decision making factors:

- **Size -** We are integrating with your existing solution and need the most compact and lightweight solution out there to give you room to manoeuvre.
- **Security -** Ensure you can use SSL/TLS and have an authentication mechanism in place if the units are accessed remotely.
- **Stability & Maturity-** Go with a solution that your peers trust and has been around the block a few times. Check GitHub and other communities to see how actively the project is contributed to and updated.
- **Agnostic -** Choose a solution that is platform agnostic. So whether now or in future you want to make a move and develop on eCos, UNIX/Linux, Windows, MacOS or take things into the app space with iPhone or Android your selected solution will let you do this.
- **Simplicity -** Plug & play. Use a library that you can easily integrate by pasting the code and just adding a few lines of code. Choosing a vendor should make your life easier.
- **Clarity -** Look for clarity when checking out the embedding API. Ideally the source should be in a single file to make embedding easy.
- **Licensing -** Clear pricing that suits your needs. Start with a product that is open source so you can test and when you are ready to commercially apply the solution, change to a commercial license. A commercial license indicates the solution is maintained and updated regularly and you have an actual company behind the product to rely on. Ideally we want a vendor that understands that the size of the project matters in pricing: are you using this for just one product, a product line or more?
- **Support-** We want a solution that is simple and straightforward so that ideally you won’t need support. But especially if this is the first integration project, the option to fall back onto a support structure is comforting and will ensure you can move your project forward fast.

References

[1] IoT – Architeture, Protocol & Services, P Sethi S Sarangi
[2] Protocol for IoT, Cyient
[3] Architecture & Protocol for IoT, Angelo, Casari
[4] IoT – Architecture, IoT-A
[5] Application Layer Protocol forIoTbyAsim
[6] Layer Architecture by Darwish
[7] Architecture for Iot –Zulfikar
[8] Security Framework for IoT – Liu, Zangg, Trapp