

IoT Generic Frameworks: What Needs to Improve

Indunil Withana
University of Westminster
Informatics Institute of Technology,
Colombo 06, Sri Lanka
w1673841@my.westminster.ac.uk

Cassim Farook
Informatics Institute of Technology
No 57, Ramakrishna Road,
Colombo 06, Sri Lanka
cassim.f@iit.ac.lk

Abstract - *Internet of things (IoT) is one of the trending technologies which is available in the current technology world. The term IoT could be described as devices which could be connected via the internet. The number of devices in the world is increasing rapidly every minute. The world is moving towards IoT enabled smart cities. To control these devices, a generic framework is needed. This paper contains a review on technologies, techniques, and domain found out by ongoing research which introduces a generic framework to manage IoT devices.*

Keywords - *Internet of Things, Architecture, Framework, Live Data Logging, Data Interoperability, Security*

I. INTRODUCTION

Data which are populated by IoT controlled devices are valuable for many reasons. People use these devices generated data for their business-related decision. [1]. The world is creating cities powered by smart devices to create a smart user-friendly living. Barcelona is testing a smart city which has been created using smart devices and millions of devices generating a huge amount of data [2]. It has been predicted that by the year 2020, 50 billion IoT devices would be in the world. [3]. Invest bank such as Morgan Stanley [4] has confirmed that the usage of IoT related technologies in the business gives an advantage over other companies which use legacy systems in their businesses. When analyzing research done in this domain, researches have been carried out in handling and processing data which is a challenging task in the IoT world.

Data has been a key resource in today's world. IT devices generate a huge amount of data. These data can be used to take advantage. Algorithms can be refined and tuned using this data. Section 2 of this paper contains the information gathering techniques used for the research. It follows by the description of the problem domain which is section 3. Section 4 compiles the related work carried out in the domain.

II. PROBLEM DOMAIN

Researches have been carried out on the problems related to IoT data management. But most of the solutions lack certain factors. Some of the solutions have not answered the IoT management. IoT is not just managing data. IoT framework must be able to manage, data, devices, and users must be able to command, control and monitor devices. Most of the researchers have identified the above factors, but they have proposed solutions for one or two sections of them rather than giving a fully equipped solution. When implementing a generic IoT framework, there are certain factors that need to be addressed. These following factors have been identified by several types of research and detailed analysis follows them.

- Sensor data storing mechanism.
- Data Interoperability in data communication
- Live information monitoring.
- Security implementation.

III. IOT GENERIC PLATFORM RESEARCH METHODOLOGY

IoT related research has been widely done in the recent past. Before starting the research knowledge capture by reading the existing literature on the domain and by performing a case study regarding the problem.

Details stated below are gathered from a case study and interview carried at Mirissa, Sri Lanka. Even at the present day, people who control boats in Mirissa whale watching, communicate using mobile phones to find the direction they should travel, and direction whales are located. Some owners go on board to provide a good service and to prevent fraud. If there is a system which could show the owner about location details of the boat, fuel level, temperature and get information about whales, he or she could communicate with the boat from land rather than being on board of the boat. A device which contains a temperature sensor, a location GPS tracker and speed sensor could easily make his or her life easier. Generic