A Comparative Study On Various Providers Of Wearable Wireless Sensor Products And Solutions

Midterm Report

Submitted to Dr. Maher Elshakankiri

Department of Computer Science

CS 890DH - Topics on Communications

Spring/Summer 2020

By

Tanu Nanda Prabhu - 200409072 - tnb735@uregina.ca

Regina, Saskatchewan

June, 2020

Abstract

Wearable wireless body sensor networks are by far the most striking and trending domains in Computer Science. Especially one of its applications such as health care is the most legatee of the Internet of Things (IoT) and Wearable Computing. In this research project, a detailed comparative study on various wearable wireless sensor products and networks specifically on health care would be conducted. As we all know, the present era comprises many technological vendors, products, and solutions that are very prominent in healthcare. So out of many, the three top providers of wearable wireless sensors as Libelium, Shimmer, and OCF (IoTivity) framework along with LoRaWAN protocol shall be chosen for the study. Overall the detailed comparison between the implemented applications or models and the ones mentioned in the research will be thoroughly explored as the result.

Introduction

Health Care

Universally, the elderly populace is growing, and the general populace is getting older [1]. Because of the ever-increasing population, unknown diseases are putting more pressure on the health care systems. Infectious, deficiency, hereditary, and physiological, and the recently emerged virus (coronavirus) that has spread and killed thousands of people are just some diseases that have caused burdens for medical workers. Not only that, but a great extent of people also suffers from chronic illnesses. The Center for Disease Control and Prevention (CDC), says that every "six in ten grown-ups in the United States have a chronic disease and four in ten grown persons have two or more chronic diseases" [2].

Wireless Sensor Network

The wireless sensor network can be characterized as a network of devices (sensors) that transmit the data (information) remotely [3]. To be more specific, the data here can be a sample of humidity, temperature readings, blood and glucose levels, etc. [4]. The information is sent through various remote sensors called hubs employing a gateway [4]. A sensor node can navigate autonomously or heavily influenced by human [5]. Figure 1 depicts the architecture of WSN model [5].

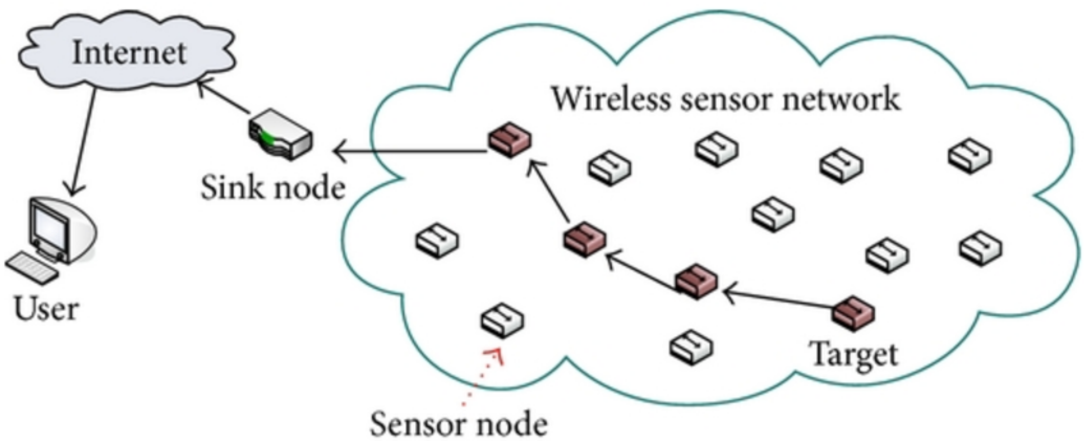


Figure 1: WSN architecture [5]

A wireless sensor network was primarily used in military applications but was later extended to health care applications. Some system models, for example, finger-worn ultrasound test, virtual sign checking in emergency clinics, indoor and versatile maturing have demonstrated the capability of wireless sensor networks to empower early recognition of clinical disintegration through constant patient observation in medical clinics [6]. With the help of this innovation, individuals need not spend a ton of cash on medical services. They can monitor and improve their health status via the applications embedded with wireless sensors.

Internet of Things

A critical number of physical things are being related with the Internet at a phenomenal rate understanding the chance of the Internet of Things [7]. The IoT makes the devices or the objects intuitive and shares information. In the present era, every device is connected to the internet. This includes cellphones, coffee makers, washing machines, headphones, etc. In simple words, we can define IoT as connecting all various devices or gadgets that relate with all characteristics of many livings to the internet, singular for transference, prepotency, and relaying of information towards making most productive decisions. Figure 2 shows the architecture of IoT.

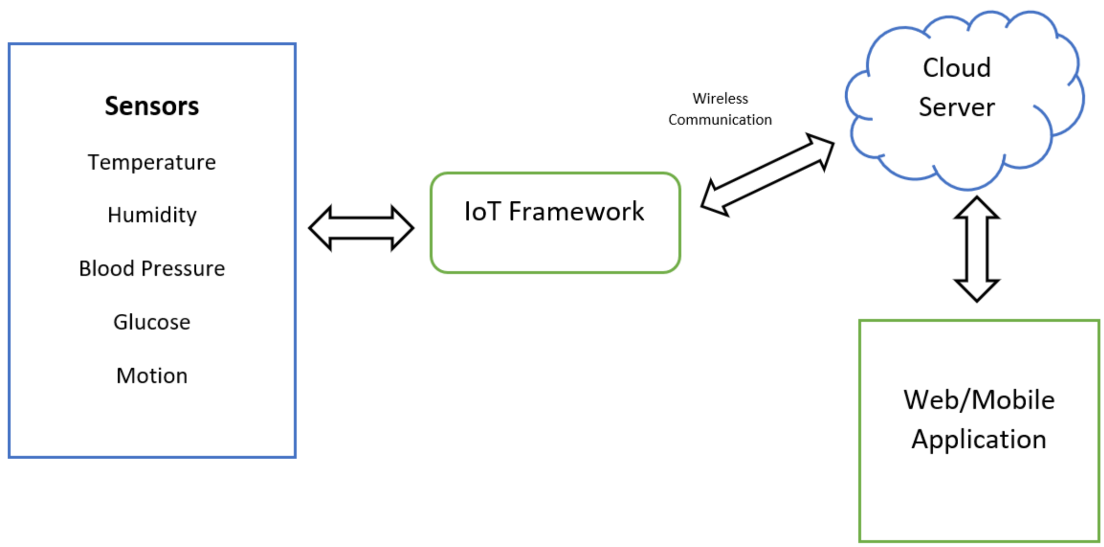


Figure 2: IoT architecture

Open Connectivity Foundation

Open Connectivity Foundation (OCF) is an open-source foundation devoted to guaranteeing a standard secure communication platform for customers, businesses, industries [8]. With the help of its open-source certification and implementation program, any devices irrespective of form factor, operating system, service provider the OCF characterizes a typical correspondence the structure that associates and astutely deals with the progression of data among gadgets to address the rising needs of the IoT [8]. Figure 3 is the building block of the success of OCF. The OCF was founded in the year 2004 and has over 400 members altogether [9]. OCF has 5 different membership lists which include Diamond Members comprises CISCO, Electrolux, Haier, Qualcomm, etc. [9]. Platinum members include Intel, Shaw, Mediacom, etc. [9]. Gold members include Lenovo, L&T Technology Services, Netgear, etc. [9]. Non-profit educational members include Kookmin University, KAIST, Seoul National University, etc. [9]. And the last but not the least basic member which include MediaTek, Microsoft, B&W Group Ltd, Nokia Networks, etc. [9]. When compared to demographics OCF is spread throughout the world, 54% in Asia, 30% in North America, 14% in Europe, 1% both in South America and the Middle East [9].

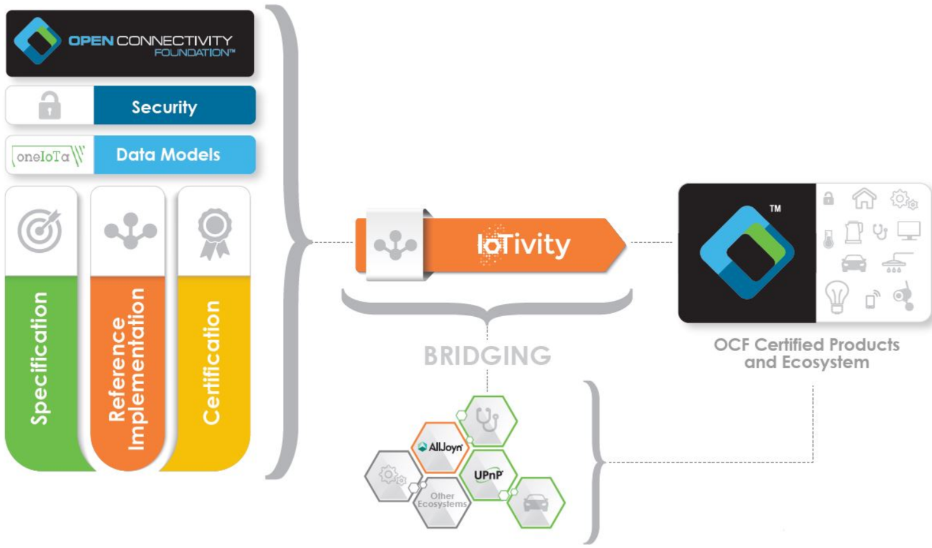


Figure 3: Building blocks of OCF [9]

OCF provides an online tool called "OneIoTa" to support the plan of interoperable gadget data models for the Internet of Things [8]. There are different data models such as Heartbeat, Blood pressure, Glucose level, etc available in the OneIoTa repository. All these data model is freely available and is in the form of JSON files. Below is the JSON file for the Heartbeat [8].


```json
{
  "swagger": "2.0",
  "info": {
    "title": "Heart Rate",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LICENSE.md",
      "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/HeartRateResURI": {
      "get": {
        "description": "This Resource describes the Properties associated with a person's heart rate.\n The unit, which is the default unit, is bpm.\n The
            heartrate Property is a read-only value that is provided by the server.\n When range (from \"oic.r.baseresource\") is omitted the default is 0
            to +MAXFLOAT.",
        "parameters": [
          {
            "$ref": "#/parameters/interface"
          }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt": [
                "oic.r.heartrate"
              ],
              "heartrate": 80
            },
            "schema": {
              "$ref": "#/definitions/HeartRate"
            }
          }
        }
      }
    }
```


Figure 4: JSON file data model of heart beat [8]

This permits new gadgets to be immediately manufactured using existing gadgets where conceivable and ensures interoperability between gadgets that utilizes other information models as of now in the database.

IoTivity

IoTivity is also an open source framework that actualizes the Open Connectivity Foundation guidelines giving simple and secure communications to IoT devices. Each day more and more web-based are coming online, adding to the ever-developing Internet of Things (IoT) [10]. Because of this security and reliability play a prominent role. The IoTivity venture was made to unite the open-source network to quicken the improvement of the system and administrations required to associate these billions of gadgets [8]. Below-shown Figure 5 is the architecture of IoTivity [10].

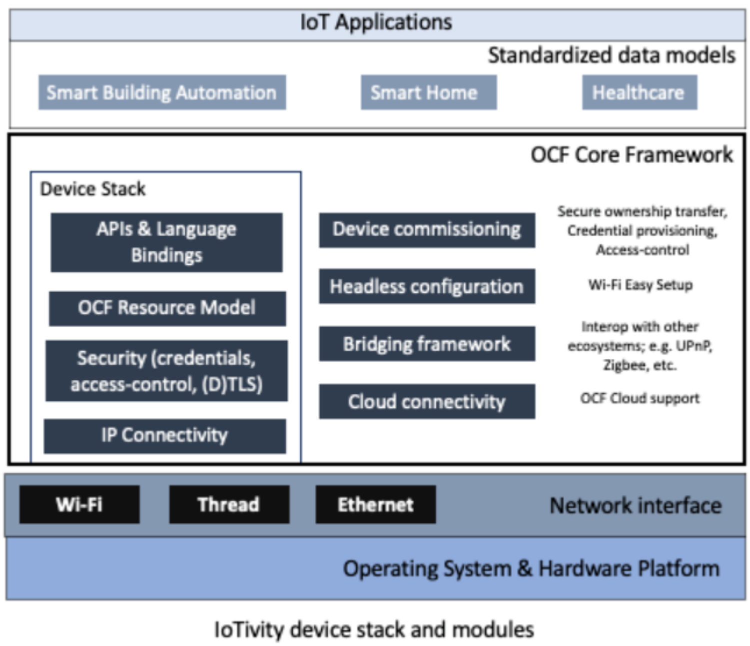


Figure 5: Architecture of IoTivity [10]

Libelium

Libelium is the producer of remote sensors for different applications, for example, brilliant homes, urban communities, medical care, etc [11]. It was founded in the year 2006 and is headquartered in Europe, with over 11-50 employees actively working. Alicia Asin, David Gascon are the prime founders of Libelium. There are several products such as waspmote, plug sense, smart parking, meshlium, Mysignals, mobile phone scanner. Below shown Figure 6 is the ecosystem of Libelium [12].

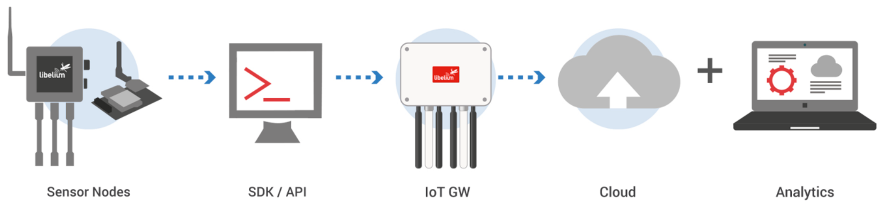
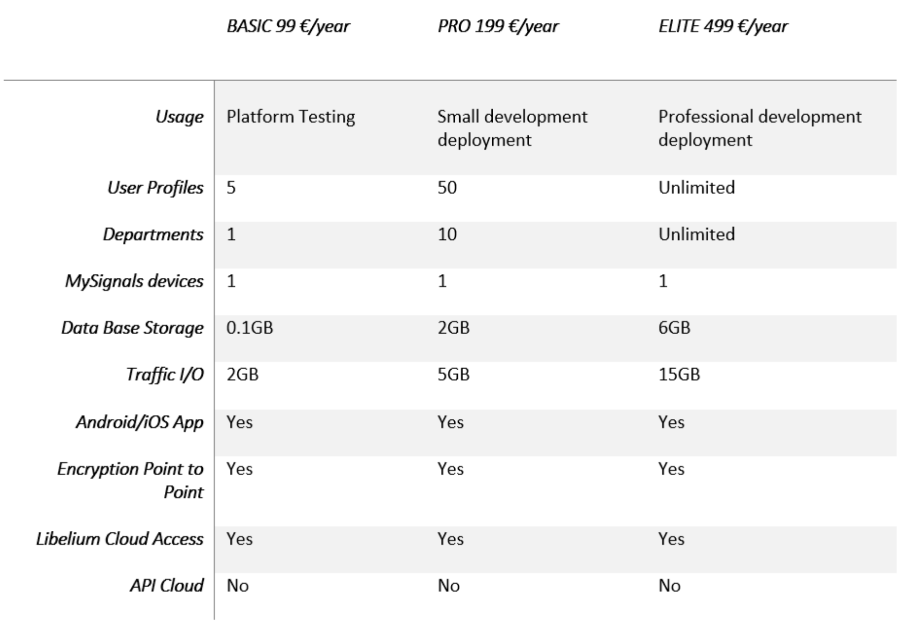


Figure 6: Ecosystem of Libelium [12]

The cloud partners of Libelium include Ericsson, Amazon web services, NEC, IBM cloud, etc. Similarly, its solution partners include Basen, eagle.io, Tata communications, etc. And hardware partners include Multitech, apogee instruments, GILL, etc [12]. Libelium MySignals is an improvement stage for clinical gadgets and e-wellbeing applicationss. MySignals measures are over 15 different biometric parameters. Since it's open-source, we can add our sensors to build new devices. Some sensors include Temperature, Glucometer, Blood pressure, body pressure, etc. With the help of its cloud services, the data gathered by my signals sensors would be encrypted and sent to the owner's account. Data can be sent via Wifi or Bluetooth to the cloud for storage [12].

User Cloud Plans



	BASIC 99 €/year	PRO 199 €/year	ELITE 499 €/year
Usage	Platform Testing	Small development deployment	Professional development deployment
User Profiles	5	50	Unlimited
Departments	1	10	Unlimited
MySignals devices	1	1	1
Data Base Storage	0.1GB	2GB	6GB
Traffic I/O	2GB	5GB	15GB
Android/iOS App	Yes	Yes	Yes
Encryption Point to Point	Yes	Yes	Yes
Libelium Cloud Access	Yes	Yes	Yes
API Cloud	No	No	No

Figure 7: MySignals cloud plans [13]

Shimmer

Shimmer is a provider of wearable wireless sensor products and solutions. It was founded in the year 2008 and is headquartered in Europe. It was founded by Paddy White. Serves around 11-50 employees [14]. Some of their successful kits are Consensys Bundle Development Kit, Shimmer3 EBio Consensys Development Kit, Verisense Activity and Sleep Kit, etc [15]. Shimmer has developed, in collaboration with partners such as QTUG, Kinesis, NeuroLynQ, etc. [14].

Related Work

H. Cha and J. Jeon et. al [8] implemented the blood pressure monitor using OCF data model [8]. The main reason for using OCF is to extend its data models available from its 'oneIoTa' repository which is open source and free to download. Shown below is the data model specification table.

Data Model Name	**Blood Pressure**
Published in	RAML
Type	JSON
Format	Swagger
Compatible with	OCF Restful architecture

Figure 8: Data Model Specification table [8]

Below is the JSON format of Blood Pressure data model:

```
{
  "id":
"http://openinterconnect.org/iotdatamodels/schemas/oic.r.bloodpressure.json#",
  "$schema": "http://json-schema.org/draft-04/schema#",
  "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc.
All rights reserved.",
  "title": "Blood Pressure",
  "definitions": {
   "oic.r.bloodpressure": {
    "type": "object",
    "properties": {
     "systolic":   {
      "type": "number",
      "minimum": 0,
      "readOnly": true,
      "description": "Systolic blood pressure"
     },
     "diastolic":   {
      "type": "number",
      "minimum": 0,
      "readOnly": true,
      "description": "Diastolic blood pressure"
     },
     "map":   {
      "type": "number",
      "minimum": 0,
      "readOnly": true,
      "description": "Mean Arterial Pressure (MAP)"
     },
     "unit": {
      "type": "string",
      "readOnly": true,
      "enum": ["mmHg", "kPa"],
      "description": "Blood pressure unit"
     }
    }
   }
  },
  "type": "object",
  "allOf": [
   {"$ref": "oic.core.json#/definitions/oic.core"},
   {"$ref": "oic.baseResource.json#/definitions/oic.r.baseresource"},
   {"$ref": "#/definitions/oic.r.bloodpressure"}
  ],
  "required": ["systolic", "diastolic"]
}
```

Figure 9: JSON file data model for blood pressure [8]

The RAML API downloaded from the OCF repository is also as shown below:

```
schemas:
 - BloodPressure: !include oic.r.bloodpressure.json

traits:
 - interface:
   queryParameters:
    if:
     enum: ["oic.if.s","oic.if.baseline"]

/BloodPressureResURI:
 description: |
  This resource describes the properties associated with a person's
Blood Pressure.
  The unit is a single value that is one of mmHg or kPa.
  If the unit Property is missing the default is a millimeter of mercury
[mmHg].
  The BloodPressure and unit Properties are read-only values that are
provided by the server.
  When range (from oic.r.baseresource) is omitted the default is 0 to
+MAXFLOAT.

 displayName: Blood Pressure
 is: [ interface ]

 get:
  description: |
   Retrieves blood pressure of an object.
  responses:
   200:
    body:
     application/json:
      schema: BloodPressure
      example: |
       {
        "rt":     ["oic.r.bloodpressure"],
        "id":     "unique_example_id",
        "systolic": 110,
        "diastolic": 85,
        "unit": "mmHg"
       }
```

Figure 10: RAMPL API for blood pressure [8]

As seen in the above JSON code, blood pressure data has two important parameters systolic and diastolic with values 110 and 85, respectively. The unit we measure them in is mmHg or kPa. The value of 200 shows successful communication between the server and the client. IoTivity can be installed on Windows and Ubuntu and different platforms. Here IoTivity was used only to create a server that is primarily used for measuring transmitted health care data and client which is used to receive data from the server. Similarly, Libelium MySignals was used to develop a prototype with its available sensors such as blood pressure monitor, glucometer, body temperature, scale.

T.Amartuvshin et.al [16] implemented the human activity recognition using Shimmer 2A wireless sensor platform . Here the 2A platform comprises sensors, data storage, communication, connection capabilities. Show below is the specifications of the Shimmer 2A platform [16].

Platform Name	Shimmer 2A
Core	MSP430 MCU
Bluetooth	Yes
Accelerometer	iMEMS tri-axis MMA7260Q
Software Operating System	Tiny OS

Figure 11: Shimmer 2A platform specifications [16]

Block diagram of the SHIMMER baseboard is as shown below [16]:

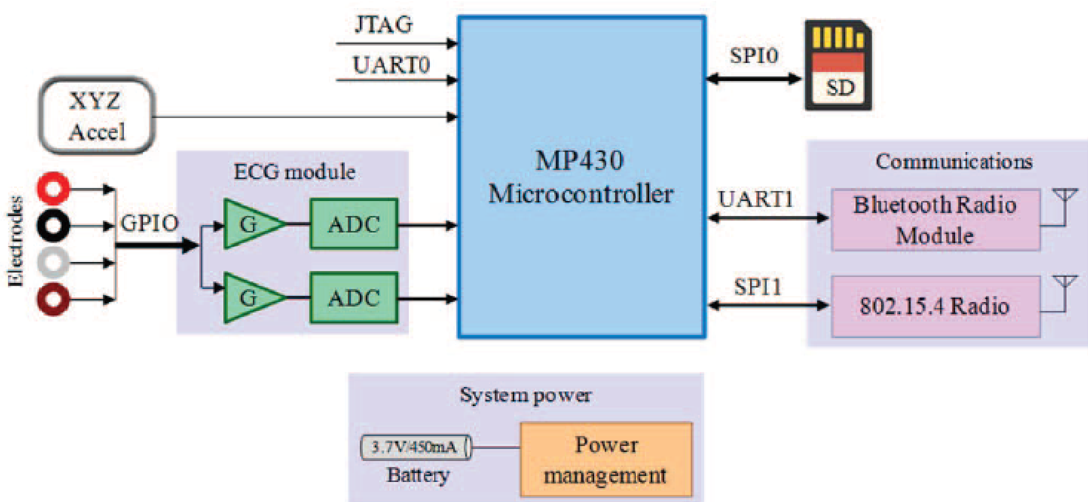


Figure 12: Shimmer 2A platform [16]

The only reason that the accelerometer is being used for measuring the patient's body movements. Similarly, the ECG (electrocardiogram) is primarily used for recording the heartbeats (activity).

Data Model

The sensor data can then be accessed on a mobile phone. With the help of this, the device can understand the movement of the patient's body such as running, walking, skipping as shown below [16]. Shimmer device in this case can monitor and record the ECG of the patient's activity, it keeps on monitoring continuously with high accuracy by allowing users to access the data on their wireless device.

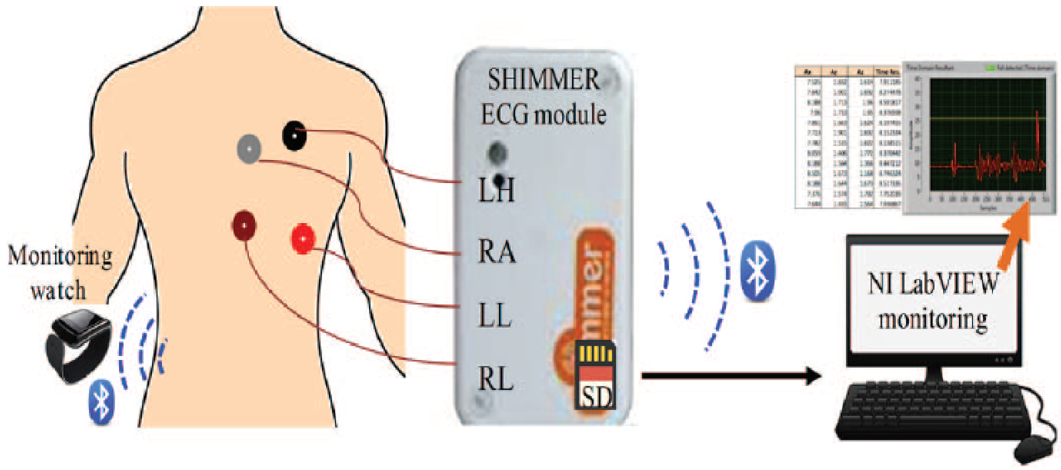


Figure 13: Block diagram of the data model [16]

Afef Mdhaffar et al [17]. implemented e-health monitoring system (along temperature, glucose monitor) in rural areas via LoRaWAN[17]. The low power wide area network (LoRaWAN), is a low-power cost, and enables secure communication between the sensors implanted to the patients and the dashboard or the User Interface at the doctor's side. The authors use IoT4HC (Health Care) architecture because they believe all the existing works use relatively high communication links, and the connection is not secure. Shown below is the architecture for IoT4HC [17].

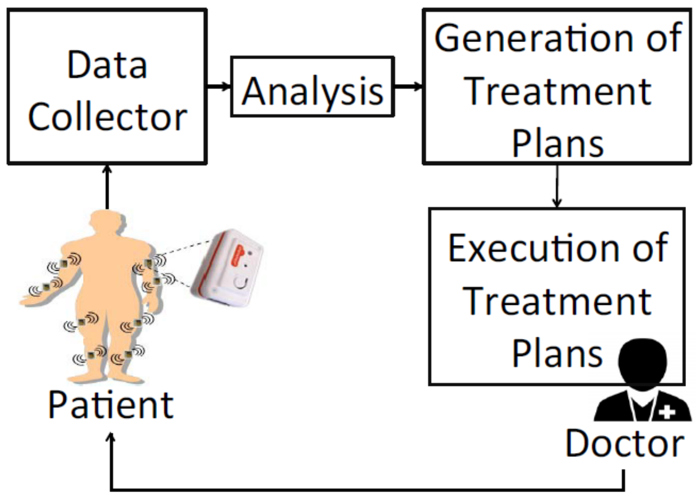


Figure 14: Architecture for IoT4HC [17]

The data collector collects the data from the implanted sensors on the patients. The analysis component gathers the data from the collector and generates diagnosis reports which then would be to provide generation treatment plans by the generation treatment plans component [17]. Last, the created reports are approved and executed by the specialists. The IoT4HC depends on the LoRaWAN convention: highly efficient communication protocol. The security of LoRaWAN is much better than the traditional GSM because it houses symmetric encryption based on AES IEE E802.15.4/2006 along with a 128-bit key. The below figure shows the based e-health monitoring framework [17].

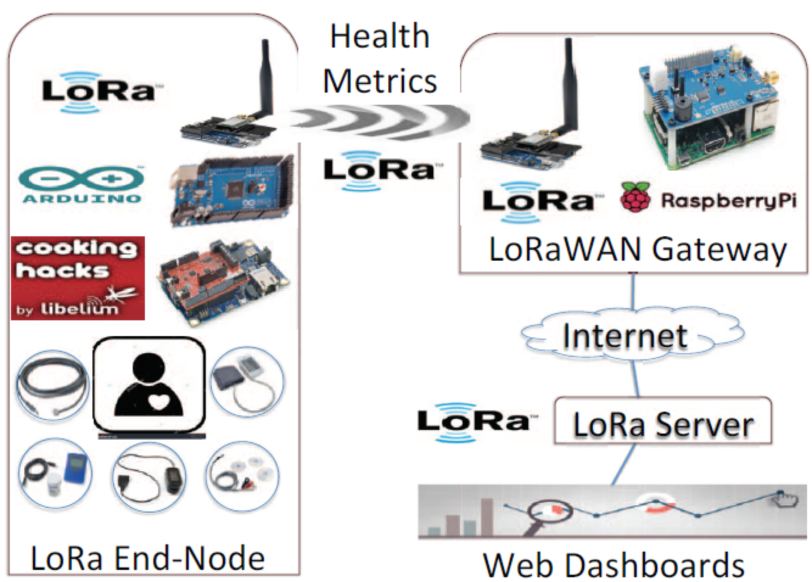


Figure 15: LoRaWAN based e-health monitoring system [17]

As seen in the above figure the LoRa end node comprises e-health sensors, a shield, along with Arduino Mega micro-controller, LoRa transceiver [17]. All the transceiver, sensors and the shield are designed by Libelium. Below is the comparison between LoRa and GPRS [17].

Proposed Solution

Comparison between OCF, IoTivity, Libelium MySignals, and Shimmer

Implementation of the blood pressure monitor using the OCF data model board with a blood pressure sensor. Blood pressure sensor is present in the MySignals Kit. Rather than comparing side by side, the comparison would be done by exploring the concepts, products thoroughly listing advantages, disadvantages, and the future scope. Firstly lets see the advantages of the above implementation.

- With the help of these frameworks (IoTivity and OCF), it could be easily extended and connected to other applications such as home applications (smart), transport applications or devices, etc. without drastic customization/transportation issue.
- The IoTivity was operated on OCF's standard data modeling, which is an official OCF reference specification implementation.
- Since due to the open-source nature of the OCF repository, any available data model can be easily edited and downloaded.

Below shown are the disadvantages of the above mentioned implementation:

- **Platform Compatibility issue**: Arduino Micro-controller is not supported by IoTivity. Arduino is utilized as the primary stage for Libelium MySignals.
- **External Shield**: An alternate option can be installing it to another operating system such as Linux, Ubuntu, etc. and then transport it to Arduino. For this successful porting an external shield should be used such as Raspberry Pi to Arduino shield.
- **Voltage level difference**: Arduino UNO and the Raspberry Pi 3 operate on two seperate voltages 5 and 3.3 V [8]. Hence to maintain the balance between them 3.3-5V bidirectional level shifter was used. The primary motivation behind utilizing a bidirectional level shifter was to arrange the voltage level contrast [8].

- **Bluetooth connectivity issue**: Also, Bluetooth (BLE) connectivity was also not supported, but since the code was open source, we could easily modify the OCF standards and make it compatible.

- **Selective operation**:Additionally, Bluetooth (BLE) availability was likewise not upheld by OCF, but only devices with three different protocols such as constrained application protocol, user datagram ptotocol, internet protocol operation only could be authenticated and certified.

As a future extension, an alternate number of OCF applications independent of areas can be worked in IoTivity. Because of the missing BLE feature, an alternate option would be using SHIMMER 2A or any higher versions, because the base version of SHIMMER provides BLE connectivity, and it is powered by MSP430 MCU which is way more powerful than ATMEGA 328.

Using shimmer 2A platform to implement human activity recognition [16]. Firstly lets see the advantages of using Shimmer 2A platform of the above implementation:

- **Direct-To-Host data transfer**: With the assistance of this, we can without much of a stretch exchange the information from Shimmer's smaller scale SD (Secure Digital) card to unload.

- **Expansion connectors**: The connectors of the shimmer board expands which allows all the baseboard hardware versions to work with internal and external sensors.

- **Tiny OS**: With the help of this software the application can be migrated via a compile time flag.

- **Wireless Data Streaming**: As mentioned earlier, the Shimmer 2A base board is equipped with Bluetooth module named RN-32, which can be connected to the MCU MSP430. This can be possible with the help of serial connection. [16]

The baseboard of the Shimmer 2A provides three different types of sensing capabilities such as kinematic, physiological and ambient. There are three channels of ADC which are used to connect to the micro control unit. The ECG provides a low power auto calibrating CMOS amplifier.

As a future scope, rather than performing a multiple test, the base data for the testing can be extended by using one of the data models from the OCF repository. Also, an extended feature such as an alert notification system can also be added, in case if the patient falls down then an automatic call or message should be given to the caretakers. Also to store all the data the Libelium Mysignal cloud environment can be used since it is one of the cheapest and provides lots of features compared to other cloud environments.

Implementation of health monitoring via LoRaWAN (IoT based). The advantages of implementing LoRaWAN rather than GPRS are:

- **More Range**: The total area covered by LoRaWAN is around 33 square kilometer, when it is placed on a altitude of 12 meters.
- **Power consumption efficient**: The power consumption of LoRaWAN is atleast 10times lesser than traditional GPRS 3G/4G. The average power consumption of LoRaWAN is 148.6 mA where as the GPRS is around 2000 mA.
- **Low installation and Infrastructure cost**: The cost of the LoRaWAN is very less compared to the GPRS. The infrastructure cost of LoRaWAN is 10 times cheaper than GPRS. Also LoRaWAN provides secure communication.
- **Low Idle mode**: The idle mode of IoT4HC is 4.7 mA, where as the idle mode of the GSM is around 20 mA. Shown below is the comparision of both LoRaWAN and GPRS.

Comparision between LoRa and GPRS

	Average Range	Energy Consumption (mA)	Minimum Hardware Cost	Maximum Data Rate
LoRa	60 km	Idle mode: 2.8 mA Continuous receive mode: 14.2 mA Send mode 38.9 mA	Transceiver: 10 $ Gateway: 250 $	50 kbps (downlink) 50 kbps (uplink)
GPRS	60 km	Idle mode: 20 mA Continuous receive mode: 130 mA Send mode 2000 mA	Transceiver: 50 $ Gateway: 10000 $	85.6 kbps (downlink) 14 kbps (uplink)

Figure 16: LoRa versus GPRS comparision [17]

Comparision between IoT4HC and GPRS

	IoT4HC using LoRaWAN protocol	**GPRS (General Packet Radio Service)**
Power Consumption	**148.6 mA**	**~2000 mA**
Current Consumption (Idle Mode)	**4.7 mA**	**~20mA**

Figure 17: IoT4HC versus GPRS comparision [17]

The disadvantages of IoT4HC is as shown below:

- **Small frame size**: Collection of data would be represented as a small frame size of 256 bytes. And the need to send frequently this may overflow the frame.

- **Continuous data**: The IoT4HC cannot send continuous information (medical). The continuous medical information in this case would be ECG, EEG

As a future scope, rather than using a LoRaWAN module for sending continuous medical data, we can use GPRS such as 4G, which is expensive, but this can send continuous data. We must also implement the IoT4HC must also in urban areas because not all people are rich and the internet connectivity is good in urban areas. A dashboard must be enabled because often patients would like to keep updated about their health, and they can monitor it regularly rather than an experienced person look over it.

Conclusion

This paper has presented a comparative study or review of wearable wireless sensor products and networks specifically on health care. The different body sensor network providers were Libelium MySignals, Shimmer the frameworks such as IoTivity, OCF, and the protocols such as LoRaWAN were studied in depth. Since this is a midterm report, only half of the comparison would be done, but the other half in-depth and more comparison would be provided in the final report. However, this is a detailed comparison between implemented and researched applications so there are no best and worst answers to choosing different frameworks, providers, or protocols. But as a whole for choosing the platform if the application is implementing a health monitoring system, then according to the study conducted the Shimmer 2A platform performs better than Libelium MySignal Arduino Uno. Similarly, for implementing the protocol resulting in low cost, secure then LoRaWAN is the best option compared to GPRS. Also to access the data model, the OCF repository is the only and the best option available. Also, in the appendix section, a small snippet of python code is provided to convert the JSON code (data model) to CSV (Comma Separated Value) format.

Appendix

The JSON file can be found here. It is open source and freely downloadable.

```python

from . import json
from . import import pandas as pd

# Loading the json file from the system
data = json.load(open('/content/data.json'))

# Converting the json file to the data frame
dataframe = pd.DataFrame.from_dict(data, orient="index")

# Saving the generated data frame to the CSV file
dataframe.to_csv('datamodel.csv')
```

Code Listing 1: Converting the json data model (Heart rate) into a CSV file

On executing this program the JSON file would be automatically converted to the a CSV file as shown below:

	0
swagger	2.0
info	{'title': 'Heart Rate', 'version': '2019-03-04', 'license': {'name': 'OCF Data Model License', 'url': 'https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LICENSE.md', 'x-copyright': 'Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved.'}, 'termsOfService': 'https://openconnectivityfoundation.github.io/core/DISCLAIMER.md'}
schemes	['http']
consumes	['application/json']
produces	['application/json']
paths	{'/HeartRateResURI': {'get': {'description': 'This Resource describes the Properties associated with a person\'s heart rate.\n The unit, which is the default unit, is bpm.\n The heartrate Property is a read-only value that is provided by the server.\n When range (from "oic.r.baseresource") is omitted the default is 0 to +MAXFLOAT.', 'parameters': [{'$ref': '#/parameters/interface'}], 'responses': {'200': {'description': '', 'x-example': {'rt': ['oic.r.heartrate'], 'heartrate': 80}, 'schema': {'$ref': '#/definitions/HeartRate'}}}}}}
parameters	{'interface': {'in': 'query', 'name': 'if', 'type': 'string', 'enum': ['oic.if.s', 'oic.if.baseline']}}
definitions	{'HeartRate': {'properties': {'heartrate': {'description': 'This Property describes the heart rate in bpm.', 'type': 'integer', 'minimum': 0, 'readOnly': True}, 'rt': {'description': 'The Resource Type.', 'items': {'enum': ['oic.r.heartrate'], 'type': 'string'}, 'minItems': 1, 'uniqueItems': True, 'readOnly': True, 'type': 'array'}, 'n': {'$ref': 'https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-schema.json#/definitions/n'}, 'if': {'description': 'The OCF Interface set supported by this Resource.', 'items': {'enum': ['oic.if.s', 'oic.if.baseline'], 'type': 'string'}, 'minItems': 1, 'uniqueItems': True, 'readOnly': True, 'type': 'array'}, 'range': {'$ref': 'https://openconnectivityfoundation.github.io/IoTDataModels/schemas/oic.baseresource.properties-

Figure 18: CSV file of the data model

References

[1] "Ageing", Un.org. [Online]. Available: https://www.un.org/en/sections/issues-depth/ageing/. [Accessed: 28- Jun- 2020].

[2] "Chronic Diseases in America — CDC", Cdc.gov. [Online].
Available: https://www.cdc.gov/chronicdisease/resources/infographic/chronic-diseases.htm. [Accessed: 28- Jun- 2020].

[3] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow and P. Polakos, "Wireless sensor network virtualization: A survey," in IEEE Communications Surveys Tutorials, vol. 18, no. 1, pp. 553-576, Firstquarter 2016, doi: 10.1109/COMST.2015.2412971.

[4] "Introduction to Wireless Sensor Networks Types and Applications". ElProCus - Electronic Projects for Engineering Students, from https://www.elprocus.com/introduction-to-wireless-sensornetworks-types-and-applications/ [Accessed: 28- Jun- 2020]

[5] Ryu, Ji Irfan, Muhammad Reyaz, Aamir. (2015). "A Review on Sensor Network Issues and Robotics. Journal of Sensors". 2015. 1-14. 10.1155/2015/140217.

[6] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis and M. Welsh, "Wireless Sensor Networks for Healthcare," in Proceedings of the IEEE, vol. 98, no. 11, pp. 1947-1960, Nov. 2015, doi: 10.1109/JPROC.2015.2065210.

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," in IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2015, doi: 10.1109/COMST.2015.2444095.

[8] H. Cha and J. Jeon, "OCF Healthcare Proof of Concept (PoC) on Libelium MySignals," 2017 European Conference on Electrical Engineering and Computer Science (EECS), Bern, 2017, pp. 356-364, doi: 10.1109/EECS.2017.73.

[9] "OCF Membership List", Open Connectivity Foundation (OCF). [Online]. Available:
https://openconnectivity.org/foundation/membership-list/ [Accessed: 28- Jun- 2020].

[10] "IoTivity Architecture — IoTivity", Iotivity.org. [Online]. Available: https://iotivity.org/about/iotivity-architecture. [Accessed: 28- Jun- 2020].

[11] "Products — Libelium", Libelium.com. [Online]. Available: http://www.libelium.com/products/. [Accessed: 28- Jun- 2020].

[12] "Ecosystem", Libelium.com. [Online]. Available: http://www.libelium.com/partners-ecosystem/.

[Accessed: 28- Jun- 2020].

[13] "MySignals Cloud — Libelium", Libelium.com. [Online]. Available: http://www.libelium.com/cloud-services/mysignals-cloud/. [Accessed: 28- Jun- 2020].

[14] "Customers — Shimmer Enterprise Customers", Shimmersensing.com. [Online]. Available: http://www.shimmersensing.com/services/customers/. [Accessed: 28- Jun- 2020].

[15] A. Burns et al., "SHIMMER™: An extensible platform for physiological signal capture," 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, Buenos Aires, 2010, pp. 3759-3762, doi: 10.1109/IEMBS.2010.5627535.

[16] T. Amartuvshin, D. Erdenechimeg, N. Chuluunbaatar, T. Enkhbaatar and D. Enkhzul, "Preprocessing techniques for daily healthcare device using SHIMMER™ wireless sensor platform," 2016 11th International Forum on Strategic Technology (IFOST), Novosibirsk, 2016, pp. 558-562, doi: 10.1109/IFOST.2016.7884179.

[17] A. Mdhaffar, T. Chaari, K. Larbi, M. Jmaiel and B. Freisleben, "IoT-based health monitoring via LoRaWAN," IEEE EUROCON 2017 -17th International Conference on Smart Technologies, Ohrid, 2017, pp. 519-524, doi: 10.1109/EUROCON.2017.8011165.