

2019

The deployment of an IoT network infrastructure, as a localised regional service

John Fox

Andrew Donnellan

Liam Doumen

Follow this and additional works at: <https://arrow.tudublin.ie/engscheleart>



Part of the [Electrical and Computer Engineering Commons](#)

This Conference Paper is brought to you for free and open access by the School of Electrical and Electronic Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, gerard.connolly@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 4.0 License](#)

The deployment of an IoT network infrastructure, as a localised regional service

John Fox

Technological University Dublin
Department of Electronic Engineering
 Dublin, Ireland
 John.Fox@it-tallaght.ie

Dr. Andrew Donnellan

Technological University Dublin
Department of Electronic Engineering
 Dublin, Ireland
 Andrew.Donnellan@it-tallaght.ie

Liam Doumen

Technological University Dublin
Department of Electronic Engineering
 Dublin, Ireland

Abstract— The Internet of things (IoT) is fast evolving with a wide range of technologies being designated specifically as IoT solutions. Studies on such solutions generally reference the specific communication medium while negating the complete architecture of the IoT system. From a system perspective, a complete IoT solution can be separated into three categories, the data collector, the communication method and the cloud platform service. The data collector relates to the embedded system device (or the ‘things’ element) at the source of the application, the communication method relates to the network protocol used to send or receive the data and the cloud platform service relates to the facility used to store and process the data collected. LoRaWan and LoRa are ‘Long Range’ technologies, which define the communication method for such IoT applied systems. LoRa defines the modulation technique, that allows for long range communication, whereas LoRaWan defines the communication and system architecture.

This paper presents the design architecture and methodology of a fully functioning LoRaWan based IoT system. Such a system can be provided as a service to a given local region, by utilizing an End Device in conjunction with a LoRa transceiver, a LoRaWan Gateway and a defined cloud platform. The presented IoT system currently serves the region of Tallaght (Dublin, Ireland) and its wider area. As a service, the system has been shown to be capable of supporting a wide range of IoT based applications.

Keywords— *Internet of Things (IoT), IBM Cloud, Node Red, LoRa, LoRaWan*

I. INTRODUCTION

The “Internet of things” (IoT) has become an emerging topic over the last number of years. It can be described as, the connection of devices or ‘things’ across the internet to deliver upon an assigned function. This mainly comes in the form of devices sending or receiving data. From a system overview, it involves the interconnection of embedded system type devices to a given cloud platform service(s), with communication completed through specific network protocols and interconnections i.e.

- A method for data collection at the source of the application - the embedded system device (or the ‘things’ element).
- A communication means to send or receive the data - the network protocol.

- A facility to store and process the information gathered - the cloud platform service.

In the context of wide area networks, the ability to provide connectivity for distribution across large geographical areas must support the balance of the delivery of long range communication, in tandem, with the longevity of the battery supported End Device. This balance can be seen in the technologies emerging under the umbrella of Low Powered Wide Area Networks. Such technologies aim to provide long-range communication distances at low power. Typically, sacrifices to achieve long range and low power affects data rate availability, latency and reliability. For LPWAN’s implemented to support an IoT system, this becomes a trade-off based on requirement. Not all IoT application requirements can be met if data rate, latency and reliability are critical factors. However, depending on the LPWAN deployed, such technologies can still meet the requirements of a vast range of IoT applications. Applications that only need to exchange a small amount of data can communicate at infrequent times and do not need, for example, high response times. Research applied to LoRaWan over time has mainly focused on its operation and performance as a LPWAN technology with fewer specific application deployments [1]–[5] in the complete system context. In this paper, an overview of the development of an IoT network infrastructure for use as a localised regional service is presented. The paper discusses the implementation of the proposed three core elements to create an IoT infrastructure as a complete system. An Arduino is used to form the basis of the ‘things’ element, LoRa/LoRaWan forms the communication method and IBM’s IoT cloud platform is implemented as the cloud entity. A design architecture and methodology is presented, which has been applied to a real application deployment, related to a 6kW wind turbine.

II. BASICS OF LORA & LORAWAN

LoRa is a proprietary spread spectrum modulation technique developed by Semtech[6]. The modulation technique is a derivative of Chirp Spread Spectrum. It utilizes 6 x orthogonal spreading factors and allows trading of data rate for range or power, within a consistent bandwidth (typically 125kHz). This is achieved across a number of defined channels, within the 868MHz ISM band. Through regulation, a duty cycle limitation (1%) is applied as the method of channel access and collision avoidance. If an End Device transmits on one

particular channel, the time of transmission and a time on air value is recorded to ensure the channel cannot be re-used for a period of time. Possible signal strengths exceeding 150dB enable LoRa to achieve long-range communication. As LoRa defines the modulation technique, which allows for long range communication, the communication and system architecture is defined by LoRaWan. Using a star configuration, a LoRaWan system includes components such as End Devices, Gateway(s), a Network Server and an Application Server. The Gateway acts as the intermediary between End Device and a central Network Server. The Network Server has a number of functions including controlling communication and providing security access of End Devices. The Application Server can be used to group End Devices and their messages as related to particular applications. Bi-directional communication is available, however, message acknowledgement can be disabled to improve performance. Payload sizes range up to 250 bytes. Using LoRa modulation, data rates range from 0.3-37.5 kbps [7]–[9].

III. MESSAGE QUEUE TELEMETRY TRANSPORT (MQTT)

In order to transfer the data messages to a storage centre, an MQTT broker is required. MQTT is a lightweight network protocol that has been adopted as an IoT connectivity protocol. Developed by IBM in the late 1990's, it is implemented on top of the TCP/IP protocol stack and is suited towards high latency and limited bandwidth network devices. MQTT operates on a publish and subscribe model, where message 'topics' are used to categorized the MQTT messages. The topic can be specific to the type of data sent by an End Device e.g. an End Device could be publishing data to a topic called "Temperature". The protocol requires a MQTT broker in order to satisfy the publish and subscribe model. Any entity that can send or receive data to the MQTT broker is known as a client. Clients can subscribe to a message "topic" in order to receive the messages published by another client [10], [11].

IV. IMPLEMENTATION

A. End Device

Overall, there are two options for the End Device choice, off shelf or modular in type. For this study, in order to support the concept of delivering an IoT system as a service, a modular type End Device was chosen. This End Device choice was based on the following desired attributes:

- Simplicity of use, in terms of the development environment.
- Versatility, in terms of application deployment.
- Reproducibility, in the case of damage or failure.
- Small device profile, for ease of deployment.
- Support for LoRaWan, within the software.

Across both hardware and software, the Arduino Pro Mini proved to fit as the microcontroller of choice. LoRa modulation was achieved using 'RF solutions', LoRa transceiver radio module [12]. An Arduino version of the IBM LMIC library for

LoRaWan is widely available [13] and has been applied in this research project. Fig. 1 illustrates the End Device produced on PCB. The modular design of the End Device allows for ease of reproducibility, where the Arduino and its development environment, caters towards simplicity of use and flexibility in functionality.

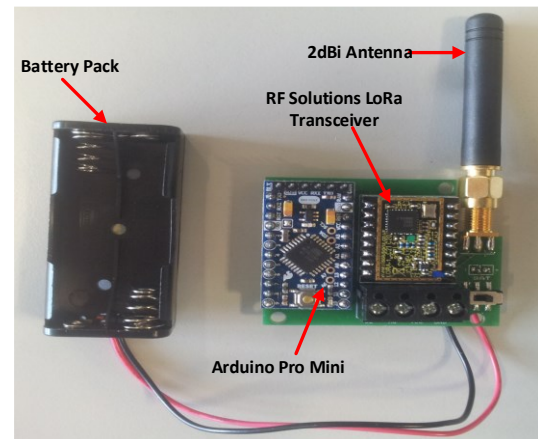


Fig. 1. LoRaWan End Device

B. Gateway

The Gateway deployed was the Multiconnect Conduit IP67 Base Station Model: MTCDTIP-LEU1-220L-868-OP. A 3dBi dipole antenna was connected to the Gateway. Transmit power of the Gateway was set to 26dBm. Fig. 2 illustrates the Gateway unit and antenna presently installed on the campus roof of Technological University Dublin – Tallaght Campus, approximately 19 metres from the ground.



Fig. 2. LoRaWan Gateway installation

The Gateway was also configured in Network Server mode, meaning the unit acts as a Gateway and Network Server. The Gateway uses Node Red as its programmable connection interface.

C. IBM IoT Cloud Platform

In order to pass data from the LoRaWAN Network into the IBM cloud platform, an MQTT broker is required. Within IBM's IoT cloud platform, the MQTT broker is already established. In addition, there are some intermediates services required to complete the connection between the End Device and the cloud entity. The following IBM cloud services were required, as part of the design architecture:

1) *Internet of Things Platform*: This provides access for registered devices/gateways to the cloud services. It allows for End Device and Gateway management, monitoring of End Device real time data, monitoring of events/alerts and it also provides secure connectivity within the IoT system.

2) *Cloudant*: This is a NoSQL JSON document store that is optimized for handling heavy workloads of concurrent reads and writes in the cloud. It is an always on service and caters towards fast growing workloads. It is a fully managed, Database as a Service (DBaaS), running on IBM cloud as a NoSQL type database.

3) *Db2 Warehouse on Cloud*: This is a fully managed SQL database service in the cloud. It is optimised for analytical workloads catered towards high performance processing of scale.

Fig. 4 represents the design architecture and methodology of the full IoT system. In order to interconnect the LoRa Gateway to IBM cloud and its services, the IoT cloud platform service must be created to which the LoRa Gateway is registered. A Node Red Cloud Foundry application instance is also created. Cloud Foundry is an open source platform as a service(PaaS), it allows for quick creation and deployments of applications on the cloud. It is not vendor specific and does not restrict the developer into proprietary software or cloud infrastructure. In this case, a Node Red Cloud Foundry application is created which is linked and connected to the IBM IoT platform service. Data is passed through this IoT platform service, from the Gateway and subsequently directed to the connected Cloud Foundry application. Within the Node Red Cloud Foundry application, a flow program is developed to push received data to IBM's Cloudant NoSql DB service. Cloudant allows for a direct connection to the Db2 Warehouse service for analysis. Data visualisations can be created where instantaneous data values logged can be presented in web form. Graphs, charts, gauges etc. can be applied to give window views of data-sensing behaviour in a real-time processing/analysis context. Node Red flows can be further developed to support:

- The creation of alerts as applied to incoming data.

- Mathematical functional pre-processing of incoming data, prior to database insertion.
- Data filtering for removal of non-pertinent data that consume valuable cloud storage.
- Data stream comparison between End Devices.
- Event monitoring.
- Application of automation triggers.

V. APPLICATION DEPLOYMENT: WIND TURBINE

A 6KW downwind self-regulating wind turbine resides within the Tallaght campus of Technological University Dublin. Within the vicinity of the wind turbine, a 15m meteorological mast has been installed for the purpose of wind speed and wind direction data collection.

The design architecture and methodology for the IoT system as described in Fig. 4 was applied to three designated application metrics located at the meteorological mast i.e. wind speed at 10m, wind speed at 15m and temperature at 2.5m. As noted in Fig. 4, Node Red as a programming interface applies at two junctures, at the LoRa Gateway and at the IBM Cloud platform.

A. LoRa Gateway: Node Red configuration

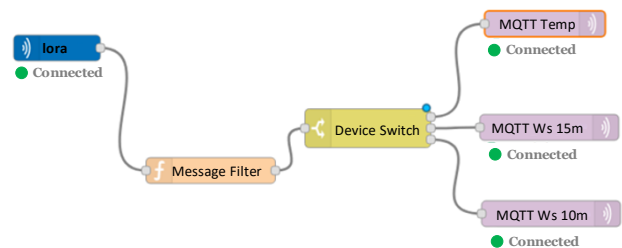


Fig. 3. LoRa Gateway: Node Red flow

At the LoRa Gateway, a Node Red program (Fig. 3); filters the message payload to remove unnecessary information, identifies and isolates individual End Device messages and finally directs the data to the IBM Internet of Things platform service.

B. IBM Cloud: Node Red Cloud Foundry application

Through the Node Red Cloud Foundry application we can apply: pre-processing to data messages, define alerts and events rules, filter messages and data values, assign data values to specific databases and create data visualisations on real time data. Fig. 5 illustrates an example of the Node Red program flow for the Cloud Foundry application deployed. The program flow inserts incoming messages, for the wind speed sensor at 10m into the Cloudant database (thus completing the path from End Device to the cloud platform). The program also provides a real-time visualisation view on the incoming data. The program flow can be directly replicated to accommodate data collection for the temperature sensor and the wind speed sensor at 15m.

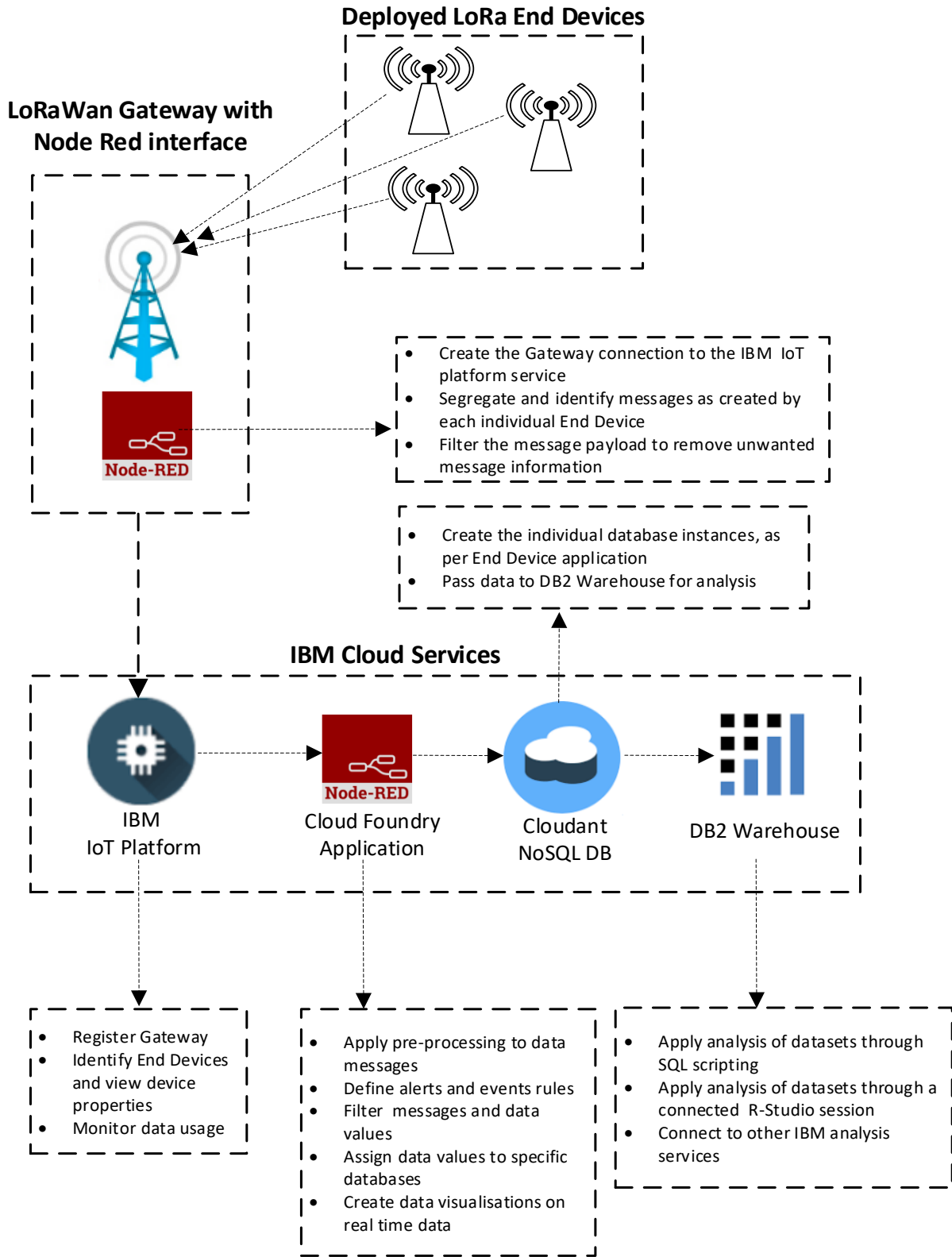


Fig. 4. IoT System: Design architecture and methodology

Fig. 6 illustrates the Cloud Foundry application visualisation page (accessed via web browser), for real time data, as executed in the node flow in Fig. 5. Flexibility of the program

flow allows additional features to be added, including event monitoring and alerts.

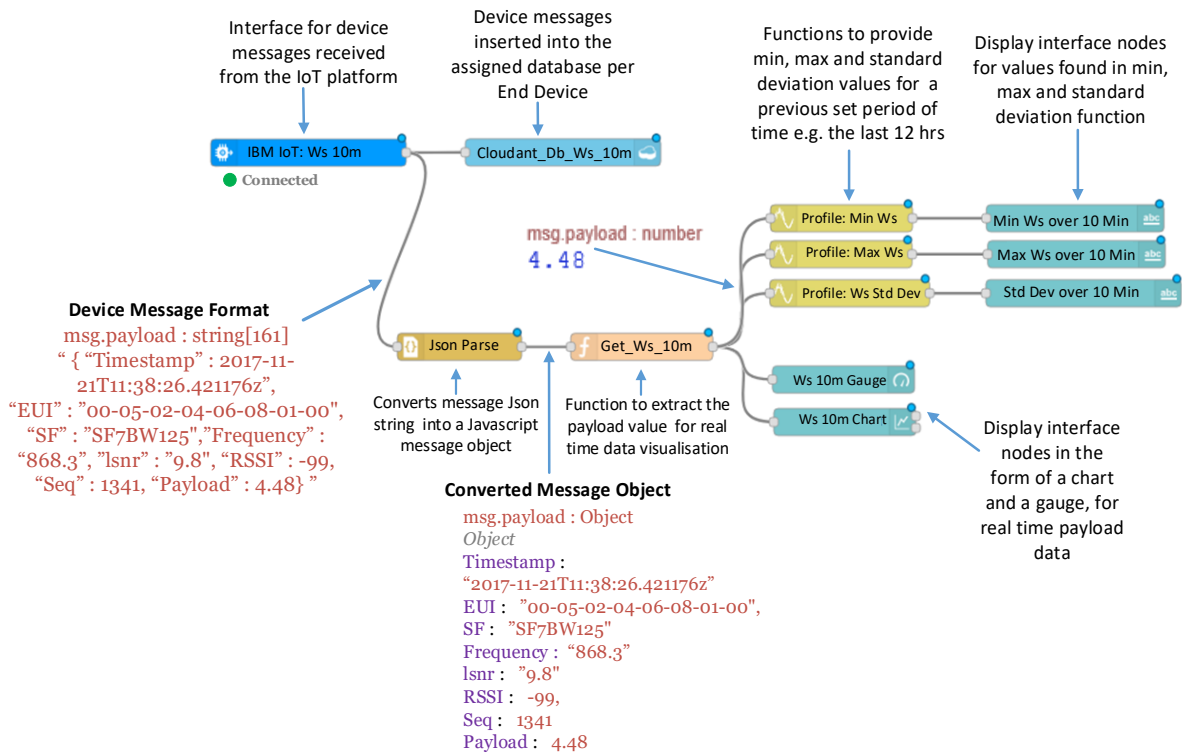


Fig. 5. IBM Cloud: Node Red Cloud Foundry application flow

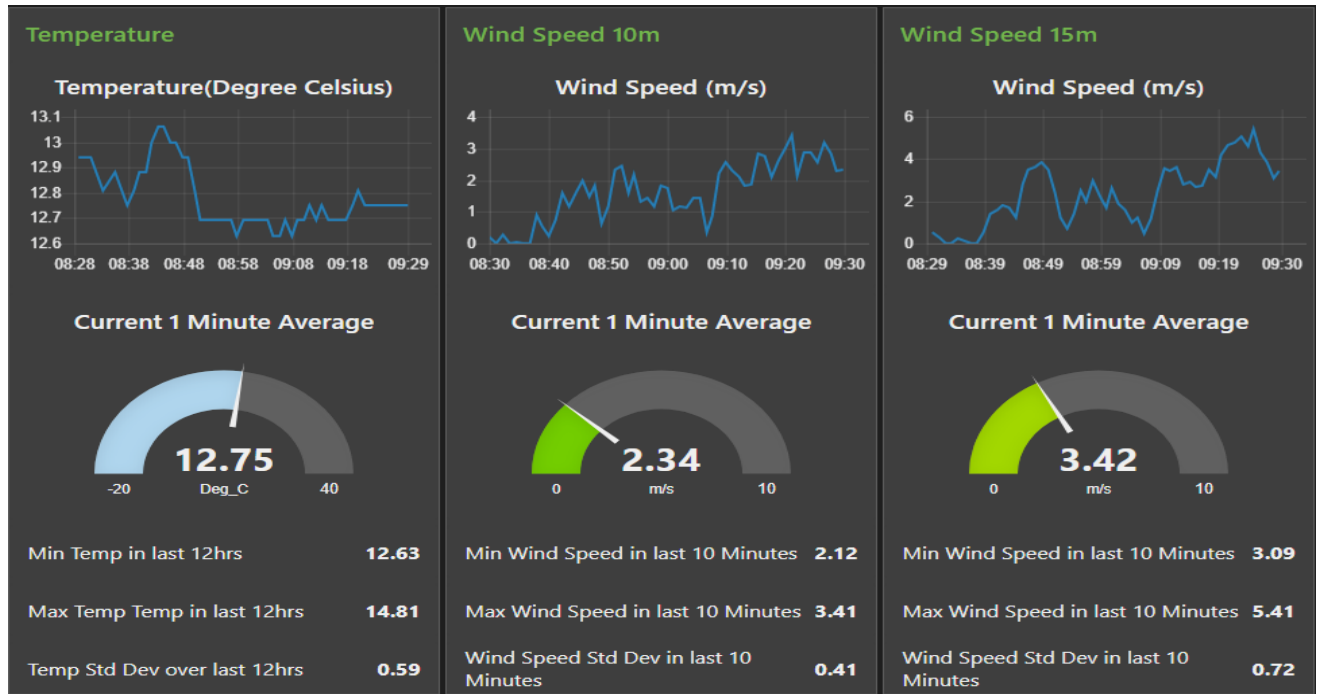


Fig. 6. Real Time Data Visualisation

VI. CONCLUSION

This paper has presented a fully functioning IoT system as a delivered service for the region of Tallaght (Dublin, Ireland) and its surrounding areas. Principally the research has focused towards the implementation of the complete system. For the IoT system currently in place, a design architecture and methodology was presented and applied to data metrics relating to a 6kW wind turbine. The same implementation can be applied to various application requirements of the region. However, limitations of LoRaWAN will restrict the number of applications served, such as, when fast response times may be required. Limitations of the End Device, such as battery longevity, are specific to each application scenario. While the majority of research applied to LoRaWAN has focused on operation and performance of the technology with fewer applied deployments, the presented research bridges the gap between the communication technology and the cloud platform entity, as an implemented system service. Future work will involve expanding the application base of the system, as an IoT service for the given region, with a focus towards End Device optimisation, within the system and future applications.

REFERENCES

- [1] S. A. A'ssri, F. H. K. Zaman, and S. Mubdi, "The efficient parking bay allocation and management system using LoRaWAN," in *2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC)*, 2017, pp. 127–131.
- [2] D. Davcev, K. Mitreski, S. Trajkovic, V. Nikolovski, and N. Koteli, "IoT agriculture system based on LoRaWAN," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2018, pp. 1–4.
- [3] A. C. Lundin, A. G. Ozkil, and J. Schuldt-jensen, "Smart cities : A case study in waste monitoring and management," *Proc. 50th Hawaii Int. Conf. Syst. Sci. | 2017 Smart*, pp. 1392–1401, 2017.
- [4] G. S. Ramachandran, F. Yang, P. Lawrence, S. Michiels, W. Joosen, and D. Hughes, "μPnP-WAN: Experiences with LoRa and its deployment in DR Congo," in *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, 2017, pp. 63–70.
- [5] Y. Liu, B. Lin, X. Yue, Z. Cai, Z. Yang, W. Liu, S. Huang, J. Lu, J. Peng, and J. Chen, "A solar powered long range real-time water quality monitoring system by LoRaWAN," in *2018 27th Wireless and Optical Communication Conference (WOCC)*, 2018, pp. 1–2.
- [6] Semtech, "LoRa." [Online]. Available: <https://www.semtech.com/lora>.
- [7] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.
- [8] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadiania, and N. Strachan, "Evaluation of LoRa and LoRaWAN for wireless sensor networks," in *2016 IEEE SENSORS*, 2016, pp. 1–3.
- [9] A. Augustin, J. Yi, T. H. Clausen, and W. Townsley, "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things," *Sensors — Open Access J.*, vol. 16, no. 9, p. 1466, Oct. 2016.
- [10] IBM, "MQTT." [Online]. Available: <https://www.ibm.com/developerworks/library/iot-mqtt-why-good-for-iot/index.html>.
- [11] MQTT.org, "MQTT." [Online]. Available: <http://mqtt.org/>.
- [12] RF Solutions, "RF Solutions LoRa Transceiver." [Online]. Available: <https://www.rfsolutions.co.uk/radio-modules-c10/frequency-c57/fm-lora-transceiver-module-pre-set-to-868mhz-p468>.
- [13] IBM and M. Kooijman, "IBM LMIC Framework." [Online]. Available: <https://www.arduino-libraries.info/libraries/ibm-lmic-framework>.