A Demo Implementation of Interactive Internet of Things

Pascal King'oku, Alieu Sowe, Qinghua Wang
Department of Computer Science, Faculty of Natural Sciences
Kristianstad University, SE-29188 Kristianstad, Sweden
pascal.kingoku0004@stud.hkr.se, alieu.sowe0019@stud.hkr.se, qinghua.wang@hkr.se

***Abstract*— Internet of Things facilitates interaction between human users and physical environment through the use of electronic sensors, wireless communication and cloud technology. In this paper, we demonstrate a scalable implementation of Internet of Things in a lab environment. A versatile set of sensors including cameras, power outlets and ordinary temperature sensors are embedded in the physical environment and connected to the cloud via local gateways. Data is processed locally and accessible via the cloud. A web service hosted in the cloud enables remote two-way interaction between the human users and the monitored physical environments.**

Keywords*— *IoT, Testbed, Sensors, Gateway, Cloud

I. INTRODUCTION

According to Vermesan et al, Internet of Things is considered an interaction between digital and physical worlds[1]. The importance of IoT is growing more and more as people embrace their wide range of applications in a multitude of sectors[2]. Future Internet(FI) predicts the number of IoT connected devices to outnumber the computers and mobile devices used by humans in a matter of years[3]. IoT's impact towards achieving the 2030 agenda for sustainable goals is measured in trillions of dollars[4].

The creation of sturdy architectural models requires that a reference model is set up to examine the interoperability of deployed IoT resources. Implementation experiments need to be set up as they make it easy for potential parties to visualize intended outcomes and to deal with avoidable issues [5] before scaling up.

Great IoT solutions have the potential to exceed implementation budgets, timelines, or even fail if the concerned parties do not have an idea of how to go about it. Various testbed suggestions have been provided over the years to help introduce concepts and ease the transition of scholars and researchers into real-life applications. Mahmoud et al explore ways to remotely implement experiments related to IoT experiments in an Experiments-as-a-Service kind of model. Their implementation utilizes DHT-11 low-cost sensors as end nodes connected via Wi-Fi to an Arduino Mega gateway. They focus on making it easier for scholars to understand how different layers of IoT setups operate[6]. In [7], a user-centric experimental testbed for IoT research with a focus on smart buildings is presented. They emphasize the importance of realism of the experimental environment to prepare the researchers for real-life scenarios. They also incorporate real-life users into their approach. Another approach presented by Hossain et al describes an architecture that connects multiple Contiki-powered IoT devices in different networks in a Testbed-as-a-Service kind of setup[8].

In this paper, we demonstrate a bidirectional implementation of multiple IoT devices on different networks and how to interact with them. It is intended as an educational guide for scholar and researchers to quickly understand the potential complexities of setting up and integrating IoT devices in multiple locations and using different communication protocols. This is one of the most common problems that tend to overwhelm researchers at an early stage[9].

This setup aims to explore the interoperability of communication technologies/devices and scalability of IoT implementations at a cloud level connecting multiple devices located on different LAN networks. By focusing on an educational context in our experiment, we are able to break down our implementation into easily-understandable parts for consumption by scholars and researchers. Such testbeds require maximum flexibility and should be able to support multiple communication protocols, networks, and use-cases[9].

This paper is organized as follows: In Section II, the design is presented. Section III describes the testbed implementation of the experiment. Section IV presents the results of the implementation. Section V presents the discussion and section VI concludes the paper presenting some potential application areas in the near future.

II. DESIGN

This involved setting up a local network that was accessible via the cloud in a way that allowed two-way communication between the nodes and gateway. The setup was also created such that it could easily scale up if required as shown in Fig. 1.

A. Architecture

1) Lab Network

This local network is inaccessible from external networks unless through specific ports. Our setup consisted of two parts. The first part, the gateway, acted as the connecting node between the end nodes and the cloud. It allows end nodes from outside the network to connect as long as they use ports allowed through the firewall.

End nodes constituted the second part and consisted of sensors and smart devices. These connect to the gateway via Wi-Fi, Bluetooth, or physical connections.

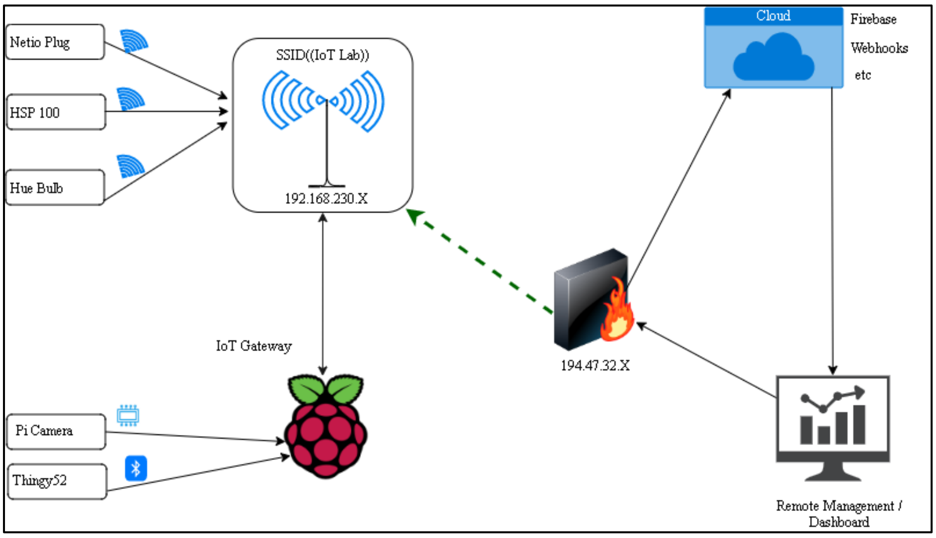

Fig 1. IOT Lab network diagram architecture

2) Firewall

This is configured to allow specific internal ports to be accessible from external networks. It also employs and enforces authentication to prevent unwarranted access to the gateway and end nodes.

3) Cloud

This allows for the storage of data, registration of webhooks, and hosting. We stored our data in real-time databases available on firebase. For webhooks, we used IfThisThenThat (IFTTT) that offered easy and free configurations. The application dashboard was hosted online on Heroku cloud.

4) Remote Network

This allows individuals outside the network to control the smart devices and read information from the sensors via an easy-to-use interface or APIs. Depicted in Fig.1 as Remote Management.

B. Equipment

The IoT components used in this demo implementation comprised of a gateway and IoT devices. Gateway is a multi-protocol device that connects a constrained network with an external network. We used a Raspberry Pi 3 to serve as the gateway for the smart devices and sensors. It is a single-board computer with 1GB RAM, 802.11n wireless LAN, Ethernet, composite video Port, Bluetooth 4.1 connectivity, and was running Raspbian OS.

IoT devices are usually embedded with actuators, lightweight services, sensors, operating systems, and more. They collect information and execute actions as configured. We used a number of IoT devices as described below.

- Thingy52: A multisensory development kit that is connected to the gateway via either BLE or Bluetooth 5.0. We chose this particular device as it supports 52 different functions and it has a documented API that allows for user configurations according to user requirements.
- Netio4: This is a 4-socket smart plug that supports connections via either LAN or Wi-Fi. In our case, it connected to the gateway via Wi-Fi. The device allows each socket to be switched on/off individually and also has a documented API that allows for custom configurations.
- Modular Pi Camera: This connects directly to the Raspberry Pi via the Camera Serial Interface (CSI-2). It contains an 8MP sensor and can record both video and still photographs.
- Philips Hue: This is one of the most popular smart bulbs on the market that connects to the gateway via Wi-Fi.

C. Protocols

We used a variety of protocols to connect smart devices to the gateway and subsequently the cloud. For the Thingy52, we used Bluetooth Low Energy (BLE) which was specifically made for low-powered devices and uses less data. Only when a connection is initiated does it exit its sleep mode.

The Netio plug uses MQ Telemetry Transport (MQTT) which is a lightweight messaging protocol used to send data between devices using a publish/subscribe model.

Wireless Fidelity (Wi-Fi) provides internet access limited to a specific range and we used it for the local area network (LAN).

Websockets provide a persistent connection between the server and client and any party can use this connection to start sending data any time they want. We implemented this using a real-time database on firebase. (See Fig 2.)

D. Technology

The software setup for this testbed employed JavaScript for both the frontend and backend. In the frontend, it was used to style components and charts while in the backend Node.JS was used to connect the required libraries that allowed the connection of sensors via Bluetooth.

III. IMPLEMENTATION

A. Scenario 1- Event–driven Video Trigger

In this simulation, we used a Pi Module Camera and a Thingy52 and we tested two ways to trigger a video recording. One method was by monitoring temperature information from the Thingy52 sensor. Every time the temperature goes below 22 degrees Celsius, the camera starts recording a video that's 10 seconds long. The second way was by using the camera and comparing frames to recognize when there was a change in the field of view.

This video is then uploaded to firebase for storage and this can be shown to the end-user via the application dashboard.

B. Scenario 2 - Smart Plug button

Here we included a button that turned a smart plug on or off every time it was clicked. We used a TP-Link Smart Plug (HS100) and a Thingy52 for the button implementation. In this scenario, the button is on the remote network and the smart plug is on the lab network. For every click, a command is sent to the smart plug on the lab network.

C. Scenario 3 – Event data Visualisation

This setup relied on sensor data from the Thingy52 where humidity, pressure, and temperature data were collected every few seconds. The sensor data is collected within the lab environment and relayed to the firebase data storage via the gateway. The information is then displayed on a remote network as charts on an application dashboard

IV. RESULTS

A. Scenario 1 - Event-driven Video Triggers

An end-user is able to use this to monitor access to a room without special equipment like a PIR sensor since the video is only recorded when motion is detected.

This helps to save on consumption, bandwidth, and storage space. These can be used to trigger actuators that then perform other tasks such as unlocking doors, triggering alarms, and more[1].

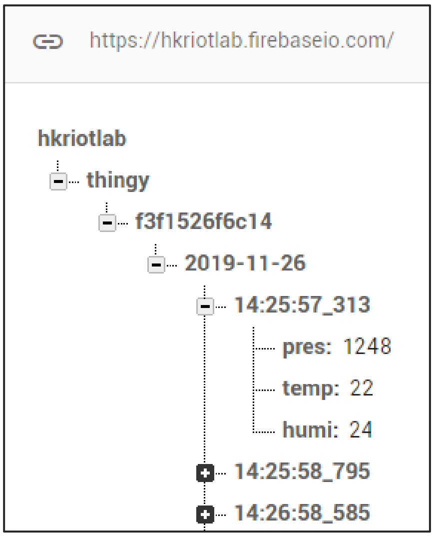

Fig 2. Firebase data storage

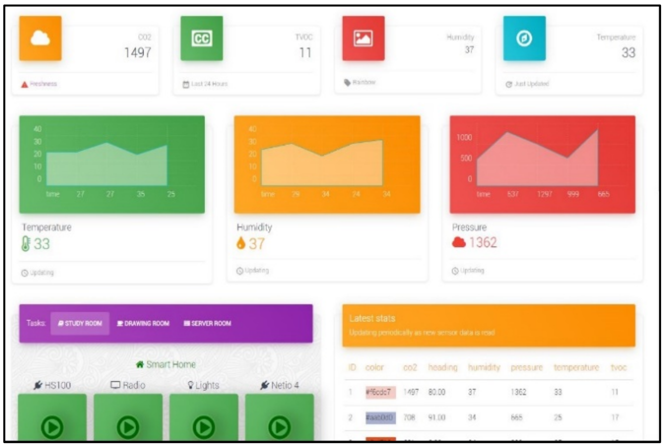

Fig 3. Implementation dashboard

B. Scenario 2 - Smart Plug button

We connected an led bulb to the smart plug so that we could easily visualize button presses from the Thingy52. The bulb in the lab network toggled on/off with every click from the button on the external network. The urban population has an ever-increasing need for convenience. In daily life, this can be implemented in various ways. It can be used to remotely unlock a house door if a repairman needs to access the house or it can be used to shut off appliances such as electric cookers

C. Scenario 3 - Environment Data Visualization

Data read from the sensors was pushed to a Firebase database every 2 seconds where it would be stored (Fig. 2) and pulled to be displayed on a dashboard as shown in Fig. 3. The dashboard displays temperature, pressure, and humidity data as received from the Thingy 52 via the real-time firebase database.

AgTech (Agriculture technology) has embraced technology to improve production over the past few years. The scenario described may be applied to help monitor conditions for crops, livestock, and pollinators which are very sensitive to changes[4], [10]. This setup may also allow for the real-time monitoring of conditions inside a workshop to ensure the optimal working environment resulting in cost-savings by allowing for real-time adjustments.

V. DISCUSSION

This implementation allows end nodes to connect to coordinator nodes regardless of whether or not they reside on different networks as shown by Fig. 1. Observer nodes then subscribe to real-time data as received. Tight security is maintained by ensuring the IoT devices are not exposed externally (by configuring firewall policies). In this setup, the application dashboard acts as the observer node, the raspberry pi acts as the coordinator node while the IoT devices act as the end nodes. The network architecture used in our implementation is consistent with the IoT template level-5 (as shown in Fig. 4) which is presented by Bahga and Madisetti in [11]. According to Bahga and Madisetti, this setup is suitable for solutions that handle big amounts of data and where the analysis requirements are computationally demanding.

The raspberry pi gateway, being a 5v device, offers more computing power as compared to the Arduino used in [6] which is also a 5v device. The setup of our testbed offers easy

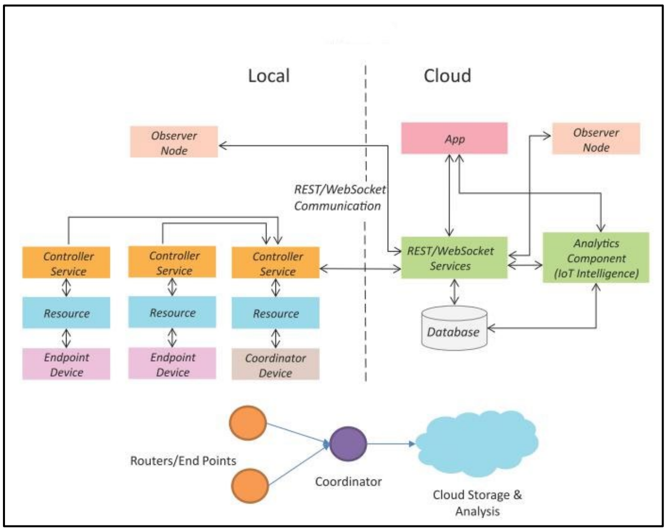

Fig 4. IoT deployment template [11].

configuration, interoperability which was also one of the main objectives for similar presentations presented in [6], [8]. Ensuring realism in the testbed setup ensures relevance of the implementation in real use-cases as was emphasized in [7]. Our setup presents similar use-cases that directly show the potential and flexibility available.

VI. CONCLUSION

Despite their very close relationship, IoT discussions now outnumber Big Data and IoT is yet to reach its peak[12]. This deployed implementation demonstrates the potential that is offered by IoT in helping humans interact with their physical environments. This paper provides insights on the significance of IoT and gives a clear picture of the ease of scaling it up to cater for different use-cases.

The testbed implementation described by this paper provides a step by step guide on how researchers, enterprises, and software engineers can configure different experimental setups according to their areas of interest in a scalable and efficient manner. It offers flexibility by allowing the extensibility of possible configurations and offers a number of examples that can be expanded or improved upon to ease the researcher into a real-world configuration.

Future works entail expansion of scope to accommodate more communication protocols and explore the implantation of such configurations in an online learning environment where scholars are in multiple locations but working on the same IoT project.

REFERENCES

[1] P. Sethi and S. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1–25, 2017, doi: 10.1155/2017/9324035.

[2] E. Gioia, P. Passaro, and M. Petracca, "AMBER: An advanced gateway solution to support heterogeneous IoT technologies," in *2016 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2016, pp. 1–5.

[3] F. Behmann and K. Wu, Collaborative internet of things (C-IoT): for future smart connected life and business. 2015.

[4] Page Louis, "How IoT can help achieve UN Sustainable Development Goals in 2020 and beyond." https://sumas.ch/how-iot-can-help-achieve-un-sustainable-development-goals-in-2020-and-beyond/ (accessed Mar. 22, 2020).

[5] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noël, "Thorough IoT testbed Characterization: from Proof-of-concept to Repeatable Experimentations," *Computer Networks*, vol. 119, pp. 86–101, 2017, doi: 10.1016/j.comnet.2017.03.012.

[6] M. AbdelHafeez and M. AbdelRaheem, "AssIUT IOT: A Remotely Accessible Testbed for Internet of Things," in *2018 IEEE Global Conference on Internet of Things (GCIoT)*, Alexandria, Egypt, Dec. 2018, pp. 1–6, doi: 10.1109/GCIoT.2018.8620157.

[7] M. Nati, A. Gluhak, H. Abangar, and W. Headley, "SmartCampus: A user-centric testbed for Internet of Things experimentation," in *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2013, pp. 1–6.

[8] M. Hossain, S. Noor, Y. Karim, and R. Hasan, "IoTbed: A Generic Architecture for Testbed as a Service for Internet of Things-Based Systems," in *2017 IEEE International Congress on Internet of Things (ICIOT)*, Honolulu, HI, USA, Jun. 2017, pp. 42–49, doi: 10.1109/IEEE.ICIOT.2017.14.

[9] S. Gvk, A. T, A. P, J. Bapat, and D. Das, "Challenges in the Design of an IoT Testbed," in *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, Jaipur, India, Sep. 2019, pp. 14–19, doi: 10.1109/ICCT46177.2019.8969009.

[10] "The World In 2030," *2030Vision*. https://www.2030vision.com/ vision/the-world-in-2030 (accessed Mar. 24, 2020).

[11] A. Bahga and V. Madisetti, *Internet of Things: A hands-on approach*. Vpt, 2014.

[12] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, "A review on Internet of Things (IoT), Internet of everything (IoE) and Internet of nano things (IoNT)," in *2015 Internet Technologies and Applications (ITA)*, 2015, pp. 219–224.