

# A Remote Sensor Network using Android Things and Cloud Computing for the Food Reserve Agency in Zambia

Mulima Chibuye

Department of Computer Science  
The University of Zambia, (UNZA)  
Lusaka, Zambia

Jackson Phiri

Department of Computer Science  
The University of Zambia, (UNZA)  
Lusaka, Zambia

**Abstract**—In order to introduce modern warehousing, improve upon the storage of grain and grain marketing business processes for the Food Reserve Agency in Zambia, a prototype of a remote sensor network was developed and built as a proof of concept for a much wider deployment using cloud computing and the internet of things concept. It was determined that a wireless sensor network would aid the Food Reserve Agency in analytics, timely action and real-time reporting from all its food depots spread-out throughout Zambia. Google's Android Things Platform was used in order to achieve the objectives. Advantages of Android Things over traditional platforms that have been used to develop wireless sensor networks were looked into and presented in this paper.

**Keywords**—Internet of things; android things; wireless sensor network (WSN); remote sensor network; Food Reserve Agency (FRA); grain marketing; modern warehousing; cloud computing

## I. INTRODUCTION

Since its establishment Zambia's Food Reserve Agency (FRA) has been a key in ensuring food security for the nation. In times of severe drought, even neighbouring countries such as South Africa and Zimbabwe have procured their staple food requirement, maize, from the Food Reserve Agency. Grain marketing is a key activity of the Food Reserve Agency. The major output of the agency though still remains in ensuring that the nation at large has enough food stored up for the future. The agency has also been key in ensuring that the commodity price for especially maize is regulated so as not to burden the citizens of Zambia with overwhelming costs [1]. This key organization has been managed by the same processes and procedures since its inception in 1995.

The agency generates a lot of data and information regarding Zambia's strategic food reserves that is a key for national planning. However, the organization has been facing a lot of challenges in managing this information due to lack of automation. In 2013, the Department of Computer Science at the University of Zambia started collaborating with the agency in an effort to resolve some of the key problems through computerization of some key business processes. The outcome of this collaboration is the current research jointly funded by the National Research Foundation (NRF) of South Africa and National Science and Technological Council (NSTC) in Zambia that will be used to automate the key business

processes of Zambia's Food Reserve Agency in order to improve on its operations. The study is jointly being carried by University of Zambia in collaboration with Tshwane University of Technology in South Africa.

The agency, despite being a very important structure in Zambia has lagged behind in the adoption of ICT's. Reports are manually generated which may result in errors and delays, warehouses are not monitored in real time and specialized personnel are required in order to assess the environment at both satellite and main depots where grain is stored. The activities at the Food Reserve Agency are mostly seasonal and hence are managed with just the minimal workforce until the next season. This entails that food storage locations and supply depots have less contact hours with individuals experienced in monitoring the environment at which stock is stored. As an example, it has been clear through preliminary interviews at two of the agency's depot in the capital city Lusaka that 24 hour monitoring of the environment and farmer produce at both the central and remote warehouses is not available. A very good example is when the central warehouses are fumigated and no individual is allowed into the warehouses under any circumstances due to the toxic nature of the chemicals that are used. Further, the 2011 Auditor General's report [2] lists the following problems as concerns FRA:

- 1) Weaknesses in grain management.
- 2) Excess purchase of crops.
- 3) Excess stock losses attributed to spillages, pilferage.
- 4) Warehouse Management – Poor Stacking making hard to take stock.
- 5) Poor State of Storage Facilities – Silos.

The 2011 audit report of the Food Reserve Agency was used because there were no subsequent audits carried out. The reasons of which might be conjecture. However, interviews and physical visits corroborate the 2011 auditor general's report findings.

The problems highlighted present huge losses to both government and risk in terms of food security for the nation at large. It is presented in this paper that all the highlighted problems can be addressed using automation and ICT's [3]. It has been shown in research that the use of ICT's introduces

efficiency and accountability in agriculture [4]. For the purpose of this research, only FRA will be targeted but the solutions presented can be applied to all institutions that deal with warehousing and food storage as well. Against this background, this paper will attempt to address the challenges.

- 1) Collecting of real time environment data from FRA warehouses.
- 2) Storing the data indefinitely.
- 3) Ensuring that the stored data is well backed up.
- 4) Generating real time environment reports.

The challenges highlighted above are addressed by a low cost and easily mass-produced remote sensor network. It is believed that the government of Zambia through the Food Reserve Agency will be able to begin to better manage the grain marketing business through the use of the platform that we have developed. Cloud computing, digital sensors and an open platform are used in order to develop a generic platform that is easily scaled and applied to various environments.

## II. LITERATURE REVIEW

Remote sensor networks have traditionally relied on four components, a sensor to collect the data, an aggregator to centralize the collection of data, an uplink network to relay the data and a server to which the data is to be sent, the wireless sensor network can be further broken down into two main components, the remote sensor network and the uplink. In this chapter, each of the main components that make up the implementation of the wireless sensor network using Android Things are discussed

### A. Remote Sensor Network

The remote sensor network is that part of the network that is composed of the sensors which have energy sources and some kind of personal area network used for localized communication. The network may also comprise a collaboration algorithm which may determine its work mode. The three main work modes that enable collaboration in a wireless sensor network are a star network, a mesh network or a ring network. The remote sensors basically sense, process and send the data either individually or through aggregator. Each of them is indicated below;

### B. Cloud Computing

With the spread of broadband internet across the globe, more emphasis has been placed on how to effectively and efficiently utilize and allocate all the available computing, processing and storage capacity available globally. A reduction in latency and increased throughput has made it possible for remote computers to be managed and applications run as though they were physically local. Resource sharing has enabled concepts such as Software as a Service, Platform as a Service and Database as a Service where the end user does not need to worry about the technical requirements of systems,

software and the technical specifications of the database but just focuses on service usage [5]. Cloud computing has made it possible for emerging companies to rollout their services at a much faster pace as the cost of datacentres is slowly becoming a non-factor with cloud based solutions such as Data Centre as a Service (DCaaS).

Cloud computing has brought about novel ideas where a computer is no longer viewed as a standalone entity but can span multiple hardware platforms and multiple geographical locations. A very good example was the so called first world computer Ethereum. The Ethereum platform relies on individual machines spread-out throughout the globe to run a client application with distributed consensus and a distributed ledger. This makes it possible for an application to be installed on the world computer and be able to run even if there was a global disaster of immense proportions. The application is able to run with just one computer that downloaded the entire Blockchain. The Blockchain is a public ledger of all transactions that ever occurred on a distributed network and is termed by many as a disruptive form of technology equivalent to the advent of the internet in the early 1990's. Blockchain technology is perhaps the de facto example of what is entailed by cloud computing as it is a classical implementation of device as a service (DaaS) and software as a service (SaaS) and Ledger as a service (LaaS).

### C. Android Things

The most popular standard for the implementation of internet of things prototypes and solutions has been to use the Raspberry Pi and Arduino with various sensors. One would then have to attach GSM, WiFi or some other wireless protocol shield in order to achieve remote connection to either a distributed or a centralized server, there has been several attempts at creating sensor networks that use a mobile device for obvious reasons of taking advantage of the use of a mobile phone and that is to send and receive data [6]. Android Things aims to bridge that gap by leveraging already existing hardware platforms and development environments by installing an operating system on those capable hardware devices. That operating system called Android Things is built with an internet of things architecture at its core and therefore supports native functions and routines that make implementing a prototype or developing a solution a very easy thing to achieve. There is no requirement to write code for multiple devices such as one that serves as a master and supporting slave nodes. All nodes including sensors are controlled from one operating system. Because Android Things is a fork of the popular Android operating system that is used in various devices around the world today, its API does not differ from that of the mobile application development and every sensor is an object level abstraction in the system. The compact nature of this solution means the development and deployment time is significantly cut down. Android Things is naturally the choice for use on the proposed remote sensor network. Fig. 1 below lists the Google Android Framework and API which is also the Android Things Framework and API.

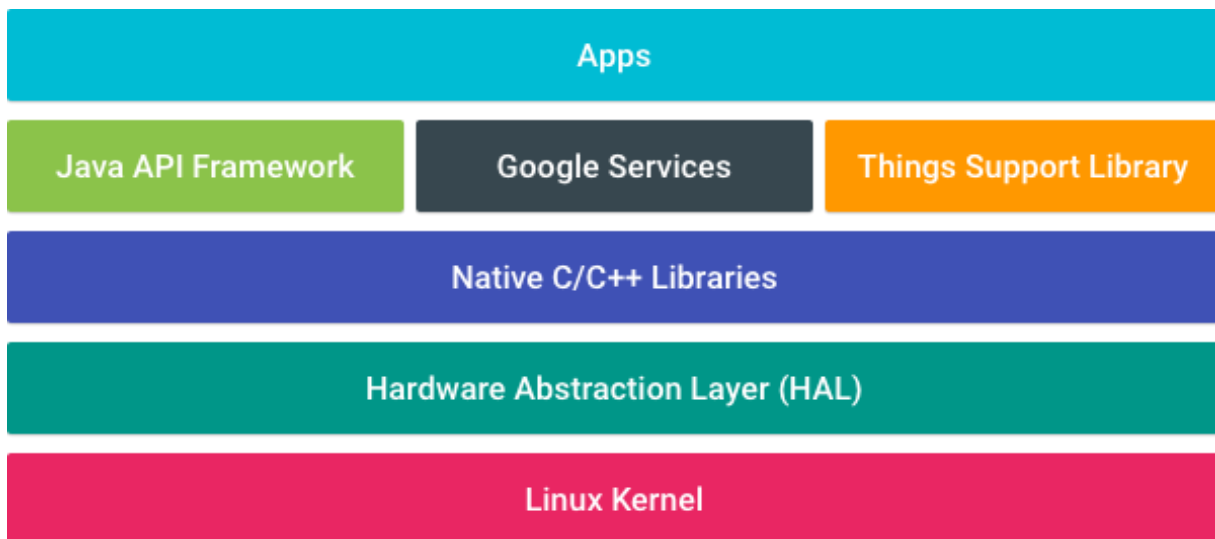


Fig. 1. Android things framework and API.

As of early 2017, about 4 devices from different manufacturers around the world support the Android Things operating system. Some of the devices include the Intel Edison, Intel Joule and the Raspberry Pi 3. While there are similarities in the framework and API to that of Google's Android, the development methods differ slightly. The Google API for Android is extended by Android Things which adds additional API's used to enable the integration of hardware devices that are not normally found on mobile phones and tablets such as smoke detectors. It is also important to note that the Android Things OS does not come with any app installed. The first app to be deployed on the platform is the app that starts-up as soon as the device is booted. This solves one major issue which is the use of system resources in remote sensor networks. It is important to make the remote sensors as lean as possible in order to conserve battery life and processing ability.

#### D. Mobile Queuing Telephony and Telemetry (MQTT)

MQTT stands for Message Queuing Telemetry Transport. The protocol is specifically designed for internet of things devices with low bandwidth requirements. It is a lightweight publish/subscribe protocol over TCP/IP with a level of quality of service defined and was originally created by IBM and Arcom [7]. The later company became part of Eurotech. It has since become an ISO standard with filing ISO/IEC 20922:2016. It is ideal for mobile applications because of its small size, low power usage, minimized data packets, and efficient distribution of information to one or many receivers. A specific implementation of MQTT for sensor networks is called MQTT-SN (Message Queuing Telemetry Transport for Sensor Networks).

By its nature, the Internet of Things relies on a messaging broker to relay information between various devices. The MQTT protocol is standardized by ISO under the filing ISO/IEC 20922:2016[8]. MQTT is a variation of Zigbee [9] which we discuss below.

#### E. Mosquitto Broker

In the typical architecture of an internet of things project,

there is the Thing which is a device connected to the internet that can be sensor or an actuator which performs an action and there is also the broker which can be a centralized or cloud based system that publishes messages to what are called topics. Each of the Things in an internet of things setup subscribes to a topic and each topic can cover a specific need or requirement such as a topic to control lights at the end of the day in a specified geographic region. The broker can publish a message to switch on lights in a specific geographic area and any switch that can be viewed as a thing that is subscribed to the switch topic will flip the switch to on. The Mosquitto Broker is an open source broker that is used in internet of things applications and was the selection for the model. The Mosquitto broker implements the mobile queuing telephony and telemetry protocol version 3.1 as of this writing. Version 3.1.1 of the broker was specifically designed with internet of things applications in mind as it implements a light weight publish/subscribe methodology.

#### F. Raspberry Pi and Arduino

The Raspberry Pi is a very tiny computer that is popularly used in prototype development for educational purposes. It has the standard computer architecture of read only memory (ROM), read and write memory (RAM), processor and bus. It is possible to extend the functionality of the Raspberry Pi with peripherals as it supports most of the common input interfaces from the general purpose input/output to the USB and peripheral input/output interfaces.

The Arduino is a very simple programmable open source electronic device that can interpret the Arduino Programming Language code which is similar to the Python programming language. The Arduino can also be programmed to do basic things such as read both digital and binary inputs to its circuit board. The Arduino differs from the Raspberry Pi in that it is a basic microcontroller while the Raspberry Pi is a full-fledged scaled down computer. Programs written in the Arduino Programming Language tend to be basic and loop. They are also functional in nature.

### G. Zigbee

Zigbee is a low power wireless protocol operating at layer 2 of the OSI protocol stack using the IEEE 802.15.4 standard [10, p. 15]. Zigbee is primarily used in the local area network of a remote sensor network. Because of its usually limited reach as concerns the wireless signal, it is further classified into a set of wireless personal area networks (WPAN). The protocol can be used to send small payloads between a sensor and its controller. Because of its design focusing on low computational complexity, it is ideal in the deployment of sensor networks communications requirements. Zigbee will be a very important part of the proposed implementation as concerned with the one of the primary goals of achieving a power efficient implementation. Zigbee is not a standardized protocol but has reached maturity in industry. Zigbee devices can be setup and configured in three distinct modes, coordinators, routers and end devices. There can only be one coordinator in a Zigbee network and this will usually be connected to a device that has access to the backhauling network. Router devices as the name suggests, aim to forward packets or frames to the targeted device while end devices are passive and only send/receive data within the cloud setup. Zigbee data can be converted to a different protocol by the use of a Zigbee gateway which is part of the standard architecture. This protocol conversion is important in remote sensor network scenarios when sending wireless sensor network aggregated data to a remote server over a different wide area network that operates on a completely different protocol stack. A Zigbee network can be set up in multiple types of topologies namely the star, tree, mesh and clustered tree topology.

### H. Products and Platforms

With respect to sensor networks, a product is defined as a tool that is used in order to build the sensor network. Various products may be used in order to achieve the goal ranging from hardware products to software products. Each of these products may have a different method of operation or rely on different software and hardware in order to get them working. For example, a wireless sensor network may be composed of a Raspberry Pi, an Arduino, a mosquito server and some communication network. Below it is defined which of these products are platforms. The broad definition of a platform is that it serves multiple groups or products or users and facilitates interactions between them. The users may be human, machines or software and each of them may be distinct, requiring individual functionality but at the same time may require exchange of data. With that broad definition, the hardware, software and other components used in a sensor network are listed and it is defined as to whether they are platforms or products by asking the following questions:

- 1) Does it serve multiple users or customers?
- 2) Does it facilitate interaction between users?

Each of the above questions is discussed below in order to show the context in which the questions were applied to determine whether an item is a product of a platform. Let us consider a practical example such as AirBnB [11]. On the question of whether it serves multiple users, the answer is yes

because it is a platform at which anybody that does not own it or does not need to know its technical workings may use its services. It also facilitates the interaction between multiple users because it brings together buyers and sellers who can then interact by exchanging goods or services for money. While looking at the current popular model used in building sensor networks, the most common devices and applications were listed in Table 1 below and those two important questions above were applied to each of them. The questions were also applied to the selected environment, Android Things. The conclusions are indicated below:

TABLE I. PRODUCTS OR PLATFORMS

Determine whether component is a product or a platform	Major components of our wireless sensor network			
	<i>Mosquito Broker</i>	<i>Raspberry Pi</i>	<i>Arduino</i>	<i>Android Things</i>
Does it serve multiple users	Y	Y	N	Y
Does it facilitate interaction between users	Y	Y	N	Y

Android things is our selected software for the development of the Food Reserve Agency's remote sensor network because it is a platform and the following benefits will be realized if a platform is used as opposed to a product or multiple products.

#### 1) Reduced technical overhead

Platforms offer reduced technical overhead because they form a base layer that facilitates interaction between several parties without the unnecessary technical complexity such as the need for specific programming languages and development environments for each component within a system. An example for the sensor network is that one programming language may be used to communicate with all peripherals and the aggregator.

#### 2) Scalability

Platforms can easily scale to support new features by simply designing them to be modular such as the case is in most operating systems. This way, new peripherals, sensors and devices could easily be supported by third parties because the platform by its definition serves the needs of multiple users hence it is assumed that for it to be a true platform, it must support extensibility and scalability third parties through access to the platform specific application programming interface.

### III. METHODOLOGY

A baseline study was carried out in order to map out the FRA business processes and also to see whether business process reengineering could be applied and the processes automated. The baseline study was also used in order to identify whether FRA would benefit from the inclusion of ICT's in the daily business activities and what perceived benefit these would have on the organization as a whole and on the grain marketing business in Zambia.

### A. Baseline Study

A baseline study was carried out between November 2015 and June 2017. Self-administered forms, interviews and observations were used in order to understand the FRA business processes. FRA also provided with the Microsoft Excel forms that were used for data capture purposes. A mix methods research methodology was employed in trying to map out the FRA business processes. A descriptive approach was also employed in order to affirm the finding from the mixed method that was initially used. It was necessary to use the two approaches because it became evident that while there were systems in place governing the daily operations of FRA, there was no documented standard and existing processes were bound to change at short notice with no log as to what necessitated the changes. The descriptive approach provided us with a broad overview of the individual business processes and how those map to each other in order to achieve the final goal of grain marketing.

### B. Illustrative System Modelling

The remote sensor network was illustratively modelled in a modular approach. This was done so as to avoid restrictions on the kind and type of modules to be used. The bus connectivity of each module was not defined and was left open to interpretation.

### C. Functional Prototyping

In order to prove the practical implementation of the remote sensor network using android things, two prototypes were developed. An iterative development approach was used. The first prototype transformed into the second prototype. This method was only useful to the author in order to assess practicality of the overall project as there was little contact with the technical team at FRA. It is suggested that for a third prototype, an agile development method be adopted so as to have constant input from the end user during the development life cycle.

## IV. RESULTS

The baseline study was carried out and this presented the challenges faced by FRA in the current manual systems employed in grain marketing and grain storage. The baseline study determined that FRA had challenges such as manual report generation, no connectivity to remote warehouses, inability to track stock on demand, theft, spoilage of stock due to lack of environmental monitoring. The business processes for the Food Reserve Agency were also mapped out and details are presented in a paper by Cynthia Muyunda [12] and Jackson Phiri [12] who were also participating in this research. As the business process mapping and business process reengineering was already completed, those will not be focused on much in this paper. However it was recognized that further efficiencies could be employed by using an agile approach with further input from the FRA staff.

There were challenges faced in collecting the empirical data that it was hoped would ascertain whether the Food Reserve Agency would require ICT's in order to better manage the grain marketing business. Over 150 questionnaires were prepared but a poor response rate of only 41%. It was therefore decided to substantiate the data with purposive visits to the

agency's management offices and warehouses. Interviews were conducted with key staff and further insight was used to create a detailed description of the business processes which were mapped and can be seen in a paper by Cynthia Muyunda and Jackson Phiri [12]. For the purpose of this paper, we focused on creating a solution for all locations that are used to store grain before it is further transported to other locations or permanently stored until sold.

In trying to develop a low cost remote sensor network solution relying on cloud computing, the requirements and components for such a network were determined and these were sensors, aggregators, an uplink and downlink network and a cloud based storage service. The most usual method of creating a sensor network would be to connect a series of sensors to a personal area network and an aggregator that has a GSM shield. This GSM shield may support the common mobile network technologies such as GSM, 3G, LTE and most recently 5G. The GSM shield is connected to the aggregator and an operating system is used in order to communicate with the shield and upload data to a remote server. The cloud based service earlier referred to for the purposes of the prototype was a broker that is used to gather all the information from the remote sensors, the most popular open source broker being the Mosquito server. Popular devices that are used for the aggregators are the Raspberry Pi or some other device with acceptable computational power and computer architecture as there would have to be some sort of operating system installed on the aggregator. The common setup was looked at and analyzed as to know whether a low cost solution based on the common architecture could be built and whether it could be improved upon. The conclusion was that since the Android operating system supported native sensors, it could be used along with its high level API calls as the operating system of choice for the remote sensor networks aggregator. The advantages were found to be as follows:

- It already supports native mobile functionality.
- It supports native calls to supported and unsupported sensors with custom drivers.
- It can be stripped down to its most basic requirements for the sensor network.
- It works well on mobile phones hence has low power requirements.
- A device could be fabricated by removing unnecessary features such as the screen, speakers, mic, etc. in order to remain with the basic operating system functionality.
- The prototype could make use of the USB port in order to interface with devices that required serial communication.

Fig. 2 shows a very early iteration in which it was attempted to connect sensors to the mini USB port of an Android device. Here an Arduino was used as an aggregator for the sensors and the Android device as the processor and communications network interface. The sensors in the figure include a temperature/humidity sensor and a motion detector. Location sensors on the Android device were made use of in order to provide localized data.



Fig. 2. Android device with temperature, humidity and motion sensor connected via USB on-the go cable (OTG).

While still trying to develop the remote sensor network, Google announced in December 2016 that they would be releasing an operating system called Android Things that would support several common hardware platforms which are commonly used in sensor networks. This meant that our research took a turn from using Android to using Android Things. There were obvious advantages such as:

- The Android Things platform was specifically built for the internet of things:
  - Low cost
  - Low power
  - Native support for the Google Cloud Platform.
- Support for several types of commonly used sensors and the ability for a developer to create custom drivers.
- Native support for mobility functions such as GSM, 3G, LTE and WiFi.
- Support for popular prototyping devices such as the Raspberry Pi 3.

For the purposes of the development, a Raspberry Pi 3 with the following sensors and support peripherals were used:

- Temperature and humidity sensor(BME280 I squared C or Serial Peripheral Interface).

- Motion sensor(Pyroelectric (“Passive”) Infrared Sensor)
- Wearable GPS sensor
- Zigbee transceivers
- WiFi access point for communication with the Mosquitto broker.

A free version of Amazon Web service AWS was used and the Mosquitto Broker was installed on it in order to collect the sensor data. Fig. 4 below is a model of our remote sensor network setup. Fig. 3 below is our model for the remote sensor network. We have only been able to implement a network with temperature, humidity and location information as a proof of concept. With our modularized design, one could add all manner of sensors and even actuators.

For the purpose of future deployment of the entire system, we developed a model for a platform and we have previously discussed the advantage of a platform in that it can be used not only by the intended user but by any other users. Our model may be utilised as a service using the Zambian government’s national data centres for storing and processing the sensor network data. Fig. 4 shows our system model. The sensor network was previously depicted in Fig. 3. Cloud computing is proposed in our model for reasons previously discussed. The Food Reserve Agency in Zambia can leverage already existing data centres and indeed the national data centre which aims to make computing resources universally accessible to all government agencies and organizations. Our model is a layered architecture where each layer may be composed of different product types with standard application programming interfaces native to the selected product. A user may for example decide to use any manner of distributed operating system in existence or yet to be implemented. A user may also decide to use any other message broker besides our prototype selection, the Mosquitto broker. This design gives true independence as to the flavor of the products selected for deployment onto the standardized platform.

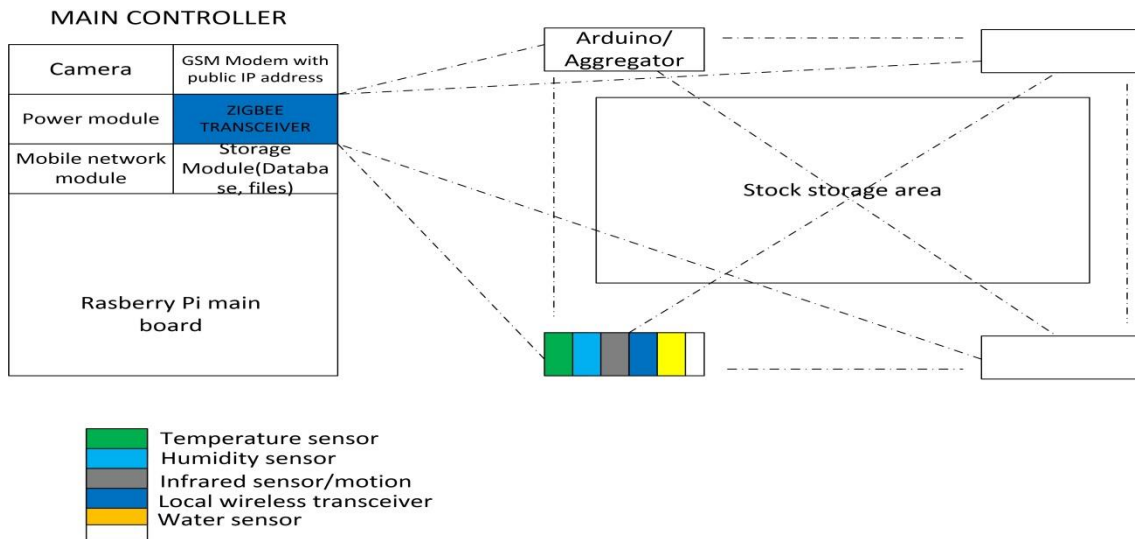


Fig. 3. Remote sensor network model.

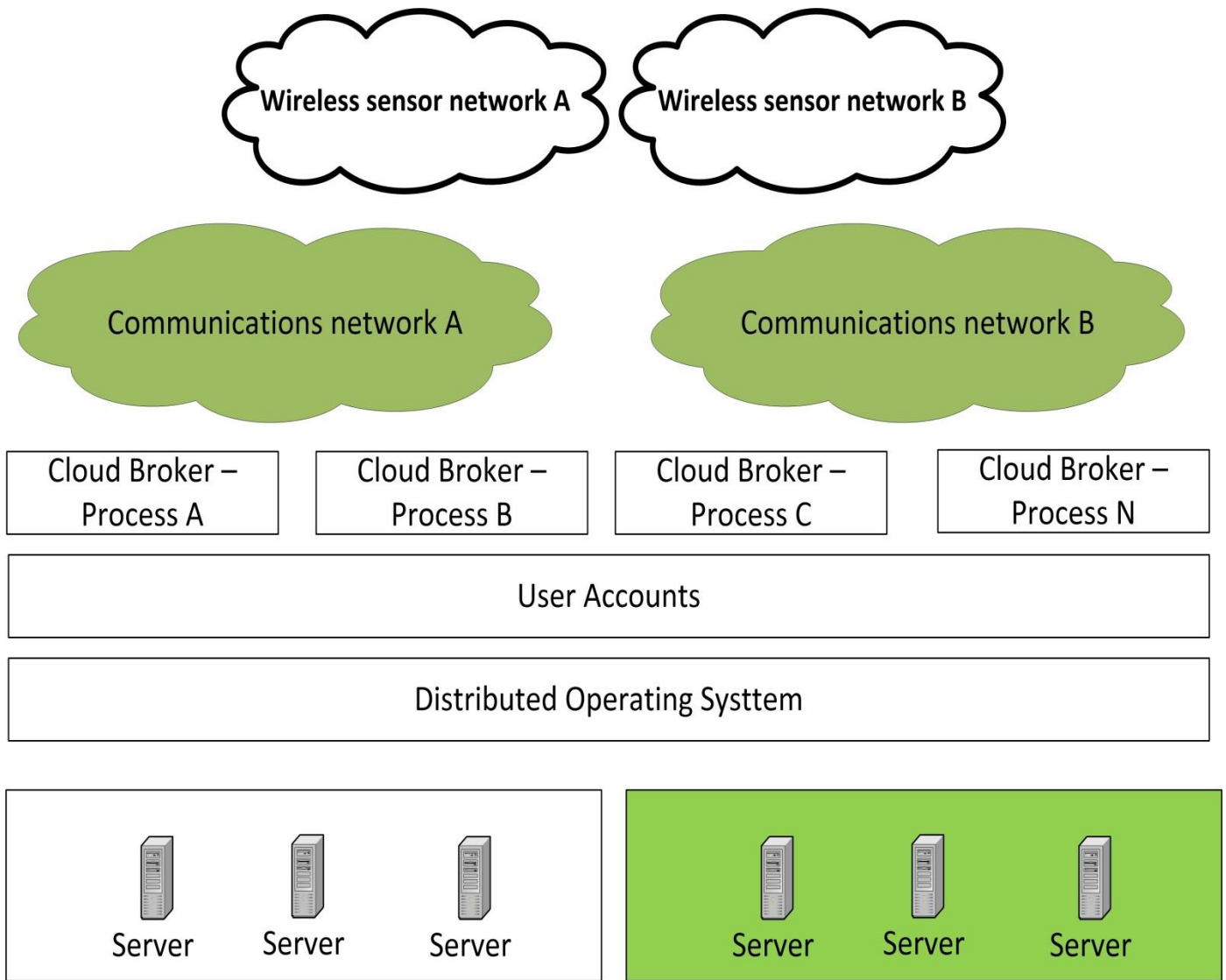


Fig. 4. FRA universal remote sensor network architecture.

## V. DISCUSSION

### A. Limitations of the Study

The study used what may be seen as an impractical implementation for a real world use case as it is unlikely that a government agency may decide to use services hosted by third parties on the internet such as the Amazon Web Services. Access to the government owned data centre would have provided the best prototyping implementation but the proof of concept stands to consolidate the practicality of the solution.

A proper cost analysis of the sensor network solution in mass production was not carried out. This is important in order to determine the financial implications of a mass rollout to all Food Reserve Agency depots.

### B. Summary

We decided to use the Google Cloud Platform because it offers a data centre as a service. There has been a lot of research done that suggests that outsourcing of the data centre can save an organization or indeed a government a lot of

money. We also came to the conclusion that a government and its institutions may consider the use of Blockchain technology for the purpose of data storage and maintenance of ledgers. We have previously presented a conference paper to that effect “unpublished” [13]. While the Google Cloud Platform was used for the prototype, we have not assessed the financial implications of its usage in an actual mass rollout of the sensor network and we believe that because the government of Zambia has an active data centre, the issue of data storage and data processing for the model sensor network can be addressed with the current facilities. This it is suggested may be taken up as additional research work.

## VI. CONCLUSION

The choice to use Google’s Android Things platform for the development of a remote sensor network is mainly due to the fact that Google has created a platform with advantages that we believe are essential towards the further development and standardization of remote sensor network architecture at both the hardware and software level. The change that we expect to

follow is equivalent to that change that followed the smart phone ecosystem following the introduction of the Android operating system which is essentially a software platform and several third parties were able to leverage Android and communicate with other parties through a standardized development environment.

It is believed that the Android Things platform will continue to develop and evolve especially since it is open source and this will in turn lead to industry self-standardizing sensor interfaces and API for seamless development and deployment. Our developed proof of concept follows the platforms model and can easily be extended by third parties.

As for how to solve the storage of data that is generated from multiple sensors and multiple warehouses, it is believed that cloud based storage holds the key. For the prototype Google Cloud Platform was used. However, cost implications may prevent the government of Zambia from utilizing that platform. It is recommended that the national data centre of Zambia is leveraged for data storage and data processing. Since redundancy and resilience are a key factor in national data storage and indeed for the Food Reserve Agency's everyday work, it is further recommend that Blockchain technology is explored as the standard for the backend storage and data processing.

A universal model for a simple and possibly low cost remote sensor network that may be implemented by the Food Reserve Agency in Zambia was developed and through literature, it is evident that modern warehousing relying on components such as sensors provide better grain storage, management, transparency of operations and hence lead to cost effective grain marketing which leads to better national food security [13].

#### REFERENCES

- [1] Nicole M. Mason and Robert J. Myers, "The effects of the Food Reserve Agency on maize market prices in Zambia," *Agric. Econ.*, vol. 44, pp. 203–216, 2013.
- [2] "Report of the Auditor General on the accounts of parastatal bodies for the Year Ended 31st December 2011." Office of the Auditor General, 2011.
- [3] Nancy Mock, Nathan Morrow, and Adam Papendieck, "From complexity to food security decision-support: Novel methods of assessment and their role in enhancing the timeliness and relevance of food and nutrition security information," *Glob. Food Secur.*, vol. 2, p. 41,42,43,44,45, Nov. 2012.
- [4] SILVIA MASIERO, "Redesigning the Indian Food Security System through E-Governance: The Case of Kerala," *World Development*, vol. 67, pp. 126–137, 2015.
- [5] S. Sharma, "Evolution of as-a-Service Era in Cloud." Center for Survey Statistics and Methodology, Iowa State University, Ames, Iowa, USA.
- [6] R. Piyare and R. Piyare, "Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone," *Int. J. Internet Things*, vol. 2, no. 1, pp. 5–11, 2013.
- [7] A. Stanford-Clark, "The importance of Internet standards and other topics," in *Open Control Systems - The Importance of Industrial Standards*, 2004, p. 7/1-711.
- [8] C. Carles Gomez, A. E. Arcia Moret, and J. A. Crowcroft, "TCP in the Internet of Things: from ostracism to prominence," Oct. 2017.
- [9] Nidhi Patel, Hiren Kathiriya, and Arjav Bavarva, "Wireless Sensor Network Using Zigbee," *Int. J. Res. Eng. Technol.*, pp. 1–5.
- [10] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802154-2015 Revis. IEEE Std 802154-2011*, pp. 1–709, Apr. 2016.
- [11] L. Zekanovic-Korona and J. Grzunov, "Evaluation of shared digital economy adoption: Case of Airbnb," 2014, pp. 1574–1579.
- [12] Cynthia Lubasi Muyunda and Jackson Phiri, "A Wireless Sensor Network Based Grain Inventory Management System for Zambia's Food Reserve Agency," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 5, no. 3, pp. 3519–3526, Mar. 2016.
- [13] Blessing M. Maumbe and Julius Okello, "Uses of Information and Communication Technology (ICT) in Agriculture and Rural Development in Sub-Saharan Africa: Experiences from South Africa and Kenya," *Int. J. ICT Res. Dev. Afr.*, vol. 1, no. 1, pp. 1–22, 2010.