Expertise
and insight
for the future

Hailu Dibaba

IoT Implementation with Cayenne Platform

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

12 December 2018

Metropolia
University of Applied Sciences

Author Title Number of Pages Date	Hailu Dibaba IoT Implementation with Cayenne Platform 35 pages + 1 appendix 12 December 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Networks/IoT
Instructors	Janne Salonen (Head of Department, Information and Communications Technology (ICT)

The purpose of this thesis is to study and present the concept of IoT and its implementation. The historical background and the progress of IoT to the present day are presented. The basic building blocks of an IoT system are analyzed and presented. Different connectivity options for an IoT solution are discussed with each option having its own strengths and weaknesses.

The different types of architectures for IoT, divided based on the number of layers involved (3 layer/5 layer architectures) and based on the location of the main data processing unit are presented. Each layer is discussed in its own sub section in this thesis.

The different protocols used in IoT applications are studied across the physical layer, data link layer, network layer and application layer. These protocols are designed to have smaller overhead, be able to operate efficiently with low bandwidth, and be able to function for a long time with low power usage. These requirements are important because of the constrained resources IoT devices have when compared with other internet connected devices.

The practical IoT implementation has been done with a platform called *Cayenne* which is provided by a company called *myDevices*. This platform enables a simpler and quicker deployment of an IoT solution. The results of the implementation are also shown in this thesis.

Keywords	IoT, IoT Architecture, IoT Protocols, Cayenne, MQTT, Eclipse Paho Project, Raspberry Pi,

List of Abbreviations

CoAP	Constrained Application Protocol
GPIO	General Purpose I/O
GUI	Graphical User Interface
IoT	Internet of things
LoWPAN	Low power Wireless Personal Area Network
LPWAN	Low Power Wide Area Network
MQTT	Message Queue Telemetry Transport
NFC	Near-field Communication
RFID	Radio frequency identification
SSH	Secure shell
SSL	Secure Sockets Layer
TLS	Transport layer security

Metropolia
University of Applied Sciences

Table of Contents

1 Introduction

The Internet of Things (IoT) is a computing system by which physical devices are connected to the internet and be able to communicate with other devices/systems. IoT devices can communicate using IP connectivity without human interference.

The devices need to have smart systems which are capable of collecting and transferring the data over the network. The collected data can be processed at the edge device or can be sent to the cloud infrastructure through different connectivity options.

There are numerous options for connecting the Internet of Things. Among these connectivity options are cellular, satellite, Wi-Fi, Bluetooth, RFID, NFC, LPWAN, and the Ethernet.
A suitable connectivity option is chosen based on power consumptions level, range that can be covered and bandwidth for the data transfer. Usually a tradeoff between these factors needs to be made when choosing the most suitable connection option.

After the data has been transferred to the cloud, software platforms perform data processing on it. The software can send back instructions to the 'thing' or it performs action automatically based on the instructions programmed on it.

There are different IoT platform options by different vendors. An IoT platform called Cayenne from a company called *myDevices* is used in this thesis. Cayenne is a GUI (graphical user interface) based IoT project builder.an Air condition monitoring system had been built using the Cayenne platform. The sensors will send air temperature and humidity values to the Cayenne cloud infrastructure. There is an MQTT connection protocol between the sensors and the Cayenne cloud. This IoT platform can be accessed from the web or a mobile application.

This thesis will discuss the theory behind the IoT operation in chapter two. Chapter three will be about the different layers of IoT architecture. Fog and Cloud computing architectures are also discussed in this chapter. Chapter four will be about the different protocols used in IoT applications. These protocols will be divided and discussed across the four layers of the Internet protocol suite.in chapter five there is a practical IoT implementation using the Cayenne platform. The results of this implementation are also discussed.in chapter six, the conclusion about the thesis is discussed.

2 Overview of Internet of Things (IoT)

The Internet of Things is a system whereby physical devices are connected to the internet and are able to communicate with other devices. This communication allows the devices to send and receive data over the network. Human-human communication is the main form of present day internet technology. The internet of things (IoT) can be considered as the next breakthrough of the internet by enabling machine-machine communication (M2M). [1]

IoT enables autonomous and secure transfer of data between physical devices and software applications. The physical devices can have different sizes, data storage capabilities and processing power. They also support different set of applications. IoT brings the physical world and the virtual world together. [2][3]

The term 'Internet of things' was first coined and used by Kevin Ashton, a British technology Innovator who was working on the implementation of radio frequency identification (RFID) technology to connect physical devices to the internet. Since then, Internet of Things (IoT) has seen rapid development and expansion into different fields. Organizations in different sectors are increasingly adopting IoT to gather more data about physical devices, their services and customers. This allows companies to operate more efficiently, improve customer experience and create new business models thereby generating more revenue.

The internet of things is a big network and is getting bigger. Different organizations provide different estimates on the number of connected devices in year 2020. The technology research firm *Gartner Inc*. forecasts that this number could be about 8 billion. [4] Ericsson and Cisco provide separate forecasts and put the number at 50 billion devices. [5-6]

According to IBM, 90 percent of all the data generated by smart devices are never analyzed. And about 60 percent of this data starts losing its value within milliseconds of being generated. [7]

The IOT concept refers to the use of objects, sensor devices, communication infrastructure, and data processing. The sensor devices can be RFID tags, electronic sensors, scanners, actuators and other sensing devices. The objects in IoT are the physical devices that are uniquely identifiable (for example with IPv6 addressing) and can be accessed through the internet. The data processing unit on the cloud helps decision making and control of the whole system. [8]

Sensors and actuators are physical devices that allow the IOT system to interact with the outside environment. In this case, the term *sensor* can be broadly defined as a device that provides inputs about the current state of the environment/system. The same way, an *actuator* can be defined as a device used to induce a change on the environment/system such as the temperature controller of a smart air conditioner.
The data collected by the different sensors can be stored and processed on the edge of the network or on a remote server. Data processing is typically more efficient when performed in close proximity to the physical device/object. This is because no time will be wasted traversing to a remote server and some decisions need to be made quickly without the need for a remote processing unit. Due to the processing capacity limitations on physical devices, part of the data or a pre-processed data can be sent to the remote server with bigger computing power for more robust and advanced analysis.

The exchange of the data happens based on protocols specifically designed for IOT applications. These applications are suitable for working with the constrained resources (like processing power, memory capacity, bandwidth and battery power) IoT devices have. The IOT is an implementation of more than just a single technology; instead it is a blending of different technologies that operate together. [9]

The following figure shows a simplified example of a typical IOT system. The system comprises three building blocks. The first is a set of IoT devices like sensors, antenna or microcontroller used for collecting data. The second one is the IoT hub used as the connectivity infrastructure on which the communication occurs. The third block represents the business applications, user interfaces or back-end systems on the server.

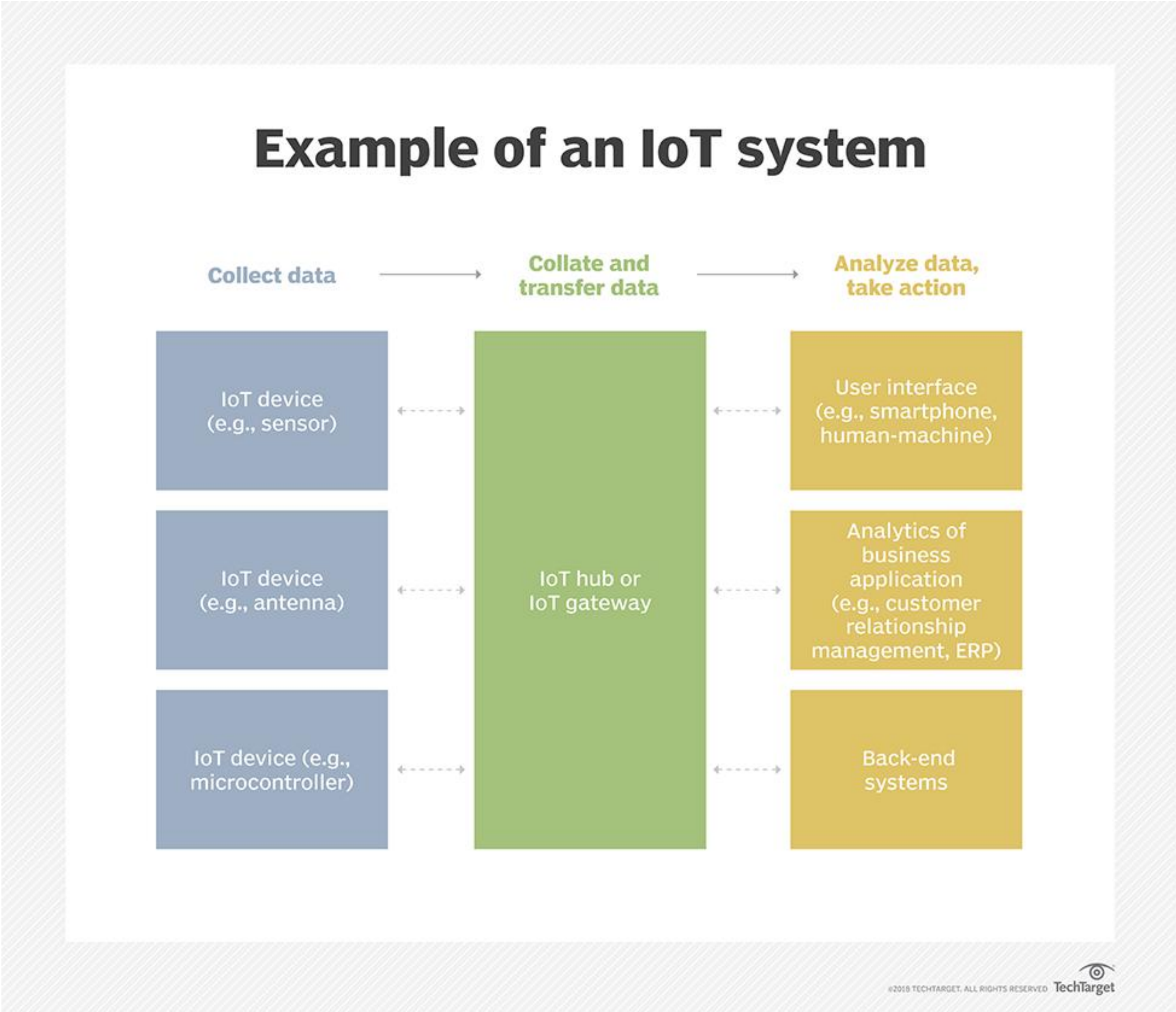


Figure 1: IoT system building blocks [10]

Figure 1 shows a simplified block diagram representation of the internet of things.

Among the numerous applications of the internet of things in the real-world are smart cars, wearable devices, smart appliances, home automation, agriculture, industrial IoT (IioT) and healthcare.

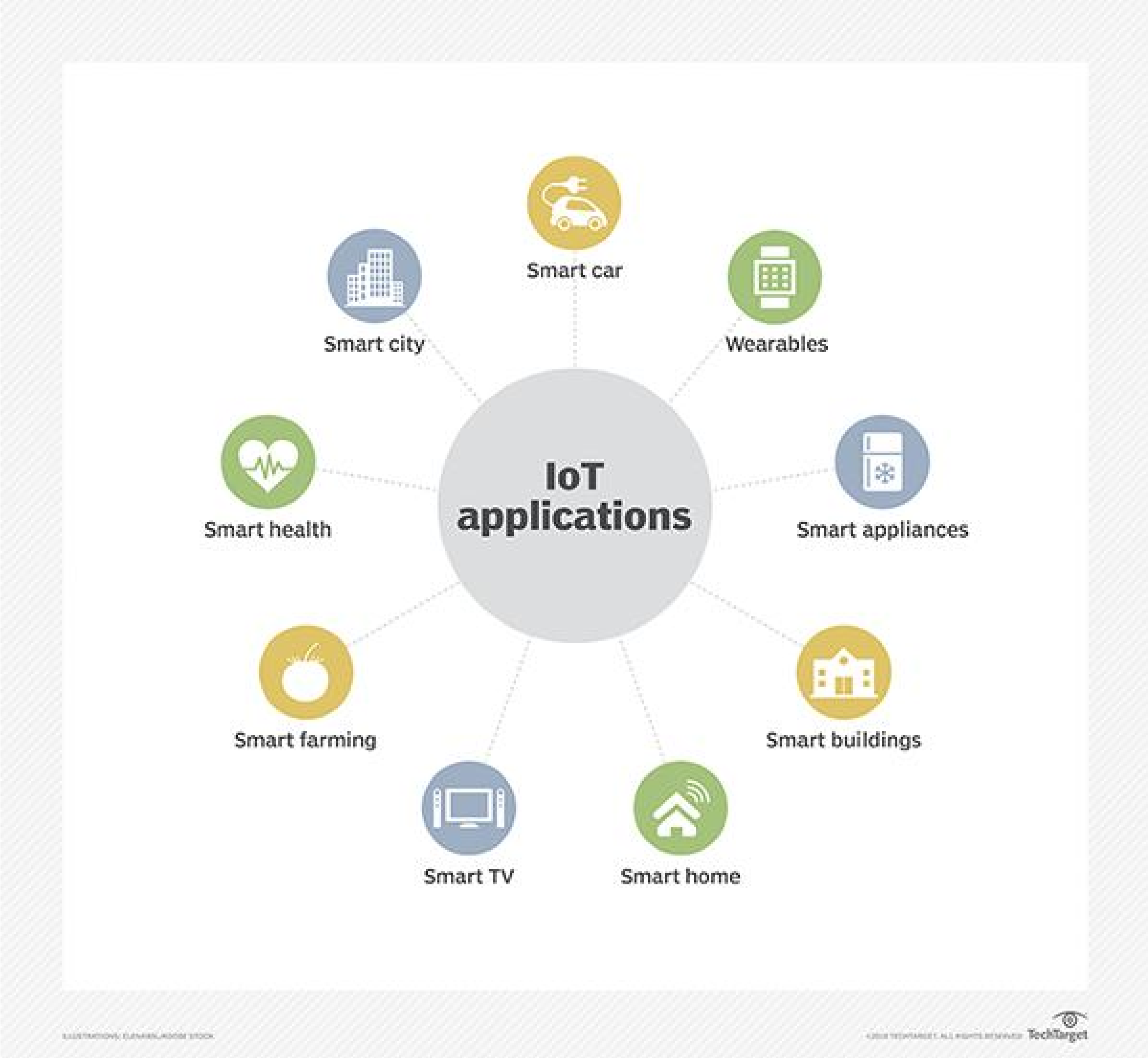


Figure 2: Example Applications of IoT [10]

Figure 2 shows that the areas of IoT applications are diverse and can affect the lives of billions of people.

Examples of IoT Applications:

Smart cars containing different vehicle sensors can be used to gather real-time data about the working condition of the car, its location, the road conditions, traffic condition, safety condition and others. IoT application in the automotive industry can improve fleet monitoring systems, logistics management, road traffic management, and safety. Preventive maintenance suggested by an IoT system can avoid operations breakdown thereby saving time and money. So a more efficient and safer running condition for cars can be achieved with proper implementation of IoT.

Wearable devices can collect and analyze data about physical fitness and health condition of a person.
Smart appliances like refrigerators can send important information to the user and also run more efficiently using IoT applications. The information can be about the refrigerator's working conditions or the amount/kind of food items inside it.

Smart buildings can increase their energy efficiency using temperature sensors and turning the air conditioner only when the temperature level goes below a certain temperature. Motion sensors can also be employed whereby the lighting bulbs will remain OFF if there is no motion detected in the rooms for a certain amount of time. The moment motion is detected, the lights will turn ON. Or smoke sensors can detect a potential fire accident and send alert message to the appropriate body.

IoT-based smart farming systems gather information about environmental conditions like rainfall level, temperature, humidity, soil moisture and others. Artificial air conditioning systems can then use this data to keep these parameters in a suitable range so that the plants will produce the most yields at the lowest cost.

Remote monitoring of patients health condition and analyzing the effectiveness of drugs can be performed using an IoT system in healthcare.

3 Architecture of IOT

The internet uses the TCP/IP protocols suit for communication between different hosts. Physical devices were not thought to be integrated into the internet at first, as mainly computers were considered to be able to do that. But the development of IoT protocols has enabled the connection of billions of physical devices globally.as more and more devices are connected to the internet, the traffic has become larger and more data storage & data processing capabilities are required. The different devices also operate with different protocols and standards. There is no uniform standard for the Internet of Things (IoT) to operate with. This poses a major challenge to the current internet infrastructure which was not designed to handle these additional billions of IoT devices. There would also be security related challenges.so when designing new IoT architecture, the issues of scalability, interoperability, and security need to be addressed properly. [11]

3.1 Three-Layer and Five-Layer IOT Architectures

Although there is not a widely agreed and accepted architecture of the Internet of Things (IoT), dividing the IOT system into three layers or five layers is the most widely used approach.

The three layer structure is considered as the basic IoT structure comprising of the perception layer, network layer, and application layer. The three layer structure is not sufficient when dealing with the detailed analysis of the Internet of things. This led to the addition of more layers and the proposition of five layer and seven layer architectures.

The five layer approach adds Middleware layer and Business Layer to the three layers on the basic three-layer architecture.

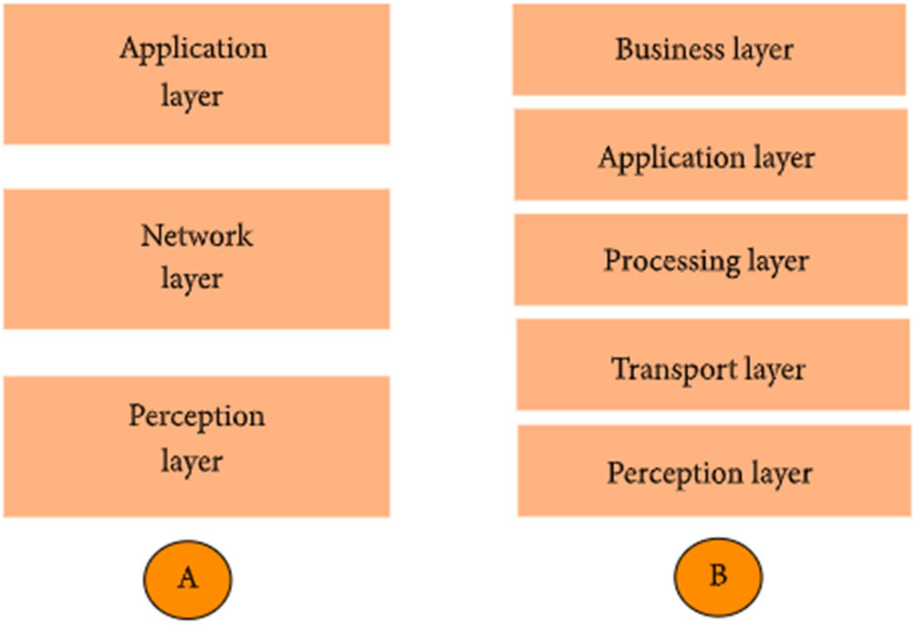


Figure 3: A (three layers) and B (five layers) [12]

Fig 3A shows the basic three layers; perception layer, Network layer and Application layer on top of each other while Fig 3B shows the five layer model.

The 5 layers of IOT architecture are discussed in the following sections.

3.1.1 Perception Layer

This layer consists of the physical devices and the sensing units. The main function of this layer is collecting and catching information. All kinds of sensors, sensor gateway, RFID Tag, camera, and GPS information can be included in this layer. Objects are also uniquely identified on this layer. Once the information is collected in this layer, it will be passed onto the Network layer.

3.1.2 Transport Layer

This layer consists of the converged network infrastructure with different kinds of communication technology. Data gathered from the perception layer is transmitted to the data processing unit through this layer.
Different kinds of communication technology are used on the network layer. Among these are 3G, Wi-Fi, Bluetooth, LPWAN, etc.

The data is then transferred to the Middleware layer. [12]

3.1.3 Middleware Layer

This layer performs information processing and management of the different devices and services in the IOT. The processed information is linked to a database. This layer can also decide which devices can and cannot communicate with each other.

3.1.4 Application Layer

This layer consists of application solutions that can bring Information technology solutions to the whole IOT (internet of things) architecture. Securely sharing information is the key issue on this layer.

3.1.5 Business Layer

This layer performs system management of the whole IOT architecture.
Careful analysis of results from this layer assist in making informed decisions about the business model. [12]

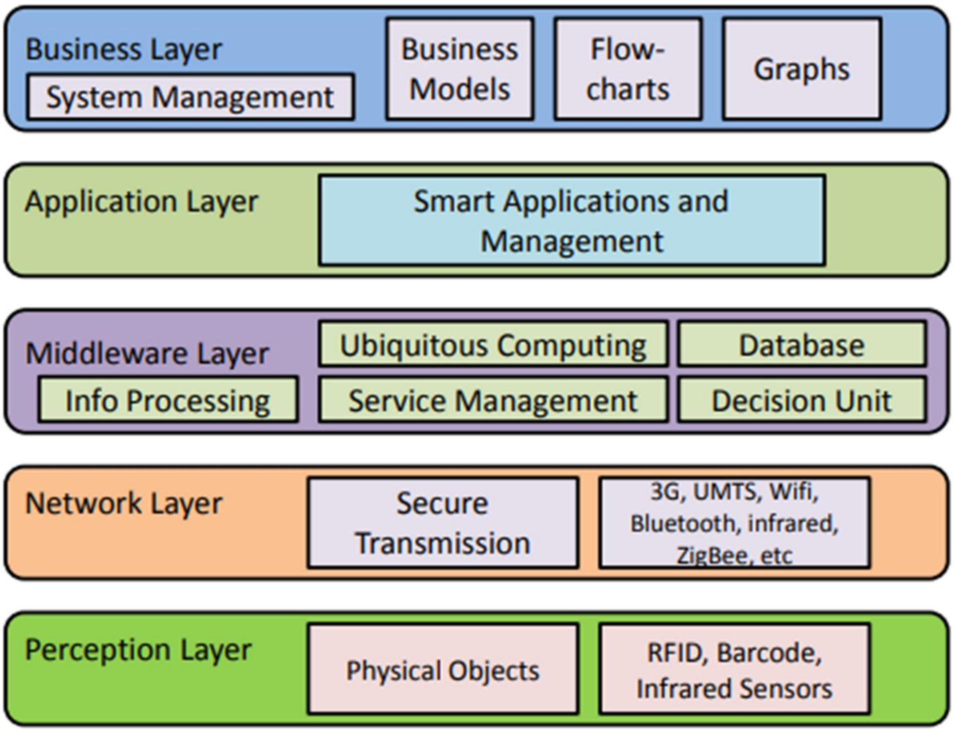


Figure 4: IoT architecture [13]

Figure 4 demonstrates the different layers and the corresponding protocols involved on the five layers.

3.2 Cloud and Fog Based Architecture

This architecture classification into cloud and Fog-based is based on the location of the data processing unit in the IoT system while the classification in the previous section(section 3.1) is on the basis of protocols used. [14]

3.2.1 Cloud Based Architecture

Cloud based architecture is a system where the data processing is done in cloud computers which usually have bigger processing power and storage capacity than machines in the vicinity of the IoT devices. [15]
Cloud based architecture offers scalable solutions as the processing power and storage capacity can easily be adjusted based on the complexity and type of the problem. The

scalability of this architecture allows application of a solution to a bigger network of things with large amounts of data to be processed.

3.2.2 Fog Based Architecture

This architecture involves data processing at the local vicinity of the physical devices. Sensors and network gateway devices perform part of the data processing.
Recently, there has been a move toward this architecture rather than relying exclusively on cloud resources located remotely. [16-18]

This architecture adds monitoring, preprocessing, storage, and security layers on top of physical layer and below the transport layer as shown in the figure below.

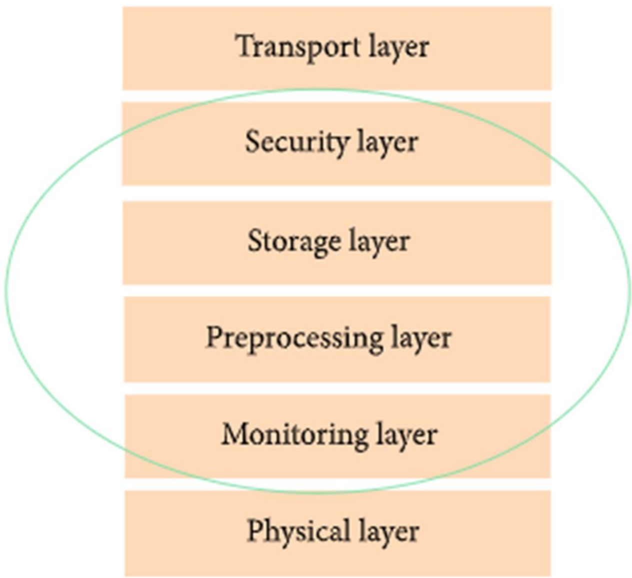


Figure 5: Fog architecture of a smart IoT gateway [19]

The seven layers used in an IoT architecture are shown in Figure 5.
The monitoring layer monitors power utilization, device resource utilization, device responses and different services.
The preprocessor layer partly performs data filtering, data processing, and analytics on data generated at the physical layer.
The storage layer performs data replication and data storage.
The security layer ensures data confidentiality and integrity of the data.

4 IoT Protocols

The TCP/IP model has 4 layers with different protocols employed at each layer.
The 4 layers in the TCP/IP stack are the Data Link layer, Network layer, Transport layer and Application layer.

4.1 Data Link Layer

This layer connects two or more physical devices together and a gateway device connects the whole set of devices to the internet. The physical devices or sensors can be thought of as devices in the Local Area Network (LAN).
The different devices within the LAN can communicate with each other without the need for the external internet. But if the communication involves a device outside the LAN, then data is routed to the internet by the gateway device.
The traditional data link layer protocols used by computers, mobiles, and tablets are not suitable for IoT application. They have big overhead frame size and consume too much power. IoT devices usually have less processing power, smaller data storage capacity and are expected to run with low power(usually on batteries) for extended time. This requirement led to the development of a different set of protocols for IoT applications. IoT protocols are normally required to have less overhead frame size and low power consumption. [20]

Some of the IoT protocols at the data link layer are:

- IEEE 802.15.4
- IEEE 802.11 AH
- WirelessHART

- Z-wave
- Bluetooth Low Energy
- Zigbee Smart Energy
- DASH7
- Long-Term Evolution Advanced (LTE-A)
- LoRaWAN

4.2 Network Layer Protocols

With the IPv4 addresses running out, the wider implementation of IPv6 addressing is enabling IoT devices to have uniquely identifiable IPv6 addresses. But since IPv6 addresses are too long with 128 bits, most IoT devices will have problem incorporating it. This led to the adoption of new IPv6 protocols for IoT devices. [20]
The following standards have been developed by the IETF for IoT applications.

- 6 LoWPAN (IPv6 over Low power Wireless Personal Area Network)
- 6TiSCH
- 6Lo (IPv6 over Networks of Resource-constrained Nodes)

4.3 Transport Layer Protocols

Different organizations have proposed and implemented different kinds of standards and transport protocols. IoT applications can use the TCP or UDP protocols for transport. But there are special transport protocols designed for IoT applications with consideration of the unique requirements in IoT devices. The protocols need to be light enough to run on IoT devices with low memory, low processing power, low power usage, low bandwidth and low cost. [21]

MQTT and CoAP are the two most common IoT transport protocols in use today.

4.3.1 MQTT (Message Queue Telemetry Transport)

MQTT is a lightweight transport protocol for IoT devices. It is also suitable for IoT devices with low and it consumes low power. It is the most widely used IoT protocol nowadays. [22]

MQTT uses publish/subscribe architecture which responds to new events. Subscribers register their interest in certain kinds of data from certain devices. Publishers are the source of new information. MQTT broker plays a central role between the publishers & subscribers and it helps in making sure that a device receives data that it previously registered for. [22]

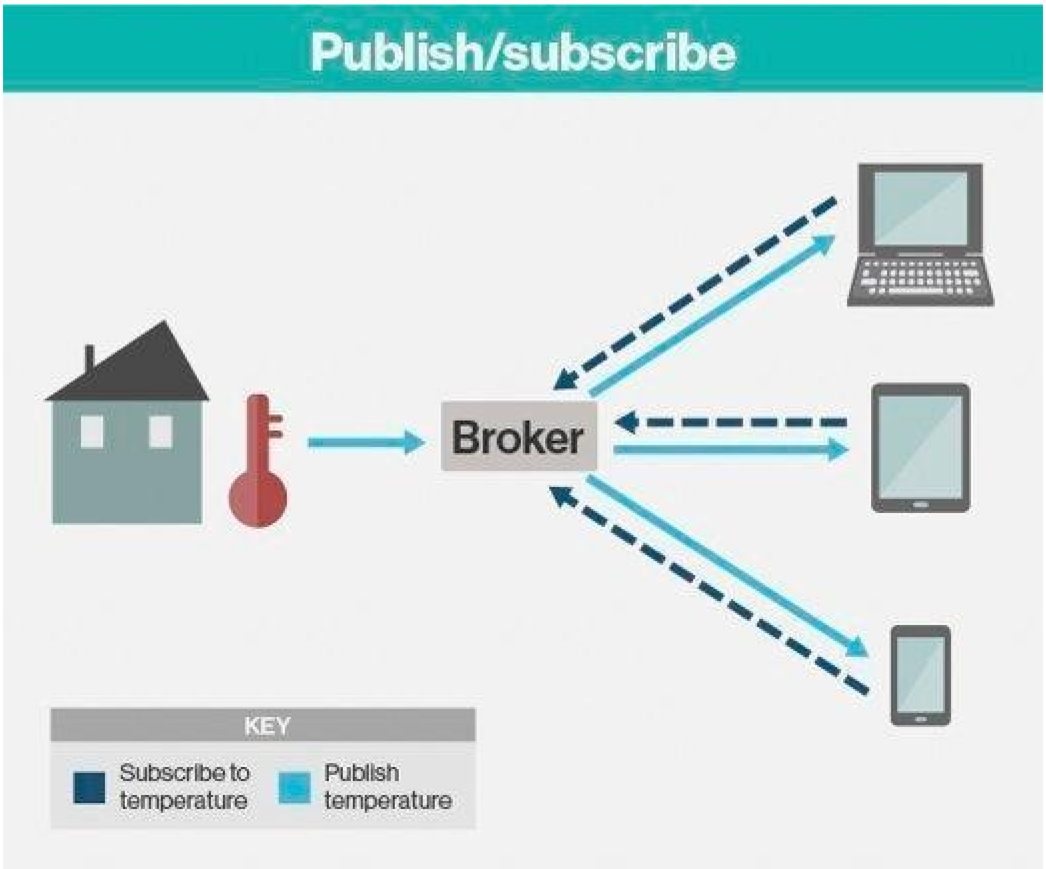


Figure 6: MQTT Publish & Subscribe Architecture [22]

Figure 6 shows that the devices on the right side first subscribe for the temperature data. When new temperature data is published to the broker, the broker publishes the data to the devices.
MQTT uses port 1883 for plain-text communication and port number 8883 for encrypted transfers.

4.3.2 CoAP (Constrained Application Protocol)

CoAP is a transfer protocol for IoT devices.it is implemented in machine-to-machine communication. It was designed by the IETF to provide lightweight RESTful interface between HTTP client and servers. But with the constrained computing and power resources in IoT, REST is not suitable for IoT devices. CoAP is designed to offer reliability in low bandwidth and high congestion.

With CoAP architecture following the client/server model, clients can make requests to servers (GET), clients can create new entries on the server (POST), servers can give appropriate response to the request (PUT) and servers can remove a certain entity based on a request (DELETE). [25]

CoAP can be represented in the following abstract layer,

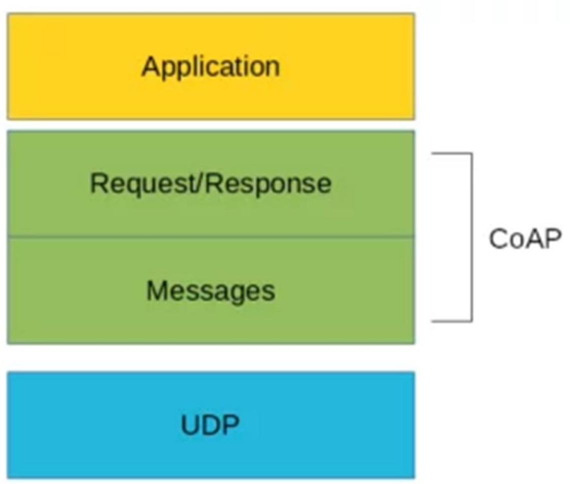


Figure 7: CoAP abstraction protocol layer [23]

As Figure 7 shows, CoAP has two sub layers: messages layer and request/response layer.

5 IoT Implementation for Air condition Monitoring

In this section of the thesis, the practical application of IoT using the Cayenne IoT platform is demonstrated by building an air condition monitoring system. The system consists of a temperature & humidity sensor (DHT11) connected to a raspberry pi. The sensor gathers the data from the atmosphere and forwards it to the raspberry pi. The raspberry pi performs pre-processing on the data and forwards it to the Cayenne IoT cloud which performs a more robust data analysis.

The special IOT session protocol; MQTT is used when collecting sensory data from devices and sending it to the cloud. A Python application for performing the MQTT session is installed on the raspberry pi. This application also performs the MQTT authentication between the publishing devices and subscriber applications on the cloud. The cayenne cloud acts as a broker for the MQTT process.
The Cayenne dashboard is the graphical user interface displayed on the Cayenne cloud application. The dashboard is available in both the web application and the mobile application (for both Android and IOS devices).

Users can get the access to the data and can control different aspects of the device operation using the cloud applications.it is possible to define rules algorithms on the dashboard using if….else statement. This allows performing a certain task when certain requirements are met. Notification and alert messages can also be configured to receive important information about the system via text message or an email.

The raspberry pi, DHT11 temperature & humidity sensor, the Cayenne IoT platform and the findings of the practical implementation are discussed in the following subsections.

5.1 The Raspberry Pi

The raspberry pi is a small single board computer used mainly for educational and prototype development purposes.it uses a Broadcom System on Chip (SoC) with an ARM-compatible central processing unit (CPU). All the series up to pi 3 model B+ have on-chip graphics processing unit (GPU).
Secure digital (SD) cards are used for storing the operating system, other programs and files.

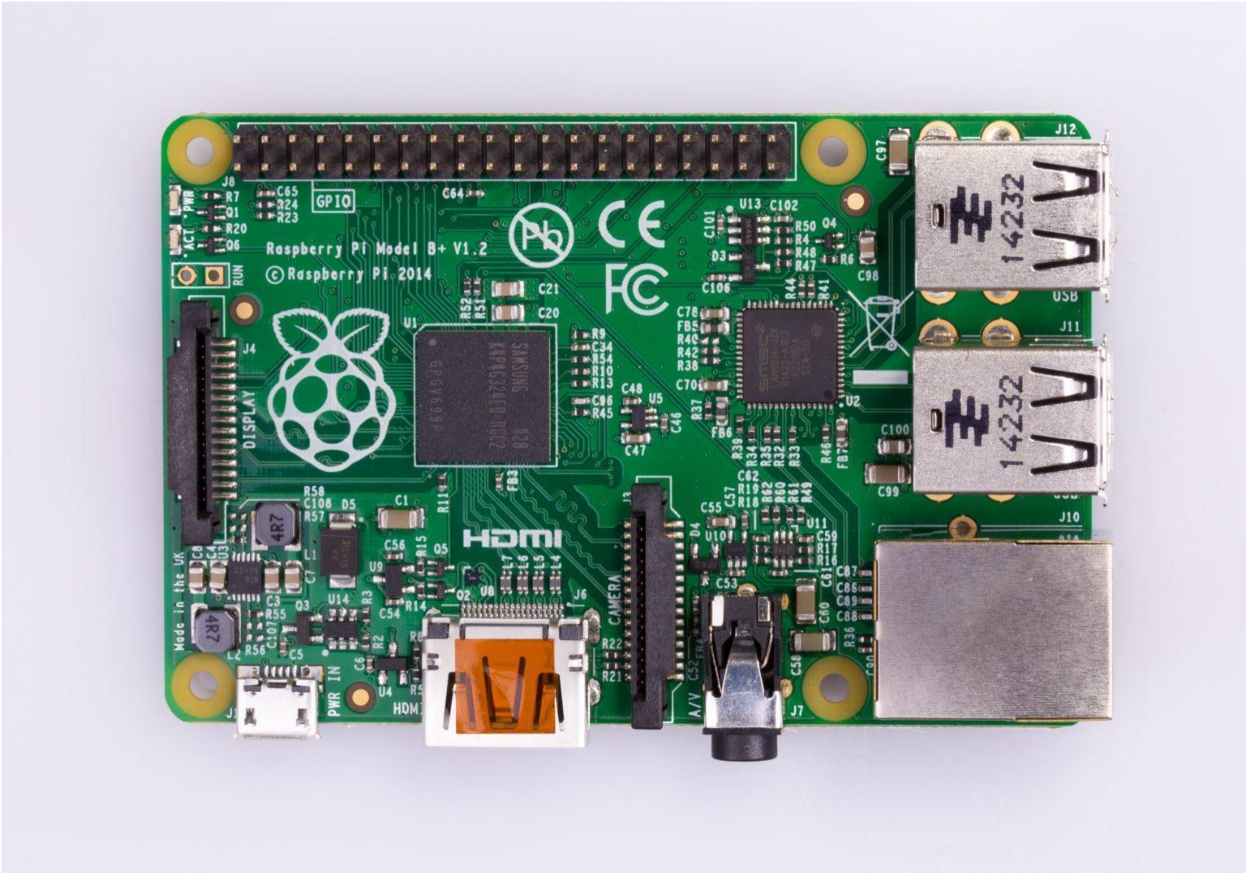


Figure 8: raspberry pi Model B+ v1.2 [24]

Figure 8 shows the raspberry pi model B+ v1.2, the different ports, the GPIO pins and the different PCB components.

The model used in this implementation project was Raspberry pi Model B+ v1.2. It was by developed and released by the *Raspberry pi Foundation* in 2014. This model has a Broadcom CPU clocked at 700MHz and 512MB RAM. It supports wireless, Bluetooth and Ethernet communication. Its specification can be summarized as follows: [25]

- Basic size, 85mm x 56mm
- Broadcom SoC clocked at 700MHz
- 512MB RAM soldered on top of the chip
- Micro USB power connector
- Dual step-down (buck) power supply for 3.3V and 1.8V
- 5V supply with polarity protection and 2A fuse
- 4 USB ports
- 40 GPIO pins

- HDMI port
- 3.5mm 'headphone' jack
- Camera and DSI(Display Serial Interface) Display connector
- MicroSD card socket
- Ethernet socket

The USB ports can be used to connect any USB-based devices like mouse, keyboard, Wi-Fi dongles, Bluetooth adapters, etc. external monitors can be connected to it using the HDMI connector.

The raspberry pi works with different operating systems; both Linux based operating systems and non-Linux based Operating systems like Windows 10 IoT core. But the Raspberry pi Foundation provides a debian-based Linux distribution called *raspbian* which can be downloaded from their website for free.

By connecting the raspberry pi to different sensors, and using its different connectivity options, different IoT projects can be implemented. The sensors will send the collected data to the raspberry pi for processing. The raspberry pi can process the data locally and make necessary decisions based on its program or if the data gathered is too much/too complex for the raspberry pi, it can send it to a more powerful remote server for a more detailed data analysis and storage.
In the IoT implementation for this thesis, the raspberry pi was connected to a temperature and humidity sensor (DHT11) and part of the data processing was performed by the raspberry pi and the remote Cayenne IoT cloud handled the more demanding and complex problems.

Locally the raspberry pi had been connected to the internet through an Ethernet cable.it could also be connected using the Wi-Fi which would require a Wi-Fi dongle connected to the USB port of the raspberry pi. The pi could be accessed using an SSH connection from another computer and programs could be installed and run remotely. SSH remote connection offers more security as the packets are encrypted during transmission. A command line interface (CLI) also consumes less processing power to the raspberry pi when compared with a graphical user interface.

MQTT was the chosen messaging protocol for the transport layer in this IOT project. The cayenne MQTT API was used to connect the raspberry pi to the remote cayenne cloud.

Once the pi is connected, data can be transferred from the sensors and displayed on the cayenne dashboard application. The data can also be displayed in the form of widgets. Commands can also be sent from the cayenne cloud to the local devices.

The cayenne cloud infrastructure acts as a broker in the MQTT architecture.it manages the sensors, actuators, applications and devices. MQTT uses a subscriber-publisher architecture model with the broker acting in the middle.

To connect and communicate with an MQTT, broker, an open-source MQTT client called the *eclipse paho project* was used. The *eclipse paho project* can be used for different languages like C/C++, Python, Java, JavaScript, Go and C#.

The eclipse paho MQTT Python library provides a client class that enables clients to communicate to the MQTT broker (the cayenne cloud) to publish messages. Different applications (subscribers) can also subscribe to different topics and receive the published messages.

5.2 DHT11 (Temperature & Humidity Sensor)

DHT11 is a temperature & humidity sensor complex with a digital signal output.it provides reliable and stable output by using digital-signal-acquisition technique. The sensor is composed of a resistive-type humidity measurement block and an NTC temperature measurement block. These blocks connect to an 8-bit microcontroller that can offer a high quality, fast response, and interference resistant capability.

It could be used in different applications such as HVAC, dehumidifier, weather stations, home automation, automotive, etc. [26]

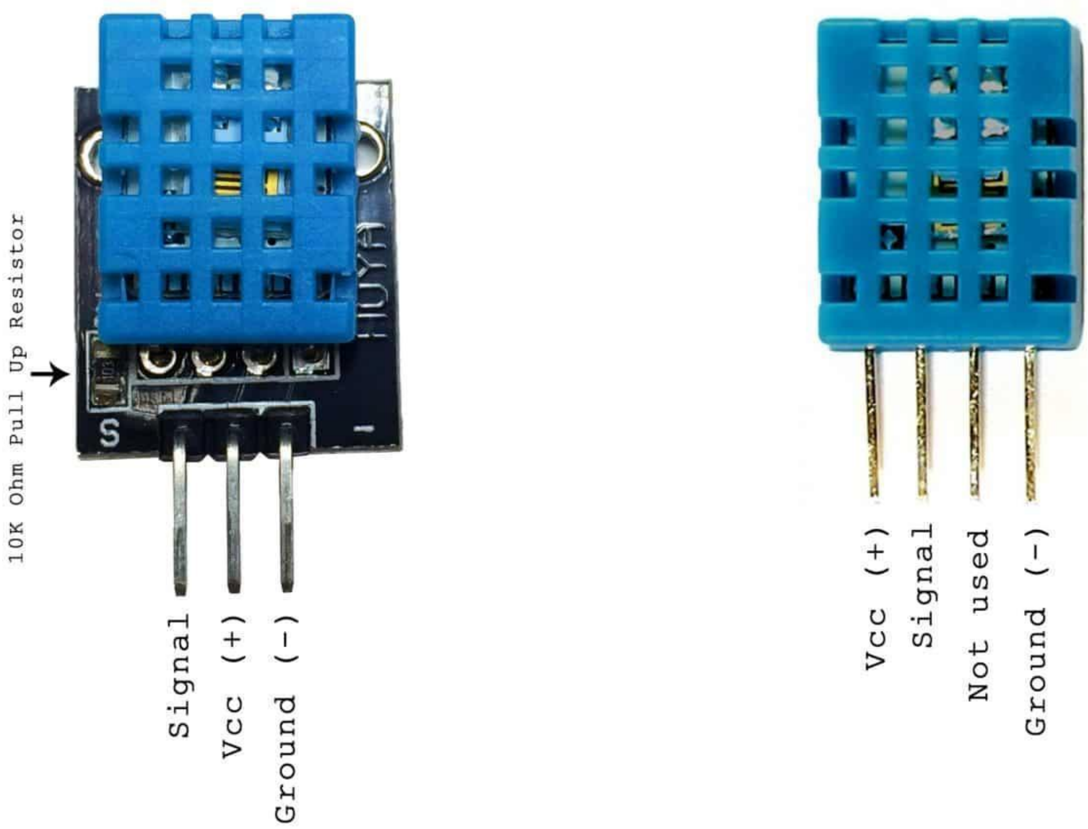


Figure 9: DHT11 [26]

Figure 9 shows the pins on the two variants of DHT11.the first one has three pins while the second has four pins.

5.3 The Cayenne IoT Platform

Cayenne is an IoT project builder for developers, designers, engineers or students and can be used in different IoT applications. It makes the IoT development process easier & faster. It has cloud based web applications as well as mobile apps for Android and IOS devices. Both the web and mobile apps have a main graphical user interface called the *dashboard*.

The Cayenne IoT platform was created by an IoT Solutions company called *myDevices*. *myDevices* was founded in 2013 by Kevin Bromer who also had been the CEO of the company.

The cayenne mobile apps can be used to monitor and control remotely located physical devices. The Cayenne dashboard contains widgets that can be configured to visualize data, create rules, schedule events, and make notification rules and more.

The Cayenne platform can be used to connect different kinds of sensors, lights, motors, valves, relay and generic actuators.it also supports the Raspberry pi, Arduino, ESP8266, LoRa devices and other development boards.

For this specific thesis, the raspberry pi was connected to the Cayenne IoT platform. A temperature and humidity sensor (DHT11) was connected to the raspberry pi and was used to monitor and control air temperature and Humidity.
The *raspbian* or any other Linux-based operating system need to be installed before starting installation of Cayenne on the raspberry pi. Cayenne installations take place in 4 steps:

1. Installing libraries
2. Installing agent
3. Installing software
4. Installing drivers

Once the installation is done, the Cayenne Online dashboard will appear automatically.it shows information about the raspberry pi model, the Operating system, CPU, RAM and Disk usage level.
Once sensors are connected to the raspberry pi and the raspberry pi is connected to the Cayenne cloud platform, advanced rules can be made on the dashboard. These rules can be about what needs to be performed when a certain reading level is reached.one simple example can be to turn ON a fan once the room temperature reading reaches a certain level.
The notification engine can send notifications through text messages or email. The rules for the notification can be configured from the dashboard as required.

Events scheduling can be used to perform certain actions during certain time intervals without the need for external action.

The Cayenne IoT cloud uses Cayenne MQTT API to connect devices. This lightweight protocol enables smooth transfer of data. MQTT is designed especially for IoT applications where there is a constraint of computing resources, battery power and bandwidth.

The Cayenne cloud acts as a broker and manages the different sensors and actuators (publishers).the applications running on the cloud act as subscribers that will receive data about events that they subscribed to.

The client devices used in this thesis interact with the MQTT broker (Cayenne cloud) using the Eclipse Paho project. The Eclipse Paho project offers open-source MQTT clients for C, C++, Python, Java, JavaScript, Go and C#. It comes bundled with the Cayenne libraries making it easy to use for this application. [27]

When establishing MQTT connection between devices and the Cayenne cloud, the MQTT credentials; MQTT Username, MQTT Password and Client ID are used. The values for each credentials need to be included in the code (Python in this case) on the raspberry pi. When the program with the MQTT parameters is compiled and run on the raspberry pi, an MQTT connection is established and data transfer starts. When new devices are added, these MQTT credentials from the online dashboard need to be used again. This ensures that the device is connected to the correct account. [28]

The Cayenne Cloud API enables interaction and development of applications using the mydevices IoT RESTful API. the REST API offered by *mydevices* uses OAuth2 protocol which offers increased security.in order to utilize the Cayenne Cloud API, the App Key and App Secret need to be obtained after signing up for the service. After using these applications credentials, the custom applications can connect to the Cayenne cloud API. Cayenne offers authorization and Authentication features to ensure data security and all transport communications are encrypted using TLS/SSL endpoints. [29]

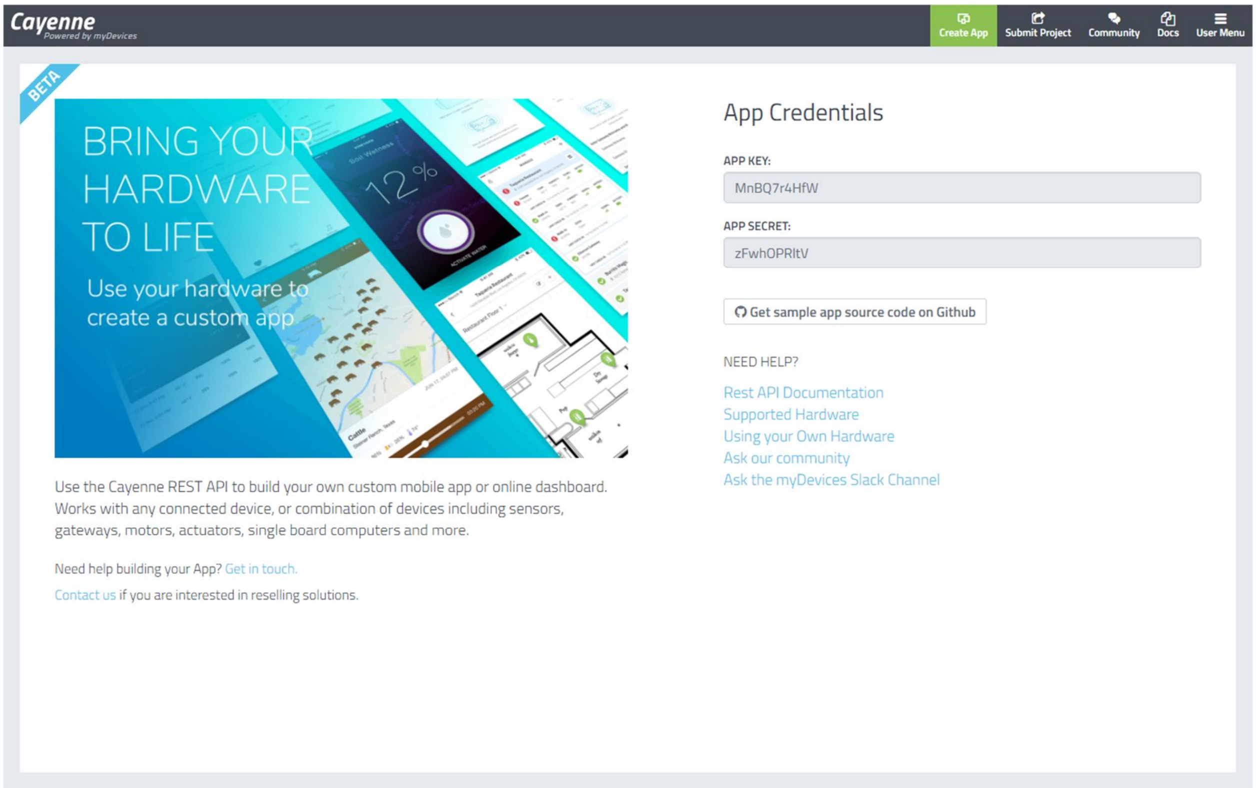


Figure 10: Cayenne Cloud API Authentication [30]

Figure 10 shows the credentials needed for the API authentication.

5.4 Cayenne Practical Implementation

This section will describe the steps involved in connecting the DHT11 (Temperature and Humidity) sensor to the raspberry pi, installing the Cayenne software on the raspberry pi and then setting up the connection between the raspberry pi and the Cayenne cloud. Cayenne application dashboard features are also shown at the end.
Once the connection is setup, then sensory data can be uploaded to the Cayenne cloud using the Cayenne MQTT Python Library. The Cayenne dashboard application allows defining rules algorithm, scheduling important events for devices and sending alert (notification) to the user. [31]

The following steps were followed for this implementation:

1. The first step will be creating a free account on mydevices Cayenne. [32] Cayenne offers paid services which are preferable for more scalable solutions with more features added.

2. Register/connect the raspberry pi to the account created earlier.this can be done either through a smartphone or the SSH terminal to the raspberry pi.the SSH terminal option was used in this thesis, and the cayenne software was installed.

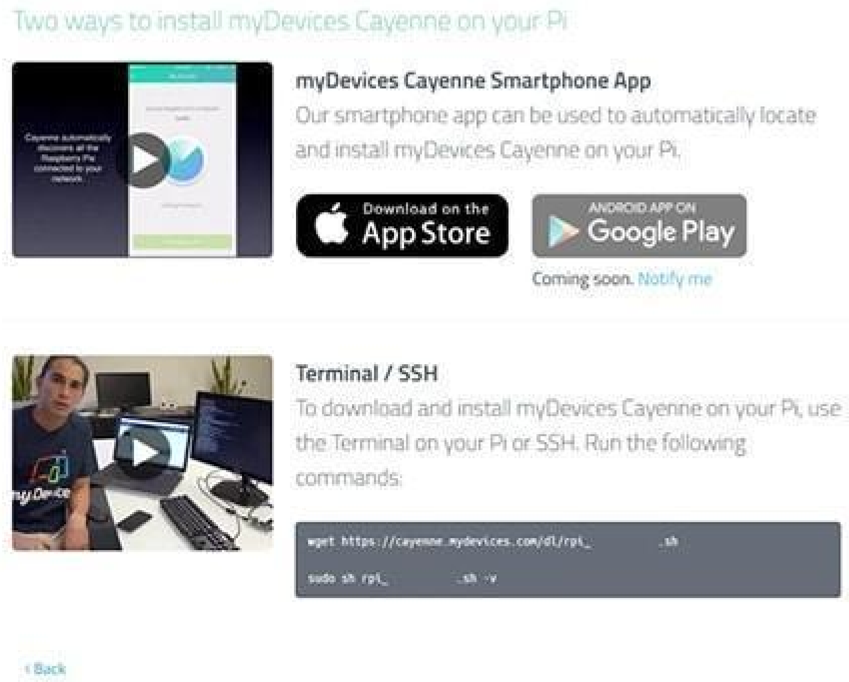


Figure 11: Cayenne Installation and connection [33]

As it can be seen from Figure 11, there are two options of installing myDevices Cayenne on the raspberry pi, using mobile phone application or through the SSH terminal.

3. Connecting the DHT11 sensor to the raspberry pi

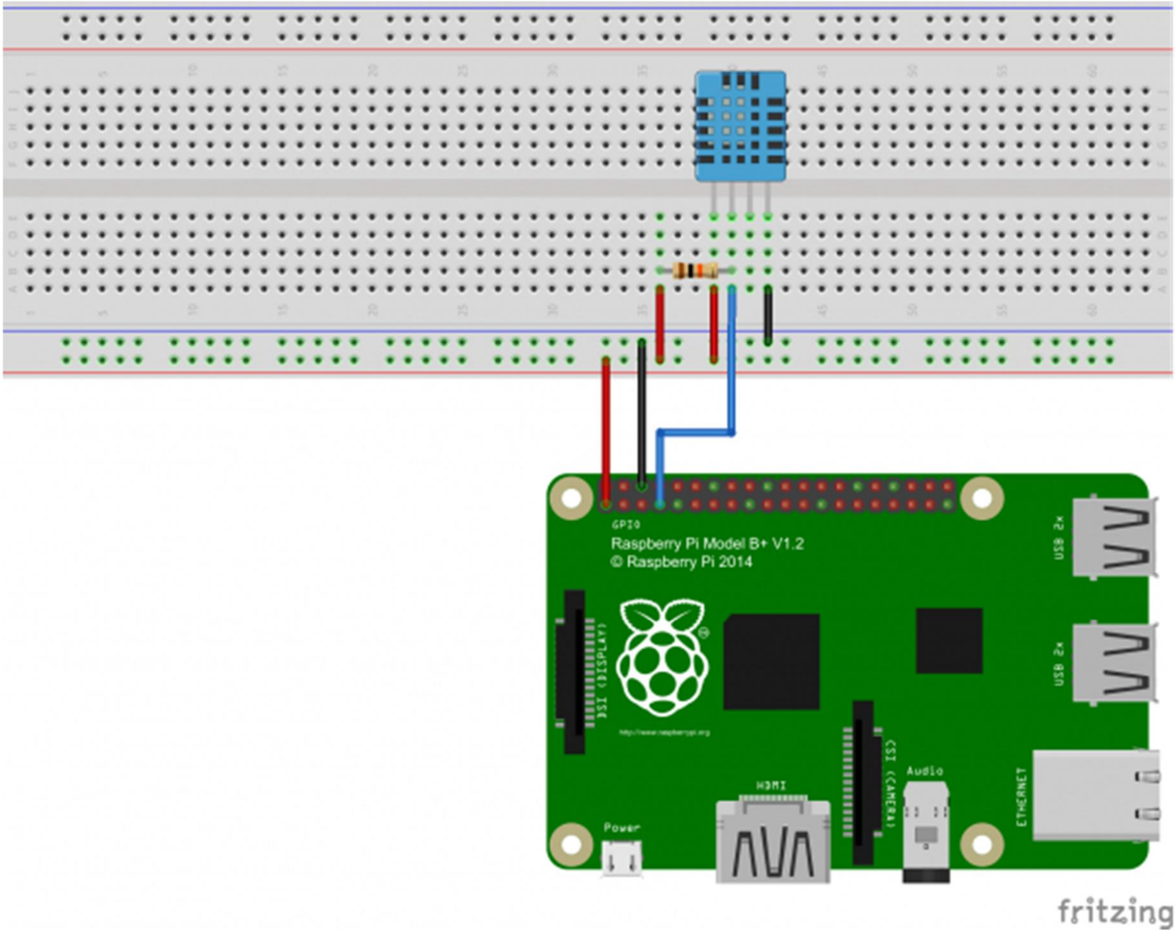


Figure 12: DHT11-raspberry pi connection [34]

Figure 12 shows the pin connection between the raspberry pi and DHT11.

4. Once Cayenne is installed, then the Cayenne dashboard will appear and show the device

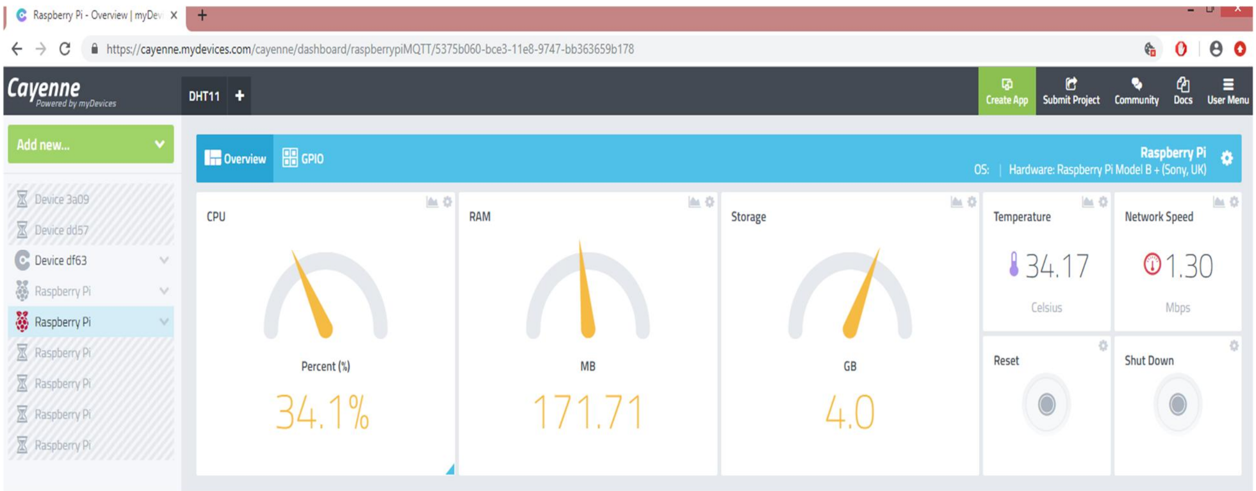


Figure 13: Device Information on the raspberry pi

Figure 13 shows basic information about the raspberry pi like the Operating System, device model, and the specifications of the device. The temperature and humidity values are also displayed.

5. Setting up an SSH connection to the raspberry pi and installing the following packages and libraries. [37]

First the following packages were installed:

```
sudo apt-get update
sudo apt-get install build-essential python-dev python-openssl
git
```

Then a pre-built library from Adafruit was used (other library options are possible) [37]

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git &&
cd Adafruit_Python_DHT
sudo python setup.py install
```

Once the library is integrated, the sensor reading can be obtained by running the AdafruitDHT.py file in the examples folder using the following commands:

```
cd examples
sudo ./AdafruitDHT.py 11 6
```

The number 11 in the second command indicates that DHT11 was used and the number 6 is the GPIO number on the raspberry pi to which the sensor is connected. The readings on the terminal can be seen as follows.

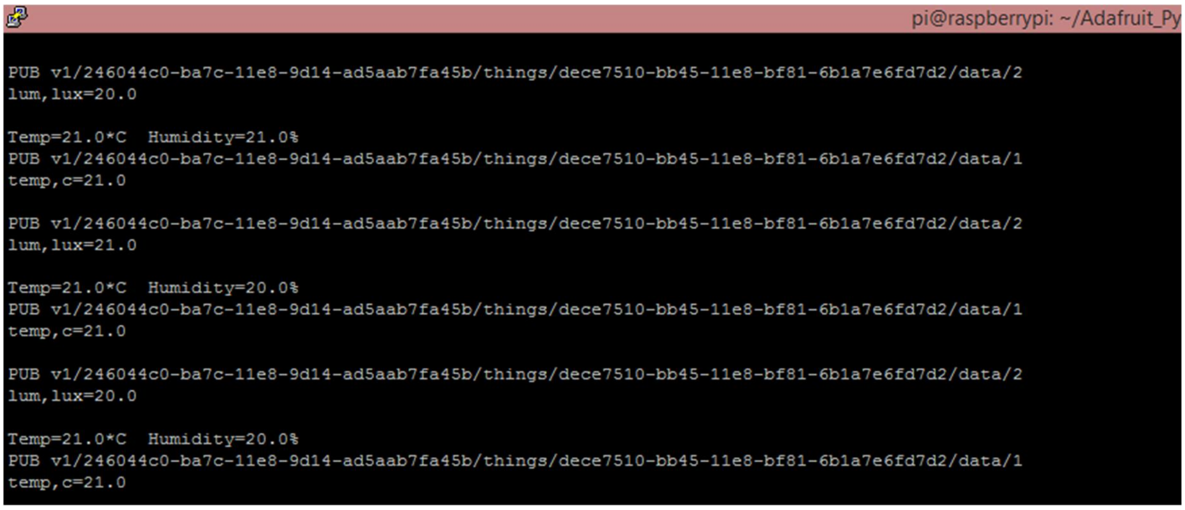


Figure 14: Sensor readings on the Terminal

The results in Figure 14 are from the SSH connection to the raspberry pi.

6. With the MQTT authentication checked to be correct, the raspberry pi connected to the Cayenne cloud properly, the dashboard setting for the sensory reading configured, the following result was obtained.

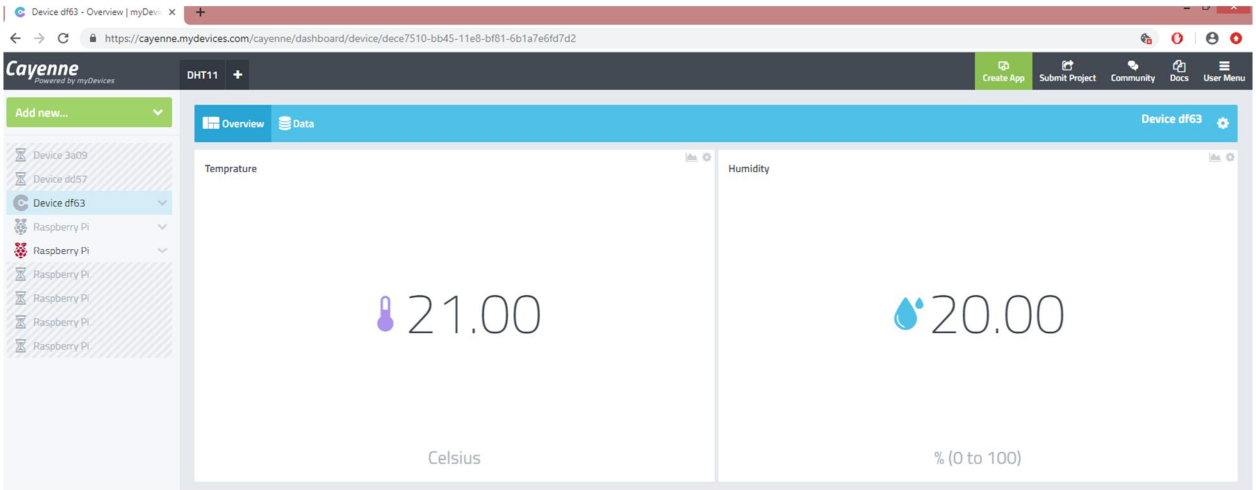
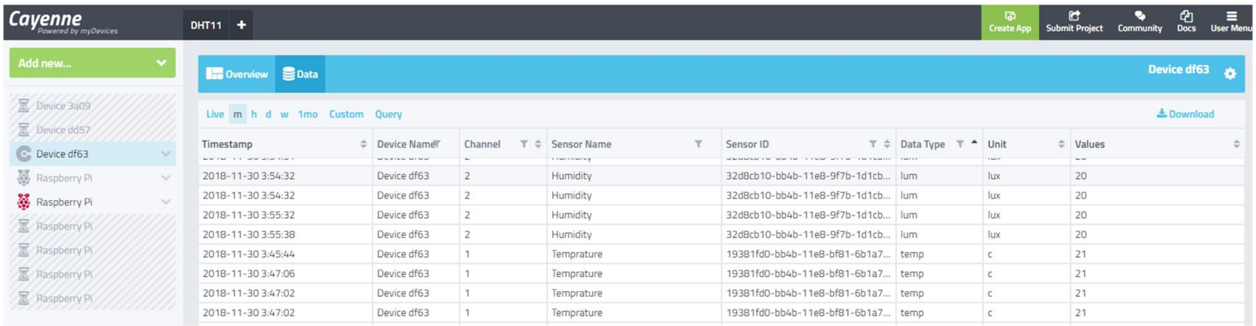


Figure 15: DHT11 reading on Cayenne

Figure 15 shows the Temperature and humidity readings from the Cayenne dashboard.

The reading with their corresponding timestamp can also be displayed in Excel spread-sheet format as follows:



Timestamp	Device Name	Channel	Sensor Name	Sensor ID	Data Type	Unit	Values
2018-11-30 3:54:32	Device df63	2	Humidity	32d8cb10-bb4b-11e8-9f7b-1d1cb...	lum	lux	20
2018-11-30 3:54:32	Device df63	2	Humidity	32d8cb10-bb4b-11e8-9f7b-1d1cb...	lum	lux	20
2018-11-30 3:55:32	Device df63	2	Humidity	32d8cb10-bb4b-11e8-9f7b-1d1cb...	lum	lux	20
2018-11-30 3:55:38	Device df63	2	Humidity	32d8cb10-bb4b-11e8-9f7b-1d1cb...	lum	lux	20
2018-11-30 3:45:44	Device df63	1	Temprature	19381fd0-bb4b-11e8-bf81-6b1a7...	temp	c	21
2018-11-30 3:47:06	Device df63	1	Temprature	19381fd0-bb4b-11e8-bf81-6b1a7...	temp	c	21
2018-11-30 3:47:02	Device df63	1	Temprature	19381fd0-bb4b-11e8-bf81-6b1a7...	temp	c	21
2018-11-30 3:47:02	Device df63	1	Temprature	19381fd0-bb4b-11e8-bf81-6b1a7...	temp	c	21

Figure 16: DHT11 readings with their timestamps

Figure 16 show the tabular form of the sensor readings.

The result can also be visualized in graphical format as shown below:

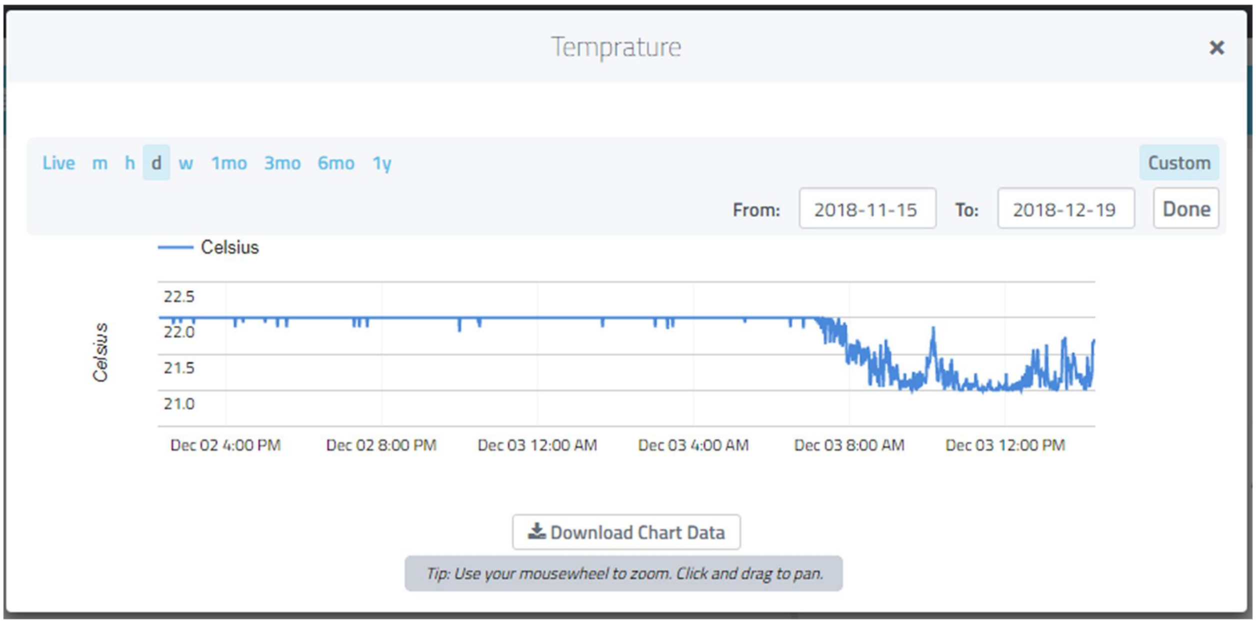


Figure 17: Graphical output of Temperature values

Figure 17 shows a graphical output for the Temperature readings.

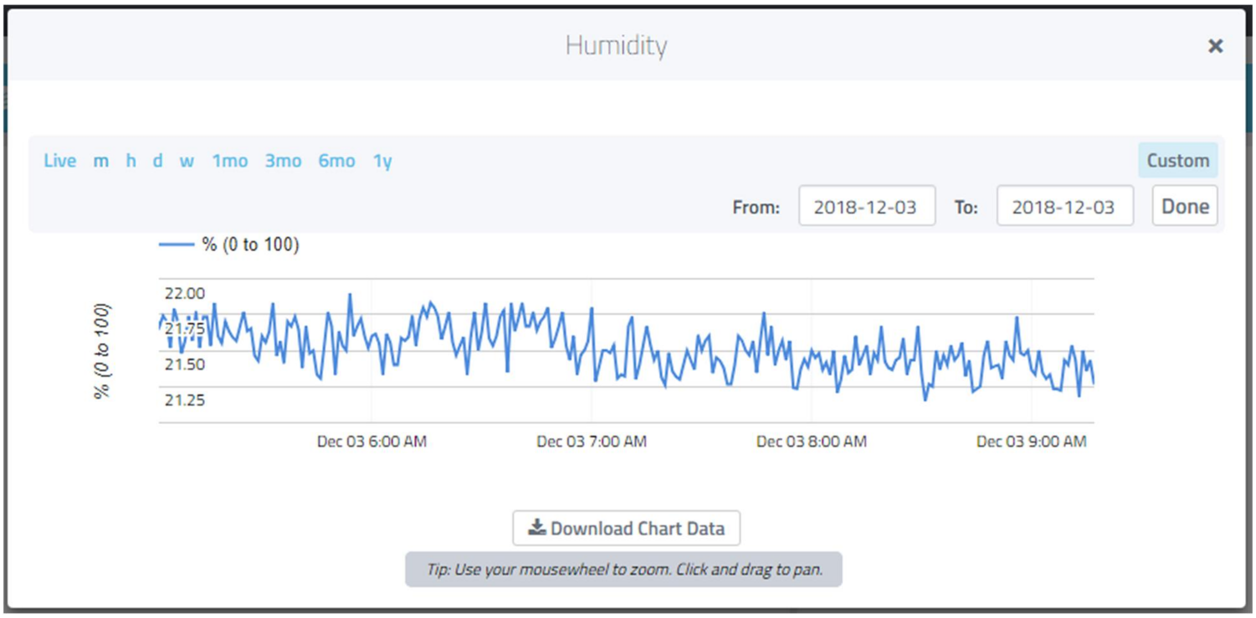


Figure 18: Graphical output of Humidity Values

Figure 18 shows the graphical output of the Temperature readings.

6 Conclusion

The objective of this thesis was to study Internet of Things and implement a practical IoT solution using Cayenne cloud platform. The background theory about IoT had been covered in this thesis.

There had been a need to come up with a widely accepted and used IoT architecture model. But no single standard had be brought into practice. Instead there are different IoT architecture models in use today. The absence of a uniform standard among the IoT industry poses a challenge in terms of interoperability of different architecture.

The three layer architecture and five layer architecture are the most widely used models for IoT. The three layer option comprises basic component blocks of IoT.
The three layer model offered only the basic blocks in IoT and there was a need for a more detailed block layering and hence the five layer architecture was introduced.

IoT devices do not have the processing power, memory and bandwidth capability that other network devices have. The different protocols in the internet protocol suite would not run smoothly on an IoT device with these communication and computation constraints. This led to the creation of different working groups in the *IETF (Internet Engineering Task Force)* standards organization. These working groups developed open internet protocol standards for IoT systems. These different IoT protocols have been discussed in this thesis.

The IoT implementation with the Cayenne platform was used to demonstrate the practical application of IoT. This showed that connecting devices, transferring data and running different applications is simpler when performed using the Cayenne platform. The platform also offers scalable solution that can be applied in a system with hundreds or thousands of devices. Since all data is encrypted during transmission, it would be more difficult for an attacker to get access to the data in case of a cyber-attack.

Although the internet of things is expanding rapidly and more devices are being connected to the internet, there are some challenges facing the IoT industry. Some of these challenges are security, interoperability, weak internet infrastructure and lack of governmental regulation. Currently different actions are being undertaken by the different stakeholders involved to address these challenges and there had been advances made.

These developments need to continue if the industry is to have a more secure and interoperable system that can support billions of devices.

References

1. Huang Y, Li G. Descriptive models for Internet of Things. 2010. URL: https://ieeexplore.ieee.org/document/5564232 [Accessed 11 Dec 2018]

2. Fan T, Chen Y. A scheme of data management in the Internet of Things.2010. 2nd IEEE International Conference on Network Infrastructure and Digital Content URL: https://ieeexplore.ieee.org/document/5657908 [Accessed 11 Dec 2018]

3. Huang Y. A Semantic Analysis for Internet of Things.2010 International Conference on Intelligent Computation Technology and Automation. May 2010.Volume 1 URL: https://ieeexplore.ieee.org/document/5523336 [Accessed 11 Dec 2018]

4. Gartner Inc. Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016. 2016. URL: https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016 [Accessed 11 Dec 2018]

5. Satpute S, Deora BS. Improving performance of Internet of Things by using local IOT controller unit. In Green Computing and Internet of Things (ICGCIoT), International Conference on 2015 Oct 8 (pp. 1328-1330). IEEE.2015. URL: https://www.computer.org/csdl/proceedings/icgciot/2015/7910/00/07380672-abs.html [Accessed 11 Dec 2018]

6. Evans D. The internet of things: How the next evolution of the internet is changing everything. 2011. URL: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf [Accessed 11 Dec 2018]

7. Singh S, Singh N. Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce. In Green Computing and Internet of Things (ICGCIoT), International Conference on 2015 Oct 8 (pp. 1577-1581). IEEE. 2015. URL: https://www.researchgate.net/publication/297712965_Internet_of_Things_IoT_Security_challenges_business_opportunities_reference_architecture_for_E-commerce [Accessed 11 Dec 2018]

8. Y. Yu, J. Wang, and G. Zhou. The Exploration in the Education of Professionals in Applied Internet of Things Engineering in 4th International Conference on Distance Learning and Education (ICDLE), October 2010. URL: https://ieeexplore.ieee.org/document/5606038 [Accessed 11 Dec 2018]

9. Sethi P, Sarangi SR. Internet of things: architectures, protocols, and applications. Journal of Electrical and Computer Engineering. 2017. URL: https://www.hindawi.com/journals/jece/2017/9324035/abs/ [Accessed 11 Dec 2018]

10. IoT Agenda. What is internet of things (IoT)?. 2018. URL: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT [Accessed 11 Dec 2018]

11. Tan L, Wang N. Future internet: The internet of things. In Advanced Computer Theory and Engineering (ICACTE), 3rd International Conference on 2010 Aug 20 (Vol. 5, pp. V5-376). IEEE.2010. URL: https://www.researchgate.net/publication/261311447_Future_Internet_The_Internet_of_Things_Architecture_Possible_Applications_and_Key_Challenges [Accessed 11 Dec 2018]

12. Sethi, Sarangi, R. S. Internet of Things: Architectures, Protocols, and Applications. 2017. URL: https://www.hindawi.com/journals/jece/2017/9324035/#B93 [Accessed 11 Dec 2018]

13. Khan R, Khan SU, Zaheer R, Khan S. Future internet: the internet of things architecture, possible applications and key challenges. InFrontiers of Information Technology (FIT), 10th International Conference on 2012 Dec 17 (pp. 257-260). IEEE. 2012. URL: https://pure.qub.ac.uk/portal/files/81384964/PID2566391.pdf [Accessed 11 Dec 2018]

14. Weyrich M, Ebert C. Reference architectures for the internet of things. IEEE Software. 2016 Jan; 33(1):112-6.URL: https://electrovolt.ir/wp-content/uploads/2018/03/Internet-of-Things-Challenges-Advances-and-Applications-ElectroVolt.ir_.pdf [Accessed 11 Dec 2018]

15. Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future generation computer systems. 2013 Sep 1;29(7):1645-60.URL: https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/ReferencesPapers.aspx?ReferenceID=1507740 [Accessed 11 Dec 2018]

16. Bonomi F, Milito R, Natarajan P, Zhu J. Fog computing: A platform for internet of things and analytics. In Big data and internet of things: A roadmap for smart environments 2014 (pp. 169-186). Springer, Cham. URL: http://www.sciepub.com/reference/122440 [Accessed 11 Dec 2018]

17. Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the internet of things. InProceedings of the first edition of the MCC workshop on Mobile cloud compu-

ting 2012 Aug 17 (pp. 13-16). ACM. URL: https://pdfs.semanticscholar.org/5fe4/c23964c165f42b4e0cd98bf686d04501e628.pdf [Accessed 11 Dec 2018]

18. Stojmenovic I, Wen S. The fog computing paradigm: Scenarios and security issues. In Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on 2014 Sep 7 (pp. 1-8). IEEE. URL: https://ieeexplore.ieee.org/document/6932989 [Accessed 11 Dec 2018]

19. Sethi, Sarangi, R. S. Internet of Things: Architectures, Protocols, and Applications. 2017. URL: https://www.hindawi.com/journals/jece/2017/9324035/#B93 [Accessed 11 Dec 2018]

20. Salman Tin. A Summary of Network Traffic Monitoring and Analysis Techniques.2018. URL: https://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/#Wave [Accessed 11 Dec 2018]

21. HiveMQ. MQTT 101 - How to get started with the lightweight IoT protocol. 2017. URL: https://www.hivemq.com/blog/how-to-get-started-with-mqtt [Accessed 11 Dec 2018]

22. IoT Agenda. What is MQTT (MQ Telemetry Transport)? . 2018. URL: https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport [Accessed 11 Dec 2018]

23 Azzola F. CoAP Protocol. 2018. URL: https://www.javacodegeeks.com/2018/11/coap-protocol-step-step-guide.html [Accessed 11 Dec 2018]

24. Raspberry pi Foundation. 2018. Raspberry Pi 1 Model B. URL: https://www.raspberrypi.org/products/raspberry-pi-1-model-b-plus/ [Accessed 11 Dec 2018]

25. Adafruit Industries. Raspberry Pi Model B 512MB RAM. 2018. URL: https://www.adafruit.com/product/1914 [Accessed 11 Dec 2018]

26. Circuit Basics. How to Set Up the DHT11 Humidity Sensor on the Raspberry Pi.2018. URL: http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi [Accessed 11 Dec 2018]

27. myDevices, Cayenne features. 2018. URL: https://mydevices.com/cayenne/docs_stage/getting-started/#getting-started-raspberry-pi. [Accessed 11 Dec 2018]

28. myDevices, Cayenne features. 2018. URL: https://mydevices.com/cayenne/docs_stage/getting-started/#getting-started-raspberry-pi. [Accessed 11 Dec 2018]

29. myDevices Inc. cayenne-mqtt-api. 2018. URL: https://mydevices.com/cayenne/docs_stage/cayenne-api/ [Accessed 11 Dec 2018]

30. myDevices. Cayenne cloud API. 2018. URL: https://mydevices.com/cayenne/docs/cayenne-api/#cayenne-cloud-api-http-api-reference-using-the-http-api-authenticating [Accessed 11 Dec 2018]

31. RasPiNews. Monitoring Temperature and Humidity Using Cayenne, Raspberry Pi and DHT11.2018.URL: https://www.raspinews.com/dht11-raspberry-pi-cayenne [Accessed 11 Dec 2018]

32. myDevices Inc. 2018. URL: https://accounts.mydevices.com/auth/realms/cayenne/protocol/openid-connect/auth?response_type=code&scope=email+profile&client_id=cayenne-web-app&state=omiym5UCmN9NGN0dmbYboFqHRmuQUxlZr7B7GSqy&redirect_uri=https%3A%2F%2Fcayenne.mydevices.com%2Fauth%2Fcallback [Accessed 11 Dec 2018]

33. Raspberry Pi Tutorials. Raspberry Pi: Measure Humidity and Temperature with DHT11/DHT22.2018. URL: https://tutorials-raspberrypi.com/raspberry-pi-measure-humidity-temperature-dht11-dht22/ [Accessed 11 Dec 2018]

34. Pi My Life Up. Getting Started with Raspberry Pi Cayenne. 2018. URL: https://pimylifeup.com/raspberry-pi-cayenne [Accessed 11 Dec 2018]

Appendix 1

1 (1)

Title of the Appendix

IoT Mind Map with MQTT