Projet S9 - SENSORA

THE SENSORA TEAM

Chloë BOURDIN	cbourdin@enseirb-matmeca.fr
Maud BUISSON	mbuisson@enseirb-matmeca.fr
Guillaume CHAPOTOT	gchapotot@enseirb-matmeca.fr
Thierno DIALLO	tdiallo@enseirb-matmeca.fr
Saad EL ABOUDI	selaboudi@enseirb-matmeca.fr
Mohamed MEKKAOUI	mmekkaoui@enseirb-matmeca.fr
Sam SANDERSON	ssanderson@enseirb-matmeca.fr

Abstract

The Internet of Things (IoT) is emerging across the globe at an alarming rate, sensory connected objects are useful tools to evaluate a room's temperature, humidity and other parameters. The project's objective consisted of designing a connected object network using Long Range (LoRa) and a comparative study of different possible services.

A LoRa-compatible gateway capable of detecting the device's messages, captures and forwarding them towards a cloud named The Things Network (TTN), a collaborative cloud associated with a database.

HTTP, MQTT and Actions on Google are different solutions used for gathering data, each solution was compared with each other. Different services were also created: a website, an application and Dialogflow, which operates with voice recognition.

Contents

Introduction **2**

1 Global Presentation **3**

2 Project Management **4**
2.1 Work Methodology 4
2.2 Team Presentation 4

3 Embedded System **4**
3.1 The Connected Object's Architecture 5
3.2 Network Communication 5

4 The Things Network **5**
4.1 Gateway 6
4.2 Application 6
4.3 Devices 6

5 Data **6**
5.1 Data gathering 6
5.2 Integration of TTN with Firebase & Cayenne 7
5.2.1 TTN - Cayenne 7
5.2.2 TTN - Firebase 7

6 Comparative Study **8**

7 Services **9**
7.1 Application Web and Mobile 9
7.2 DialogFlow 9

Conclusion **9**

List of Figures

1 Sensora's Global Architecture 3
2 Cayenne's Dashboard 7
3 Architecture of TTN data extraction with Firebase 7
4 Cayenne and HTTP Integration's Comparative Study 8

Introduction

Our society has experienced a major progress in the world of IoT, which is the interconnection between the Internet and objects: these objects could be devices, buildings, vehicles and other various embedded systems. This interconnection will allow these objects to collect, share and exchange different types of data to various destinations and applications.

Over the last few years, the number of connected objects has exponentially increased. In 2025, it is expected for the amount of connected objects to reach over 75 billion worldwide. Connected objects and the IoT have invaded people's daily lives, their conception is a subject at the heart of innovation and business. The objective of SenSora is to realize a comparative study of different solutions for the creation of a connected object network and, more particularly, of its software layer.

Three main aspects were taken into consideration for the SenSoRa project's objective. The first part was on the embedded side of the project, to understand and manipulate a connected object in order to collect and transmit data. The second was to implement the Data section, which has the role of directly exploiting the collected data. The third was to create different services to interact with the exploited data for the end user. Each method for the various sections are then compared to one another for their advantages and disadvantages.

1 Global Presentation

Figure 1 represents the project's global architecture with the different components and links between each entity which are explained throughout the rapport.

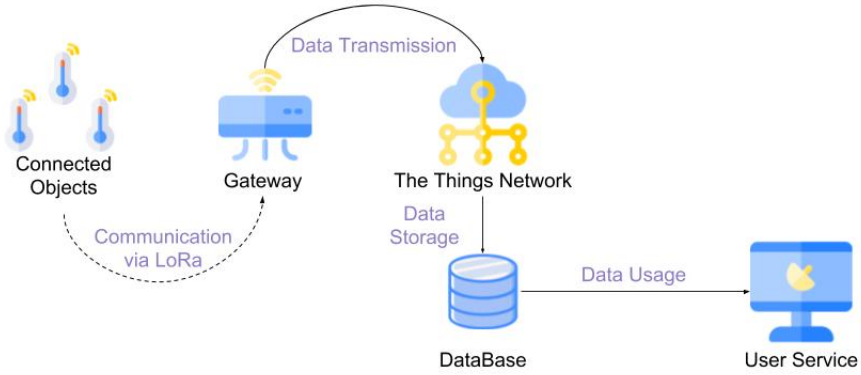

Figure 1: Sensora's Global Architecture

2 Project Management

2.1 Work Methodology

The project group decided to use an agile methodology because the team was required to be responsive to the customer's requests that evolved over time throughout the progress of the project.

To be able to listen to the client's requests and to meet with the client frequently, the team decided on organizing weekly meetings which the client could participate in, allowing us to show the project's progress. As well as this, the meetings were used to establish future tasks required for the following meeting, the client could participate during these meetings to help with the development and improvement of the the project's specifications. With this method, the project group tried to meet the customer's needs and various requirements.

Throughout the project, for a clear organization in this work format, different tools are available and applied to the project. As a project management tool, Trello was selected which allowed the team to assign tasks to each work group and to each person, to follow the progression of all the tasks and to fix deadlines. Along side Trello, Slack was decided on for the team's main communication tool. Various discussion groups were created, one for each work group and a general discussion group to transmit meeting schedules, important information, share notes and important documents.

2.2 Team Presentation

Concerning role distribution, roles evolved over the course of the project due to different work groups finishing their part of the project and their help was required elsewhere in the group.

Two poles were created to start the project's start, each with their own pole manager. The poles were respectively responsible for the embedded section and the data gathering/processing section. The "Embedded" pole was composed of Chloe and Sam whereas the "Data" pole had Maud, Guillaume, Thierno, Saad and Mohammed as members. Once the "Embedded" part ended, Chloe and Sam joined the team "Data" to strengthen it, this team was divided into several small teams, each team was led to develop a different service.

3 Embedded System

One of the project's key components were the connected objects that are able to use its sensors to collect data on its surroundings and send them wirelessly to a nearby gateway. The collected data from the objects consists of information to monitor the comfort conditions of a room.

3.1 The Connected Object's Architecture

The connected objects used throughout the project are composed of: a Raspberry Pi, which is a nano computer with an embedded Linux operating system designed for prototyping; a Chistera-Pi, a shield that enable LoRa and Long Range Wide Area Network (LoRaWAN) connection; and a SenseHAT, a board containing various sensors such as an accelerometer, a gyroscope, a magnetometer, a temperature, a barometric pressure and an humidity sensor.

Every fixed amount of time, a data sample is composed from the ambient temperature, humidity and pressure measured. The gathered information is encoded in order to be sent to the gateway, it also contains identification information. These messages are then transmitted over the Internet and more precisely to TTN that reads the message and associates it to the specific connected object which originally sent the message.

3.2 Network Communication

The embedded systems use LoRa to communicate and transfer data. LoRa is a type of wireless radio telecommunication network connected to Internet via gateways. It allows data to be sent with low power which is an important for connected objects. Data sent via LoRa can travel over 10 km in rural areas (the coverage is almost equal to the Global System for Mobile Communications (GSM) coverage).

For this project, a gateway was placed in a room within ENSEIRB. It's LoRa-compatible allowing an access to a LoRaWAN which transfers towards a cloud named TTN. In order to be identifiable by TTN, the LoRa message must contain several parameters :

- DevUI: identify the end device;
- DevAddr: device address on LoRa network;
- AppEUI: application identifier (explained in TTN section);
- GatewayEUI: identify the gateway.

4 The Things Network

TTN is an open-source decentralized infrastructure that enables low power devices to use long range gateways to exchange data with applications using LoRaWANs. To use the TTN network, an account must be created where different applications were created and fitted to the project's needs. Various options are available once the console has been opened: gateway configuration, application creation, devices registration, and the ability to forward information towards other applications.

4.1 Gateway

The Things Network Packet Forwarder uses an authenticated and encrypted TCP connection to the network server and has support for Linux based gateways. To register a gateway, a Gateway ID, protocol and frequency plan must be defined.

4.2 Application

Applications can deploy a group of devices using the same protocol and whose data must be forwarded to the same destination. An application is identified by its $AppEUI$ which must be known by each participating device. When an application is created, a new $AppEUI$ is generated from TTN.

4.3 Devices

Once the application was functional, each device was registered. Various keys and IDs are necessary to connect a node to the network and to start collecting data. The $DevEUI$ and $DevAddr$ can be generated on the device's creation on the cloud or they can be added manual if they are known beforehand.

In order to perform an HTTP integration, we need to make use of our application details and when this integration is set up, all decrypted data will be forwarded to our end point url and devices will automatically get added when they transmit data. Added devices should be able join the TTN network when turned on, and its data should be visible in the application's Data page.

5 Data

5.1 Data gathering

In this part, the aim was to be able to extract data sent by the sensors in order to use them. TTN offers 3 possibilities to gather those data :

- API handlers : This is the most basic way to integrate an external application with TTN by using standard protocol libraries to connect directly to TTN's Handler APIs. (e.g: HTTP, MQTT)
- Software Developer Kits (SDKs) : Uses common programming languages tools like Java, Node.js and Go. By using these tools, it is possible to integrate TTN APIs in an application.
- Integration : This method is the easiest way to connect TTN with an application. In fact, in integration platform APIs are already implemented and directly connected to TTN. (e.g: Cayenne, Azure IoT Hub, AWS IoT, HTTP).

5.2 Integration of TTN with Firebase & Cayenne

5.2.1 TTN - Cayenne

Cayenne is an integration platform using the MQTT protocol and requires two parameters for configuration: the sensor type (e.g: Lora) and the device identifier. Anytime sensors send data to TTN, the data is immediately push to Cayenne, thanks to the publish/subscribe system. Figure 2 below shows how Cayenne displays the data to the user.

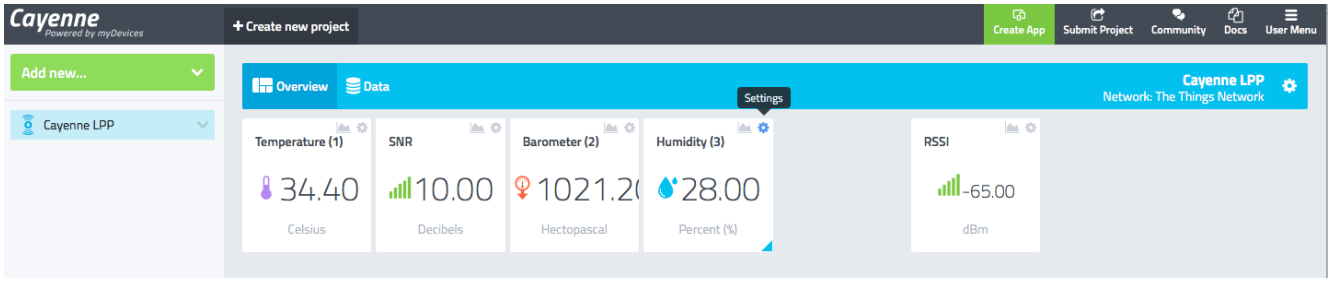

Figure 2: Cayenne's Dashboard

5.2.2 TTN - Firebase

Stored data on TTN wasn't a sustainable storing solution since the data is automatically set to be deleted after seven days. Instead, the decision was made to extract the data from TTN and to store it in an external database, but the issue was deciding what sort of database to use. After some research, Firebase appeared as the best solution, since its a real-time database and possesses cloud functions. Figure 3 presents the architecture implemented to extract data and send it to Firebase.

Figure 3: Architecture of TTN data extraction with Firebase

6 Comparative Study

Two different integration methods were selected and implemented: one integration method using the Cayenne application and the other using HTTP. A third method based on Actions on Google was studied but wasn't implemented. The following table (Figure 4) shows each method's advantages and disadvantages, relevant to data exploitation and service creation.

	Advantages	Disadvantages
Cayenne	• A mobile application to configure, track and control objects and sensors everywhere and in real time • Easy integration • Interface already implemented • A rule engine in order to trigger automatic actions between connected objects • Graphical input / output management	• Real time data • Easy integration with TTN • Freedom in adding functionalities • Free implementation of the visual interface
HTTP Integration	• Interface quite unmodulable • Unable to add features • Two devices corresponds to two Cayenne accounts	• Longer implementation (implementation of a Web application and request to the database) • Need to set up an interface

Figure 4: Cayenne and HTTP Integration's Comparative Study

A method based on Actions on Google was studied following a request from the client. Actions on Google offers a link to Google's open environment as well as the Smart Home Action service. This allows a user to control and interact with their devices: connect, query and control it through Google's cloud infrastructure.

However, this method is set for devices compatible with Google technology and the project's devices were governed by the LoRaWAN protocol and couldn't connect with Actions on Google.

A more suitable solution for the project's devices would have been to use Actions on Google to make request on a API endpoint. This API endpoint would have given the user information about the devices via GET requests and would pilot the device via POST request. The gateway's characteristics couldn't permit this type of interact since it is only capable of receiving data and not emitting data, so this method was not viable with the project's resources.

7 Services

One of the goals of the IoT is to collect data. The communication chain setup makes it possible to meet this need. It is necessary to create services that allow customers to use and interact with their data. The interface between the user and their data is an essential element: it must be easy to use, intuitive and efficient. It's format can be diverse and each service can be different. No instructions were given regarding the type of service required. Knowing this, after a bit of research and brainstorming, the SenSora team came up with several services to be created.

7.1 Application Web and Mobile

The first service implemented was a website. In a business, the main working tool is the computer. If a company wanted to manage a network of connected objects, a web interface would be an ideal solution to meet their needs. This type of service makes it easy to manage the data and to have desired information in real time. The website allows the end user to display, in real time, the temperature, pressure and humidity detected by the sensors.

A mobile application was also created. Nowadays, almost everybody has a smart phones, and its usage is very common thing. Via an application, the data can be directly available from the phone, it can then be consulted at any time with ease. Similarly to the website, the mobile application displays the data received from the sensors.

7.2 DialogFlow

The final service created was via Dialogflow. It's an interface that allows a user to use a Google interface: it is possible to use speech recognition thanks to the service's Cloud Natural Language API. With machine learning and recorded speech, Google recognizes the sentence and launches an adequate response. The use of voice assistants is widely used today and is useful and popular for many people.

Creating a service that meets this demand is importantand with Dialogflow, an adapted application was created. It allows end users to interact with speech recognition for information on the temperature, pressure and/or humidity from one of the devices. With these services, customers can easily use and process their data however they want.

Conclusion

With the help of a modular team management, the project's objectives have been completed. This comparative study demonstrates how data can be integrated in an IoT Network, revealing that there are an unimaginable amount solutions.

However, in the given context, only a few of these solutions are applicable. In SenSora's case, the limited number of solutions was due to the communication protocol and the LoRa Network Server. The team managed to create a complete communication chain from collecting data from sensors on various devices to providing an online service for users.

Glossary

GSM Global System for Mobile Communications. 5

IoT Internet of Things. 1, 3, 9

LoRa Long Range. 1, 5, 10

LoRaWAN Long Range Wide Area Network. 5, 8

TTN The Things Network. 1, 2, 5–7