MASTER THESIS

Academic year 2019/2020

BoT: Blockchain of Things

Author: Amaia Lizaso Lorente
Supervisor: Prof. Guido Perboli
Cotutor: Prof. Stefano Musso

Table of contents

List of tables and figures

List of tables

List of figures

Abstract

This Project presents a root analysis made on a production process, and the improvements taken step by step in order to reach to the final solution. Specifically, how to improve the process of tracking olive oil bottles by an ESP32 microcontroller located on each pallet. The applicability of this, shows much relevance when it comes to adapting the current system in todays changing world and ensures an outstanding outcome, not only in terms of costs but also in the usage of hidden data. The objective is to develop a modular, highly flexible, and low-cost solution that can be configured based on the cost and necessary benefits, in order to store IoT data in Blockchain chains, such as IOTA and BigchainDB. The methodology followed was analytical and experimentational, as it was needed a very detailed picture of the actual process in order to analyze the possible improvements in order to fill this important gap in the interrelation between Blockchain and IoT. The final result responds to a combination of a microcontroller ESP32 and a sensor DHT11, which captures environmental information and tracks the supply and delivery of the products, without the necessity of additional devices in the supply chain. This will contribute to a new era of devices that will be capable of tracking and reaching invisible information in order to simplify and ensure information transparency and an optimal supply chain system.

Acknowledgements

Firstly, I would like to thank the pillar of my life, my family. They have supported me in all the possible ways through my path, encourage me to face situations without fear and to value myself.

My deep gratitude also extends to Prof. Guido Perboli and Prof. Stefano Musso, the supervisor and cotutor of my thesis, who have guided me during my thesis, remarking her fan to help and generosity.

A huge appreciation to Politecnico di Torino, for teaching me along my academic journey, but also for sharing its values and principles.

Lastly, to my colleagues among this journey. For showing me the great value of friendship. Remarking the last two years of happiness, stress, exams and friendship.

This goes to you, Ariadna.

1. Introduction

We can clearly say that Internet of Things and Blockchain are two of the unruliest technologies of today's world. Blockchain, which primary started as cryptocurrency accounting, now is used by hundreds of well-known companies thanks to the multiple number of advantages it presents. Since the birth of Bitcoin and the implementation of Blockchain as an accounting tool for these cryptocurrencies, there has been an increase in the number of cases of Smart contracts that continues to increase and will definitely boost a growth in investment, design in the sector and new innovations.

In the other hand, IoT is the legitimate evolution of the Internet to the systems, it has been expanding widely since its initials. According to Gardner, in 2020 it is expected more than 20.8 billion connected devices. So, the question formulated was, why not combine both technologies in order to obtain a modular solution? And this is what we did.

From an identifying a necessity, we developed a modular solution. We wanted to combine both of the most innovative technologies of today's world and offer an alternative to tracking processes, naming it Blockchain of Things, BoT.

The main objective of BoT is to design and develop a system of low-cost physical devices that can proceed and store IoT information captured in a Blockchain chain. The specific objectives of this solution are focused on the capabilities that these devices require, at a technical level, and the cost contained therein. The project contemplates that the system is modular and can fulfill the previous main objectives, even if the data processing capacities of some parts of the system were not sufficient enough.

Among the analysis of the project and the improvements made on it, we will see how these adaptations of new devices are made on the process, and the impact of each one of those. By the approach of different point of views and contrasting knowledge, we finally came up with the most optimal and reachable solution, which presents a quintessential combination of microcontrollers, sensors and the newest technologies. This ideal combination includes devices such as microcontrollers ESP32, with a sensor, that measures temperature and humidity. One key point of this final solution, is that all the set of information captured, will automatically be uploaded and saved to BigchainDB and IOTA, two of the most recent Blockchain networks.

The combinations of these both technologies, Blockchain and IoT, result in a very scalable product and solves the problems related to fraud and illegal cross-selling.

As an initial application of this technology, this device will be designed to provide the olive oil sector with a tool that allows obtaining information about all measurable biometric data. It will also be used as a logistics tool to offer an accurate product tracking service.

1.1. Origin and motivation

It all started when I finished my studies between Universitat Internacional de Catalunya and Politecnico di Torino, I developed a strong interest in coding, programing and its endeavors. It just fascinated me how with a line of code you could manage to set up a web page easily.

From that on, I started to craft myself and extend my knowledge in a field that I barely knew. And that was motivation, get to know the process and the hole understanding of it. I wanted to get deep, as everything it was very unfamiliar for me. As I was learning and getting to know the insights, I started to get small knowledge in other fields, such us Artificial Intelligence, Machine Learning and Blockchain. I did hear before about Blockchain, but it wasn't until that moment that I started to get interested in it.

I have been always a very technological person, in other words, I have always found my passion in operational processes and technological aspects. I think that everything can be technically explained or exposed. From where I stand, I do believe my academic background has given me the idoneal path in order to discover my passion and who I want to become as a leader in a future. The combination of my double degree, in both institutions, Universitat Internacional de Catalunya and Politecnico di Torino, has taught me to discover my passion and commit to it wholeheartedness. And this is what I did with BoT.

Specifically, in both institutions I could find subjects related to the field, especially in terms of IoT (Internet of Things) and how to lean processes and systems. At UIC, we took part in a subject called Sistemas de Información de la Empresa and Operations, where we developed a lot of sessions and conferences about IoT, information systems and business intelligence. On the other side, in Politecnico di Torino, we did a subject called Strategie e strumenti per l'Innovazione, where we self-

developed a project and all its endeavors, from lean thinking, going though innovation management to Technologies based companies.

I am a strongly believer that you have to pursue what awakes you. That is the reason why I wanted to go through BoT, not just because I do find particularly amazing how new technologies are making a change in our day-to-day life's, but also because blockchain and IoT (Internet of Things) are two of the most promising technologies at the moment. Blockchain is considered as a lawless technology and IoT represents one of the rowdiest technologies of this century. It is a natural evolution of the Internet to systems. Therefore, why not go through a combination of both?

With this analysis of the process of BoT, I hope I can transmit the deepest insights of Blockchain and IoT though it, by itself. Finally, I hope I have been able to convey my enthusiasm and eagerness for the know-how of these newest technologies and processes.

1.2. Aims and scope

The aims for this project are as follow:

- Design a new tracking system that allows you to identify a set of products and the location at all times.
- Quantify the improvements of the improvements of the actual solution while eliminating the dependencies of local databases (MySQL).
- Identify and track all the conditions that have occurred while transportation and a timestamp of when it happened.
- Make an economical implementation in terms of investment needed and calculate the correspondent reduction in costs.
- Within the design and architect structure of the process, assure fully transparency in all information, secureness and resistance to software falls.
- Deeply understand the process and the operations of this type of decentralized systems

As we can see the aims for this project is basically the understanding of itself and account the effects of the improvements made among the process. Starting from a general approach in order to give a global vision and slightly centralizing into the process to study, the Blockchain of Things. In order to

understand BoT and its endeavors, it is crucial to understand the technology itself and how it works, in order to see how the process can be improved and enhanced. For this we need to know not just how it works, but also the design, the technical aspects and the compliance with standards and approvals.

Subsequently, the scope for this project is the improvement in the basic process of Blockchain of Things, as it vegan as an idea. This is accomplished by the improve and the upgrade of the tracking systems, sensors, devices and decentralized networks used in the process itself.

As said in the aims previously, with the understanding of the process, we can define, get an insight of the process and the know-how of it. But the scope is the improvement of this process, in order to see how this improvement can achieve a significant growth in revenues, an enhancement and simplicity in the hardware and software of the process and lastly, the increase in information transparency and a decrease in costs.

In other words, how this improvement from the actual process can be measured quantifiably and qualitatively.

1.3. Structure of the document

The structure of the document is divided in eight different chapters:

Chapter 1: Introduction. Focused on the description of the motivation to do the project and the aims and scope.

Chapter 2: The company. In this first chapter, it will be briefly explained the company in order to understand better the process and find the relation within the project.

Chapter 3: The actual Solution. After the briefing introduction about the company, it will be explained the actual solution that is now applied. Also, it will be explained the characteristics of the solution and the choices we have made in terms of programs, for the deployment.

Chapter 4: Solution that it wants to be accomplished in a future. After the actual solution, it will be explained in a more general approach, the modular solution that it wants to be accomplished, in order to have a brief idea of what it should be like.

Chapter 5: The proposed technical solution. In this fifth step, it will be exposed the technical solution that it wants to be achieved. In this chapter it will be explain deeply and technically all the

components that take part in the future solution. Not to forget, beside the components, it will be explained the networks used, the characteristics that those present, and how these will be employed in the process improvement.

Chapter 6: Economic assessment. After explaining to detail the technical solution and the networks used in it, it will be analyzed the economic impact of these improvements.

Chapter 7: Execution. Sequentially, an execution will be shown and explained. It will contain the sequential phases for the correct execution of coding and programing of the microcontrollers and computer boards, in order to reach the proper execution of the process.

Chapter 8: Future lines of application. In chapter 8, new future applications of this process of BoT will be boarded. Also, what measures can be taken in consideration in order to evolve this process.

2. The Company

BlockTac is a company created in Barcelona in April 2018 dedicated to the development of tools based on Blockchain technology to carry out certifications and digital seals, turning the documentation and products treated into forgeable and immutable.

The company is the market leader in this sector, with more than 300,000 certificates committed since its creation. The products rest on Blockchain technology and respond to their own developments. Because it uses blockchain's largest chain for its products, Ethereum, ensures unparalleled security levels for any other solution.

The company business model is "Blockchain as a Service" - BaaS, a variation of SaaS's most general models (Software as a Service).

The company's business model is currently based on the development of three services/products based on Blockchain technology and their commercialization through distributors or by themselves in national and international vertical markets: Digital certificates for institutions, digital seals for the companies that manufacture different goods like clothing, accessories Fast Moving Consumer Goods, oil and so on.

An example of two different types of certificates:

Figure 1: Group Pascual

Figure 2: Group Oils Ferran

3. The Actual Solution

The actual situation of the process it can be showed graphically for a better understanding:

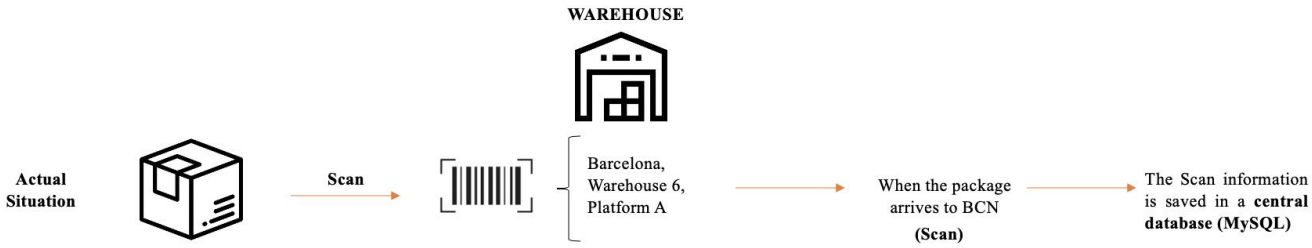


Figure 3: Example of the actual system of the process

As seen in the picture above, the actual situation is composed by the items or boxes that arrive to the warehouse, a code, the corresponding scanning and a centralized database. In order to understand how the actual process works, we are going to expose an example of olive oil bottles. Imagine we are an intermediate warehouse, and a pallet with different boxes of olive oil arrive to the warehouse, at the point of arrival, the first step to do is scan the label or code of each package in the pallets. When receiving, it is crucial to audit what it enters to the warehouse and to know the location of the product or item in the warehouse itself.

Once we have scanned the code of the product in the receiving, the following step is where we must locate the product in the warehouse. At this point, the necessary information it is known from the scanning previously done, which will reboot us the information of location. In this case, we suppose that the package itself has to go to Barcelona, at the warehouse number six, which is located in platform A. When this package leaves the intermediate warehouse and arrives to Barcelona, which is the package final destination, it will be scanned again and delivered to the final destination. All the final scan information will be stored in a central database, which right now it is MySQL.

The pain point of this solution is that, in order to upload information from the scanning and the location of the packages, each one of the operators among the chain has to collaborate and put the information in the central database. Also, we may say that this model is susceptible to attacks and

there is a need to do backups periodically, otherwise all the information will be lost. Not to forget, all the time and money that the operators and the company itself has to spent within this process.

Central Database

One of the biggest American computer technology company, in their recent article "What is a database?", defines a central database as "*an organized collection of structured information, or data, typically stored electronically in a computer system*". ("What is a database?", 2020)

In other words, is a set of information that is stored and located in a single spot, technically, it is a central computer that has all the information. Normally, a central database is a Central Processing Unit (CPU) or a central computer. Different users from remote places, can access the central database.

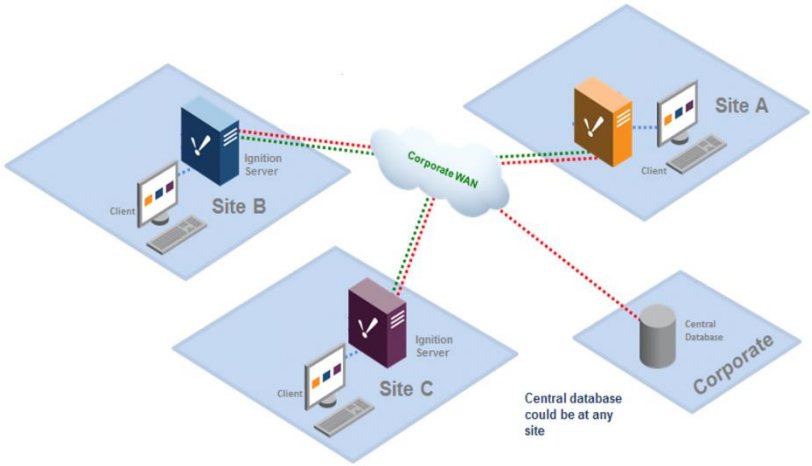


Figure 4: An example of Central Database

As we can see in the example above, the central database itself can be located at any location or point. All the information that is stored in the central database has to be managed by a *Database Management System (DBMS)*. This management system is used for a better administration, in terms of organization, update, recapture and a correct consolidation of the information itself. Normally, this interface controls all the database, counting with restore and back-up tools.

In our case, the fact that we are using a central database in the different stages of the bottles of olive oil production makes that farmers, oil mills, cooperatives, bottlers, transporters and retail stores must cooperate and insert the information in the database.

Thanks to the cooperation of these protagonists, a series of data will be added to the central database. The most important step is that everyone in the line of production, management and distribution has to cooperate among the tracking of the data and record the data in the system. If the data is not record correctly or there is no cooperation, therefore information won't be accurately nor precise, leading to a decrease in benefits and an increase in costs.

Advantages and disadvantages of a centralized database

The types of database that a company uses depends on the necessity of the company in terms of usability or the variability of the information stored. Having a centralized database has many inconveniences as well as disadvantages. ("Centralized Database Management System", 2020)

The advantages we can find in a centralized database are: ("Centralized Database Management System", 2020)

- **Ultimate integrity:** It is simpler to unionize the data, due to the fact that all the information it is stored in a unique central location.
- **Merest redundancy:** As the information does not have different locations of storage, the redundancy levels remain minimal.
- **Maximum security:** The security of a centralized database is at its best, as there is no dispersion of information. There is only one place where it is stored, therefore this equal to a high level of security.
- **Higher cost efficiency:** The maintenance and power supply are less costly rather than a decentralized database.
- **Fastest searcher:** As it doesn't have to look up information in multiple databases.

As many of the advantages, there are also some disadvantages while talking about centralized databases: ("Centralized Database Management System", 2020)

- **High probability of bottlenecks:** Due to the fact that the information or data it is stored in the same location, the freight may be higher.
- **Synchronously access:** Access points are limited when talking about centralized databases, as if many users are trying to access to the same array of data. This problem can result in a decrease in productivity and performance.
- **Back-up:** As the set of information it is stored in a single location, if there is no back-up of the information or the back-up it is not done accurately, there is the probability that the information stored can be lost.

Our choice for the centralized database: MySQL

In our actual solution, the main characteristic can be exposed in one word: MySQL. As explained previously, in this process we are using a centralized database, which basically means that everyone in the process line of the bottles of olive oil has to insert the information of tracking and put it up, in order for the central database to get all the data and update it.

As explained Oracle ("What is a database?", 2020), MySQL is categorized as *"Relational Database Management System"* (RDBMS) and nowadays it is a product of Oracle, but it was originally created and developed by IBM. By now, MySQL is one of the major players on its league. It was thought to simplify the process of queries and transactions.

MySQL responds to the language *Structured Query Language (SQL)*, that it is used for the management of databases ("SQL Introduction", 2020). SQL can do multiple actions, for example it can create databases, tables, tables in databases, manage all kind of records, update them, establish permissions on tables and so on.

The query language SQL is the most used specific language when programming. As CISSP Study Guide explains, SQL language has two subdivisions of directions: (Conrad, Feldman & Misenar, 2016)

- **DML:** This stands for Data Manipulation Language. As the name itself, this command is for the manipulation and administration of the data/information. It can develop queries and renew data that already is stored.
- **DDL**: This stands for Data Definition Language. It is used for managing and structure the data of the database. With this command, the user can create and modify any tables.

The user interface: phpMyAdmin

In order to manage and administrate MySQL, we have chosen phpMyAdmin. Thus, is a tool written in Hypertext Preprocessor, also known as PHP language. The difference about this type of code is that is carried out on the server, and sequentially sending the produced HTML to the client. It is basically to access to the data. ("PHP: ¿Qué es PHP? - Manual", 2020)

We have chosen phpMyAdmin due to its easiness when using it. Also, it results easy to learn how to manage it and it is predictable. Normally, it is a very easy-to-use tool and it has also complex commands and features, which is remarkable when managing and administrating the data. Finally, a positive point is that is free, so there won't be any kind of cost associated to this, only the electricity needed in order to run the computer.

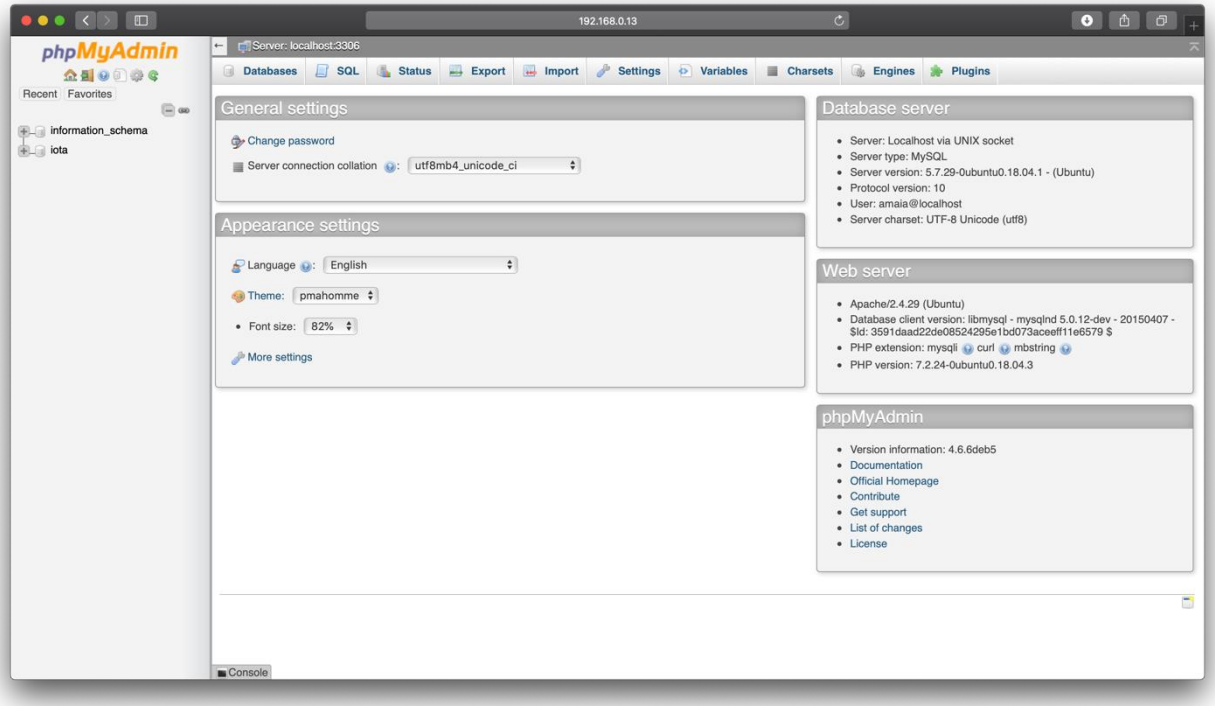


Figure 5: A phpMyAdmin initial screen

4. Solution that wants to be accomplished in a future

Pain points about the actual situation

From the first actual solution that, as explained, it was based on a centralized database, it was thought about what new measures or improvements could it be made in order to improve the actual solution, in order to achieve a meaningful enhancement in the process, more efficiency and obviously, a reduction in costs.

By this, we first thought that even though there were multiple advantages on having a centralized database, for our process it was getting obsoleted. Firstly, there was a dependency in terms of scanning. As mentioned previously, all the pallets that arrived whether the intermediate warehouse, whether the final destination, had to be scanned. The scanning information had to be putted up, in order to complete all the tracking information in the central database, which was identified as a drawback. Not to forget, that all the workers among the production, management and distribution lines had to cooperate among the tracking of the data, thus sometimes can result in poor cooperation or in the worst case, a re-sell on un-official markets.

Secondary, the time that was lost due the scanning. As it is said "times is gold", therefore all the time that was lost among the action itself was crucial in order to increase efficiency. So, why not look up for a solution that eliminates the time spent in all the scanning process?

Parallelly, we wanted more accessibility to a larger number of users. One of the disadvantages that the actual solution presents is the bottleneck probabilities when a considerable number of users access the same set of data. Normally there are no problems when connecting a large amount of users, but we wanted to assure that everyone could connect without difficulties or errors.

To sum up, transparency was a must. The feedback that we were getting at the time was basically about data reachability and how to make automatic updates of the data itself. This was complicated to get when talking about a central database, as nobody but users could access to that data, which is reasonable.

Intermediate solution

When we started to realize these blind spots, the initial idea was to adapt the process to these market needs that were stepping out. Hence, a first process improvement was made, leading to this first model:

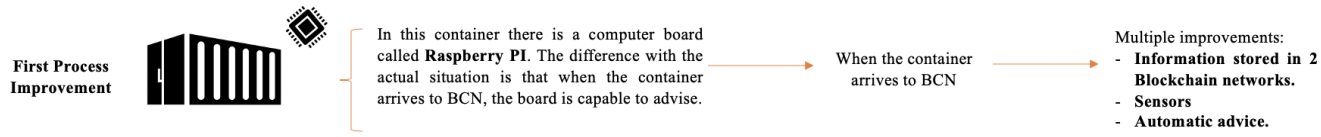


Figure 6: First Process Improvement

As seen in Figure 6, for us, the barcode was completely gone as it was one of the main pain points, due to the amount of time lost when scanning. It wasted a huge amount of time and it was getting out-of-date. It was set as a requirement to look for other alternatives.

Because it was decided to get rid of the barcode (in our process, not from the boxes), it was thought to look for even a better solution that could accomplish more security. Among all the possibilities that were on the table, it was thought to insert a computer board inside each pallet of packages, named Raspberry Pi. The advantages of this computer board were multiple, and increased security, which was a point to reconsider as well.

The function of this computer board was easy. This board was collocated inside the container, exactly in each pallet. When these arrive to its medium or final destination, the board itself, by a connection through the Wi-Fi, advise the database, which was distributed. The Raspberry Pi has beacons inside, meaning that it has GPS on it. When those beacons connect to the Wi-Fi, this get the location and send the information of it.

But it did not end here. It was thought that it could be interesting to catch more invisible data, and it was then when it was decided to insert sensors to the Raspberry Pi, by the Sense Hat. The function of this added board was to capture any kind of added data apart from the tracking information. For example, in the production and transportation of olive oil, these sensors in the board can catch up multiple information as the temperature of the container, for accomplish a better preservation of the

qualities of the oil. Another example would be, the presence of any chemical factors in the air of the container.

Lastly, all this information would be recorded in two of the newest Blockchain networks, which were BigchainDB and IOTA. The recording into these networks ensure transparency, which it wanted to be accomplished.

So, in this process we have four improvements:

1. The centralized database is switched to a distributed database. All the data is registered automatically in BigchainDB and IOTA, two of the newest Blockchain networks.
2. There is no more bar codes and scanning. The pallet or container advises automatically.
3. A new computer board is inserted in the process, called Raspberry Pi.
4. Introduction of sensors in the Raspberry Pi, thanks to the Sense Hat.

These four new improvements in the process where key in order to fulfil the requirements of the market, but the Raspberry Pi is a powerful computer board, which can do a huge amount of functions that weren't needed at the time. Parallelly, the hardware cost of the computer board was too high for the necessities.

Final solution

As commented, due to the fact that the Raspberry Pi has some mismatches with what it was needed at the moment, it was thought a replacement of the board into one that had less power and did the exact needed functions.

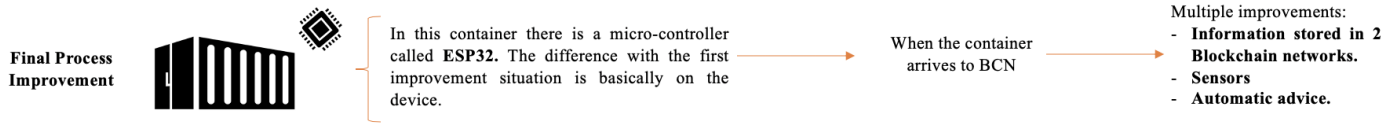


Figure 7: Final Process Improvement

As a result of the research done to determine the most suitable family of microcontrollers for the design of the device itself, finally it was decided to choose a microcontroller ESP32. This microcontroller had just the right amount of power necessary for what we wanted to accomplish, and it was way more economic than the Raspberry Pi.

Likewise, it can be re-used multiple times for future operations, which was also a key point for the final decision. The sum of the parallel characteristics that this device offered, resulted in the design of the final solution for this process.

Another key point was the change in the type of connectivity, from WI-FI to LoRa, though LoRaWan. The connectivity with Wi-Fi was the first choice, but in terms of battery was not acceptable. With the WI-FI on it lasted about one hour approximately, when using it to the fullest. Whereas with LoRa the scope was greater, the duration of the battery can last nearly for 10 years or more, but it has a much lower connection capacity, which is not a problem because what is received and send is basic information. In order to send the information, all the set of data is transformed into a MQTT message, which is defined as a communication protocol. Once this MQTT message has been developed, it is sent though LoRa. LoRa it'ss sent through radio frequency; therefore, it is needed also what is known as a *gateway.* The gateway receives the set of information and sends it though Wi-Fi to the router, which is consequently sent to the server.

As same as the previous improvement, the data was recorded and registered in BigchainDB and IOTA, in order to assure transparency.

Lastly, we could define the process as: a set of devices that capture information from the environment through sensors and through its own software integrate it into a Blockchain chain, which will allow the data to be present in a public infrastructure, and also in a secure way to monitor precisely a process or product.

In other words, the process is a modular system initially applied to a vertical market such as the olive oil production and supply chain, but adaptable to practically any market and production/commercial process, and which is capable of processing differentiated volumes of data, adjusting its speed and costs. This solution is a physical IoT device capable of incorporating the information to be written by signing the transactions and incorporating it into a Blockchain, public or private, where it can be independently exploited by third-party systems. In this way, the chosen Blockchain becomes a secure and distributed repository where variables and events from IoT devices are stored.

The system would be as it follows:

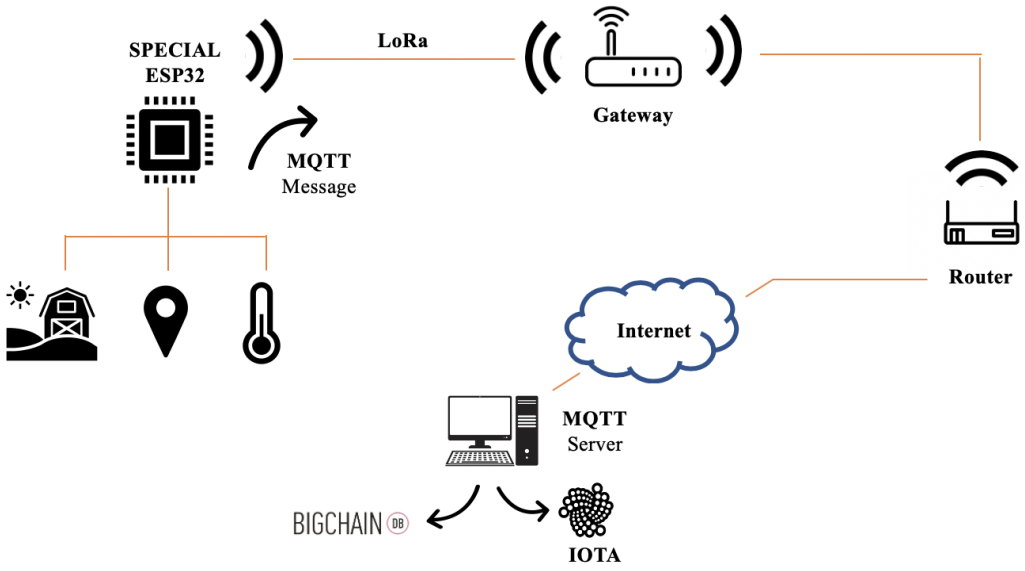


Figure 8: Architect structure of the system.

5. The Proposed Technical Solution

In the previous chapter an overall view of the actual situation of the process and the improvement made was explained. In this chapter, it will be explained the technical characteristics of the process in extent, in order to understand it fully. It will be explained all the elements that take place on the improvement of the process. These are:

- Internet of Things (IoT)
- Blockchain
- BigchainDB
- IOTA
- Raspberry Pi 4
    - Sense Hat
- Microcontrollers ESP32
    - Sensors
- Lora
    - LoraWAN

5.1. What is IoT?

IoT stands for *Internet of Things,* in a simplest way, it is a group of sensors or devices that are mutually connected among them, thus collects external data precisely without the need of any human intervention, and this information can be perfectly analyzed in a future. As in the article of *"Blockchain in Internet of Things: Challenges and Solutions"* describe IoT as the evolution of the Internet itself. (Dorri, Kanhere, & Jurdak, 2016)

All the information collected from the sensors can lead to an improvement in all kind of services, operations and processes, since some of this data is copious to capture without these sensors. IoT presents also a sort amount of security problems, since all this imperceptible information is captured, prepared and refined from the bosom of population, from their private lives. This attends to a problem of protection and confidentiality.

This technology presents, as mentioned before, a wide range of positive and concrete enhancement to operations and mechanisms, resulting in an increase in efficiency. These sensors have converted

the industry and the methodology to understand to the fullest the data that is invisible, such as temperature, gaseous residue, sound, ultraviolet radiation, chemical alterations and so on. But, inside all this good information, what it wants to be accomplished is to read between lines all this information by the means of finding patterns, in order to realize and find out spikes of critical information. These spikes will foreground the crucial missing data, leading to a better outcome for users. (Dorri, Kanhere, & Jurdak, 2016)

This is good terms when it is applied in industry processes and operations, but not when it comes to population, and this is when the privacy and confidentiality problems arise. When analyzing a set of data, you can detect a behavior of numbers and algorithms, resulting in a determination of human patterns of behavior, conduct, attitude and lifestyle. These patterns are no likely to be known by any unbiased observer.

When IoT was first introduced, there was no awareness about these issues that this evolution of Internet presented at the time. In fact, there is no doubt that a multitude of IoT first devices presented in the market had this absence and scarcity in security, limiting the technology IoT in a limited scalability.

As Gartner Forecast explained in its article *Internet of Things — Endpoints and Associated Services Worldwide*, from the years 2016 to 2021 IoT will have an annual growth at a CAGR of 32% (Gartner Inc, 2017). Having real-time data of users with these sensors has been crucial for companies, leading to a digital transformation that won't go back in time. Corresponding to Gartner, in 2021 it is expected that a 64% of these sensors will be customer implementation.

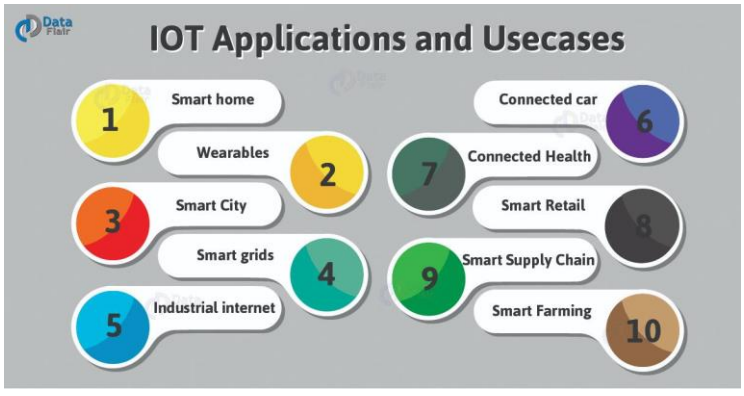


Figure 9: Main IoT Applications

5.2. Blockchain

We may say that Blockchain is one of the emerging technologies of the last years. The first knowledge of this disrupted technology came up in 2008, with a white paper signed by Satoshi Nakamoro, Actually, Nakamoro was a pseudonym for an individual or a group of individuals, whose identity has never been clear. In this white paper Nakamoro explained the functionality of Blockchain and consequently, of the cryptocurrency Bitcoin.

Blockchain is understand as a distributed database within different nodes or participants, which the data is stored in the blocks, which are protected cryptographically, and these are organized mathematically related to each other. The hole Blockchain operates under the protocol and the consensus among users. As the name itself defines, Blockchain is a *block of chains.* We may define the basic elements of this technology:

- **Block:** The blocks is where the information of the transactions is stored. Thus, is stored chronologically.
- **Chain:** The set of the blocks in a particular immutable order.
- **Hash:** The hash is a combination of numbers and letters that is a mathematical calculation. It is used to apply security measures. We may say that the hash is the identity number for each transaction, in order to not be modified by a third party. This is what ensures non-counterfeiting and modification of what is has already been signed in Blockchain. Any change in a generated transaction, creates a completely new hash. By this mathematical calculation is easy to calculate the output hash, but it is not easy to do it backwards, from the output hash to the input hash, in mathematics is called a nondeterministic polynomial time problem (NP). Likewise, it is nearly impossible or computationally very arduous to previous determine the hash combination for a future transaction, this is called preimage resistance.

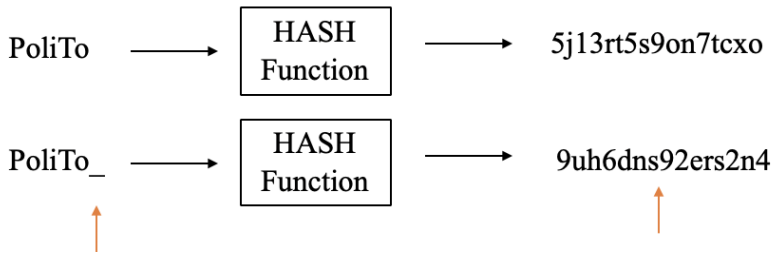


Figure 10: Hash Function

A seen in Figure 10, with a minimun change in the name or number, give us a total different hash combination.

The Hash combination it not only depends from the hash of the actual transaction, but also from the hash of the latest transaction, in order to verify the non-modification of this.

There is a protocol called Merkel Tree, which is based in the *double-hash verification.* It consists, as seen in Figure 11, in a pyramidal structure of hashes, where in the function of each hash it is also verified the previous hashes. Thus, results in the obtainment of the *root hash* of the block. This method is in order to verify the non-modification, the integrity and cohesion of all the data. (Nakamoto, 2008)

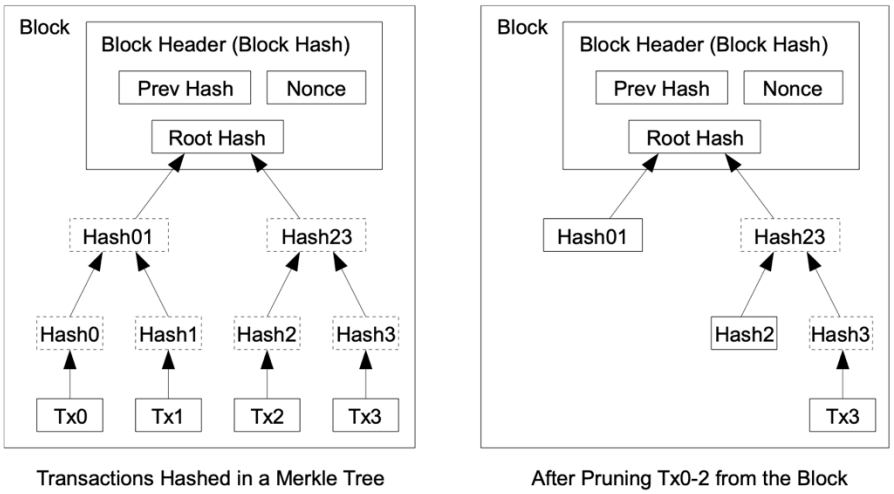


Figure 11: Merkle Tree

- **Node:** It is a point of connection, in other words any member (computer) that is active in Bitcoin. With the protocol Peer 2 Peer (P2P), the nodes can communicate within each other. Likewise, in the nodes there is a copy of the stored data along with the programs to carry out the Blockchain executions and storage protocols. (MLSDev, 2019).

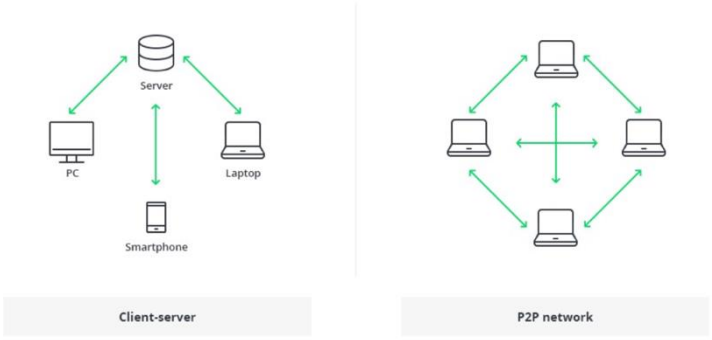


Figure 12: Structure of a Client-Server and a P2P Network

- **Miners:** The miners are the ones in charge of creating the blocks and confirm transactions. Normally the duration to find an empty block is approximately 10 minutes in Bitcoin, conducting computational problems. It is a complex process, therefore the reward for mining one block is around 12.5 BTC, which is around 87.161€ (February 2019). Thus, takes into account the mass use of energy expenditure, time, computing and hardware. This is called *Proof-of-Work*. (Nakamoto, 2008)

Main characteristics of Blockchain

According to Don Tapscott and Alex Tapscott, the main characteristics of Blockchain are: (Tapscott, D., & Tapscott, A, 2017)

1. Distributed ledger

The major characteristic of Blockchain is that this is what is called a distributed ledger. This means that all the data in the Blockchain can be validated and shared among members.

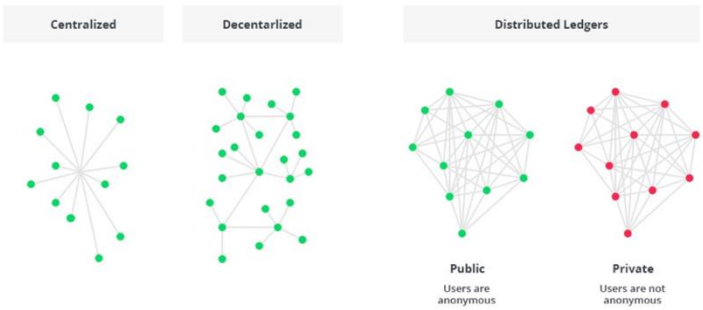


Figure 13: An example of Centralized, Decentralized and Distributed Ledgers

There is no need for a centralized approval in order to execute these actions. In like manner, if the execution of any node it is not correct, all the other nodes still operate correctly.

2. Cryptography

Asymmetric cryptography has 2 keys:

- **Public Key:** This key is public; everyone can see it. A combination of numbers.
- **Private Key:** A large combination of numbers that it is supposed to not be shared with no one. Every user has one private and one public key. This combination is, probabilistically, impossible to decipher.

The private key is derived from the public key. The algorithms used in order to extract the public key are done in order that it is impossible to decipher the number backwards, from the private key to the public key.

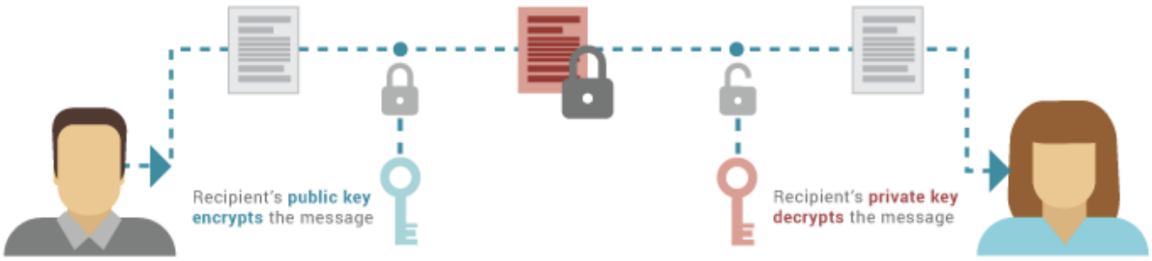


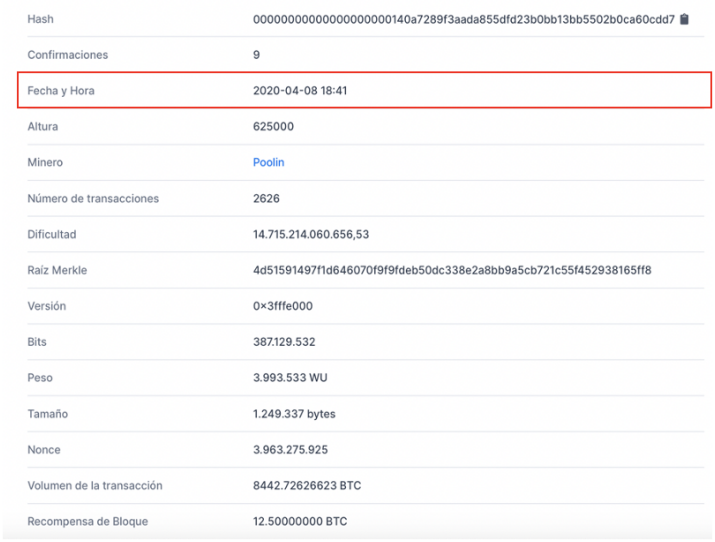
Figure 14: Example of Public and Private Key

For example, if we suppose that there are two people that want to exchange a message, the first will encrypt the message with the second's public key. Now the message is encrypted, once the second has received the message, she will decrypt the message with her private key. In other words, the message is encrypted with recipient's public key and decrypted with the recipient's private key.

What ensures the cryptography is that all the data has been verified and inserted in Blockchain cannot be modified, eliminated nor copied, never. Likewise, thus eliminates the risk presented by third parties when being the intermediate of any swap.

3. Time-Stamped

All the transactions that are made among Blockchain have a timestamp. Any user, if it is a public chain can check up the date and the exact hour that a transaction has been executed. Parallelly, there is also a timestamp when a block is found. This ensures that the block won't be copied nor duplicated, neither the hash.



Hash	00000000000000000000140a7289f3aada855dfd23b0bb13bb5502b0ca60cdd7
Confirmaciones	9
Fecha y Hora	2020-04-08 18:41
Altura	625000
Minero	Poolin
Número de transacciones	2626
Dificultad	14.715.214.060.656,53
Raíz Merkle	4d51591497f1d646070f9f9fdeb50dc338e2a8bb9a5cb721c55f452938165ff8
Versión	0x3fffe000
Bits	387.129.532
Peso	3.993.533 WU
Tamaño	1.249.337 bytes
Nonce	3.963.275.925
Volumen de la transacción	8442.72626623 BTC
Recompensa de Bloque	12.50000000 BTC

Figure 15: Example of a Timestamp

As seen in figure 15, the timestamp of one block, has to add the timestamp from the previous block in its hash, emphasizing the preceding timestamps. This is known as a *"Proof of Existence"*, therefore that the information has existed in time. (Nakamoto, 2008) (Haber & Stornetta, 1991)

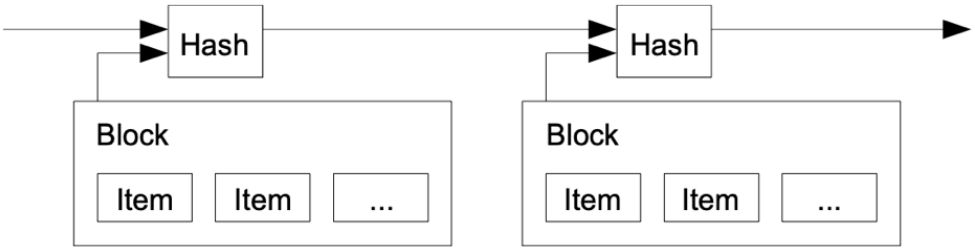


Figure 16: How a Timestamp works

4. **Transparency and fraud protection**

   One of the major benefits from Blockchain, is that you can check up any transaction that has been done in a public chain. By this, you are enabling everyone to be part of the chain. To fins transparency in accountability has been always a challenge, as most of the times hasn't been clear at all. Leaving to a side the fact that third parties as mediators are excluded (including its risks), this could reduce human-right exploitation and corruption.

5.3. BigchainDB

From February 2016, this open source software presents characteristics from Blockchain, such as decentralization and immutability, and also characteristics from a database. In other words, BigchainDB is a net of computers, where normally if you want to make a deployment you will have to build a private node, and, in each node, a MongoDB server is raised. MongoDB is a database, and in order to manage the huge amount of data that is handled the natural solution is to choose MongoDB. In like manner, is based in Tendermint, giving this the agility and quickness when making transactions. (Pon, 2017)

BigchainDB can come up with 1 million writes per second. If we compare it with Bitcoin, BigchainDB presents a large throughput and capacity in terms of storing, rich admission, query competences and low abeyance among other characteristics. (BigchainDB GmbH, 2018)

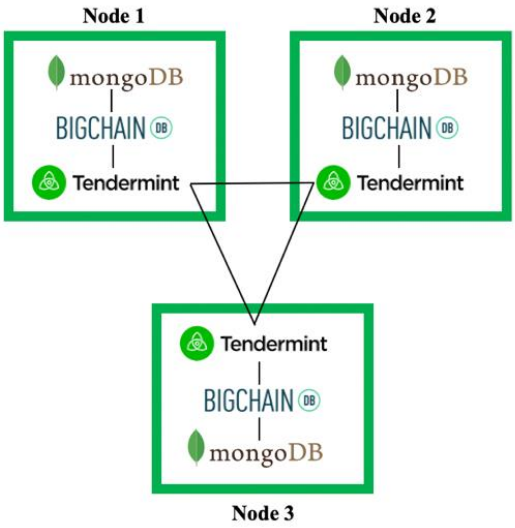


Figure 17: Structure of BigchainDB

Main aspects of BigchainDB

Since its launch in 2016, BigchainDB always promoted itself as a database with Blockchain features. But when it first launched there were some inaccuracies, some of them even could annul the whole database, but with the version 2.0, that was developed and executed in order to deal with these problems, made those blunders vanished. The characteristics are:

- **Non-centralized:** This an aspect that comes from the Blockchain features. Each node is independent, meaning that if one of them breaks down, the others will still function properly. The methodology is uncomplicated, each node uses MongoDB as the database, in order to manipulate the stored data, and the communication in-between the nodes is accomplished over Tendermint. The good part is that Tendermint owns the characteristic of being what is called *"Byzantine Fault Tolerance"*, which means that in interest of that faults to not happen, there is a consensus in order to ensure the correct function of the system with those components that do not work properly. (BigchainDB GmbH, 2018)

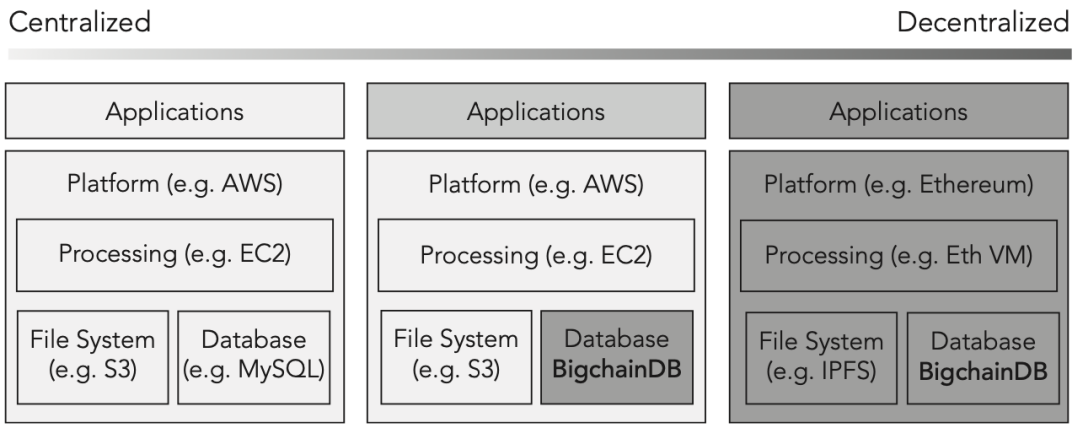


Figure 18: From a centralized to a decentralized network

As in this example, we see how this is exactly one part of the improvement it wants to be accomplished and implemented in the process. The first solution implied a centralized database, such as MySQL, which the process will completely get rid of.

- **Immutability:** Immutability is one of the major words when talking about Blockchain and this characteristic comes also from it. We may say that when a new set of information is introduced to Blockchain, it may never be modified nor eliminated, and if in the worst case it is, it can be identified rapidly. (BigchainDB GmbH, 2018)

  But, in order to attain this characteristic there must be some plan of action, BigchainDB responds for:

  - There is no APIs in furtherance of modify nor wipe out the stocked information.
  - There is a copy of the stored information in each node, through MongoDB. Meaning that if a node gets demolished or altered, the information won't get lost.
  - As explained previously as a Blockchain main aspect, each transaction is signed cryptographically, which attains to a not counterfeit.

- **Proprietor- controlled Assets:** This means that only the proprietor or proprietors of the assets can transfer them. The main characteristic is that in BigchainDB there is the possibility of creating new *"tokens",* which is completely forbidden in chains like Ethereum or Bitcoin,

because the amount is already fixed. Although, the commands will change depending on the finality of the usage:

  - **CREATE Transaction:** This will be used when generating new *Tokens.* These new *tokens,* later on have to be introduced to BigchainDB. Thus, is done through BigchainDB HTTP API. Each CREATE transaction must have one input only, which indicates who is issuing the new *tokens*.
  - **TRANSFER Transaction:** This command is just for transferring the *tokens* to another user. Each TRANSER transaction must have at least one input.

- **Immense transaction rate:** The transaction percentage in terms of speediness is outstanding. It can undertake enormous amount of transactions within seconds, this is fulfilled thanks to Tendermint and its low latency.

- **Structured Information:** We already mentioned that each node in BigchainDB contains MongoDB, which is a database. The manipulator of the node has completely access to MongoDB and can index and query the information stored.

- **Sybil resistance:** In BigchainDB in order to combat these Sybil attacks, the commanding governance has a supervision and controls the list of users. Hence, despite other structures such as Bitcoin that anybody can adjoin a block, in BigchainDB there is no need to put more restrains due to the design of the network.

Fields of Application

BigchainDB has many fields of applications, starting from data governance, through supply chain to audit trails. As seen, the immutability and security that all Blockchain networks give is nowhere to be found in the solutions that were present before. (BigchainDB GmbH, 2018)

If we focus on supply chain, in every chain of processes there are a lot of active users and operators that have to manage, synthetize and publish all kind of information. Thus, might present and cause merge difficulties of data. In all these processes, to have the data updated is crucial and key, and it is

difficult for everyone working actively in the process to work and ensure a correct function of the data emission.

Another key point is the blind trust. In some cases, not all the actors in the chain are willing to work according to the rules, causing problems of distrust and in the worst case, a selling of the products in other illegal channels.

On that account, having any Blockchain network in the supply chain will increase loyalty and trust, not only because once a set of information is emitted it cannot be eliminated nor modified, but also it helps to emerge and structure the amount of data. In the case of BigchainDB, it simplifies the process of structuring information and produce reports.

5.4. IOTA

IOTA was born specially for Internet of Things and with it, new cryptocurrencies, named *IOTAs*, and a new structure, named *Tangle*. The reason was due to the fact, that in every Blockchain transaction, rules are, that you have to pay a fee in every transaction, and sometimes these fees may be bigger than the amount of the proper transaction. Therefore, IOTA proposed a system where these fees vanish from the equation.

Tangle, which is the model proposed, is what is called a *Directed Acrylic Graph (DA*G), this graph follows a specific order, its vertices are displaced linearly. The mechanism is simple, when a new transaction is issued, thus must accept two previous transactions. In other terms, Tangle is composed by:

- Edges: One edge has two vertices. Thus, can be:
    - **Directed:** Straight lines
    - **Undirected:** Arcs. This occurs when a transaction is obliquely approved.

- Vertex: The vertices are the squares, which are transactions.

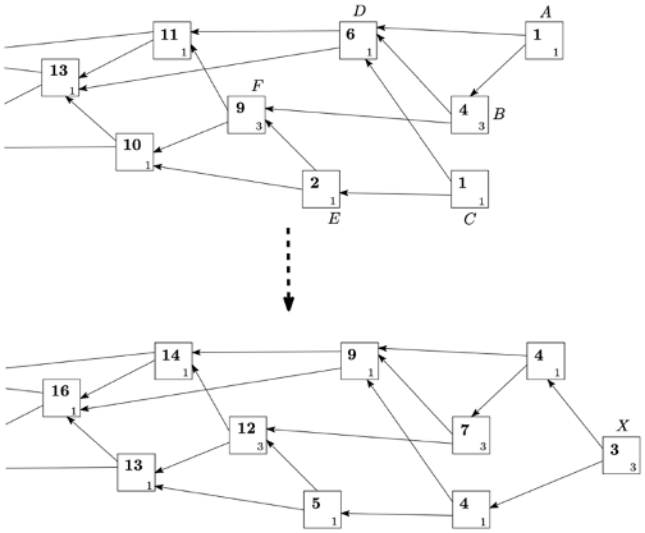


Figure 19: IOTA Network

As we can see in Figure 19, the squares are what we called vertices and the arrows are edges. The squares are the transactions, starting from the right, the big number is the amounted weight, and the small number is the weight of each specific transaction. So, as we can see from right to left, the total amount of weight is bigger. The first transaction is denominated *genesis.* Therefore, as in figure 19, the most recent transaction will be A and the oldest, the genesis.

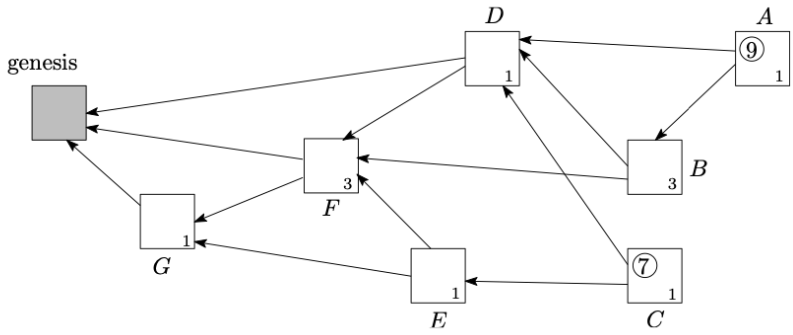


Figure 20: Genesis

As in Bitcoin, the network itself is composed by nodes. Because there are no fees, there are no incentives for the active users in IOTA, therefore the fact that they have to validate two transactions every time when issuing a transaction, commits to the safety and protection of the network itself.

Jointly, the nodes check the transactions, and if the node itself notice that the transaction may not be correct or safe, then the node won't approve this transaction, and will alternatively look for another

transaction to validate. When the node has validated those two transactions, then in order to issue the new one, it has to resolve and figure out a cryptographic problem. (Popov, 2018)

Snapshots

In order for the tangle to be lightweight, there are *snapshots.* Those snapshots are used in order to eliminate and get rid of the addresses that their total amount is zero. With this, the tangle gets lighter, and within the net, it is only remained the addresses with a positive balance. This is done approximately every two months.

With this method, IOTA has the fully history register of the valued addresses and manages to work to a greater level of efficiency and velocity.

Tips in *Tangle*

In IOTA, *tips* are known to be the transactions that hasn't been validated so far. For example, if we assume that we are standing on transaction nine, all the red squares are the ones that have been validated, and all the blue squares are tips, the transaction that hasn't been approved nor validated yet.

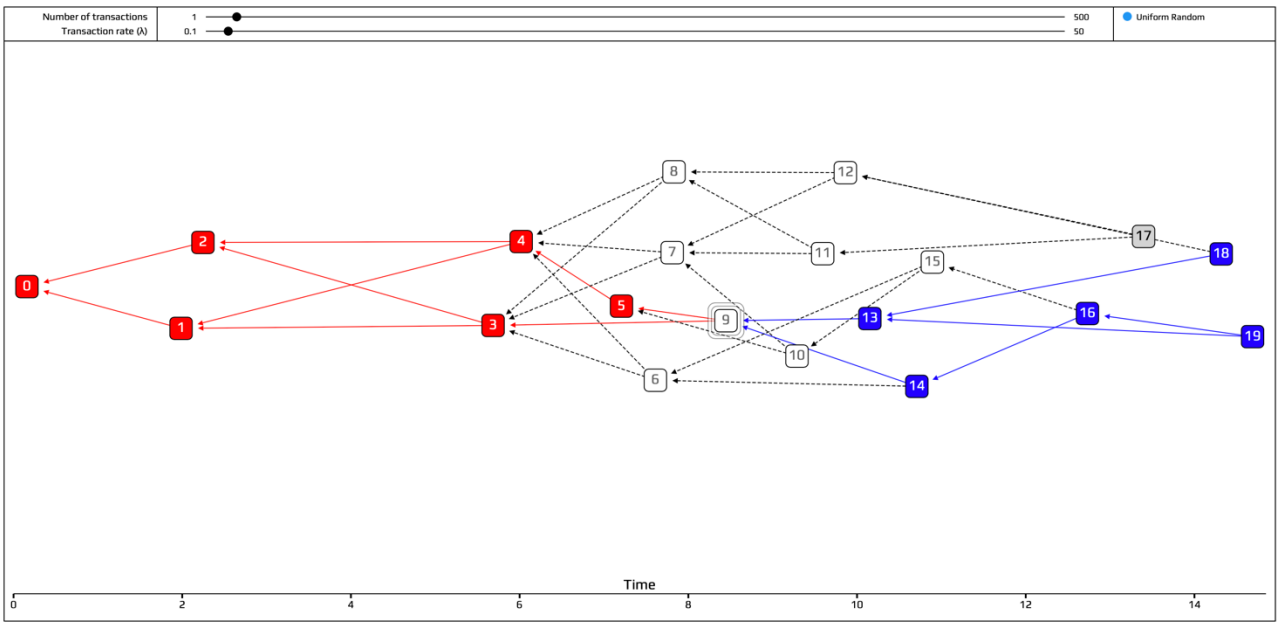


Figure 21: IOTA Tips

This means, that the following transactions will have to validate these blue tips in order to execute their transaction correctly.

Transaction rate and lag period

So far, we have explained briefly the different aspects of IOTA and the *Tangle*, such us snapshots, tips and the genesis among others, but how does the transaction rate affect the *Tangle* itself?

Here it is introduced what is known as *Poisson Point Access,* which is mathematical model that distributes these points randomly among the mathematical area. The transaction rate will be named lambda (λ). In the previous examples we didn't take into consideration the rate in which these transactions came into the structure, assuming a constant rate. Nonetheless, depending on how busy these structures are, λ can take different values among the *Tangle*: (Gal, 2018)

- **Small Value of Lambda:** When there is a small number of transactions, the number of tips become smaller. Considering the simulation shown below, the number of transactions is fixed to a total of 10 and the transaction rate (λ) at 0,1. When having these modest numbers, the validation is done only to one tip, as a result of the low latency in the structure. Computers do work speedy and agile and the amount of transactions is not that big to spend more time than average.

  As seen, in figure 22, transactions two and three have come between a closer arrival rate. Consequently, the arrival rate has spaced itself for transaction number four. Followed by the other transactions, which are the fifth and the sixth had a minor difference between its arrival rates, again. This is an example of the time differences among the transactions done, and how this creates this irregular path in the *Tangle.*

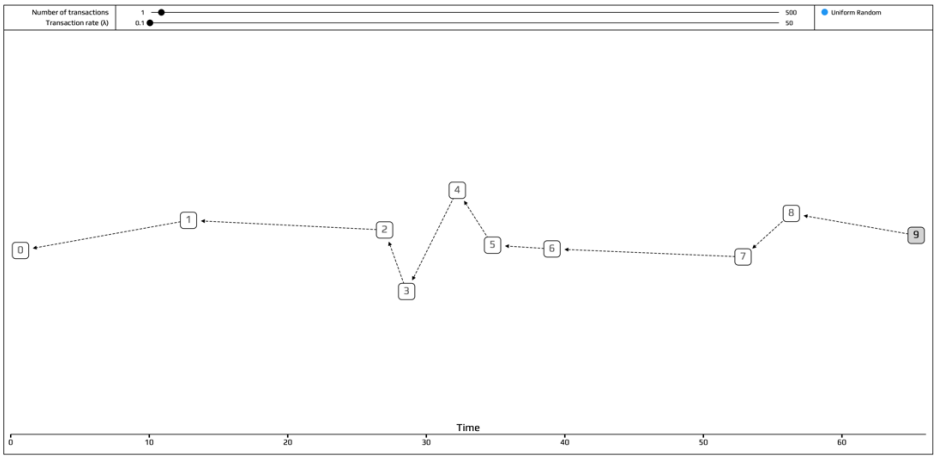


Figure 22: Simulation for a low transaction rate in the Tangle

- **High Value of Lambda:** In this case, when there is a high transaction rate (λ), this means that the in-between range of time is minimum. It is very interesting to see how this affects the structure of *Tangle.* Setting the number of transactions to 13 and the transaction rate (λ) at 12.7, the result is as it follows:

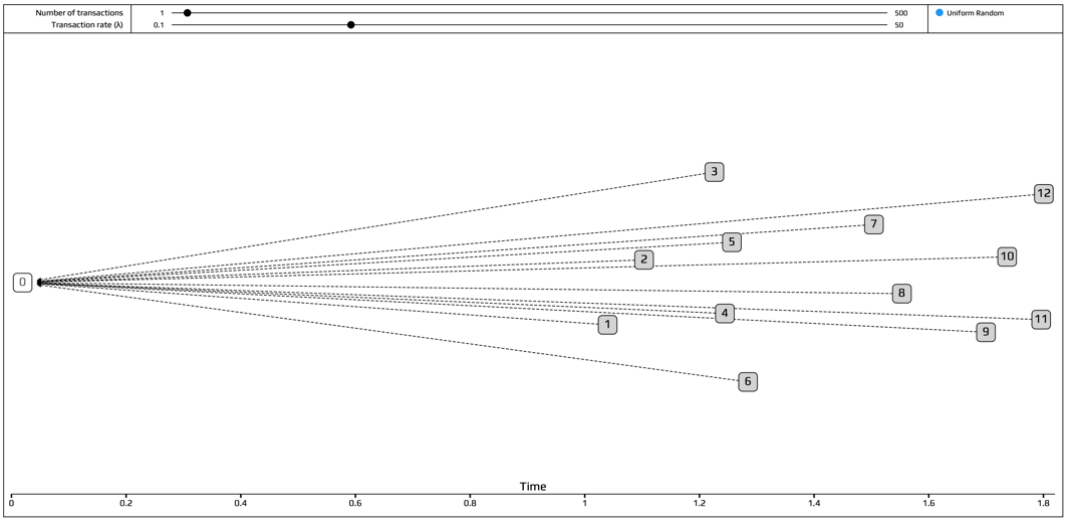


Figure 23: Simulation for a high transaction rate in the Tangle

This graphical result relies on what is called the *lag period*, referred as the variable *h.* When a new transaction is done, it takes some time in order to execute and complete the

computations for that transaction to emit. Therefore, *h* attributes for the time period that it takes to carry out these executions. When performing these actions, these new transactions disappear from the public, and that is why when there is a high transaction rate, the only previous transaction is the genesis. According to Gal, *h* is one of the keys in the *Tangle* model, terms of closeness to real life and unpredictability. (Gal, 2018)

What are random walks?

When it comes to validate a transaction, the rule is to validate two older transactions, but there is not a rule asserting that it has to be the newest transactions emitted. It is not as Blockchain, where there are incentives to the miners that search for the new blocks and for validating the transactions, in order to not find double spending nor fake cryptocurrencies. In IOTA there are no miners, the responsible to validate the transactions are the users itself.

We have knowledge about the random way to select and choose the tips when emitting a new transaction, without any restriction in terms of choiceness. But, due to the fact that there are no specific rules in terms of choosing the transaction to validate, there is one selection strategy in order to combat the so called *"lazy tips"*, called the "Weighted Random Walk". These doesn't validate the most recent transactions, but the oldest. As an example, transaction number fourteen is identified as a lazy user, because it doesn't validate transaction number eleven nor eight, but validates the earliest ones. (Gal, 2018)

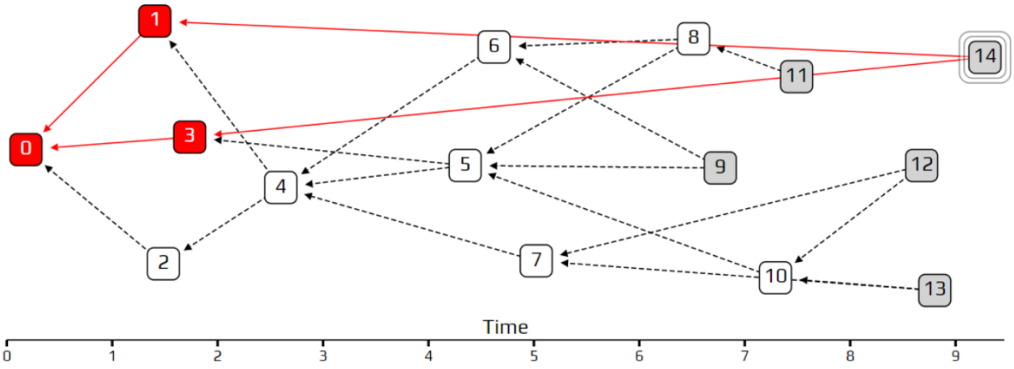


Figure 24: Example of a Lazy Tips

In order to manage these situations, they introduced the *random walks.* There are two types:

- **Unweighted random walk:** In these, the conduct is easy. The algorithm follows all the path, the red line is the path that has gone through and the blue lines are the ones that will follow. This goes from the genesis to the most recent transactions, from left to right in this specific case. Hence, when it arrives to each transaction, it validates it. For example, in Figure 25, it has now validated transaction number two. But the question is, how does this algorithm choses which direction to follow?

  In this case, all the transactions to go have the same probability to be elected, in this case the probability comes up to one third. This means that they all play equally, without any preference of choiceness.

  The transactions that are lighter located in the right side, are the newest transactions.

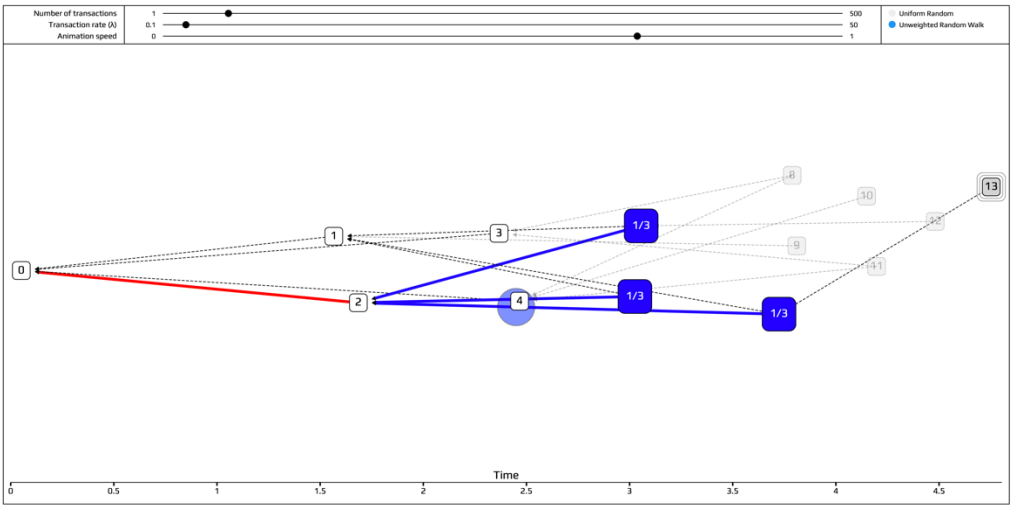


Figure 25: Unweighted Random Walk

- **Weight random walks:** This is the method chosen in order to face the so-called lazy tips. As we have seen there are no rules nor obligations, beside the two required verifications. In this part, it will be introduced the “cumulative weight”. The function is at follows: the probability to be chosen in order to be verified is greater if the cumulative weight is bigger, directly and indirectly. For example, as seen in Figure 26, among the three options to follow the blue direction, number six will have a bigger probability to be chose, and indeed, it choses transaction number six.

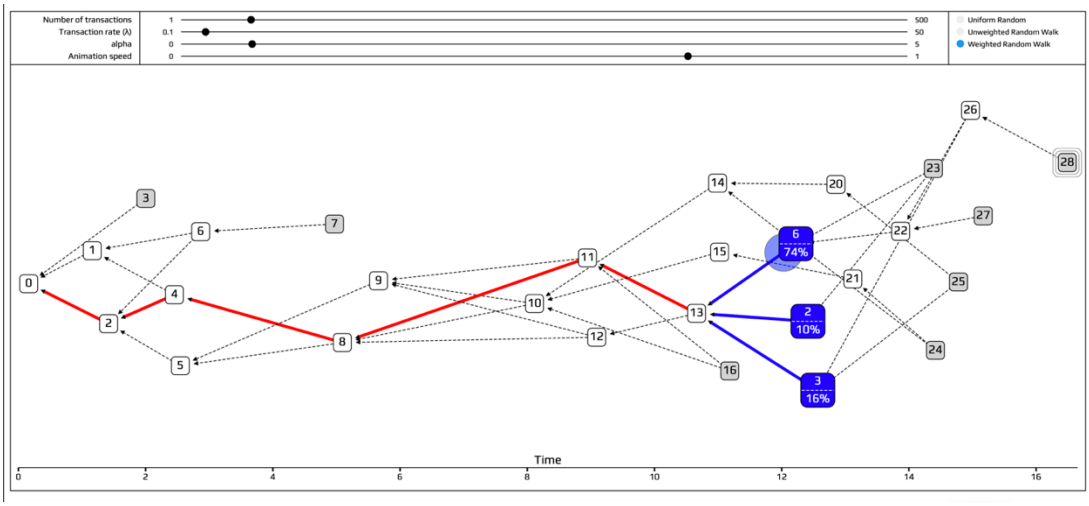


Figure 26: Weight random walk

Beside that this is a very good methodology in order to vanish these lazy tips, it won't be accurate to maximize this methodology, because it would look like this:

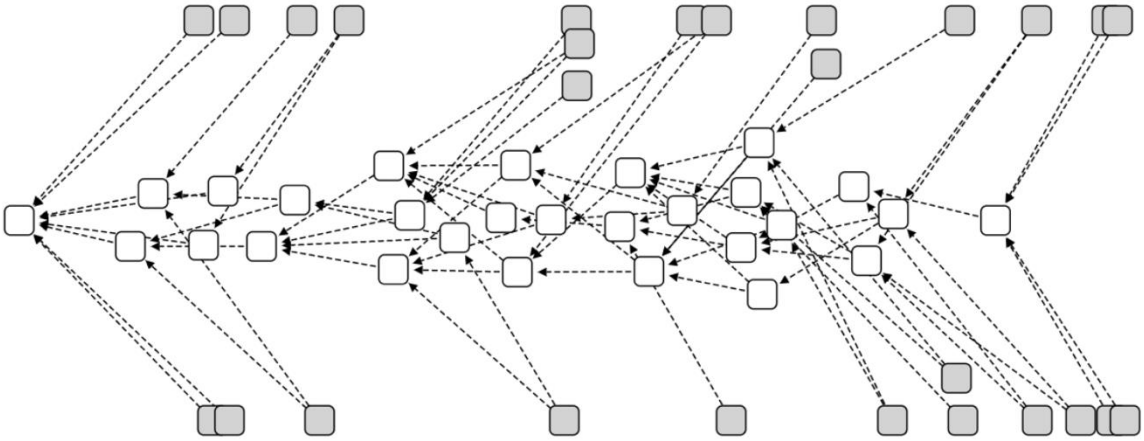


Figure 27: Enlargement of the methodology weight random walks

The grey transactions (tips) are the ones that as a result of their lower value of cumulative weight have been left aside, and the white ones have been the chosen ones, thanks to the high number of cumulative weights. If executing this methodology in extent, there is a danger to

forget about the transactions of a lower cumulative weight, and therefore instability of the structure.

This spread of the transactions is not normal and doesn't lead to a correct functioning of IOTA. The appropriate and equitable way would have some tips on the right side, but not in the above and lower position of the linear structure itself, as it is shown in the figure above.

In such wise, the solution to this irregularity relies on the parameter α , which evaluates the relevance of the aggregated weight. It does not only consider the quantity but the importance.

This parameter has to be adjusted, it has to find a fair value avoiding extremes in order to find the right balance between the lazy tips and the tips left aside. It is crucial to understand that each step is independent from another, and that the decisions in each one of these steps do not rely on the decision-making of the others.

With the introduction of IOTA among the different main characteristics, an equalized and efficient *tangle* should look like this:

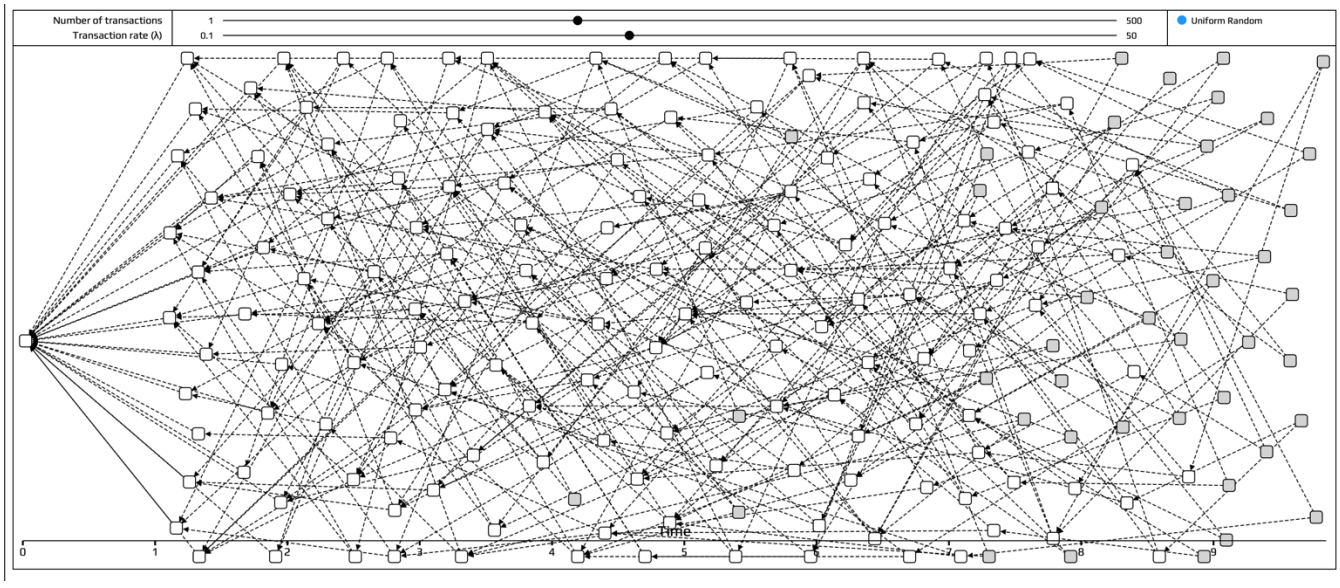


Figure 28: An example of a Tangle

An example of a recorded transaction in IOTA:

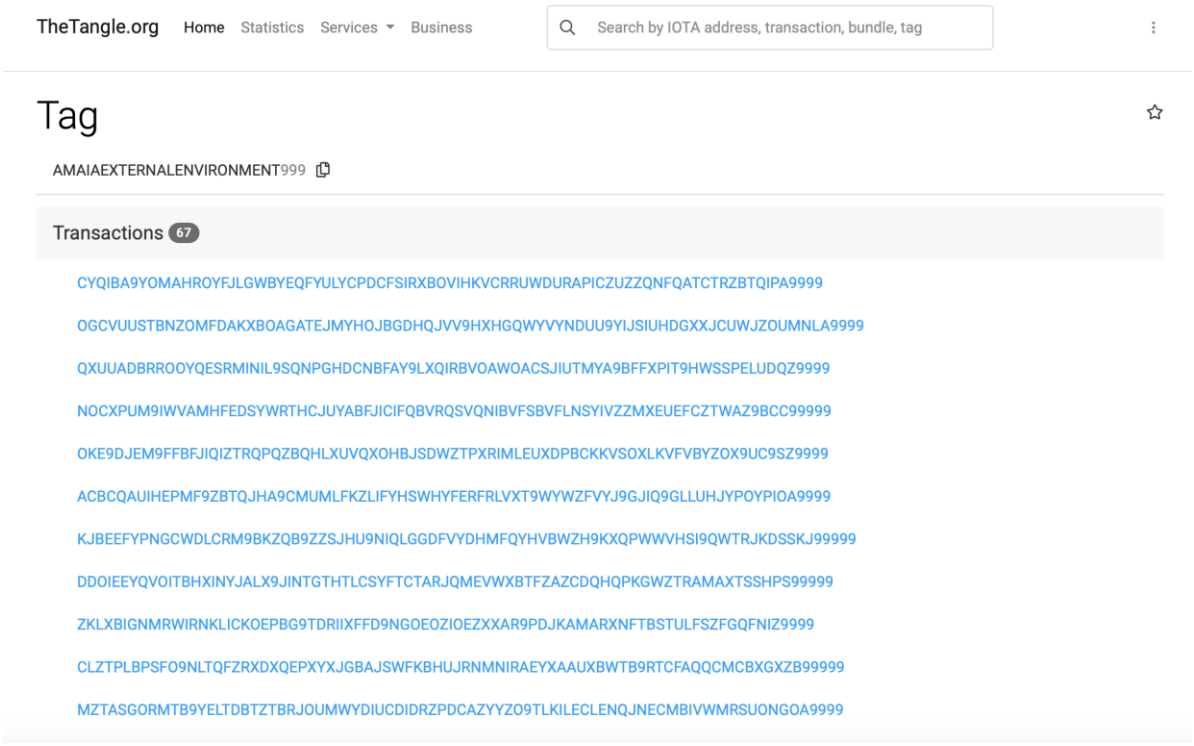


Figure 29: Tags in Tangle

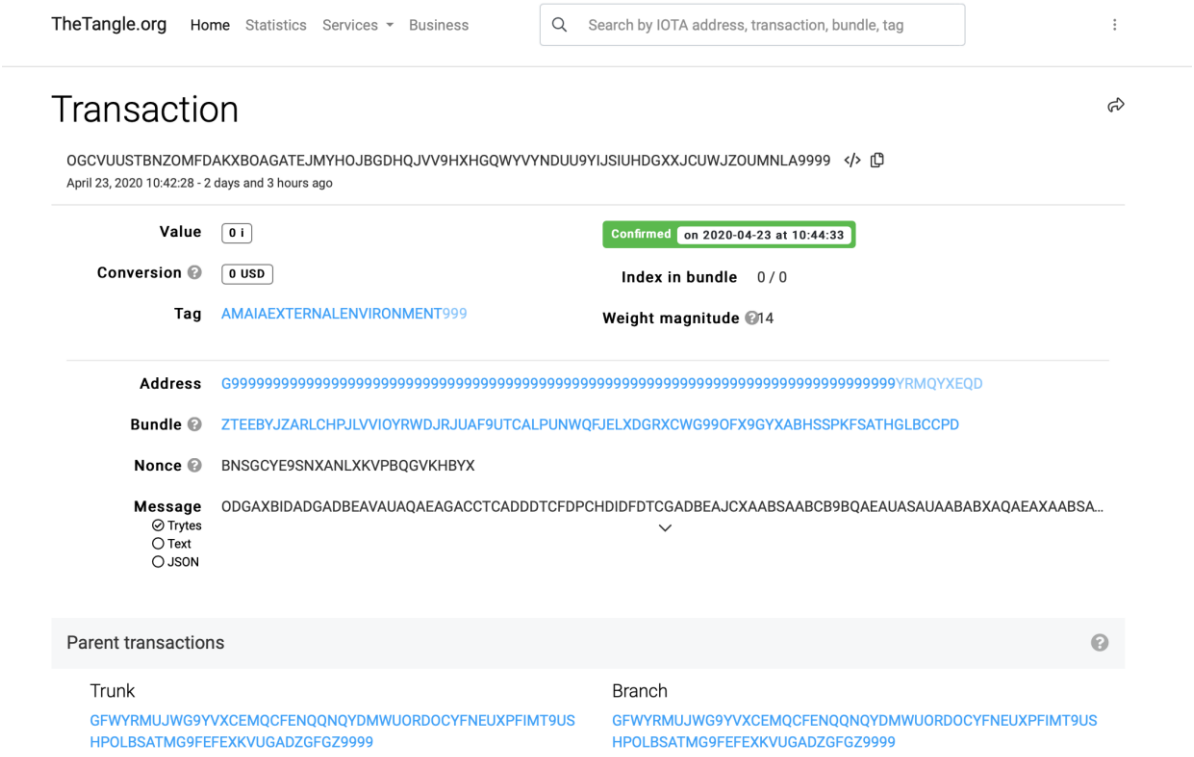


Figure 30: A transaction in Tangle.

As seen in Figure 30, the confirmation of the transaction is exposed.

5.5. Raspberry Pi

The Raspberry Pi is a compacted single computer board which was first launched to the market in 2012. It had an impressive acceptance in the market due to many advantages that presented, but also due to the low cost that it had, which was approximately 25$.

This computer was thought in order to boost the knowledge of computer science in fields such as education in schools, but also in an indirect way, such as for physical experiments in other subjects as science. It works as a computer, but it comes without keyboard nor mouse. Normally, the Raspberry Pi is used in robotics or in weather stations among others, partly thanks to its portability, capability and size, which is shocking taking into account that we are talking about a computer. (Raspberry Pi Foundation, n.d)

Figure 31: Raspberry Pi 4

The board itself runs on Raspbian which is the official operating system for the Raspberry Pi, which hangs from the family of Linux, which is a free operating system (open sourced). Thus, comes in a small SD card, in the same box as the Raspberry Pi. The board has everything as a computer, the Office package, games, coding and programming programs among others. (Raspberry Pi Foundation, n.d)

Hardware of Raspberry Pi 4

In our process, the improvement from the actual solution to the first final solution was the introduction of the Raspberry Pi 4, due to the powerfulness, the capabilities and especially for its dimensions. Therefore, it will be explained only the technical characteristics of the fourth version, being:

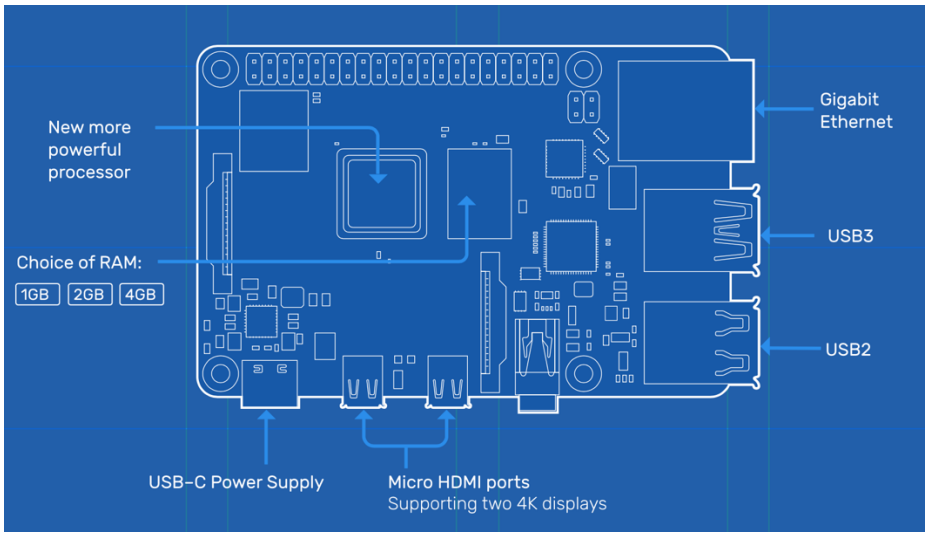


Figure 32: Technical Characteristics of Raspberry Pi 4

- 2 USB ports
- Gigabit Ethernet
- USB-C Port. Being the input for the power supply.
- 2 Micro HDMI ports
- The RAM capacity can be chosen between 1GB, 2GB or 4GB
- MicroSD port

The installation of the operating system in done through a MicroSD, which is included in the box of the Raspberry Pi.

What is the Sense HAT?

The Raspberry Sense HAT is an additive board to the Raspberry Pi. This board was designed to be joined to the computer board and with the sensors that it has on the surface can get data from the

environment, having also a LED screen and a joystick. The data that can get is: (Raspberry Pi Foundation, n.d)

- Temperature. It measures temperature in Celsius degrees.
- Orientation. Measured by a gyroscope
- Barometric Pressure. It is measured in millibars.
- Acceleration. Measured by an accelerometer. When it is not moving, it measures gravity and direction. When it is on the move, it measures the acceleration force and the direction.
- Magnetism. The single angle comes in between 0º and 360º
- Humidity. It is represented as a percentage.

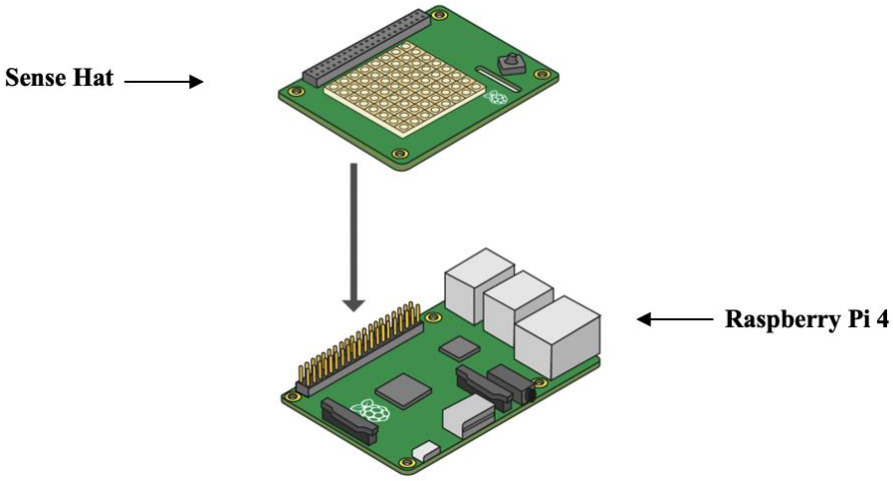


Figure 33: The Sense Hat

This board it was firstly design for the Astro Pi Challenge and it started to be used in the International Space Station. The idea was fun, in order to promote coding and programing with the Raspberry Pi among teens, the best codes executed with computer board would be send to the space in order to develop investigations. (Doutel, 2016)

5.6. The microcontroller ESP32

The most novel and relevant part of the process improvement corresponds to the design and development of the autonomous device, with a long useful life, very low cost and capable of recording various data on Blockchain chains.

In the final improvement of the process, the decision was to substitute the Raspberry Pi 4 and the Sense Hat with a microcontroller ESP32. This decision was made basically because the Raspberry Pi was way too much of what it was needed for the process, its capacities and powerfulness were beyond expectations. Also, taking into account that it was for a tracking service, we had to keep in mind the price of the solution, therefore the added value presented when choosing the Raspberry Pi and the Sense Hat was quite big and it wasn't realistic at all when talking about costs.

Consequently, it was decided to forget about the solution with the computer board and to look for another board or microcontroller which would present a great combination between cost and capabilities. And for that reason, it was chosen the ESP32 for providing a processing capacity that is considered adequate, together with communication capacities that may be sufficient in some areas. This microcontroller is programmable from Arduino IDE.

The ESP32 is a microcontroller which belongs to the company Espressif, focused on the realization of chipsets among other products. This company released the two most famous microcontrollers for IoT, ESP8266 and ESP32. (Espressif, 2019)

The general characteristics of ESP32 are:

- It has integrated classic Bluetooth, low energy Bluetooth and low energy Bluetooth that can be connected to Smartphones and reduced energy beacons emissions.
- It works in environments between -40º degrees and +125º degrees Celsius.
- Due to the fact that it was design especially for IoT and wireless connections, the durability of its battery is extremely high with a very low consumption.

The latest series, which is ESP32 (S2) does not, in principle, incorporate two CPU cores, although it has a higher computing capacity. Also, BLE is not available. However, it disables radio frequency when it is not needed (resulting in potentially longer battery life) and integrates support for RSA and AES256 (crypto algorithms).

As for the Espressif family, the other least capable and oldest processor it can provide is the ESP8266, whose computing power we do not think is sufficient for the process. However, it is possible to connect it directly to a hub or server, in which case the heavy part of the computation would be done outside the microcontroller.

Figure 34: Microcontrollers ESP32

5.7. Sensors

The typology and cost of the sensors do vary, those that can be connected directly to the microcontroller are extremely cheap. Alternatively, autonomous sensors can be used, connected to the microcontroller via Bluetooth. The price of the latter varies, but the simplest can be obtained at prices compatible with our requirements.

A separate mention deserves the GPS positioning sensors. For tracking solutions, where a single device can track a pallet, the price may not be relevant. For other solutions where positioning must be carried out in closed spaces or where it must place other people or machinery in close proximity, equivalent results can be obtained by using associated Bluetooth beacons to people or machinery, and captured by devices deployed nearby, at a cost more than acceptable.

In the first process improvement the sensors used where an additional board, called Sense Hat. This board, as explained previously has multiple sensors in order to capture great amount of information. Because of the price of the board, which wasn't the appropriate for the deployment at the time, it was figured out another alternative.

This alternative, takes part in the second process improvement, and it is called DHT11. Thus, is a sensor that measures temperature and humidity, and will be joined to the ESP32. This sensor needs a logical converter (3.3V-5V) to work with the ESP32.

Figure 35: DHT11 Sensor

5.8. Communications: LoRa

In terms of communication, there was a total of four possibilities: Wi-Fi, BLE (Bluetooth), LoRa and Narrowband-IoT (included in LPWA 5G). A priori it was ruled out directly the use of GSM despite its ubiquity, due to the battery consumption it requires.

When using the Espressif family, the availability of Wi-Fi and BLE is immediate in ESP32. The problem will be in the case of Wi-Fi in determining the frequency of communication to maximize the life of the battery, without excessively penalizing the characteristics of the device. Fortunately, Espressif provides a deep-sleep mode that help greatly decrease processor consumption.

The use of the BLE seems adequate for a modular architecture where we separate the microcontroller from the sensors. This type of architecture greatly limits current consumption and allows the deployment of the sensors to be further simplified, especially in the case of tracking people or machinery (proximity to them) which can be useful in the application in the oil of olive, or in the tracking of sub-elements of a pallet, for example (intelligent packaging).

The specific case of the Narrowband-IT modules allows communications to any of the microcontrollers, at a larger distance and with reduced consumption, although they require a data subscription. To put a comparison, we have the following: (Zidek, Janacova, Hosovsky, Pitel, & Lazorik, 2018)

Standard	Range	Pros	Cons
NB – IoT	From 10 to 15 kilometers	Sensor reading, tracking, fleets	Subscription and sim card
Bluetooth	20 meters	Proximity reading, *Smart-homes*	High frequency of communications
GSM	35 kilometers	Elevated distance	Higher consumption of battery

Table 1: Differences among NB, BLE and GSM

However, the attractiveness of this type of communications, the costs of the necessary modules exceed the maximum budget set at that point. This would limit this type of communication to change for a mixed architecture.

After analyzing all the possible modules of communication, firstly, in the first improvement solution there was a preference to develop the communications through Wi-Fi. This idea was also turned down, because of the low durability of the battery, therefore it stand-out a new idea: LoRa

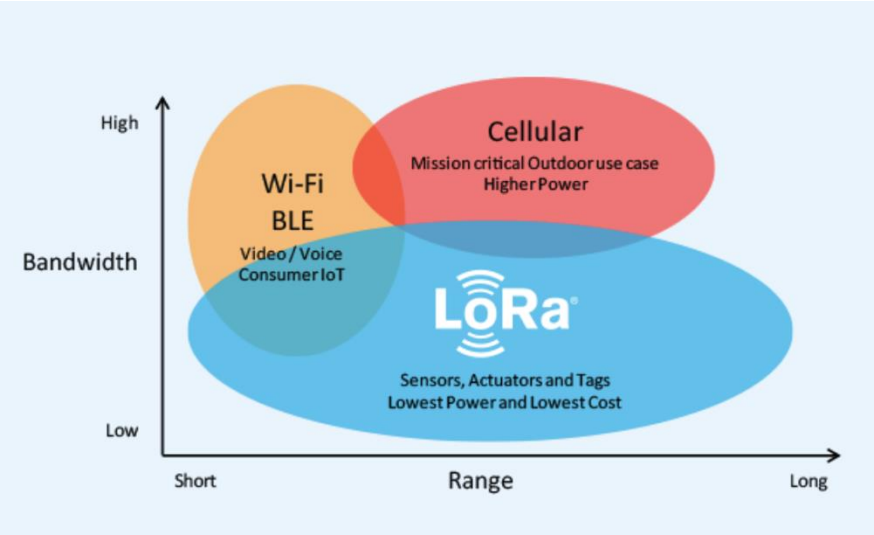


Figure 36: Wi-Fi, BLE and LoRa

What is LoRa?

LoRa comes from what is known as *long-range* information links, basically is a type of radio frequency modulation used in *low-power wide-area networks.* According to Haxhibeqiri, Poorter, Moerman and Hoebeke, it has shown as a need, the ability to communicate within kilometers, not just meters, and this is why we have chosen LoRa. Potentially, you can create links between very extensive networks as allows long distance communications in metropolitan areas and even larger in agrarian areas, arriving up to 15 kilometers.

The function is easy: the devices, like ESP32, that are complemented with LoRa (physical layer), can send messages through radio frequency. Thus, means that the message will be translated in a signal and will be sent to what is known as the Gateway, that we will explain deeply further. (LoRa Developers Portal, 2019)

The technology LoRa presents multiple advantages and characteristics:

- As said previously, it has a wide range of distances when it comes to communications, depending on the localization of the end nodes. The expected coverage range in metropolitan areas is between two and five kilometers and about fifteen kilometers in agricultural areas.
- The life of its battery can come up to ten years.
- It uses a *Chirp Spread Spectrum Modulation*, in other words, this allows to encode information. The result of this modulation presents a higher resistance to interferences and also a higher difficulty to interfere.
- It has 6 "*spreading factors*", that go from SF7 to SF12 (in Europe). Normally when there is an inferior spreading factor, a bigger amount of data can be encoded in unit of time. And when there is a greater spreading factor, a smaller amount of data can be encoded per unit of time.

LoRa has been a major player on its league, it has been deployed majorly for IoT due to the opportunity to create and control a large network of communications within physical layers. Indeed, this physical layer must have what is called a communication protocol.

Is LoRa the same as LoRaWAN?

Actually, major people confuse LoRa with LoRaWAN, and they are not the same. We understand as LoRaWAN the communication protocol that is defined for LoRa, in order to manage its devices. Behind the physical layer it is essential to define a good architecture of the network, to maximize the connectivity and minimize the inaccuracies.

LoRaWAN is composed by:

- **Nodes:** These are the final devices (also known as end-nodes), that send information to the gateway.
- **Gateway:** Is in charge of sending and receiving set of information to the nodes.

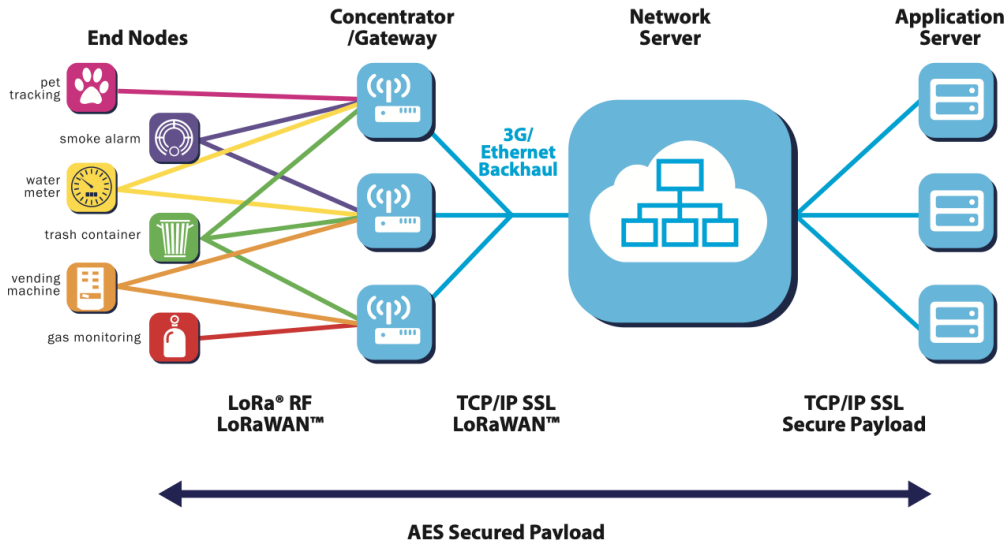


Figure 37: How LoRaWAN works

According to Figure 35, the gateways are the ones in charge of relay the messages between the end-nodes and the network server. There are multiple gateways and the end-nodes do not correspond to one specific gateway. This means that any gateway available at the time can receive the set of information. Thus, results in a better and more efficient usage of the gateways, thanks to their openness. (LoRa Alliance, 2015)

The functionality is simple and accurate. The end-nodes send the information to the gateways. These receive the signals of the encrypted information and send it to the network server, which is located in

Holland, and finally this information arrives to our server, located in Barcelona. In order to put up a personal Gateway you do not need any kind of license.

The Gateway chosen for the process is from *The Things Network,* which attend as far as thousands of nodes.

Figure 38: The Things Network Gateway

When taking into account the network server, it has to be quite powerful in order to absorb in all the data received from the different gateways. Thus, is accomplished by a "*multichannel multi-modem transceiver",* placed in the gateway. (LoRa Alliance, 2015)

The duration of the battery is enlarged, due to the methodology of communication. The fact that each end-node is put in a completely different scenario defines the network itself as nonsynchronous, meaning that they will send the information whenever it is ready or planned.

Not to forget, it has geolocation without the obligation of using a GPS, which is the key point for the project. All the set of data is encrypted by a methodology of encryption, in both directions. Hence, it complies with the protection and security standards.

5.9. Communication Protocol: MQTT

Message Queue Telemetry Transport is known as a communication protocol (open source) for IoT devices, for machines to machines. It was created by IBM, around the nineties. As the name defines itself, MQTT is a type of transport for messages and it is based on what is known as *Internet Protocol Suite* or TCP/IP. Nowadays it has converted itself in the principle communication protocol in the world of IoT. (Llamas, 2019)

In our process improvement, MQTT takes place in the phase of sending the messages. In other words, the ESP32 with LoRa sends the set of data in a MQTT format through LoRaWAN. These signals are received by the gateway and then processed.

The MQTT broker chosen for the project is Eclipse Mosquitto, which is an open source broker.

How does MQTT works?

In MQTT the key words are: (Llamas, 2019)

- **Publisher:** Publish the message.
- **Subscriber:** Receive the messages that has been subscribed to.
- **Broker:** In charge of receiving the messages published, and then redirecting those to the correct location, in other words managing and guiding the information.

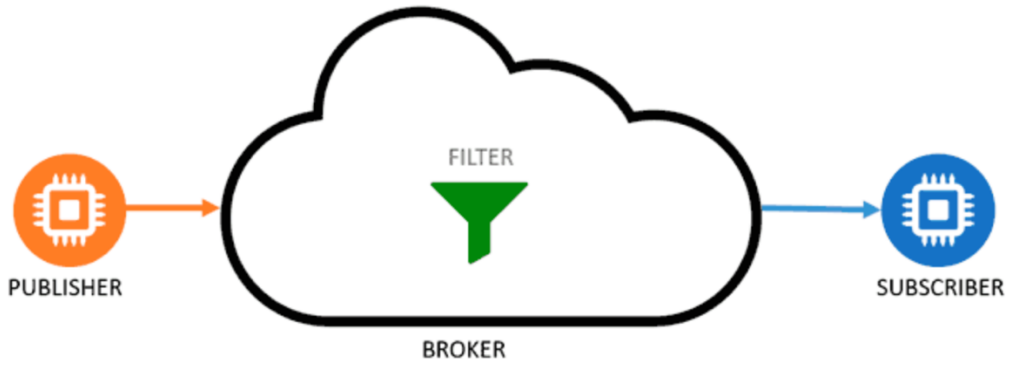


Figure 39: MQTT System

We may say that MQTT is a push service, this means that the publisher contacts the broker, and this manages the information. The key part in this system is the process of filtering the information, it is nearly the base of MQTT. Clients emit those messages with a specific "theme", the broker receives it and manages the data. Later, subscribers that have previously subscribed to the topic will receive the information, so in order to receive messages the client must be subscribed to. These topics are hierarchically predisposed. (Llamas, 2019)

The published and the subscriber doesn't have direct connection, it is the publisher that contacts the broker, and the subscriber that is contacted by the broker, and vice versa.

There are different types of commands, the principals are:

- **CONNECT and CONNAK:** When a publisher wants to connect with the broker, it sends a CONNECT message with all the necessary information, such as the client identificatory among others.
  The broker responds with a CONNAK message, which is the response of the requirement sent previously.
- **SUBSCRIBE, UNSUBSCRIBE and UNSUBACK:** In order to subscribe to one specific topic, the client must send the message SUBSCRIBE. And, in order to cancel the subscription, it must be sent UNSUBSCRIBE.
  UNSUBACK is when the subscription has been effective.
- **PUBLISH, PUBACK and PUBCOMP:** In order to publish an amount of data, the publisher must send the message to the broker by PUBLISH and when the publication has been accepted, a message of PUBACK will be received. When the topic has been published, the corresponding message would be PUBCOMP.
- **PINGREQ and PINGRESP:** In order to confirm the connectivity, clients send PINGREQ and the broker has to respond PINGRESP, if it is still active.
- **DISCONNECT:** The final step in order to cut off the connection is sent DISCONNECT.

What is the structure of a MQTT message?

As Light comments in his article *Mosquitto: server and client implementation of the MQTT protocol,* MQTT is lightweight, meaning that the message won't take over much bytes and it is not design for taking it. Those messages are composed by three parts, one compulsory and two optional:

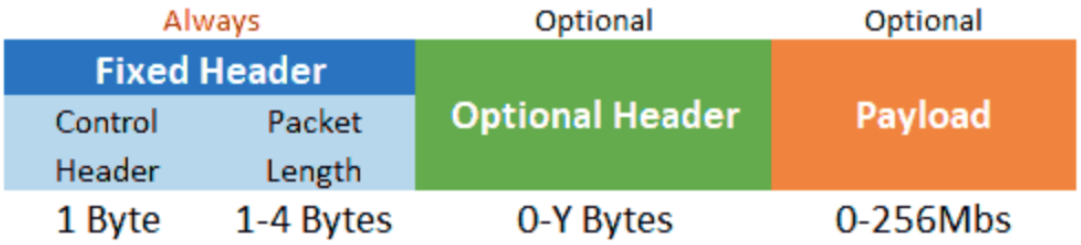


Figure 40: A structure of a MQTT message

- **Fixed Header:** This part is compulsory. The length is codified between one and four bytes. The control header occupies one byte, and it is in charge of identifying the message.
- **Optional Header:** This part may contain additional information about the message, sometimes this kind of part is also mandatory.
- **Payload:** The actual content of the message. The volume can come up to 256 megabits per second, but in real implementations it is normally up to 4 kilobits per second.

QoS: Quality of Service

In order to manage the quality of service of a message, in other words, the robustness of those message deliveries, MQTT has stablished three different levels: (IBM, n.d)

1. *"QoS 0: At most once delivery":* Meaning that the message has been sent once, and any further measure in order to ensure the transfer of it, won't be taken into account.
2. *"QoS 1: At least once delivery":* Better than QoS 0. Meaning that there will be a retry in sending the message just before an acceptance of it has been received.
3. *"QoS 2: Exactly once delivery":* Each message is assured to be sent once to the subscriber.

6. The improvements in numbers

In order to deeply understand the improvements made in an economical approach we are going to chronologically explain how these enhancements impacted financially.

6.1. Costs incurred for the Actual Process

As commented previously, the first and the actual solution is based on a centralized network (MySQL) and the example for the production line is olive oil bottles.

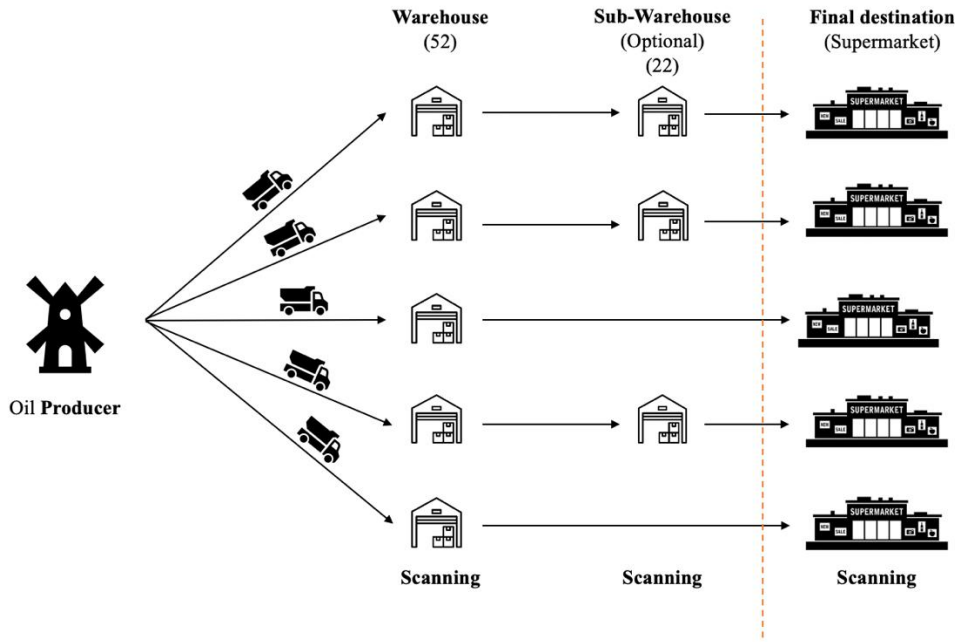


Figure 41: The layout of the actual solution

Starting with a freshly reminder of the process itself, the oil producer oversees producing the bottles of oil, assuming that in the production line all the hand-operated procedures and steps depend on them, for example the harvest of olives, the bottling and the transport.

Once the oil producer has carried out the processes and operations, it is sent to various warehouse among Spain. As known, Spain is one of the greatest producers of olive oil, not just nationally but internationally. Hence, it is assumed that the numbers of provinces which the oil is sent to, comes up to a total of 52 (as seen in Figure 39).

To follow up, there are some provinces that have sub-warehouses, which are located in order to deliver in terms of regions, for example, in Barcelona there would be a sub-warehouse for delivering to Vallés Oriental, Vallés Occidental, Barcelónes or Maresme among others. So, according to these, the estimation of the total amount of sub-warehouse comes up to 22. As seen in the sketch, this step is optional, meaning that not all the deliveries will have a sub-warehouse.

Coming to the end of the process, the final step is the deliver from the first warehouses or sub-warehouses to the supermarket. In all the parts of the delivery process, the methodology carried out to track the packages, is the most commonly known procedure, which is the scanning. Thus, is what is majorly used in nowadays delivery services, which is quite expensive, as one hand-computer device has an average price of a thousand euros.

With this said, let's get deep in the economic approach:

Actual solution of the process	
Number of provinces delivering	52
Number of sub-warehouses	22
Average price of Hand Laptops	1.000 €
Number of Hand Laptops needed	104
TOTAL	**104.000 €**

Figure 42: Economical approach of the actual solution

The numbers are not that complicated in this first approach, as commented in advance, the number of provinces delivering is 52, and the number of sub-warehouses comes to 22. Hence, the sum of both is 74 different warehouses across Spain.

We have to take into account the average price of the portable hand laptops, which is around 1.000€. When considering the number of these needed, the first approach was 74, one for each warehouse, but we must take in consideration more displays in case any of them broke, does not work properly or it just gets lost. Consequently, a concrete amount of 30 devices where summed up to the first 74, resulting in a total amount of 104 portable devices.

Lastly, multiplying the average price of the one portable hand laptop to the number of devices needed, it comes to a total amount of 104.000€ of initial investment. Taking into consideration that the network is centralized, and MySQL is free, there are no costs incurred for this.

6.2. Costs incurred for the First Process Improvement

From the actual solution to the first process improvement there is a major step. The centralized network gets completely obsolete and new devices are introduced.

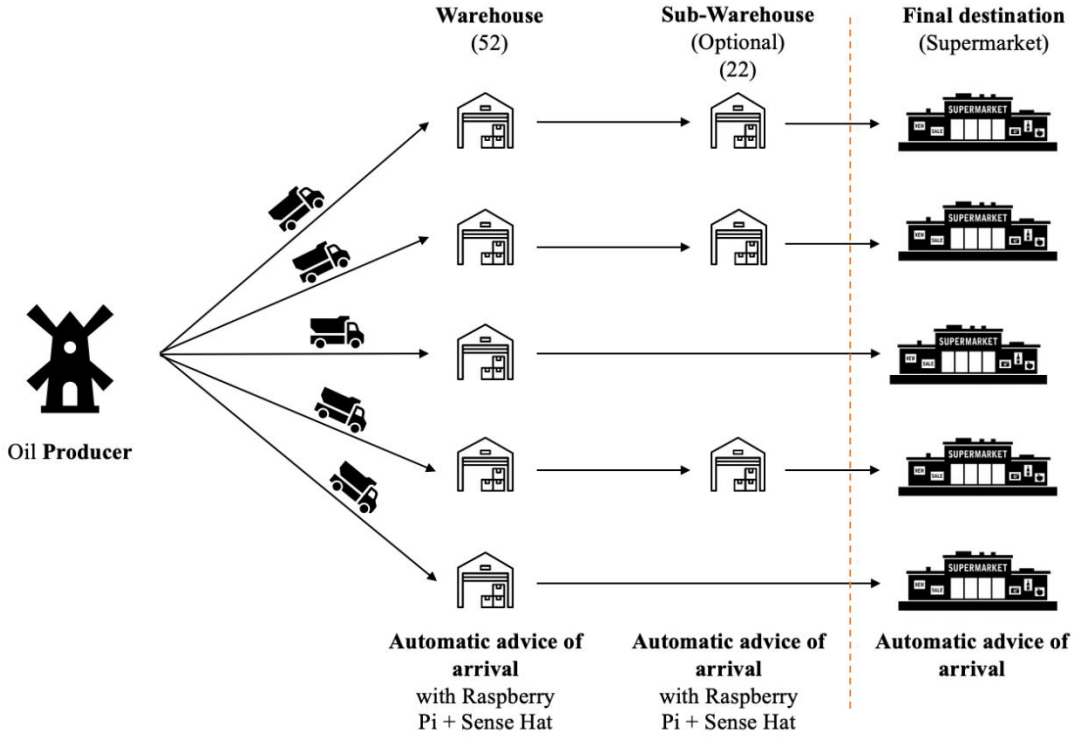


Figure 43: The layout of the first process improvement

The process here follows the same line but with some differences, the oil producer keeps performing its main handed action in terms of production until the delivery process comes in, the change is basically in the device. Once the packages have arrived at the different warehouses and sub-warehouses the Raspberry Pi, which has a beacon inside, will advise of the arrival through the Wi-Fi.

All the information about the tracking will be put up in the network automatically, also the information of the sensors from the Sense Hat.

This device is complemented with a Sense Hat, which is basically a board of sensors. These sensors collect added information of the tracking process, for example the temperature and the humidity in the truck.

At last, the packages will arrive to its final destination, the supermarkets. The process here differs as well, in order for the operator to confirm that the packages have arrived at the final destination, he/she will have to confirm it through an App, and this confirmation of arrival will be uploaded directly to BigchainDB, IOTA and to the server.

First Improvement of the process	
Number of provinces delivering	52
Number of sub-warehouses	22
Price of Raspberry Pi 4	39,45 €
Price of Sense Hat	27,90 €
Price of Batteries	0,85 €
Price of Battery Holder	0,30 €
Number of Raspberry Pi needed	1040
Number of Sense Hat needed	1040
Number of Batteries needed	1040
Number of Battery Holder needed	1040
TOTAL	**284.960 €**

Figure 44: Economical approach of the First Improvement of the Process

In terms of costs, we have to take into account a couple more things. Knowing that this initial investment is for one year, the number of provinces and sub-warehouses are maintained.

Focusing on the hardware part of the solution itself, each solution has the four components that appear in Figure 42. Therefore, a complete solution of Raspberry Pi and a Sense Hat comes up to 68,50€ each solution. In terms of quantity, knowing that each of these modular solutions will be located in

each pallet, we need at least an average of 20 per truck. In this case, we have multiplied 52 (which is the number of provinces) per 20 (devices per truck), resulting in 1.040 devices needed.

To sum up, the total price of the solution is 284.960€. It is way much than the actual solution, due the fact that the number of devices needed is expanded, and the Raspberry Pi presents way more power than the needed for this type of solution. Therefore, there was a need to find an alternative.

6.3. Costs incurred for the Final Process Improvement

The final process improvement is seen as it follows:

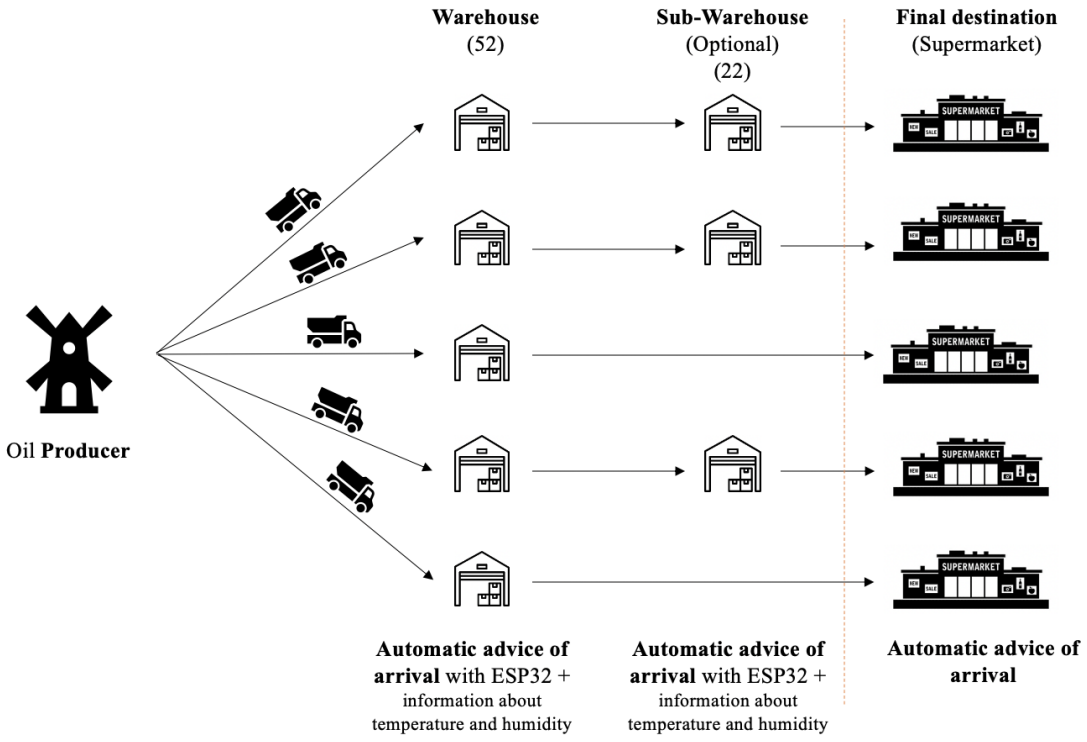


Figure 45: The layout of the first process improvement

The process itself is the same as the previous ones, the oil producer, manages all the production line of these and then the delivery service is made through trucks, so it presents the same structure as the other ones.

The difference here is the modular solution, which is an ESP32 and a sensor. This device instead of being a computer board, is a microcontroller, so in financial terms, it is way cheaper and copes with the necessities needed. Another difference is the sensor of temperature and humidity, which considering that the bottles are filled with olive oil, these parameters do matter. Finally, each of these devices needs a Gateway, which is the one to collect the information that is send through radio frequency, though LoRa and LoRaWAN.

Here the need was to develop a solution that was way more economical than the last one. Resulting in:

Final Improvement of the process	
Number of provinces delivering	52
Number of sub-warehouses	22
Price of ESP32 with LoRa+ LoRa battery+ battery holder	12,80 €
Price of Sensor DHT11	0,52 €
Price of Gateway	200,00 €
Logical converter 3.3V-5V	0,65 €
Box for the modular solution	0,80 €
Number of logical converters	1040
Number of boxes	1040
Number of ESP32+DHT11 needed	1040
Number of Gateway needed	100
TOTAL	**66.082,40 €**

Figure 46: Economical approach of the First Improvement of the Process

As seen, the price of the modular solution including the ESP32, the battery and the battery holder costs exactly 12.80€ which is more than 50€ of difference between the first improvement of the process. In terms of units required, it was followed the same procedure; we calculated an average of devices needed per truck, which we calculated 20 devices per truck. Also, we multiplicated these by 52, which are the provinces, resulting in $52 * 20 = 1040$ pieces of each component.

In order to protect the device, we included boxes and also the sensors DHT11, in order to measure temperature and humidity, which are the main information that clients request. Not to forget, the main

cost are the gateways needed, which we only need one in each warehouse or sub-warehouse, ergo we put 100, to have 26 in case there is any sort of problem or error.

The total amount is 66.082,40€, which is the best result among the three options. Therefore, this solution is not only the best in terms of hardware, but also it is the most simple, compact and precise solution among the three, but also it presents the best initial investment needed.

Lastly, among the three options, seeing that the first one was 104.000€, the second one 284.960€ and the third 66.082,40€, we clearly see the which one to choose financially speaking. The first one is an acceptable system, but it doesn't present the enough adaptability to today's market. Also, the portable devices are devices that do need periodically maintenance, in terms of hardware, time and money, which at the end it always impacts negatively financially speaking.

The second option it was the first idea improvement, it was just a sketch of what it could be achieved. But, the needed powerfulness and the capacity of the Raspberry Pi didn't match with our needs, it was way more than what we needed at the time. The fact that these devices had to be located on each pallet, forces you to set the producer a marginal price, in order for these not to affect the end price.

It is difficult to overestimate the possibilities of a technology that allows connecting an IoT data source with Blockchain, at a very low price. Even more, if we consider that the technology can work in the opposite direction, acting in some way based on the specific content recorded on Blockchain.

At this point, with this improvement of the process, it is not only way cheaper than the first solution, but also it reduced the amortization of the devices, an increase in information transparency, an increase in connectivity and an increase in quickness among others. All this, results in what is nearly the most important point, which is the reduction in the initial investment and the reduction of the total costs giving the best solution, which at the end is one of the key objectives of the overall improvement.

7. Deployment

In order to understand the process at its finest, we are going to show the execution when it comes to coding and programing. We are running in an Ubuntu 18.04.

This deployment is done on the computer of the client.

7.1. Firstly: Installing Apache, PHP and MySQL

The first thing to do is install the Apache, in order to install a web server:

```
sudo apt-get update

sudo apt-get install apache2
```

Now, we have to install PHP:

```
sudo apt-get install php libapache2-mod-php
```

Then, we will install mySQL:

```
sudo apt-get install mysql-server

sudo apt-get install python-mysqldb

sudo apt-get install phpmyadmin
```

Once installed, we have to configurate MySQL. This is to create a new iota user and a new iota database. By this it will be ready to be connected:

```
sudo mysql -u root -p
```

Later on, you will have to connect to the localhost, and create a new table. This table will be linked to the sensors and data obtained from them.

7.2. Installing BigchainDB

We decided not to install through a docker and neither through a docker compose, but simply because it wasn't functioning properly. Hence, finally we installed it step by step.

Firstly, we update the packages:

```
sudo apt update

sudo apt full-upgrade
```

We proceed with the installation of libraries need for Python3 (pip and ssl related):

```
sudo apt install -y python3-pip libssl-dev
```

Now, we have to install the latest versions of Pip3:

```
sudo pip3 install -U pip
```

Following up, we proceed with the installation of BigchainDB:

```
sudo pip3 install bigchaindb==2.0.0

bigchaindb configure
```

Then, we install MongoDB:

```
sudo apt install mongodb
```

And consequently, Tendermint:

```
sudo apt install -y unzip

wget https://github.com/tendermint/tendermint/releases/download/v0.31.5/tendermint_v0.31.5_linux_amd64.zip

unzip tendermint_v0.31.5_linux_amd64.zip

rm tendermint_v0.31.5_linux_amd64.zip

sudo mv tendermint /usr/local/bin

tendermint init
```

Lastly, we have to install Monit:

```
sudo apt install monit

bigchaindb-monit-config

sudo chown kitchen:kitchen
/home/kitchen/.tendermint/data/blockstore.db/CURRENT.bak

sudo chown kitchen:kitchen
/home/kitchen/.tendermint/data/blockstore.db/CURRENT

sudo chown kitchen:kitchen
/home/kitchen/.tendermint/data/state.db/CURRENT.bak

sudo chown kitchen:kitchen
/home/kitchen/.tendermint/data/state.db/CURRENT
```

And now we start up BigchainDB node:

```
monit -d 1
```

In order to generate public and private keys for the user, we have to generate them. Thus, is achievable by typing this is the terminal:

```
python3

from bigchaindb_driver import BigchainDB

from bigchaindb_driver.crypto import generate_keypair

bdb = BigchainDB('http://192.168.0.19')

amaia = generate_keypair()

print("pub:",amaia.public_key)

print("prv:",amaia.private_key)
```

The result should look like this:

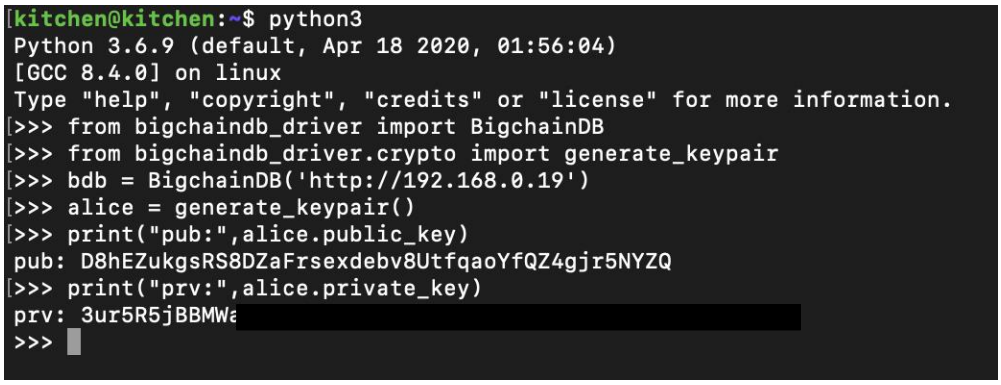


Figure 47: Generation of Public and Private Keys in BigchainDB

The source code for the BigchainDB is attached on the Annex 2.

7.3. Installing the MQTT Broker

First of all, we have to install the broker of MQTT, by typing: (Sopapun, 2018)

```
sudo apt-get update

sudo apt-get install mosquitto

sudo apt-get install mosquitto-clients
```

Now, we have to secure the broker with a password, as follows. Then, you have to generate a password in order to secure the broker:

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd mqtt_user_name
```

Now we have to direct the file we just created in order to generate a composition file:

```
sudo nano /etc/mosquitto/conf.d/default.conf

allow_anonymous false

password_file /etc/mosquitto/passwd
```

Finally, we have to restart it, in order to proceed the configurations made:

```
sudo systemctl restart mosquito
```

The source code for the MQTT broker is attached in Annex 3.

7.4. Installing IOTA

Firstly, we have to generate a seed, in order to have an identification inside Tangle:

```
cat /dev/urandom |tr -dc A-Z9|head -c${1:-81}
```

Then, we install Python drivers, in order to assure better functionalities:

```
pip install pyota[ccurl]
```

And finally, we install Pandas, which is a library to do numerical calculations. Therefore, is in charge of analyzing the set of data collected and execute a statistic summary:

```
pip3 install pandas
```

The source code for the IOTA is attached on the Annex 4.

7.5. Programing the device: ESP32

Previously, we have seen the installation that it would be made in the case we did the first improvement option, Raspberry Pi. Now, we are going to focus on the steps to follow, technically speaking, in an ESP32.

We have followed the steps of The Things Network, as the getaway that we decided for this improvement was from them.

According to *The Things Network*, the steps to follow are: (The Things Network, 2020)

1. Firstly, we have to download the program for programing the LoRa devices. This is the *Arduino Software (IDE).*
2. Then, we must download the according library, which is named *"TheThingsNetwork"*.
3. Now we can proceed with the connection of the device with the computer, in order to execute the programming on it. Hence, once connect, we must select the port "*Arduino Leonardo"*.
4. Once selected, we have to include the library previously chosen to the sketch. The code such run like this:

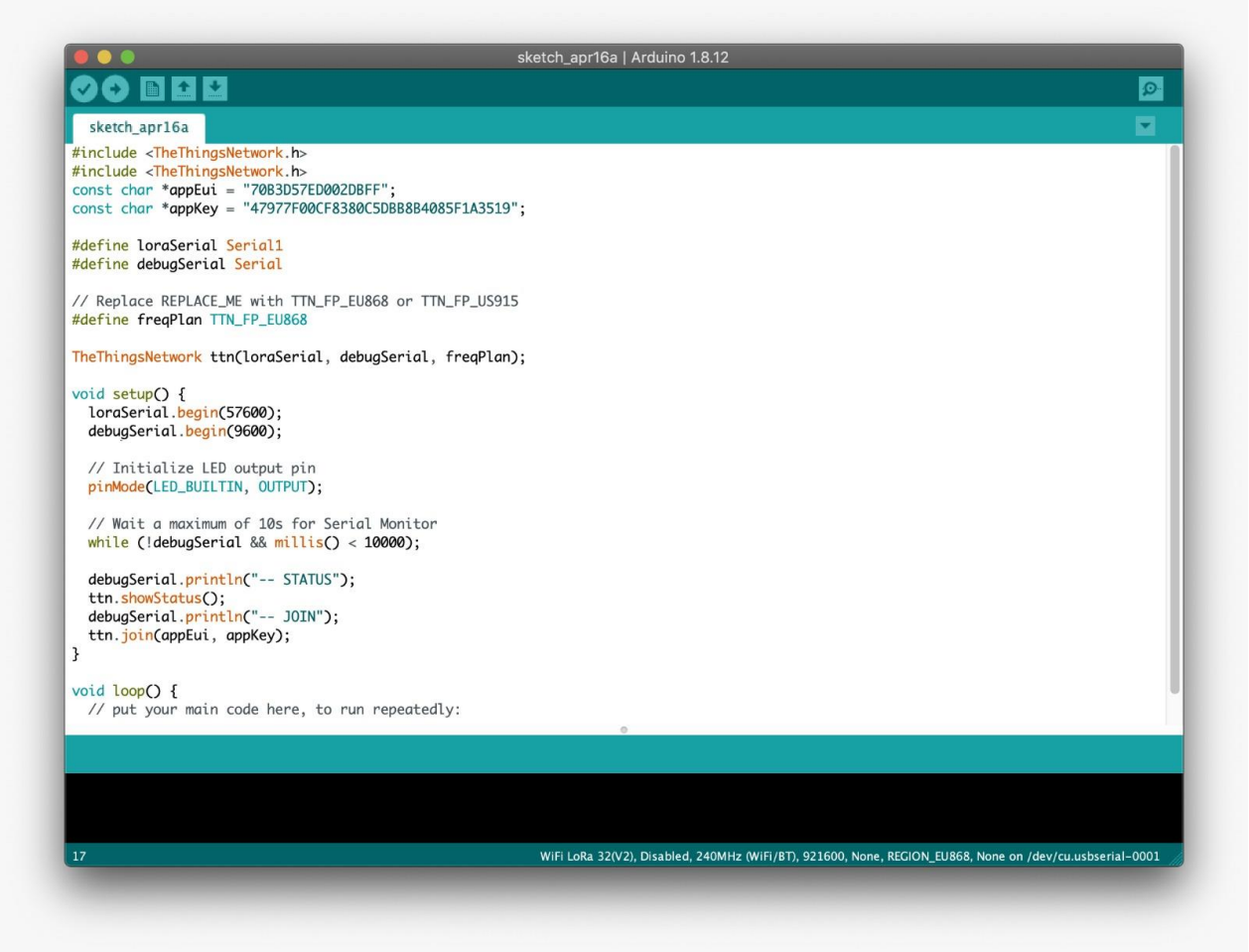


Figure 48: Software Installation of the ESP32

We have defined TTN_FP_EU868 because we are going to run these devices in Europe. Also, *"void setup",* is used when the code must run only one time. In these are defined the baud rate, which is the one you have to set up in the Serial Monitor in order to see the data collected by the ESP32.

5. Later on, we have to get the *"Extended Unique Identifier"* or EUI, which as the name describes itself, it's an identifier number, in *The Things Network.* As seen in the above Figure, we already put it down to the sketch. In order to obtained it, we just have to upload the file and go to the serial monitor. Thanks to the command we have introduced *"ttn.showStatus()",* we can get to know the information about the device.

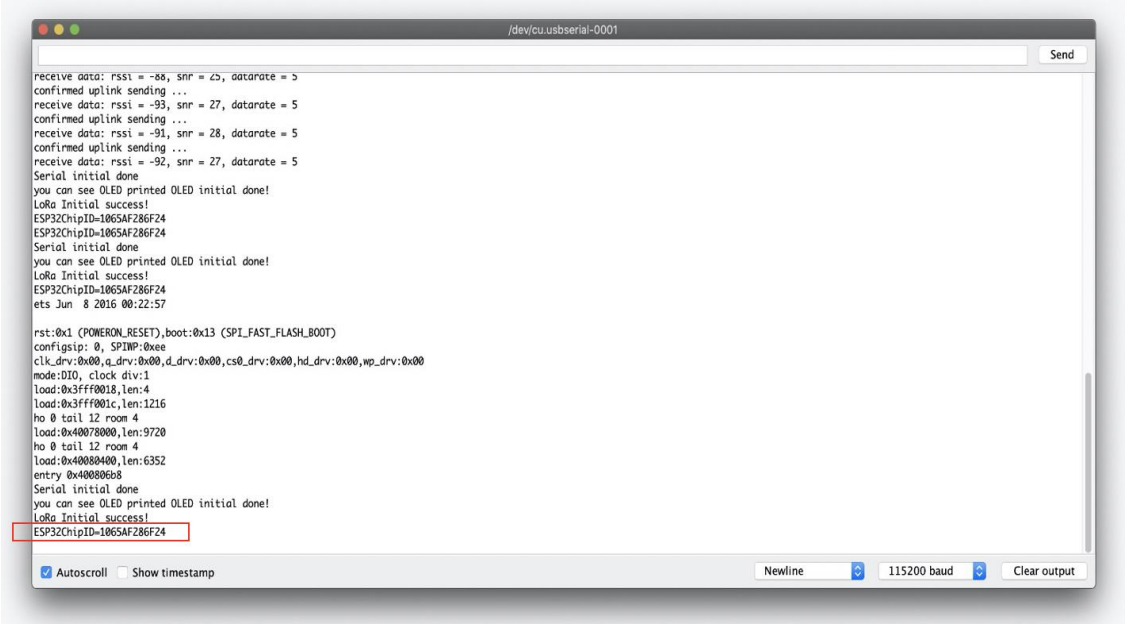


Figure 49: EUI of ESP32

6. This may be the simplest steps: the creation of an account and an application of the device Gateway. Firstly, we have to create an account, later on we have to adjoin an application, in order for the devices to communicate. As seen in the Figure above, this was the application for the device it was used for development. Every application has also a number of EUIS, as seen.

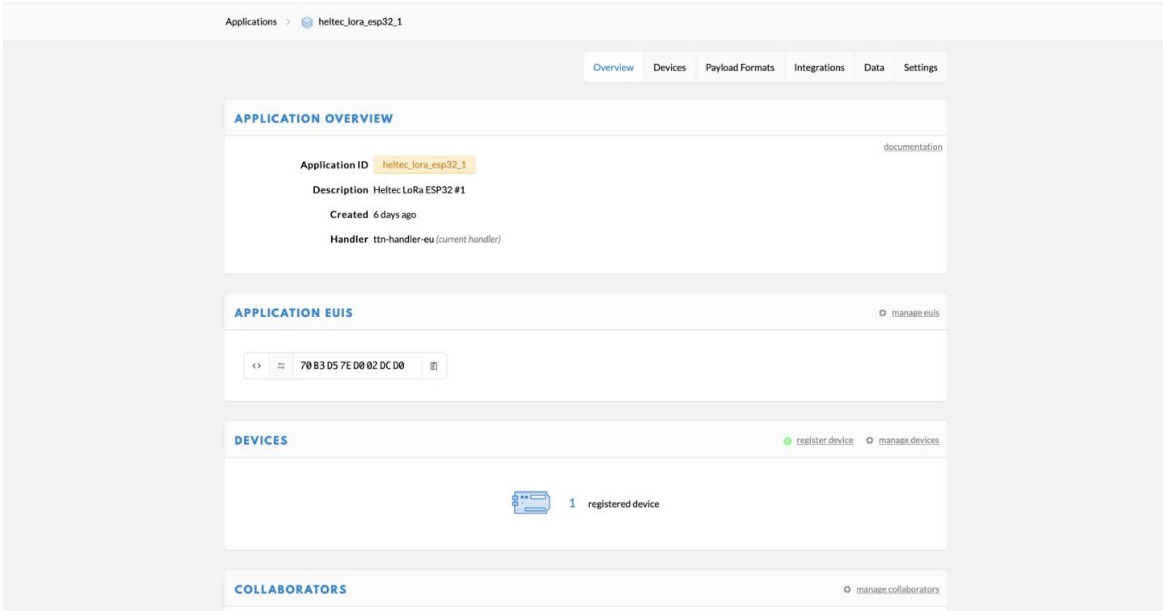


Figure 50: Application Overview of the device ESP32

7. The following step is registering the device. You must put a name for an ID, the EUI previously found and the APP key that it will be generated when registering the device. This is an example of the overview of the device and all the obtained field from the registration. As seen, the App Key is private, the Network Session Key and the App Session Key.

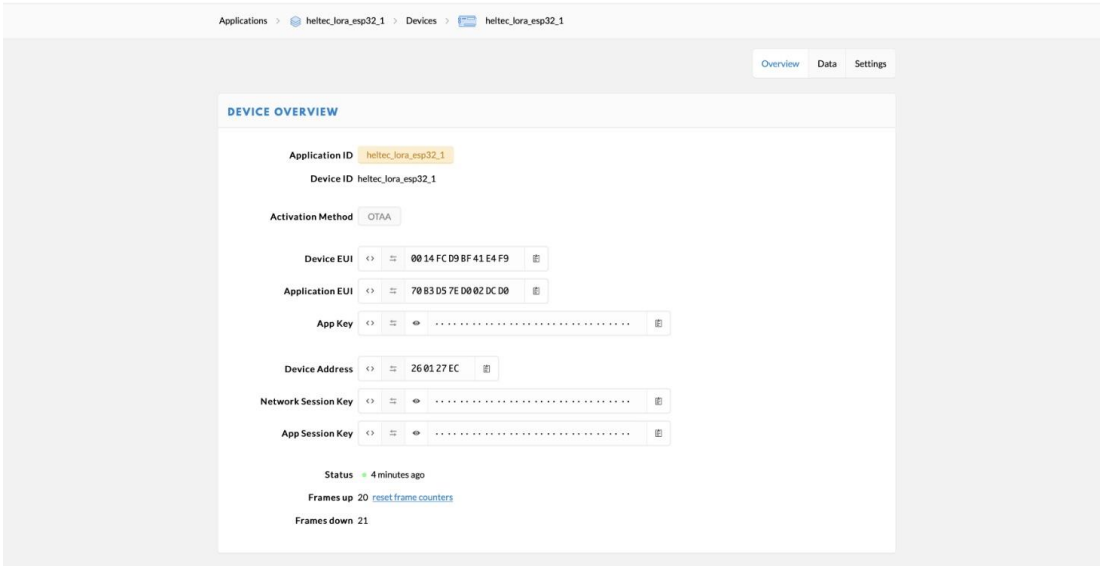


Figure 51: Device Overview of ESP32

8. Last but not least, we must activate the device. This is also known as *Over The Air Activation*, in order to communicate using the App Keys. In annex 1 we can see the hole code that was generated for the deployment of the ESP32.

   One interesting part is that the EUI of the devices and the applications is hexadecimal, this means that the identification number of the device in OTAA changes by adding 0x in the front of the number. If we compare the previous Figure with this one, the Device EUI is the same but with these two added combinations of number and letter (0x).

```
/* OTAA para*/
uint8_t DevEui[] = { 0x00, 0x14, 0xFC, 0xD9, 0xBF, 0x41, 0xE4, 0xF9 };
uint8_t AppEui[] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x02, 0xDC, 0xD0 };
uint8_t AppKey[] = { 0xC3, 0x04, [redacted] };
```

Figure 52: Device EUI, Application EUI and App Key for ESP32

The way the ESP32 works is different from the Raspberry Pi. In the Raspberry Pi, in order to catch the data, a command must be typed, which we will explained later. On an ESP32, the device doesn't have to be activated by a command, it just has to be connected to a battery. In Figure 51 we can see an example of how data is cached by an ESP32 and it is showed in the terminal. This data is collected by the microcontroller, then it is sent to our gateway and trough Wi-Fi, the central servers from *The Things Network* catches this information, which is rebooted to our server.

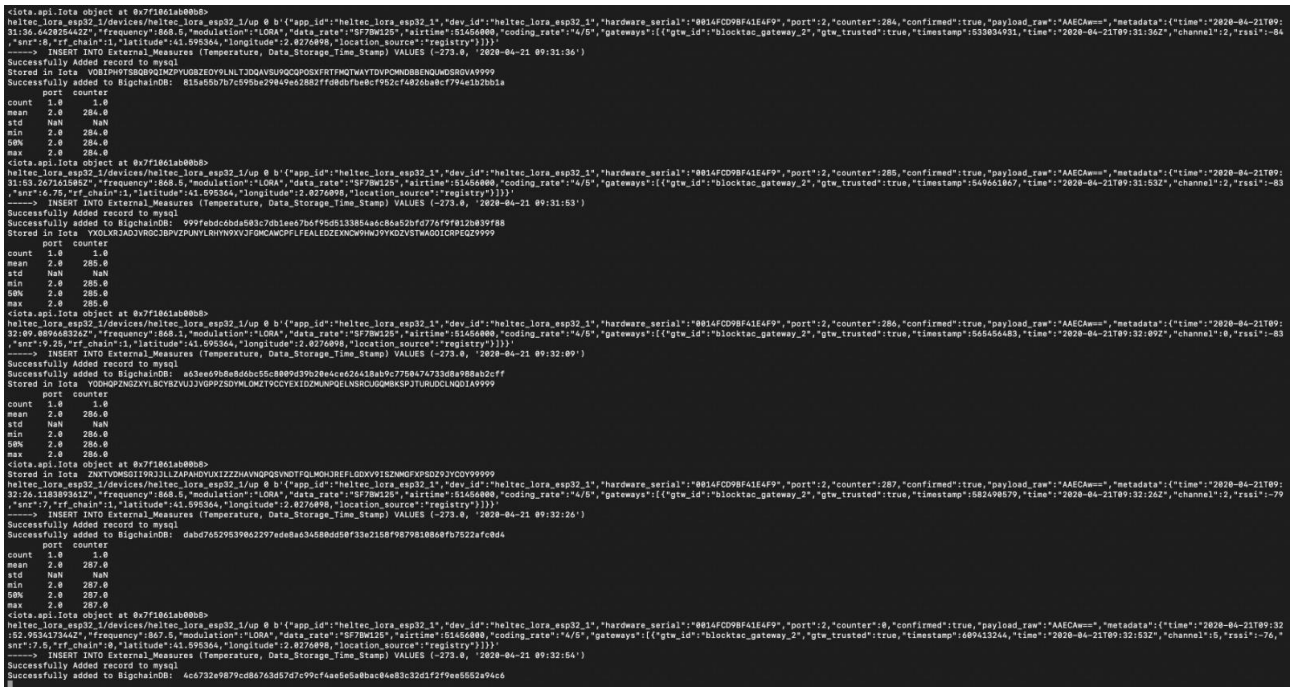


Figure 53: Result of the data collected by a ESP32

The source code for the MQTT of LoRa is attached on the Annex 5.

7.6. Programing: Raspberry Pi

The programing of the Raspberry Pi is the easiest part, as it is a device that is prepared for this. Therefore, the steps to follow are:

Firstly, the packages installed have to be update (if there is one):

```
sudo apt-get update
```

Then, the corresponding libraries of the Sense Hat and MQTT:

```
sudo apt-get install sense-hat

pip install paho-mqtt
```

With this, the programing of the Raspberry Pi is finished. This is a good point in terms of a massive installation in multiple devices, as it is not hard nor complicated and it can be done consecutively.

This is the result of the data collected from a Raspberry Pi:

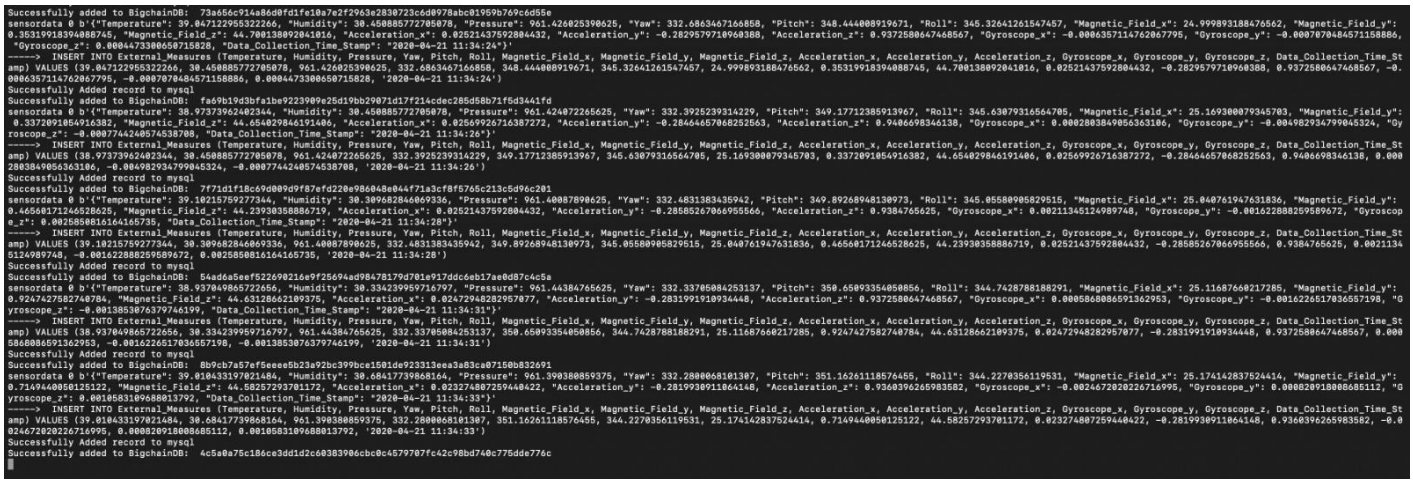


Figure 54: Result of the data collected by a Raspberry Pi

This data is actioned by the command:

```
python3 pubsensordata.py
```

Once this command is typed, the Raspberry Pi starts to collect the data. The collection of information can be done in the period of time desired, for example every 5 seconds or every 30 seconds, depending on the amount of data wanted.

The source code for the Raspberry Pi is attached on the Annex 6.

8. Future lines of application

As seen, this device presents great scalability, thanks to its capabilities, low-cost and innovation. There is a growing trend in the application of Blockchain in IoT, and therefore, Blockchain technologies can help overcome IoT challenges and difficulties when it comes to security. Due to the great diversification of Blockchains applications, all of them perfectly comprehensible for the device proposed.

Ergo, we are going to propose future line of applications of this device:

- **Putting a gateway in the final receiver:** In the process itself, we have seen how the last part of the project, when the pallets arrive to the supermarket, the operator has to advice through an application that the pallets have arrived. Hence, the proposal would be to put a gateway in the warehouse of the supermarket, therefore the operator wouldn't have to advise through the application, because the server would be automatically advised by the gateway.
  This results in the eradication of the application and thus, in a reduction of costs or a reduction in the initial investment needed.

- **Recollection of the biological data:** When this project was firstly introduced, the environmental data to take into consideration, was mainly temperature and humidity. When working closely with employees of the agricultural field, it was identified a necessity, which was the information required and needed about the biological data. In case these devices where put up in different locations such as olive oil trees, or to measure soil productivity, harvest volume, location and changes in it and so on. This information will be accessed by different members of the community, (farmers, oil mills, cooperatives, bottlers, transporters and shops selling to the public). Thanks to the cooperation of these operators, a series of data will be recorded on the entire process from the cultivation, plot, variety of olive, the use of fertilizers, insecticides or other care of the cultivation, the passage through the oil mill, the additives added to the product, packaging, transport times and conditions and in general any data for which there are sensors. Hence, this creates the opportunity for these devices to be the "eyes" among the process and control everything that goes through. Not to forget, this traceability offers a guarantee on food security, which the key point.

- **Intelligence massive fabrication:** In today's world, Big Data plays a very important role when it comes to the automatization of production. When it comes to a production line, a lot of information is created among the line, for example in the distribution phase or in the same manufacturing. Nevertheless, this amount of data is not done at the same time and it is fragmentated, making it difficult and complex to achieve a complete analysis. These security problems can be vanished thanks to these devices. As normally is, in a fabric or production center, the devices IoT have to be updated periodically, in order to solve security and regulative violations. With a decentralized system and the devices, if programed accordingly, the firmware can be updated automatically and being based on Smart Contracts, resulting in more efficiency.

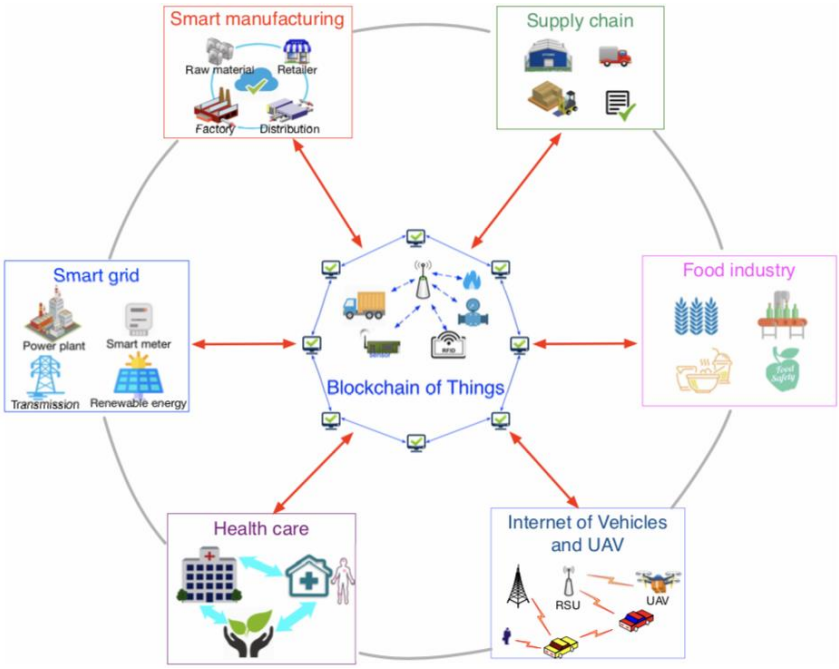


Figure 55: Blockchain of Things

9. Conclusions

In this chapter we will focus on the final conclusions about the process, taking in consideration the objectives set in the first stage of the process. Also, zooming in the experience had and the problems faced.

9.1. Final remarks and implications

From where I stand, the fist conclusion we can mention is that we did design and structure a new tracking system in order to determine the position of the pallets at all time. The system developed has been accomplished with small devices that do perfectly fit not only in terms of price but also by its precision, which it was one of the main objectives. To find this combination of separated devices in order to achieve the final modular solution, we had to invest a lot of time searching for the microcontroller that fitted us the best. It wasn't easy, as there are a lot of computer boards, microprocessors and microcontrollers that manage to do the same tasks but in different approaches.

Not to forget, the fact of putting these modular solutions into each pallet, accomplished one of our objectives, which was the location at all times. By having a geolocation sensor, the pallets have a tracker of where they have been. We did realize that by now, we don't need more exact information of the location of the package, but there would be a problem if we wanted a geolocation in terms of localization within different floors. In other words, if we want to have an exact location, we must add an extra GPS sensor, because the already sensor included in the ESP32 doesn't give us this exactitude.

Secondly, an important point is also the initial investment needed when deploying these new systems. As seen previously in the financial aspects, it fulfills the expectations when it comes to the reduction in costs, in the final process improvement. When we first developed the first improvement in the process, the cost where even higher than the actual process, reaching nearly 290.000€. This was a problem, because the improvement couldn't have more incurred costs, but it had to be nearly marginal for the client. Because of this, we started to look up for a substitute for the Raspberry Pi, which presented the main cost of the whole architecture. Finally, the choice made was the swift to the ESP32, and this led to a reduction in costs, summing a total of nearly 66.100€ for the initial investment.

Thirdly, once decided that we wanted to work with the ESP32, we wanted to know more hidden information that was important and wasn't considered. In consequence, we decided to put a temperature and humidity sensor that showed us this sets of data, that were quite important when

talking about olive oil. There were no negative consequences, as the data was taken precisely and there were not any range of errors.

In the same line, when the first improvement was made, we realize that the sensor of temperature in the Sense Hat of the Raspberry Pi, wasn't really giving out the exact temperature. This was probably because the temperature sensor was heated by the same Raspberry Pi. This was another of the reasons why, at the end, we didn't take into account this device. But it is excellent in terms of capabilities, capacity and powerfulness. We may consider this device for other deployments as it was really easy to work with and it is perfect for development.

Fourthly, we accomplished information transparency by the introduction of BigchainDB and IOTA, the two Blockchain networks. It was a great choice, as we wanted all this set of data visible by everyone. Also, it is an outstanding point that present this crucial characteristic of time-stamped, as we wanted to assure the non-falsification, non-modification and precision.

Focusing more into the technical aspects, BigchainDB and IOTA, both of them are very young networks and from where I stand, I do think that sometimes they are not quite reliable. In fact, when we were doing development with the device, sometimes IOTA wasn't functioning correctly as it stopped receiving the data. This was a point that in order to deploy correctly the device and the service, it must be supervised. IOTA and BigchainDB are no more than five years old and they have nothing to be compared to Ethereum or Bitcoin, as are the main Blockchain networks and both of them present an amazing network architect, functionality and structure. Nevertheless, these two newborns do need to expand, and this is only about time.

Adding up, the communications among LoRa and MQTT are outstanding. The protocol MQTT is not only fast in terms of sending and receiving messages, but also the architecture of its deployment and functionality makes it trustworthy. In the other hand, LoRa and LoRaWAN are the best choice in terms of remitting the information, as it solved all the problems of durability of batteries presented in the first improvement made.

BigchainDB was quite laborious to install on the server, as the docker wasn't functioning for us. Instead, we had to install it step by step on an Ubuntu 18.04. Nevertheless, at the end it was quite quickly to install it.

To sum up, I do believe these devices and the combinations of these two disruptive technologies such as Blockchain and IoT have a lot to offer to this new stage of applied science. It is outstanding how

these small devices can come up to a full tracking and shipping or in some cases, even the full monitoring of the production line.

As a general conclusion, I consider that in an overall approach this improvement of the process presents not only an economical reduction in costs and initial investment, but also in terms of simplicity of the deployment itself. As seen, the intermediate solution didn't match exactly the path we set as an objective, as costs where extremely high and the Raspberry Pi presented much powerfulness not needed. By this, the implications of adopting this modular solution would not only increase the transparency of information among all the line of supply chain, also it would diminish initial investment and the amortization of the devices, it would enrich the obtainment of other invisible information among the transportation and last but not least, it would present highly security, as sets of data are inserted in Blockchain networks which its main characteristic is immutability and non-modification.

9.2. Future research and limitations

As commented previously, these devices have many future reachability as it is a very scalable product that can swing to other industries, such as health care and internet vehicles. These devices do not require a high initial investment and are on the loop of the latest technology on the market.

Nevertheless, I do feel some markets are evolving in slow motion and are not ready to face or adopt these new technologies. For example, it was noticed that some companies didn't want to adapt new measures in order to confront counterfeit, and neither wanted to identify the products that were being resold in an illegal market. Because they knew there were being resold, but they couldn't stop it, because they couldn't firmly confirm which operator among the supply chain was acting against the company.

The fact that these special microcontrollers can monitor with the sensors all the entire information among the entire supply chain, creates a whole new perspective. Not to forget, the invisible information that it hasn't been important for years, now it opens a key door, which is mandatory if you want to stand out.

These devices can face, for example, the control of food intoxication. By having information about all the components, sites and track of the path, the company ensures a hounded percent compliance

with the sanitary regulations. For example, here in Spain, there was an identification of meat that was contaminated, and it was a scandalous situation that it's still being analyzed by the Spanish court.

In terms of limitations, I do feel that since it is one of the newest technologies, people don't really understand it fully. Blockchain has come to change the way contracts and accountancy is done, among others. It is the most secure network, which everything is completely transparent and reachable, and nobody can modify it once it is registered. Big corporations and using it in terms of controlling the information and the track of their products, and it is giving amazing results.

Also, in this system is always needed the best connectivity for LoRa, which if in a future, these devices are located in agricultural fields it may be hard to catch the best connection. Also, the duration of the batteries of the microcontrollers can be limited, but it can also be extended with an additional power bank. Those limitations are present when the deployment of the service. (Butun, Pereira, & Gidlund, 2018)

According to Sheikh Ferdoush and Xinrong L, in the publication *Wireless Sensor Network System Design using Raspberry Pi and Arduino for Environmental Monitoring Applications*, these devices present also barriers in terms of long-term implementations, as it hasn't been on deployment for many years. We are talking about technologies that have been on the market for a total of three years as a maximum, for example IOTA.

These microcontrollers have nearly a marginal cost for the company, as it gives the optimal results and completely transparency. In addition, a future research could be the incorporation or modification of these devices with the incorporation of different sensors, in order to catch the information that it is useful for the clients, for example chemical gases or contamination.

In conclusion, I do assume that from now and in not so many years, systems that are now used will completely change its methodology and will face a 360º turn. With this situation that we are living nowadays, we are being *"forced"* to work more virtual than never, and so far, it is resulting favorable. Adaptation takes time, and it just a matter of it.

10. References

10.1. Publications

A Light, R. (2017). Mosquitto: server and client implementation of the MQTT protocol. The Journal of Open Source Software, 2(13), 265. https://doi.org/10.21105/joss.00265

BigchainDB GmbH. (2018, May). BigchainDB 2.0 The Blockchain Database. Retrieved April 12, 2020, from https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf

Butun, I., Pereira, N., & Gidlund, M. (2018). Security Risk Analysis of LoRaWAN and Future Directions. Future Internet, 11(1), 3. https://doi.org/10.3390/fi11010003

Conrad, E., Misenar, S., & Feldman, J. (2015). CISSP Study Guide (3rd ed.). Amsterdam, Netherlands: Elsevier.

Dorri, A., Kanhere, S. S., & Jurdak, R. (2016). Blockchain in Internet of Things: Challenges and Solutions. Retrieved from https://arxiv.org/pdf/1608.05187.pdf

Gartner Inc. (2017). Forecast: Internet of Things - Endpoints and Associated Services, Worldwide, 2017. Retrieved from https://www.gartner.com/en/documents/3840665

Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. Journal of Cryptology, 3(2), 99–111. https://doi.org/10.1007/bf00196791

Haxhibeqiri, J., De Poorter, E., Moerman, I., & Hoebeke, J. (2018). A Survey of LoRaWAN for IoT: From Technology to Application. Sensors, 18(11), 3995. https://doi.org/10.3390/s18113995

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

Popov, S. (2018). The Tangle. Retrieved from https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf

Tapscott, D., Tapscott, A., & Salmerón, J. M. (2017). La revolución blockchain: descubre cómo esta nueva tecnología criptográfica transformará la economía global. México, Ciudad de México: Ediciones Culturales Paidós.

Zidek, K., Janacova, D., Hosovsky, A., Pitel, J., & Lazorik, P. (2018). Data optimization for communication between wireless IoT devices and Cloud platforms in production process. Proceedings of the 3rd EAI International Conference on Management of Manufacturing Systems, 3. https://doi.org/10.4108/eai.6-11-2018.2279672

10.2. Websites

Gal, A. (2018, February 7). The Tangle: an illustrated introduction. Retrieved April 14, 2020, from https://blog.iota.org/the-tangle-an-illustrated-introduction-c0a86f994445

Pon, B. (2016, February 14). What is BigchainDB? Retrieved April 10, 2020, from https://blog.bigchaindb.com/what-is-bigchaindb-38aff031bf51

Doutel, F. (2016, February 29). Sense Hat, Uno de los mejores periféricos para experimentar con tu Raspberry Pi: Análisis. Retrieved April 16, 2020, from https://www.xatakahome.com/trucos-y-bricolaje-smart/sense-hat-uno-de-los-mejores-perifericos-para-experimentar-con-tu-raspberry-pi-analisis

Espressif Systems. (2019). Modules | Espressif Systems. Retrieved April 16, 2020, from https://www.espressif.com/en/products/modules

IBM. (n.d.). IBM Knowledge Center. Retrieved April 18, 2020, from https://www.ibm.com/support/knowledgecenter/SSMKHH_10.0.0/com.ibm.etools.mft.doc/bc62020_.htm

Ignition. (2020). Wide Area SCADA - Central Database - Ignition User Manual 7.8 - Ignition Documentation. Retrieved April 2, 2020, from https://docs.inductiveautomation.com/display/DOC/Wide+Area+SCADA+-+Central+Database

Llamas, L. (2019, April 17). ¿Qué es MQTT? Su importancia como protocolo IoT. Retrieved April 18, 2020, from https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/

LoRa Alliance. (2015, November). A technical overview of LoRa® and LoRaWAN. Retrieved April 17, 2020, from https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf

LoRa Developers. (n.d.). LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL. Retrieved April 17, 2020, from https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/

MLSDev. (2019, March 7). Blockchain Architecture Basics: Components, Structure, Benefits & Creation. Retrieved April 8, 2020, from https://medium.com/@MLSDevCom/blockchain-architecture-basics-components-structure-benefits-creation-beace17c8e77

Onsman, A. (2018, August 3). Centralized Database Management System. Retrieved April 2, 2020, from https://www.tutorialspoint.com/Centralized-Database-Management-System

Oracle. (2020). What is a database? Retrieved April 2, 2020, from https://www.oracle.com/database/what-is-database.html

PHP. (2020). PHP: ¿Qué es PHP? - Manual. Retrieved April 3, 2020, from https://www.php.net/manual/es/intro-whatis.php

Raspberry Pi. (n.d.). Buy a Sense HAT – Raspberry Pi. Retrieved April 15, 2020, from https://www.raspberrypi.org/products/sense-hat/

Raspberry Pi Foundation. (n.d.). What is a Raspberry Pi? Retrieved April 15, 2020, from https://www.raspberrypi.org/help/what-+is-a-raspberry-pi/

Sopapun, M. (2018, December 15). Install MQTT Broker on Ubuntu 18.04 & 14.04. Retrieved April 23, 2020, from https://medium.com/@aegkaluk/install-mqtt-broker-on-ubuntu-18-04-15232ab0ee42

Team D. F. (2018, September 15). IoT Applications | Top 10 Uses of Internet of Things. Retrieved April 7, 2020, from https://data-flair.training/blogs/iot-applications/

The Things Network. (2020, May 6). Quick Start. Retrieved April 23, 2020, from https://www.thethingsnetwork.org/docs/devices/uno/quick-start.html

Yuan, M. (2018, December 5). Conociendo MQTT. Retrieved April 18, 2020, from https://www.ibm.com/developerworks/ssa/library/iot-mqtt-why-good-for-iot/index.html

Appendix

Annex 1: Code for LoRaWAN

```
#include <ESP32_LoRaWAN.h>

#include "Arduino.h"

#include "heltec.h"

#include "blocktacVariosLogo.h"

#define BAND      868E6   //you can set band here directly,e.g. 868E6,915E6



/*license for Heltec ESP32 LoRaWan, quary your ChipID relevant license: http://resource.heltec.cn/search */

uint32_t    license[4]  =  {0x0139A3F5,  0xA972071A,  0xD073D8BE, 0x611CB951};


/* OTAA para*/

uint8_t DevEui[] = { 0x00, 0x14, 0xFC, 0xD9, 0xBF, 0x41, 0xE4, 0xF9 };

uint8_t AppEui[] = { 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x02, 0xDC, 0xD0 };

uint8_t AppKey[] = { 0xC3, 0x04, 0x2B, 0x76, 0x21, 0x6E, 0x99, 0xB3, 0x46, 0x63, 0x7C, 0xE3, 0xC0, 0xF6, 0x6F, 0x14 };


/*LoraWan Class, Class A and Class C are supported*/

DeviceClass_t  loraWanClass = CLASS_A;


/*the application data transmission duty cycle.  value in [ms].*/
```

```
uint32_t appTxDutyCycle = 15000;


/*OTAA or ABP*/
bool overTheAirActivation = true;


/*ADR enable*/
bool loraWanAdr = true;


/* Indicates if the node is sending confirmed or unconfirmed messages
  */
bool isTxConfirmed = true;


/* Application port */
uint8_t appPort = 2;


/*!
* Number of trials to transmit the frame, if the LoRaMAC layer did
  not
* receive an acknowledgment. The MAC performs a datarate adaptation,
* according to the LoRaWAN Specification V1.0.2, chapter 18.4,
  according
* to the following table:
*
* Transmission nb | Data Rate
* ----------------|-----------
* 1 (first)       | DR
* 2               | DR
```

```
* 3                 | max(DR-1,0)
* 4                 | max(DR-1,0)
* 5                 | max(DR-2,0)
* 6                 | max(DR-2,0)
* 7                 | max(DR-3,0)
* 8                 | max(DR-3,0)
*
* Note, that if NbTrials is set to 1 or 2, the MAC will not decrease
* the datarate, in case the LoRaMAC layer did not receive an
  acknowledgment
*/
uint8_t confirmedNbTrials = 8;

/*LoraWan debug level, select in arduino IDE tools.
* None : print basic info.
* Freq : print Tx and Rx freq, DR info.
* Freq && DIO : print Tx and Rx freq, DR, DIO0 interrupt and DIO1
  interrupt info.
* Freq && DIO && PW: print Tx and Rx freq, DR, DIO0 interrupt, DIO1
  interrupt and MCU deepsleep info.
*/
uint8_t debugLevel = LoRaWAN_DEBUG_LEVEL;

/*LoraWan region, select in arduino IDE tools*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;

static void prepareTxFrame( uint8_t port )
```

```
{
    appDataSize = 4;//AppDataSize max value is 64
    appData[0] = 0x00;
    appData[1] = 0x01;
    appData[2] = 0x02;
    appData[3] = 0x03;
}

void logo(){
  Heltec.display -> clear();
  Heltec.display                                                    ->
  drawXbm(0,5,blocktacVariosLogo_width,blocktacVariosLogo_height,(
  const unsigned char *)blocktacVariosLogo_bits);
  Heltec.display -> display();
}

// Add your initialization code here
void setup()
{
  Heltec.begin(true /*DisplayEnable Enable*/, true /*LoRa Enable*/,
  true /*Serial Enable*/, true /*LoRa use PABOOST*/, BAND /*LoRa RF
  working band*/);

  logo();
  delay(300);

  Serial.begin(115200);
```

```
  while (!Serial);
  SPI.begin(SCK,MISO,MOSI,SS);
  Mcu.init(SS,RST_LoRa,DIO0,DIO1,license);
  deviceState = DEVICE_STATE_INIT;
}

// The loop function is called in an endless loop
void loop()
{
  switch( deviceState )
  {
    case DEVICE_STATE_INIT:
    {
      LoRaWAN.init(loraWanClass,loraWanRegion);
      break;
    }
    case DEVICE_STATE_JOIN:
    {
      LoRaWAN.join();
      break;
    }
    case DEVICE_STATE_SEND:
    {
      prepareTxFrame( appPort );
      LoRaWAN.send(loraWanClass);
      deviceState = DEVICE_STATE_CYCLE;
```

```
      break;
    }
    case DEVICE_STATE_CYCLE:
    {
      // Schedule next packet transmission
      txDutyCycleTime = appTxDutyCycle + randr( -
  APP_TX_DUTYCYCLE_RND, APP_TX_DUTYCYCLE_RND );
      LoRaWAN.cycle(txDutyCycleTime);
      deviceState = DEVICE_STATE_SLEEP;
      break;
    }
    case DEVICE_STATE_SLEEP:
    {
      LoRaWAN.sleep(loraWanClass,debugLevel);
      break;
    }
    default:
    {
      deviceState = DEVICE_STATE_INIT;
      break;
    }
  }
}
```

Annex 2: Source Code for BigchainDB.

This is the code in order to post to BigchainDB.

```
from bigchaindb_driver import BigchainDB
import time

### BigchainDB Connection

bdb_root_url = 'http://localhost:9984'
bdb = BigchainDB(bdb_root_url)

### BigchainDB Keys

amaia_prv = 'knUFRJmcyQkbMH2PqS1UQxsgRsePgm4CL4MQ2zwTiou'

amaia_pub = '44H4WgLm7pWXbbvGox1vs2xtNo42X87h1ymWmDuYKeCn'
### Raspberry Data

raspberry_asset = {
    'data': {
        'raspberry pi': {
            'serial_number': 'RPI01',
            'owner': 'POLITO'
        },
    },
}

raspberry_asset_metadata = {
    'External_Measures': 'Here goes sensor data'
}

### Sending Data to BigchainDB
def send_to_bdb(sensor_data):

    raspberry_asset['data']['raspberry pi']['Storage TimeStamp'] = str(time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()))
    raspberry_asset_metadata = sensor_data

    prepared_creation_tx = bdb.transactions.prepare(
        operation='CREATE',
        signers=amaia_pub,
        asset=raspberry_asset,
        metadata=raspberry_asset_metadata
    )

    fulfilled_creation_tx = bdb.transactions.fulfill(
        prepared_creation_tx,
        private_keys=amaia_prv
    )

    #print(fulfilled_creation_tx)
    print("Successfully added to BigchainDB: ",fulfilled_creation_tx['id'])

    bdb.transactions.send_commit(fulfilled_creation_tx)
```

```
'''
tx = bdb.transactions.prepare(
    operation='CREATE',
    signers=alice.public_key,
    asset={'data': {'message': 'Blockchain all the things!'}})
signed_tx = bdb.transactions.fulfill(
    tx,
    private_keys=alice.private_key)
bdb.transactions.send_commit(signed_tx)
'''
```

Annex 3: MQTT Broker Raspberry Pi

This is the code for subscribing to the MQTT broker, getting data from the Raspberry Pi:

```
import paho.mqtt.client as mqtt
from tobdb import send_to_bdb
from tomysql import store_mysql
from toiota import send_to_tangle
import json

### MQTT SETTINGS ###

mqtt_broker = "192.168.0.19"  # MQTT broker IP Adress

mqtt_port = 1883  # MQTT port

sub_topic = "sensordata"  # receive messages on this topic

pub_topic = "other"  # send messages to this topic


### WHEN CONNECTION IS DONE ###

def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
    client.subscribe(sub_topic)


### WHEN RECIEVING A MESSAGE ###

def on_message(client, userdata, msg):
    print(msg.topic + " " + str(msg.qos) + " " + str(msg.payload))

    sensor_data = json.loads(msg.payload)

    if "Temperature" in  sensor_data:
        # Send data to MariaDB client
        store_mysql(sensor_data)

        # Send data to BigchainDB Client
        send_to_bdb(sensor_data)

        # Send data to IOTA's Tangle
        send_to_tangle(sensor_data)
    else:
        print("Data from ESP8266, processing delayed")

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.username_pw_set("amaia", "carmex")
client.connect(mqtt_broker, mqtt_port, 60)
```

```
### Start the MQTT forever loop
client.loop_forever()
```

Annex 4: Source Code for IOTA.

This is the code in order to post to Tangle.

```
""
MQTT client for sensor data, computes a temporal statistic summary and publish it to IOTA's tangle

Summary's format is as follows: {"NumMeasurements":int, "Parameter1":[mean, std, min, median, max]}

It has been set up this way due to the message's content length limitation to 2187 Trytes, with proper changes would be shown as:

"Pressure":
"count": 5
"mean": 942.471
"std": 0.027
"min": 942.439
"50%": 942.475
"max": 942.51


"""
from iota import Address, ProposedTransaction, Tag, Transaction, Iota, TryteString, json
import pandas as pd
import json
import time
import iota
import threading

##### IOTA SETTINGS #####

SEED = 'VAHM9FFIQIWFWFRVVOFWNRSMBJQTLHNVLMSPNQZUTIVQQWXRJMFHIMBKBZMKJEGRZFWYUFGVYXYZ'
#api = Iota('http://173.212.218.8:14265/', SEED)                # POW node, SEED IS NEEDED TO DO THE SIGNING
api = Iota('https://nodes.devnet.iota.org:443', SEED, testnet = True)
api = Iota('https://nodes.thetangle.org:443', SEED)
security_level = 2

tag = iota.Tag('AMAIAEXTERNALENVIRONMENT')          # Tag for in the message.
#address = api.get_new_addresses(0, 1)['addresses']            # Generates a list of 1 address
address = api.get_new_addresses(index=0, count=1, security_level = security_level)['addresses'][0]     # Generates a list of 1 address

##### PANDAS DATA FRAME SET #####

data_frame = pd.DataFrame()

######

activate_summary = True

def send_to_tangle(sensor_data):
    global data_frame

    data_decoded = sensor_data
    df1 = pd.DataFrame([data_decoded], columns=data_decoded.keys())
    data_frame = pd.concat([data_frame, df1])
    #print(data_frame)
```

```
    global activate_summary
    if activate_summary:
        thread.start()
        activate_summary = False
        print("iota - starting summarization")

    return message


def wait_time():
    while True:
        time.sleep(10)
        send_summary()


def send_summary():
    global data_frame
    new_data = {}

    ## IOTA'S IMPLEMENTATION
    data_summary = data_frame.describe([]).round(4)
    print(data_summary)

    dict_summary = data_summary.to_dict()
```

```
    for key, value in dict_summary.items():
        new_data["Num"] = int(value.pop("count"))
        new_data[key] = list(value.values())

    data_frame = pd.DataFrame()

    json_summary = json.dumps(new_data)
    #print(json_summary)
    #print(len(json_summary))

    trytes = TryteString.from_string(json_summary)  # Code Json data in trytes

    #print('Message sent to tangle:', trytes.decode())  # shows decoded msg sent to tangle

    #print('sleep...')
    time.sleep(1)

    print(api)

    #t0 = time.time()

    result = api.send_transfer(
        depth=3,
        transfers=[ProposedTransaction(address=Address(Address(address[0], )),
                                       value=0,
                                       tag=tag,
```

```
    #t1 = time.time()
    #print('Time:', time.strftime('%H:%M:%S', time.localtime()), ' data transfered! in:', round((t1 - t0), 1), 'seconds ')
    print("Stored in Iota ",result['bundle'].tail_transaction.hash)


### THREAD FOR COMPUTING AND SENDING DATA SUMMARY ###

thread = threading.Thread(target=wait_time)
```

Annex 5: MQTT Broker LoRa

This is the code for subscribing to the MQTT broker, getting data from the microcontroller LoRa:

```
import paho.mqtt.client as mqtt
from tobdb import send_to_bdb
from tomysql import store_mysql
from toiota import send_to_tangle
import json

### MQTT SETTINGS ###

mqtt_broker = "eu.thethings.network"  # MQTT broker IP Adress

mqtt_port = 1883  # MQTT port

sub_topic = "heltec_lora_esp32_1/devices/heltec_lora_esp32_1/up"  # receive messages on this topic

pub_topic = "other"  # send messages to this topic


### WHEN CONNECTION IS DONE ###

def on_connect(client, userdata, flags, rc):
    print("Connected to The Things Network with result code " + str(rc))
    client.subscribe(sub_topic)


### WHEN RECIEVING A MESSAGE ###

def on_message(client, userdata, msg):
    print(msg.topic + " " + str(msg.qos) + " " + str(msg.payload))

    sensor_data = json.loads(msg.payload)

    if "payload_raw" in sensor_data:
        # Send data to MariaDB client
        store_mysql(sensor_data)

        # Send data to BigchainDB Client
        send_to_bdb(sensor_data)

        # Send data to IOTA's Tangle
        send_to_tangle(sensor_data)
    else:
        print("Data from LoRa_ESP32, processing delayed")

client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message
client.username_pw_set("heltec_lora_esp32_1", "ttn-account-v2.rkJZZhvurYvzdvQ27BBcH5yFQkxf9PZf8yO_T0b1VwI")
client.connect(mqtt_broker, mqtt_port, 60)
```

```
### Start the MQTT forever loop
client.loop_forever()
```

Annex 6: Source Code for Raspberry Pi

This is the code to publish into a MQTT broker, from a Raspberry Pi:

```
##### Libraries #####
from datetime import datetime
from sense_hat import SenseHat
from time import sleep
from threading import Thread
import paho.mqtt.client as mqtt
import paho.mqtt.publish as publish
import time
import json

sense = SenseHat()

##### Logging Settings #####
FILENAME = ""
WRITE_FREQUENCY = 100
TEMP_H=True
TEMP_P=False
HUMIDITY=True
PRESSURE=True
ORIENTATION=True
ACCELERATION=True
MAG=True
GYRO=True
DELAY=2

##### MQTT Settings #####
Broker = "192.168.0.13"

sub_topic = "test/instructions"    # receive messages on this topic

pub_topic = "sensordata"       # send messages to this topic

##### FUNCTIONS #####

def get_sense_data():
    sense_data={}

    if TEMP_H:
        sense_data["Temperature"] = sense.get_temperature_from_humidity()

    if TEMP_P:
        sense_data["Temperature"] = sense.get_temperature_from_pressure()

    if HUMIDITY:
        sense_data["Humidity"] = sense.get_humidity()

    if PRESSURE:
        sense_data["Pressure"] = sense.get_pressure()
```

```
    if ORIENTATION:
        o = sense.get_orientation()
        sense_data["Yaw"] = o["yaw"]
        sense_data["Pitch"] = o["pitch"]
        sense_data["Roll"] = o["roll"]

    if MAG:
        mag = sense.get_compass_raw()
        sense_data["Magnetic_Field_x"] = mag["x"]
        sense_data["Magnetic_Field_y"] = mag["y"]
        sense_data["Magnetic_Field_z"] = mag["z"]

    if ACCELERATION:
        acc = sense.get_accelerometer_raw()
        sense_data["Acceleration_x"] = acc["x"]
        sense_data["Acceleration_y"] = acc["y"]
        sense_data["Acceleration_z"] = acc["z"]

    if GYRO:
        gyro = sense.get_gyroscope_raw()
        sense_data["Gyroscope_x"] = gyro["x"]
        sense_data["Gyroscope_y"] = gyro["y"]
        sense_data["Gyroscope_z"] = gyro["z"]

    sense_data["Data_Collection_Time_Stamp"] = time.strftime('%Y-%m-%d %H:%M:%S', time.localtime())

    return sense_data


##### MQTT Section #####

# when connecting to mqtt do this

def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))


client = mqtt.Client()
client.on_connect = on_connect
client.username_pw_set("amaia", "carmex")
client.connect(Broker, 1883, 60)
client.loop_start()

while True:
    client.publish(pub_topic, json.dumps(get_sense_data()))
    time.sleep(DELAY)
```