

BACHELOR PAPER

Term paper submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Smart Homes and Assistive Technologies

Analysis of Mycroft and Rhasspy Open Source voice assistants

By: Carlos Lumbreras Sádaba

Supervisor 1: Ing. Martin Deinhofer, M.Sc.

Vienna, 2020-22-06



Declaration of Authenticity

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz/ Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

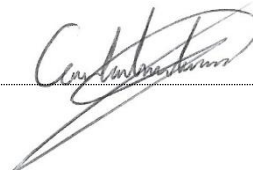
I hereby declare that I completed the present work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.”

Sesma, 2020-06-22

Place, Date

Signature

A handwritten signature in black ink, appearing to read 'C. Sesma', written over a horizontal dotted line.

Kurzfassung

Der technologische Fortschritt hat die Sprachsteuerung von Maschinen bzw. intelligenten Geräten für den Durchschnittskonsumenten zugänglich gemacht.

Da immer mehr Sprachassistenten auf dem Markt angeboten werden, ist es notwendig, diese zu analysieren, um herauszufinden, welche Vor- und Nachteile sie jeweils haben, und um festzustellen, welcher für den jeweiligen Benutzer am Besten geeignet ist.

Die beliebtesten Sprachassistenten folgen den gleichen Prinzipien beim Zugriff auf Benutzerdaten, um einerseits die Produktentwicklung zu erleichtern und zu verbessern und andererseits einen zusätzlichen Nutzen aus dem Umgang mit diesen Daten zu ziehen. Das Aufkommen von Open Source Alternativen mit unterschiedlichen Ansätzen hat gezeigt, dass es möglich ist, qualitativ hochwertige Alternativen mit einem anderen Schwerpunkt zu haben.

Bei den Optionen Mycroft und Rhasspy handelt es sich um zwei Open-Source-Alternativen, die eine Reihe von Vorteilen in Bezug auf Datenschutz und Zugänglichkeit bieten, die die bisherigen Alternativen nicht bieten. Während sich Mycroft auf ein System konzentriert, das in der Cloud arbeitet, aber dem Benutzer die Kontrolle über seine Daten erlaubt, schlägt Rhasspy die Verwendung eines Systems vor, das vollständig lokal arbeitet.

Die Verwendung dieser Assistenten zusammen mit einem openHAB-Server als Schnittstelle zu den Smart Home-Geräten hat die Entwicklung eines sehr leistungsfähigen Systems ermöglicht, das in der Lage ist, all diese Geräte per Sprache zu steuern.

Es hat sich gezeigt, dass diese Systeme eine sehr einfache Konfiguration ermöglichen und gleichzeitig eine sehr optimale Leistung in Bezug auf die Erkennung von Befehlen und deren anschließende Ausführung erzielen.

Schlagwörter: Sprachassistent, Heimautomatisierung, Rhasspy, Mycroft, openHAB, Open Source

Abstract

Technological advances in recent years have made machines and intelligent devices made accessible by voice control to the average user.

More and more voice assistants are being offered in the market. It is necessary to analyse them to see what advantages and disadvantages each one has in order to identify which one is more convenient for each specific user.

The most popular voice assistants follow the same approach of accessing the users' data to facilitate and improve the product development, and on the other hand, get an additional benefit from the handling of this data. The emergence of alternatives with different approaches has shown that it is possible to have quality alternatives with a different focus.

Analysing the different Open Source options in the market, the Mycroft and Rhasspy systems have been selected for further analysis due to the characteristics they present. They are two alternatives that offer a series of advantages related to privacy and accessibility that are not offered by the big players in the market. While Mycroft focuses on a system that works in the cloud but allows the user to have control over their data, Rhasspy proposes the use of a system that works entirely locally.

The use of these assistants with an openHAB server has allowed the development of a very interesting system. The openHAB server acts as an intermediate platform between voice assistants and intelligent devices. This makes possible to create a system that allows to control all these devices of the home by voice.

It has been seen that these systems allow a very simple configuration. At the same time is obtained a very optimum performance in relation to the detection of commands and their subsequent execution.

Keywords: voice assistant, home automation, Rhasspy, Mycroft, openHAB, Open Source

Acknowledgements

First of all I would like to thank my supervisor Ing. Martin Deinhofer, for his support and orientation to make this bachelor's thesis possible in a situation as delicate as the one caused by covid-19.

I would also like to thank Dr. L. Serrano for the support he has given me in solving all the different problems during the semester and making this experience possible.

For my colleagues at UPNA and FH Technikum Wien I would just like to say thank you for the time we have spent together. I especially want to name my friend Edu for all his support and understanding over these years.

As it could not be otherwise I wanted to thank my family. To my mother, Valen, my father, Jesús Miguel, and my brother, Óscar, thank you for showing me all that hard hard work, effort and humility means.

Finally, I did not want to forget my friends from Sesma, my town. Life goes beyond university and therefore you are indispensable. Thanks.

Table of Contents

1	Introduction	5
2	Related Work and State of the Art	6
2.1	Voice Assistants	6
2.2	Smart Homes	17
2.3	Open Source Alternatives.....	20
3	Method	25
3.1	Interconnection platform	25
3.2	Smart Home Appliances.....	27
3.3	Installation and configuration of openHAB	28
3.4	Mycroft	32
3.5	Rhasspy	39
3.6	Testing and calculation of statistics	45
4	Results	48
5	Discussion.....	53
6	Conclusion	54
7	Future Work	55
	Bibliography.....	56
	Appendix A: Rhasspy sentence configuration	64
	Appendix B: List of extra commands.....	67
	Appendix C: Raw evaluation data	68

1 Introduction

Home automation is one of the topics that have generated most interest to me in the technological field. It is an idea based on the user being able to create a technological environment in the home to facilitate the control of different devices.

A few years ago I started trying to automate different elements of my house, such as the watering control, using configurations based on the use of an Arduino¹. I didn't like it just because I had a more modern house, what especially impressed me was the fact that I could do the whole development process myself so that the whole system would work.

In the last few years, when different voice assistants started appearing in the market, I thought that using these tools could be the perfect complement for controlling these home automation devices. Voice control offers many advantages such as hands-free operation, which is interesting for both normal people and people with disabilities in particular.

However, after trying to use devices from the big companies (Amazon Alexa, Google Home) I was very disappointed by the lack of possibilities to learn how these tools work. This in turn leads to privacy concerns.

After discovering the existence of different voice assistant options in the market, I was particularly interested in learning more about them. It is clear that each system has some advantages and some disadvantages, and that is where the special interest comes in, to see if any of the different alternatives available offer more positive things than negative ones.

It is clear that based on what I had been most interested in the previous years, which were systems where you can know how they work and try to develop things for them, I am especially interested in the Open Source alternatives on the market.

¹ <https://www.arduino.cc/>

2 Related Work and State of the Art

2.1 Voice Assistants

Voice assistants are a set of tools that form a system capable of interpreting the human voice, processing a question or a command, and answering with a synthetic voice. They are based on the idea of managing to establish human-machine interaction, which has been possible in different ways throughout the history of technology, simply by using voice.

Ever since computers were invented, human beings have dreamed of being able to talk to machines. A few decades ago, this was simply something futuristic in science fiction, as in Knight Rider with the interaction with the KITT car² through voice. That is why voice interaction between man and machine has always been considered a field of special attention for large technology companies. This idea has been developed over many years with different approaches and ideas to make it possible for such a complex system to work.

Impact

The key to the development of these technologies has been the reduction in price that has brought the technology closer to the average user.

An Outerbox study states that there are currently more than 10 billion mobile devices in the world [1]. This clear example reflects the large and rapid impact that a new technology can have on society.

In turn, the fact that the impact is so great has both positive and negative aspects. On the one hand, it implies that until a technology is sufficiently developed to be accepted by the average user there is no point in putting it on the market. However, on the other hand, it makes it easier for these companies to get more data which allows them to develop their technology.

A clear example of this increased interest can be seen in the increase in articles related to voice interaction in recent years. In 'This Week in NLP' where the most important technological articles of the last years are collected weekly, it can be seen that voice assistants are recurrent topics among these articles [2].

Most systems on the market have opted for the same basic operating principle. They are based on the idea that the system needs to detect a wake word, a keyword that will make the system wake up, after which the user can say the command he wants the machine to do. Subsequently, the machine extracts the real intent of the command and maps it to a skill, which can be defined as a capability (e.g. switching lights on).

² <https://www.hagerty.com/media/car-profiles/how-kitt-became-a-trans-am/>

Architecture of the systems

The fundamental architecture that all voice assistants follow, at software level, is the same. Already in 2005 *Pieraccini* wrote an article that exposed these bases, naming the whole as a system of spoken dialogue [3]. He divides the process into different phases, a first phase that is based simply on capturing the raw audio as such from the users and converting it into a sequence of words. In the second phase the sequence is forwarded to a natural language processor, deducing grammar and meaning of the command. Finally, a dialog manager responds to the user and triggers the execution command.

In 2018 *Merdivan* proposed a system with a very different idea, instead of using words as the basis for the understanding phase, he proposed the use a two-dimensional image [4]. It showed that this change made the system more efficient but it has not really been implemented in commercial systems until now. The main advantage of this approach is the fact that no pre-processing or previous training is needed.

Evolution

At first, tools based on machine interaction through voice started to be introduced only in some computers, the next key step was their implementation in smartphones. Today, voice assistants are available on different hardware platforms. One advantage of these systems is that they can be introduced into a wide variety of systems, from Smart Homes to computers.

The advantage that they can be used in such diverse systems means that they can be used by many people.

Each of the companies focuses on development with different methods and strategies, although all starts from a similar initial structure. Some companies focus their development more on the accuracy of the system, others on the quality of the audio, and still others focus on the highest possible compression capacity. After all, they all have the same objective but they try to reach it by different ways. This objective is to make a device that makes the spoken conversation between a man and a machine as similar as possible to a conversation between people. Also is important to make an effective system in relation to the users' requests [5].

All voice assistants follow two basic processes. On the one hand, there is the Speech To Text (STT), which is the process in which the voice message that comes out of the user's mouth is converted into text that the machine is able to interpret. On the other hand there is the Text To Speech (TTS) which is the opposite step to convert the text that the software system is able to generate into a synthetic voice signal that the user is able to hear and understand.

Involved technologies

All the systems available today are based on Artificial Intelligence (AI) as the basic tool on which all their operation is based. AI is something quite complex, some cases requires training

through Machine Learning (ML). It is not a clear and strict operation but it acts according to how a certain system has been trained and which data was used.

Most of the options on the market, especially those of large companies, are based on the idea of cloud computing. In other words, they are based on the idea of sending the user's data directly to the cloud and that is where the corresponding AI tools developed in each system come in. The idea of making an implementation more local is another feasible option.

Like most technological tools that present different alternatives for use, the case of placing intelligence locally or in the cloud has its respective advantages and disadvantages [5].

Advantages of Local systems:	Advantages of Cloud systems:
No Internet needed	High computational capacity
Privacy	Tons of training data from all users
No latency	Always updated

Table 1: Advantages of local systems vs advantages of cloud systems

Cloud computing is something that in recent years has allowed to bring very complex technologies such as Big Data, or in this case, AI and ML, to the end user. This is the clear advantage of this option [6]. The main disadvantage of this is that the user loses control over what happens to his data, the information becomes non-local and passes through a cloud service that cannot be controlled.

General problems

One of the main problems with all voice assistants is simply that language is not universal. There are a lot of languages with their respective dialects that make the variability of different voices incalculable. The idea of these systems is that they work for the greatest number of possibilities. This means that a great deal of training is needed for these systems to be able to function in as many cases as possible.

At this point it is possible to see the difference between cloud systems and local systems. Cloud systems should include as many possibilities as possible. Local systems, that have more limited training possibilities, should only be trained to support the language(s) that the user will be able to use in each case.

In all these systems English has been used as the fundamental language, because it is the most used language in the technological field worldwide. Depending on the capabilities (economic and data accessibility) of the companies for development, the systems are able to support more or less additional languages. Improving the capacity of these systems to manage this variability is one of the key areas where most progress has been made in recent years.

Options in the market

In relation to the different options that there are in the market it is necessary to emphasize that mainly there are 4 assistants specially known in the market that are the assistants of Google (Google Assistant (GA)), the one of Amazon (Amazon Alexa), the one of Apple (Siri) and the one of Microsoft (Cortana). There are also other options such as Samsung's, Xiaomi's, or different solutions from smaller brands that have focused their efforts on developing quality alternatives, with different advantages and disadvantages, to those of the four big companies.

The first appearance of these assistants on commercial devices was Siri in 2011 in devices with iOS. Shortly thereafter, in 2013, to match the competition, Cortana appeared on Microsoft devices. But the greatest impact of these technologies began in 2014 when a fairly inexpensive alternative appeared on the market, with a product based primarily on the voice assistant, as was the Smart Speaker Amazon Echo. These devices included the well-known assistant now Alexa. Following Amazon's idea, in 2016, Google launched a Smart Speaker called Google Home that included the company's own assistant. They took advantage of the great impact that the brand has on mobile devices to introduce its voice assistant in all Android smartphones [7].

The ease with which Google found access to users, due to the fact that the majority of mobile devices on the market uses its operating system, meant that it soon became a leading company in the sector. The other company with a greater impact among users is currently Amazon. This is mainly because it has greatly facilitated the work of a large number of developers so that they can develop specific skills for their assistant. This has led to more than 100,000 skills being currently dedicated to this system [8], p.129.

Another relevant fact in relation to these systems was commented by *Anavi* at the *Embedded Linux Conference Europe*, reflects that given the very positive impact that these initial proposals from large companies have had, many more companies would try to find their place in this market [9]. According to a study by *Canalys* the systems with Amazon (74.7%) and Google (24.6%) dominated the whole market in 2017. One year later, although they were still the two most used systems, between them they only reached 61.7% of the entire market. This year new possibilities began to appear, such as those created by large Chinese technology companies like Baidu or Xiaomi, or other less important assistants with more different objectives [10].

The specific issue of Smart Speakers as such is somewhat peculiar, although they are the devices that have really caused voice assistants to be considered as something important and useful, the data indicates that they are not really used as much by people. A study by *Pew Research* on the use of voice assistants in the United States indicates that although more than 45% of people in the country use voice assistants, which is a really high value, only 8% use smart speakers. One of the main justifications given for this is the issue of lack of need and, above all, the issue of privacy.

Effectiveness of assistants

The effectiveness can be measured by the recognition rate, which is the ratio of correctly recognized messages and total number of messages.

The evolution of the data indicates that in 2017, the assistant with the best performance was that of Google, with 74.8% of correct answers and 99% of queries understood. However in 2019 these data became 92.9% and 100% respectively, which are data indicating the effectiveness of these systems [11] [12].

To obtain this data, a methodology based on asking the same 800 questions to each assistant was used. Different types of questions were asked (Local, Commerce, Navigation, Information, Command). The tests were performed using Siri on iOS 11.4, GA on a Pixel XL, and Alexa and Cortana using the iOS application.

The other systems that follow it in the data are Siri and Alexa, which have gone from having data in relation to 65% of correct answers and 95% (Siri) of understood queries to having values close to 80% and 100% (Alexa) respectively. The development in the basic processes of TTS and STT are those that reflect these overall improvements in the operation of the system.

A detail that is also observed in these annual analyses is that while the Microsoft voice assistant (Cortana) appeared in the analyses of the first years, doesn't appear in the current ones. The opposite has happened with the Amazon system. So it reflects the changes that have occurred in the market in this field in relation to the main companies [12].

Another notable advance in recent years has been related to the fact that current systems do not require the user to mention wake word continuously. This approach results in greater interactivity between the user and the machine.

The fast development in recent years also creates additional problems. In this case, all the development that these systems have undergone has made the operation of the current systems much more complex than that of years ago. The consequences of this have been that the development of the different skills, both the adaptation of the existing ones to the new functionalities and parameters and the development of the new ones, has been made difficult.

Amazon provided the Alexa Conversations option that allows developers to connect the different independent skills with the Alexa proprietary dialogs.

Other uses of these tools

The fact that these systems have acquired such a level of quality has made companies from other sectors use systems of this type in very different areas such as call centers. There are studies, as reflected by *Chen and Metz* in the article "*Google's Duplex Uses A.I. to Mimic Humans (Sometimes)*" in *The New York Times* [13], that show that currently only 15% of cases require human intervention in customer service calls. This allows companies to save a great

deal of staff while maintaining an equally efficient system. However, the idea of using voice assistants in these situations instead of humans can even pose an ethical dilemma for the companies themselves [8], p.129.

It is even noteworthy that by law, in some places, such as California, companies were forced to let users know that they were in contact with a machine and not a real person [14]. These examples demonstrate the great capacity of current systems to have a conversation as close as possible to a real person.

These devices have even been considered for use in very specific and sensitive environments, such as a laboratory. *Cambre, Liu, Taylos and Kulkarni* in 2019 designed a system that they called *Vitro* to improve work in a laboratory through the use of a voice assistant [15].

Data and privacy

In addition, the signs for the future of these systems are very encouraging. The growth in recent years has been very high both in terms of investment, and in terms of implementation, purchase and use by users. A highlight may be that it is even estimated that by 2022 there will be 5 billion voice assistants collecting from smartphones to dedicated devices such as Smart Speakers [16].

The amount of data that can be generated with voice assistants is very high. Nowadays, Big Data is one of the leading technologies in the technological field. It is based on the idea of obtaining a great amount of data in order to analyse it and extract useful information.

Data collection is one of the main reasons why it is so interesting for large companies to introduce these devices into users' homes. Users, however, see their privacy compromised by the fact that they lose control of their data when they leave the local area.

It is clear that the use of these assistants in our lives can bring us many advantages such as facilitating many tasks. However, many users question whether it really compensates for losing control of our data in relation to the advantages that the system can offer us. This is one of the most recurrent reasons by users to justify the non-use of devices of this type.

This is perhaps the most sensitive issue that companies have faced in this period of trying to introduce these novel technologies into the market. This is also why it is one of the most studied topics in relation to the assistants. There are very diverse opinions in relation to this topic, especially in relation to whether or not it compensates to assume this risk in comparison to the advantages that they offer. Nevertheless, most of them warn of the lack of transparency that occurs with data in large companies' systems. This is a rather sensitive and worrying issue for users.

Furthermore, companies make economic profit from the data collected [7]. Anyhow, the fact of seeing the benefit that companies can really get in return is something that really makes the average user think about whether it compensates or not. The idea of third parties also appears in relation to this aspect. None of the four large companies related to voice assistants has

denied the use of external companies to carry out the analysis of the data collected. Even some companies such as Amazon have assured that they really use third parties in their company [17].

Apart from the fact that privacy is a very important fact for users, the feeling of security or insecurity generated by the use of these assistants in people is even more important. In this case, the problem of data control is compounded by the idea that microphones are considered one of the most intrusive sensors in homes (the one that is the most after video cameras) [7].

Proposals for improvement of current systems

Some possible short-term feasible improvements for its introduction in these tools are, for example, the introduction of an incognito mode, such as in browsers for more compromised topics. Another interesting idea would be to offer the user real and precise information about what happens to their data when they leave the local area, as well as being able to choose what happens with them [7].

One of the tools that have been included in the latest versions of the most famous Smart Speakers has been the introduction of a mute button that allows to stop allowing the device to be listening. It is an interesting idea, but in many cases, given the lack of real information about what really happens with data, there are many people who don't accept the idea that the button works exactly as it's supposed. It is very common for users to unplug devices directly, and not use the Mute button, when they are going to have very private or compromised conversations [18].

Additional information collected in voice message

It should also be noted that in spoken conversation there is much more information than just the text with which each command can be identified. Additional information can be obtained in a voice message based on the tone of voice or other data collected by a person's voice. Very specific situations can be identified for each person such as their emotional state, confidence, stress level, physical condition, age, gender and other personal traits [19]. A lot of extra information is also collected due to the large amount of metadata attached to each message.

All this extra information obtained from a voice message in these cases is an issue to be taken into account in relation to privacy. If companies have access to this data they have access to much more information about users than just voice commands, which already provide enough information. Many of the providers actually use this additional information that these situations offer.

The importance given to data is also key, because many times data alone is not particularly useful. When you manage to join different data in the right way you can obtain very accurate and precise information.

It's even been proposed the introduction of a prior filter between users and cloud services to try to eliminate the entire emotional component of a voice message. The final result of the

system's response to the user should not be influenced by the emotional component in any way. So they are based on the idea that if you lose control over the data that ends up arriving at large companies, that, at least, this information is as little as possible without affecting the operation of the system.

From the point of view of the companies, they have perceived that the additional information in the voice messages can have a great interest. More and more efforts are being put into trying to improve technologies that identify emotional state.

Google has even, in relation to its STT, patented a technology capable of detecting emotions and the mental conditions of users, as indicated in its official documentation³. It is even capable of detecting diseases such as Parkinson's, the user's stress conditions, whether they are smokers or not, and several other parameters of a person's general health conditions.

The privacy paradox

The issue of privacy from the user's point of view is very difficult to analyse. Most people are very aware of the issue of privacy and it is always considered important, especially before buying and using a certain product. When the user buys a certain device and introduces it into their daily lives, most users largely forget to try to protect personal data [20], p. 121, [21], p. 1038. This is something that has always happened with websites for example where people always talk about how much companies know about us. In contrast, the most common is to accept all the cookies raised by the provider without even reading them.

In the case of voice assistants, it is not common to use any of the tools, even if they are very few, that providers offer for the control of privacy issues. The most common tool is the visualization of the logs (information about part of the information eventually sent to the cloud).

In relation to this, *Ammari* in 2019 conducted a study in which he concluded that although about 70% of people he analysed knew that voice assistants have access to log history, only 10% actually studied them. In the same way he concluded that most users have no knowledge of the specific terms of privacy [18].

It can also be considered a completely opposite point of view based on the idea that people accept without any problem that this happens. For example, there are people who accept it because they consider that the information that a voice assistant can collect cannot have any negative impact on them; although surely in most people with this opinion there is also a part of the issue of lack of knowledge about the subject.

The idea that users are compensated if companies can access their data to obtain more useful and personalized information is another opinion that is reflected [22], p. 26.

The fact that the user is provided with more personalized advertising is really something interesting for everyone, both for the companies that advertise themselves and the users. The

³ <https://cloud.google.com/speech-to-text/>

dilemma is whether it is really in the users' best interest to lose control of so much information about them with the only advantage of getting more personalized publicity.

The other factor that can benefit the user from companies collecting all this data is that they will have more data to train and improve their systems. The more data that is available, the faster and easier it is for companies to develop these technologies. In the same way here comes the big question of whether it really compensates the user to lose control of so much information about him in exchange for the company's system improving faster and having more personalized advertising information?

Also, the issue of personalized advertising is not so clear that it can be a clear advantage for the end user. Large companies may try to incite the user to make certain expenditures based on both the interests that the user may have as well as his or her weaknesses. In addition, in the case of the four large companies, they tend to encourage access to their own services by using their own assistants. As in the case of Amazon, for example, to purchase on their website, or in the case of Google, to use their own services, such as Play Music or Play Books.

Functionalities

The inclusion of the use of a voice assistant in the daily life of people, both in smartphones, tablets or computers, as well as in dedicated devices, can offer a great amount of very useful functionalities for users of different types. A very clear example of this can be the help that such a system can provide while making a recipe or something similar that allows from setting timers, to viewing the recipe or playing music, all this with the hands free.

In more specific cases you can go a step further, for example in the case of people with dementia, the use of a voice assistant can be very useful because it allows you to repeat the same thing over and over again without any problem. Another idea of this type is the clear use that people with vision problems can give to devices of this style since it allows them to be able to access documents, news or books that could not do anything in other circumstances without the help of another person. It is clear that in the case of books there is the possibility of audio books, but with the TTS tool available to voice assistants, all the written documents that are needed could be converted to synthesized audio [23], p. 81.

Other possibility that these systems can incorporate in the short term, using the tools they already have, could be real-time interactive translation. This could be a radical change in relation to translation.

These systems can be leveraged for very specific cases by preparing and training them to work in the best possible way in different areas. In relation to health issues, for example, the system can be prepared to help a particular patient follow a specific therapy with reminders, or for people with certain disabilities, so that they can do more things that allow them to have more autonomy and security, such as being able to control things in the house with their voice. These topics are very interesting and have been specially studied and researched in recent years in the face of the many possibilities that voice assistants can offer in these cases.

A paradoxical issue is also that, although for many people these systems can provide them with greater autonomy, there are people who consider that the opposite effect occurs. This is especially true for older people who believe that these systems can cause them to lose some of their autonomy. This is because they lose some control over what happens due to lack of knowledge about how they work [7].

Specific uses (Health)

A concrete real practical use that has already been considered using these technologies has been given in the UK where the National Health Service (NHS) is collaborating with Amazon. The voice assistant is used as a tool to provide people with professional medical information quickly and easily [24]. This idea in turn has a clear advantage for the health system as such as it would reduce the workload of the centres in relation to problems with common diseases.

This idea raised in the United Kingdom is based on the idea of promoting telemedicine. It is something new in the industry that is increasingly having a greater impact on society with the idea of improving the current system. The biggest problem for the development of these technologies is that being such a sensitive issue it needs authorizations from government institutions.

In the US, the use of a vital constant monitoring device has been authorized, complementing both technologies is being studied as the perfect union to improve the current system. This combination allows real and adequate control of the situation of people in their respective homes. This could allow the use of fewer medical resources in hospitals with the advantages that this can bring.

For the correct functioning of this type of system, it would be key to facilitate and take maximum care of the patient-caregiver relationship. As much as for the control of the data that the patient contributes about his situation, as, on the contrary, the information that the doctors or carers want to give to the patients about the medication or any other reason [25].

The key to the development of this is AI in all aspects related to it from the basic functioning of attendees to advanced topics in relation to medical prescriptions [26].

One approach that has been named in some studies to protect the most sensitive data in relation to voice assistants is homomorphic encryption of data trying to make access to the most sensitive and private data of users as limited as possible [25]. To this end, the approach of converting voice to text would be followed in order to facilitate its handling and storage [27].

If voice assistants were actually introduced as a complement to medicine, the information these systems would work with would become even more sensitive, making the idea of strengthening security and guaranteeing data privacy even more important. Here there are two distinct parts, on the one hand the need for data encryption in view of the confidentiality of patients' medical data, and the other part, related to the use of external systems in relation to intermediaries between the medical centre and the patient/user.

If all the information really has to go through the cloud of the companies providing these services, it would be essential to know what happens to this data from the moment they leave the local area.

Hacking and spoofing problems

In relation to general privacy issues, there are also problems at other levels, such as the ease of hacking that these systems are always considered to have. These technologies are not yet fully established and protected in relation to the large amount of data that they are capable of obtaining from an individual or a household.

The danger that this can pose is very high [7]. Problems related to spoofing may arise as there is no added protection if you manage to get within the audible range of the speaker. An example is a child who ordered thousands of cookies using an Alexa device [28]. Google is trying to develop a voice printing system to solve these problems [23], p. 81.

Indirect problems due to lack of privacy

A problem related to privacy, in which technical or technological opinion does not interfere at all, is the one that takes into account how people's behaviour is modified when they are in a room where they perceive that there is a Smart Speaker or any other voice assistant in its different forms. There are people who have no problem with this, however, there are people who knowing that they are in a place where there is a microphone can generate an additional stress situation [18].

Normally these people increase their insecurities when they do not know when the device is actually turned on and when it is not, the dilemmas about the "always on". This also occurs due to lack of knowledge about what data is reaching both the companies that provide these services in the cloud and the third-parties. This idea is also clearly reflected in the use of assistants depending on the context.

In relation to all of the above, the key to voice assistants appears, analyse if compensates or not for using the typical voice assistants in which we no longer have control of our data according to the different advantages they present. Here comes into play the emergence of alternative voice assistants that are gradually entering the market in recent years. This assistants have more difficulty in making a name for themselves in the market due to the greater difficulty they have in being known by the average user, since they are not derived from large technology companies. However, there are really interesting alternatives, with their respective advantages and disadvantages. The typical systems mentioned above, which follow the same idea of cloud computing and do not allow users to have any control over what happens to their data, are working very well and bringing many advantages at all levels to these companies.

2.2 Smart Homes

Smart Homes are homes equipped with different smart devices in order to make life easier, simpler, safer and more automated for the people who live in them. The Internet of Things (IoT) devices can be implemented to end up sensing and controlling various things in the home. All of this can be approached for different purposes, from the basic idea of home automation to more specific issues such as healthcare or security [29], p. 321.

Like the issue of controlling machines using voice, the idea of Smart Homes is something that has always been named as a futuristic idea, in books and movies, from the end of the last century until it actually became a reality [30], p. 321.

Smart Home Appliances

The fact that IoT devices as such are causing greater interest among people is why many companies have opted to bet on the manufacture of these devices: blinds, lights, locks, thermostats and many other products. This in turn has led to a great deal of competition in the market which has caused companies to put great effort into improving their products. This fact is making the devices as such increasingly better (smarter) in the process of how they interact with each other and with humans.

You can even talk about own brands like Philips, Sonos or Next4, which are brands with a lot of history in technology issues that have bet on the development of these devices.

A detail to comment also is that most of the users of IoT devices assure that they want to continue increasing the amount of devices connected in their homes. Many of them, assure that they do not invest more in these elements because they do not own the home in which they reside [18] [31], p. 46. The devices related to lights are usually quite cheap and quite easy to change from one home to another, however other devices such as blinds are more expensive and above all more difficult to put on and take off.

The IoT sensors that collect information from households and users residing in the home can be divided into 3 types basically [29], p.321.

- Cameras: Not normally used because they are considered too intrusive devices.
- Wearables: Individualized control but are not effective to be connected continuously (e.g. in health-related issues such as monitoring of vital signs).
- Binary and continuous sensors: Most used (not considered so intrusive devices) Are usually cheaper and provide enough information to perform multiple added functionalities in a Smart Home.

Integration of the different devices

Today's IoT platforms in homes are often very heterogeneous with devices from different brands, which makes it difficult to interact with them [32], p. 232, [33], p. 670, [34], p. 27.

One effective solution that has been given to this problem is based on using hubs or gateways as an intermediate point in which to connect to all devices [34], p. 27. Some common options on the market are those of openHAB⁴ and Home Assistant⁵. These solutions offer many facilities to users for the control of devices and in turn allow to create more integrated systems that generate a greater sense of Smart Home, not just having different smart devices scattered around the home.

User interaction, voice assistants

Typically the interaction with these IoT devices has been using the computer, tablet or smartphone. The introduction of voice assistants in this field goes a step further and provides a much more interesting and effective interaction with the user. The combination of these devices with the voice assistants ends up being very interesting for all the devices involved. On the one hand, it makes it much easier to control the devices, and on the other hand, it generates really interesting utilities with which to make better use of all the technology that these intelligent voice control systems integrate.

A Microsoft study reinforces this idea of the effectiveness and interest that the joint use of voice assistants with IoT devices for home control has caused [35].

Already in 1990 appeared a real idea of how to try to make an intelligent room controlled by voice. MIT created a project to be able to carry out this idea, you could have a first idea of how you could design and use a concept like this [36]. Nowadays it wouldn't make much sense to take as a base the technologies and principles used in this project because actually more prepared tools have appeared. This project was something that was characterized by creating an initial idea but that was not continued developing throughout the following years [18].

With the application of these devices a great deal of information can be obtained, which can be used in many cases in a positive way. One example is to take advantage of the occupation information of the different rooms of the house for energy saving issues. It is key to achieve an optimal and effective use of all this information to be able to relate the data in an appropriate way to end up having useful information.

Privacy

Again, as with the independent use of voice assistants, there is a very important aspect to consider in these cases, such as privacy. Against more devices are placed at home connected to the voice assistant this allows us to have many more features but also this makes it collect much more data about users, so it takes even more importance the issue of privacy.

The lack of transparency about what happens with this data, which in turn is something intended by some companies as much of its benefits actually derive from the handling of this

⁴ <https://www.openhab.org/>

⁵ <https://www.home-assistant.io/>

data, is somewhat worrying. Appears the idea to see if it is possible to consider other alternatives that are on the market more focused on ensuring the privacy of users.

The combination of a microphone and IoT devices can provide a very comprehensive check on a person's life. It is possible to know exactly what time all the residents of a house are getting up, when they are at home and when they are not, even the time a person actually spends doing certain household things such as watching TV or cooking.

An alternative use of this data within the local area serves for the control between the own tenants of a house. This idea on the one hand can be a positive approach in the case of being able to control the small children. Nevertheless this can become a very complicated subject in relation to if something similar happens between adults. The approach is similar to having your data accessed by an anonymous person but the opinion of the people is different when a known person is accessing this data without their knowledge. This can lead to secondary problems such as mentally affecting the different people living in the home and even cause them to stop using such devices.

Primary and secondary users

Another issue to be considered in the use of voice assistants in the home is the differentiation between primary (active) and secondary (passive) users. The opinion that the two different types of users may have about the introduction of these devices in the home may even be confrontational.

For primary users, the use of these systems provides some benefit because they really have a use that interests them, however, secondary users do not see anything positive in the intrusion of these devices in their homes. These systems not provide them with any direct benefit and, at the same time, depending on the case, it can lead to problems related to privacy.

In relation to this differentiation between users, the problems associated with the lack of protection of the accounts configured on these devices also appear. In these systems there is no additional tool available to control which user actually accesses these accounts. With a view to the future of these technologies it is essential to improve in these aspects, both in trying to differentiate users and in trying to generate some benefit to secondary users so that they are not so opposed to the use of these voice assistants [7].

Common use of these technologies

A very important aspect to take into account is to analyse how people actually use these technologies and devices. At first there are people who justify not buying devices like Smart Speakers with the idea that the functionalities they offer are not too many. However, in the studies that analyse the real behaviour of people using these devices in their homes, such as those reflected by *Lau*, *Ammari*, and *Huxohl* in their respective articles of 2018, 2019 and 2019, it is possible to see how the functions most used by users are the most basic ones, such as

controlling lights, music or setting reminders. The most advanced interconnectivity and other functionalities are used much less [7] [18] [6].

It is also worth mentioning that when someone buy a device of these characteristics it is very common to use at first the most advanced functions because of the novelty that it represents. Typically, when its use is really introduced in the daily life of people, its use is focused on the most basic functions.

It is also remarkable that analysing the reasons for the customers' purchase of these products, many of them justify the initial purchase by mentioning the idea of the control of the IoT devices, in order to achieve the maximum possible automation of the home. The union of these devices generates a sensation of perfect complementation generating a concept of Smart Home that is identified with modernity.

2.3 Open Source Alternatives

From the analysis of the market situation, around the different possibilities of being able to use a voice assistant, it has been possible to observe which are the main advantages and disadvantages of the use of the voice assistants of the big companies.

In view of this, the idea of considering the different minority alternatives on the market has been put forward to see if they can offer different advantages that compensate for their use as opposed to the better known options. These different proposals are based on trying to improve certain key aspects of the operation of these systems in order to try to make a more suitable system for the end user.

Most of these voice assistants are focused on improving privacy issues for the end user, because this is one of the main complaints that are usually made about the most used systems in the market, such as Google or Amazon options. The alternatives that have been focused on trying to improve this are based mainly on two different approaches.

On the one hand the approach of making the system work completely in the local environment, and another alternative, that follows the idea of working in the cloud but offering many more possibilities to the users about what they want to happen with their data. The key to this second type of options is to offer the user clear, concise, and easy to handle information about their data as opposed to the typical options that are characterized by their lack of transparency in relation to these issues.

Generally, the main problem with all these different alternatives is the scarcity of data available to train their systems. Because they have a much smaller user base, their capabilities to access the huge amount of data that large companies have access to be more limited. A clear example of this could be the use of TED talks, open databases from Mozilla or voluntary recordings, which is the process Mycroft follows to train its systems as indicated by its CEO Joshua Montgomery [37].

For example the difference can be seen clearly between an assistant like Google's compared to any alternative that does not have a platform on which to install its assistant by default. GA was installed on all Android mobile devices from the beginning.

Deriving from the previous idea of the difficulty in obtaining a large amount of data, one of the main problems with all these alternative options is that their use is limited to a very small number of languages. Most of these systems start from an initial version that only works in English and then, gradually, expand their range of possibilities. The main voice assistants on the market for their part have the advantage that they work in many languages, which in turn, allows them to reach people who would not be able to use a language other than that spoken in their day to day to use this type of device.

The fact that these systems have the ability to work in many different languages requires a very high investment in terms of both money and time. This investment is unbearable in most systems that do not come from large commercial companies.

For example, in the market there have been appearing lately alternatives of Samsung or big Chinese brands like Baidu or Xiaomi, but all these options that end up coming to the market follow an idea practically similar to those of the four big initial brands already mentioned. They are taking the same basic idea, the lack of transparency in the handling of data to get the biggest amount of data possible since for them this part is indispensable in their business. These alternatives do not present any special advantage so they deserve to be analysed in this case [9].

In this case, this analysis focuses on seeing the behaviour presented by some Open Source alternatives available in the market as alternatives to the typical systems already mentioned. The fact that they are open systems presents great advantages such as being able to know the complete functioning of what is happening in these systems. For people with knowledge on the subject, these options even offer the possibility to participate and contribute to the development of these tools. Therefore, in many cases, the problem of lack of investment ends up being alleviated thanks to the collaboration of people in these systems who do not have as their main objective the economic issue but rather to go a little further.

In this case two main types of proposals can be seen, on the one hand initiatives more focused at the research level that their main idea is based on analysing the behaviour that can generate certain systems with some very specific characteristics. And on the other hand, those that focus on developing systems that have certain advantages over traditional systems but that can end up serving as many people as possible.

In the case of ideas based on very specific topics, for example, the proposal of *Popović* and *Pakoci*, of November 2015 is based on the idea of creating a system that works in Serbian and also using an acoustic model to try to improve the response of this type of system to noise. Some very specific characteristics are proposed to see the impact that they can have but they

are not systems that are made with the idea of extending at first more than the scope of a certain investigation [38].

Another interesting proposal may also be the system called Aretha, created by *Seymour in 2019*, whose main purpose was to make a prototype for a voice assistant in the case of a Smart Home to improve the knowledge and skills of the user in relation to privacy and security issues [39]. This system tries to offer the users information about what happens with their data so that they realize the risks that can be involved in using one of the typical cloud-based systems where control over the data is completely lost. In this specific case it is based on using the technology of IoT Refine which is based on controlling the information that goes through the network [40]. But in the same way as the previous proposal, it is focused on research rather than continued use.

Analysing the different possible options you can see how more alternatives have been appearing in the market of Open Source voice assistants. After making a first review of the different options it can be seen that the two most known alternatives were Snips and Mycroft.

The Snips⁶ system, was perhaps the most advanced system on the market due to its good functioning, its large community and its great number of functionalities. However, the problem arises in that this last year it has been acquired by the Sonos trademark and has closed all its open repositories so that it can no longer enter the different Open Source alternatives.

Taking into account the Snips system, it is worth mentioning Alice's project⁷, which in its beginning was thought as a complement to the Snips system to create a set with as many functionalities as possible. After the purchase of Snips this project had to reinvent and change all this part of its system for other alternatives. An example could be the change of the Snips ASR for the pocketsphinx⁸ or Google ASR alternatives. In most cases these assistants allows the user to choose which process components they want to choose, similar to what happens in Mycroft as well. However, Alice's system has three main problems, the first of which is that it uses a GPL license, which limits its usage possibilities as mentioned below, the second main problem is that currently it only works on Raspberry Pi with Raspbian, and most importantly, that this system tries to do many things by itself from communicating with the "gateway" of the smart house outside the basic functionalities of doing all things and voice skills.

The other main option, Mycroft, is based on a cloud system but offers users full control over what happens to their data. It is a system that tries to work as much as possible locally but subsequently uses cloud services to try to offer a system with greater possibilities. Being an Open Source system you can verify what happens with your data simply by analysing the behaviour indicated in the code itself.

⁶ <https://snips.ai/>

⁷ <https://github.com/project-alice-assistant/ProjectAlice>

⁸ <https://pypi.org/project/pocketsphinx/>

As a remarkable detail it is possible to say that Mycroft even manufactures Smart Speakers, which integrate its voice assistant called Mark I and Mark 2. The main purpose on which the project is based is not to create products for sale to the end user but to create a quality system to cope with the great alternatives on the market by putting as strengths the privacy. The fact that it is Open Source results in being a neutral system without background interests [9].

Following with other options comes into play other less known options as you can see the Almond⁹ voice assistant. However is not yet designed to work properly on a Raspberry which greatly limits its possibilities when considering the use of these systems on independent devices. It is a system specially designed to work using its web interface or the Linux system using Gnome. This system can be recommended for its simplicity for beginners but does not offer as many possibilities as other options, it can be considered as a very new option still that perhaps needs a lot of development.

It is similar to other alternatives such as Linto¹⁰, although this one is specially designed for businesses, or Athena¹¹, in the field of manufacturing work, which are options that have been created for more specific purposes but have in common that they are very new projects that still need a development process.

Other similar projects that are still in their early stages of development so they are not yet highly optimized nor do they have a large community behind them are Kalliope¹², Stephanie¹³ and Jarvis¹⁴.

Another alternative that has appeared is Susi.ai¹⁵, which offers a great deal of possibilities in relation to skills and ensures that they are committed to privacy. Nevertheless, they send the data for voice processing to Google so it does not have any obvious privacy advantage.

Some of the main platforms known for using local operation as a base are Picovoice¹⁶ and Rhasspy¹⁷. In this case the option of Rhasspy is analysed since it is an Open Source alternative while Picovoice is a private alternative which makes it difficult to access information about its operation. The Rhasspy alternative is taken into account mainly to be able to compare the advantages and disadvantages of a system that works only locally, such as this one, with a system such as Mycroft which, although specifically created to improve privacy conditions, bases its operation on a cloud service.

⁹ <https://almond.stanford.edu/>

¹⁰ <https://linto.ai/>

¹¹ <http://athenaworkshere.com/>

¹² <https://kalliope-project.github.io/>

¹³ <https://slapbot.github.io/>

¹⁴ <https://www.hackster.io/blitzkrieg/j-a-r-v-i-s-a-virtual-home-assistant-d61255>

¹⁵ <https://dev.susi.ai/>

¹⁶ <https://picovoice.ai/>

¹⁷ <https://rhasspy.readthedocs.io/en/latest/>

A very positive point about using the Rhasspy system is that it is a solution that combines all sorts of solutions for all the parts of a voice assistant like the ASR, the TTS, or the NLU (Natural Language Understanding), and does not care about the home automation solution the user is using. Unlike alternatives such as Mycroft or the Alice project, it do not have a skills shop.

Another point to take into account in these cases is that some assistants are based on GPL licenses while others are based on Apache or MIT licenses. In this case under the approach of using these tools at home does not affect too much, but if you take into account the use of these tools to modify the code in order to create some variation of it, as may be to create a commercial product, that is something to consider. For example it can be said that the case of Mycroft follows the Apache 2.0 license and Rhasspy the MIT license while options like Project Alice or Kalliope are GPL so it is a fact to take into account in an analysis like this. GPL licenses limit the commercial possibilities much more than MIT or Apache licenses, but on the other hand, it also guarantees that it and all it's derivatives remain Open Source (copy left principle).

In the same way, most of these Open Source systems have a basic configuration based on the use of certain tools. However, most of them allow the system to be configured manually, allowing the user to choose the tools he considers best for his own situation. There are different options for the wake languages, for the TTS and STT processes, and even for the handling of attempts. The Figure 1 shows a wide range of the different options available.

PRIVATE BY DESIGN: Free and Private Voice Assistants						
Options for a DIY Open Source Voice Stack						
	Software	Comments	Open source software license	On device or requires connectivity?	Auspecting organization	Support available?
WAKE WORD SPOTTERS	PocketSphinx	Works on-device, low memory footprint. Available on several platforms. Support via SourceForge forum. Available only for English, long-term feasibility of project questionable.	Uses its own specific license, very similar to the BSD license	On device	Carnegie Mellon University	Community-supported forum (sourceforge.net/p/cmuspinx/discussion)
	Precise	Can be used to train custom wake words. Free for personal and commercial use. Only available for Linux. Has multiple dependencies and complex setup process.	Apache 2 license	On device	Mycroft AI (a for-profit company)	Company-backed forum and online community (community.mycroft.ai/search?q=precise and chat.mycroft.ai/community/channels/wake-word)
	Snowboy	Easy to train; has wrappers for several languages. Very small CPU footprint. Free for personal use; commercial use requires a commercial license.	Apache license	On device once trained, requires internet access to Kit.AI platform to train	Kit.AI (a for-profit company owned by Baidu)	Community-supported forum for those without a commercial license (groups.google.com/a/kit.ai/forum/#forum/snowboy-discussion)
	Porcupine	Very small CPU and memory footprint, wrappers for several languages. Several development language examples given. Only free for personal use, commercial use requires a commercial license.	Apache 2 license	On device. Customized training requires internet access to Picovoice.AI platform	Picovoice.AI (a for-profit company)	No support forum, but GitHub issues are available to browse (github.com/Picovoice/porcupine)
SPEECH TO TEXT	Kaldi	Mature software product, available for multiple languages. Able to train on own language data. Very difficult to install and configure; steep learning curve.	Apache 2 license	On device	Johns Hopkins University, with support from many other research bodies	Community-supported forum (sourceforge.net/p/kaldi/discussion)
	DeepSpeech	Backed strongly by Mozilla, will run on-device on an RPi 4 using pre-trained models or after training. Bleeding edge only runs on Linux, has significant dependencies for installation. Word Error Rate (WER) of 7.5%	Mozilla Public License 2.0	On device, but only for capable hardware (Rpi 4 minimum)	Mozilla (a for-profit company)	Community-backed forum (discourse.mozilla.org/c/deep-speech)
	CMU Sphinx	Mature software product, models available in several languages. Difficult to install; long-term feasibility of project questionable.	BSD	On device	Carnegie Mellon University	Community-backed forum (sourceforge.net/p/cmuspinx/discussion)
INTENT PARSING	Adapt	Free for both personal and commercial use. Uses slot-matching approach.	Apache 2 license	On device	Mycroft AI (a for-profit company)	Company-backed forum and online community (community.mycroft.ai/search?q=adapt and chat.mycroft.ai/community/channels/adapt)
	Padatious	Free for both personal and commercial use. Uses neural network approach.	Apache 2 license	On device	Mycroft AI (a for-profit company)	Company-backed forum and online community (community.mycroft.ai/search?q=padatious and chat.mycroft.ai/community/channels/padatious)
	NLTK	Robust linguistic software for analysis, provides advanced parts of speech tagging. Available only in Python.	Apache 2 license	On device	Not auspiced by an organization	No support forum, but GitHub issues are available to browse (github.com/nltk/nltk/issues)
	Rasa	Provides intent parsing based on conversational context. Well documented, easy to get started with. Customization of context requires a commercial license.	Apache 2 license	On device after initial training; custom training requires internet connectivity	Rasa (a for-profit company)	Company-backed forum (forum.rasa.com)
	Rhino	Well documented, easy to get started with. Customization of context requires a commercial license.	Apache 2 license	On device after initial training; custom training requires internet connectivity	Picovoice (a for-profit company)	No support forum, but GitHub issues are available to browse (github.com/Picovoice/rhino/issues)
TEXT TO SPEECH	eSpeak	Robust software, support for over 100 languages. Works on device. Quality of voice varies.	GPL	On device	Not auspiced by an organization	Community-backed forum (sourceforge.net/p/espeak/discussion)
	Festival	Works on device, but is dated.	X11-type license, unrestricted commercial and non-commercial use	On device	University of Edinburgh	Forum and mailing lists currently offline
	Mimic2	Has a range of tools to train your own voice. Too computationally intensive to work on device.	Apache 2 license	No, but can be hosted by user	Mycroft AI (a for-profit company)	Company-backed forum and online community (community.mycroft.ai/search?q=mimic2 and chat.mycroft.ai/community/channels/mimic)
	Mimic	Works on device, but has limited range of available voices.	Apache 2 license	On device	Mycroft AI (a for-profit company)	Company-backed forum and online community (community.mycroft.ai/search?q=mimic and chat.mycroft.ai/community/channels/mimic)
	Mary TTS	Range of languages supported. Runs on Java, complex to install.	LGPL	On device	DFKI (a German for-profit company)	Mailing list (www.dfki.de/mailman/listinfo/mary-users)

Figure 1: Summary table of different Open Source tools that make up the voice assistants (Source: [41])

3 Method

The goal of this project is to make a detailed analysis of the Mycroft and Rhasspy system in order to analyse the behaviour of two of the most interesting voice assistants in the Open Source landscape. In this case, two systems with a very different approach are analysed. Mycroft works with the cloud, however, Rhasspy is a system that works entirely locally on the device.

At the same time, an analysis of the ease of setting up these assistants is added. The design and introduction of a customised skill in both systems is also considered.

In this case, the idea was to create a standalone device with the ability to function as a voice assistant, using a Raspberry Pi. The selfmade voice assistant should be configured to control lights, blinds and the stereo of a Smart Home. In course of the project the appliances of the Smart Living Lab of the UAS Technikum Wien¹⁸ were used.

3.1 Interconnection platform

All IoT systems nowadays are very heterogeneous because there is a great variety of devices from many different brands. It is very important to incorporate an intermediate point where all these devices can be connected for easy handling. There are several options on the market regarding these hubs that act as an interconnection point between all the connected devices.

It is possible to say that there are different Open Source alternatives to act as this intermediate platform such as HomeGenie¹⁹, MyController.org²⁰, Indigo²¹ or MyNodes.NET²². In this case only the three most prominent alternatives were taken into account. The options of openHAB²³, Home Assistant²⁴ and Domoticz²⁵ are considered the three most important platforms because of the large community behind them.

Comparing the possibilities offered by these three systems it has been seen that Domoticz is the system, of these three, that has the least amount of possibilities to connect to different devices. The other two alternatives have a very similar set of advantages and disadvantages. It is key that both systems have the ability to connect to most devices available on the market. Another strong point of both systems is that they offer a friendly interface for the end user.

¹⁸ <https://youtu.be/xBFL0PRD6rE>

¹⁹ <http://www.homegenie.it/>

²⁰ <https://mycontroller.org/#/home>

²¹ <https://www.indigodomo.com/>

²² <https://github.com/derwish-pro/MyNodes.NET>

²³ <https://www.openhab.org/>

²⁴ <https://www.home-assistant.io/>

²⁵ <https://www.domoticz.com/>

The Home Assistant platform is more attractive at a visual level but it loses a little bit of stability with respect to openHAB. In this project, the option of openHAB was definitely selected. Therefore the main reason that justifies this choice is the great flexibility that this option offers thanks to the fact that it has the largest community in terms of this type of platforms. This may be due to that it is the system of this type that has been developed for most years.

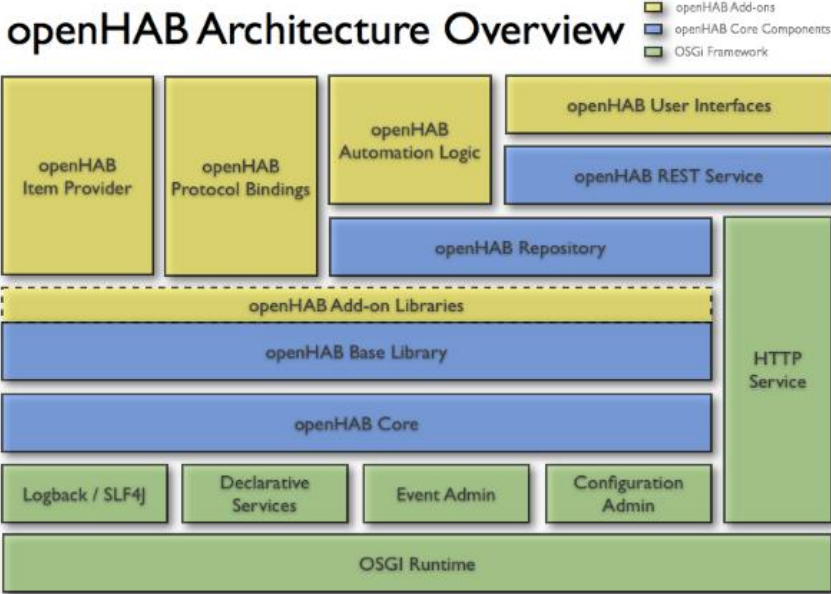


Figure 2: OpenHAB Architecture Overview (Source: [42])

The openHAB architecture is developed in Java, more specifically in Eclipse's Smart Home²⁶ environment, this allows it to run on a multitude of devices. It also generates an environment based on Apache Karaf²⁷ and Eclipse Equinox²⁸ that allows establishing an OSGI (Open Service Gateway Initiative) runtime and enables the construction of modular application components (bindings). In addition, another clear advantage is that this robust architecture that has been created is capable of running on various platforms. OpenHAB can be installed on all typical computer operating systems such as Linux, Windows or macOS up to a Raspberry Pi or a PINE64.

The architecture of the system itself reflects three clearly differentiated parts (see Figure 2). First, there is the part related to OSGI. Second, there is the kernel of the system itself, which includes the main operation of the system. Finally, there are a number of add-ons needed to accommodate each device.

Therefore, this analysis will use the openHAB platform as an intermediate connection point for all the available IoT devices as well as for the voice assistant analysed in each case.

²⁶ https://www.eclipse.org/projects/archives.php?utm_source=recordnotfound.com
²⁷ <https://karaf.apache.org/>
²⁸ <https://www.eclipse.org/equinox/>

3.2 Smart Home Appliances

Different types of IoT devices have been considered for this system. Devices such as different types of lights or a complex sound system have been installed, with all the extra functionalities that this requires. Lights are the most common element for which people use these systems so they are of special interest.

Philips Hue is the name of a set of lights and lamps that are designed as IoT devices for Smart Homes that can be controlled wirelessly. It was considered interesting to add these devices in the system.

These lights allow different additional options and not only turning the lights on or off, but also being able to control the percentage of light in even changing the color of all the bulbs.

There are different formats for these lamps, e.g. lights that need to be connected to a bridge via ZigBee. The bridge acts as a WiFi access point and makes the lamps controllable by Smartphone or other types of remote control. There are other type of Hue lamps that can be directly controlled by Bluetooth.

Philips bulbs can be controlled directly with the official application but can also be connected to an intermediate platform, such as an openHAB server. The use of a platform like this allows the creation of a graphical user interface for the control of all devices in the home. It also allows the option of connecting all devices to an external voice assistant.



Figure 3: Philips Hue devices (Source: [43])

These lights are, like the ones shown in Figure 3, one of the best examples to get into the Smart Homes environment because of their simplicity, their price and their great possibilities. This is the main reason for trying to introduce these devices into the system, to see if they can be implemented properly.

A series of KNX lights available in the Smart Living Lab of the *FH Technikum Wien* was also introduced into the system, distributed in the different spaces such as the kitchen or the living room. These lights offer the option of on/off and dimming.

As it has been previously indicated, the lights are the most used type of device for the control of the homes, so it is of special interest that the system is able to allow the control over all these devices.

Similarly, something similar happens with the blinds. All the blinds distributed in the different rooms of the laboratory can be controlled as intelligent devices, so there has also been an attempt to introduce them into the system.

The laboratory is also equipped with some intelligent switches. These switches allow to control the on/off switching of the electrical current that reaches certain domestic appliances. The control of these devices is similar to the control of the lights. So there has been an attempt to include these devices in the system as well.

It was considered essential also the control of a music device because it is another of the aspects that more people demand in this type of Smart Homes systems controlled by voice. In this case it was tried to introduce in the system the control of the different functionalities of a sound equipment Yamaha RX-V685 Music Device²⁹. These sound systems allow the control of many different functions. It allows control from the basic functions of Play, Pause or Stop, to more advanced functions such as mute the system or change between different modes (bluetooth, cd, radio or usb).

3.3 Installation and configuration of openHAB

First of all, on both systems it is necessary to install the openHAB server part in order to be able to connect all available devices. To do this, the steps in the official documentation of openHAB are followed [29], p. 321. In this case the correct functioning of the system has been checked both by using a separate device to launch the server, as well as by launching the server on a personal computer.

Following the established steps, after launching the server, it ends up being launched on port 8080 of the corresponding machine. To access the console where it is possible to see what is happening on the server (requests, responses and internal processes) an Open Source tool such as PuTTY³⁰ is used (KiTTY³¹ would be another alternative). The necessary parameters to access this console are the following:

- IP: localhost or external IP
- Port: 8101 (for the console)
- User: openhab
- Password: habopen

²⁹ http://yamaha-es.com/en/products/audio_visual/av_receivers_amps/rx-v685/index.html

³⁰ <https://www.putty.org/>

³¹ <http://www.9bis.net/kitty/#!/index.md>

```
Hacker - PuTTY
login as: openhab
Keyboard-interactive authentication prompts from server:
| Password authentication
| Password:
End of keyboard-interactive prompts from server

openhAB
2.5.3-SNAPSHOT
Build #52

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown openHAB.

openhAB>
```

Figure 4: OpenHAB console using PuTTY

In Figure 4 it is possible to see how the console of the openHAB server looks like using PuTTY. The operation can be checked without any problem from an external device. In order to do this, it is simply necessary to change the IP address of the localhost to the external IP that the machine has in the network.

The next point that appears is the configuration of the different IoT devices with the openHAB platform. The website on port 8080 is used to carry out this configuration.

Paper UI is used to make the configuration, HABPANEL or Basic UI are intended as user interface for the end user wanting to control the appliances. Basic UI offers an option based on Material Design Lite from Google, whereas HABPANEL propose a panel control specially designed for touch screens.

Within the Paper UI environment, four different concepts must be clearly differentiated [44]:

- Things: are the devices that connect to the server as such
- Items: are the different functionalities that can be assigned to each thing
- Channels: are capabilities of things, e.g a temperature sensor
- Bindings: are plugins or add-ons adapting access to the devices and their functionality (software adapters)

The use of different extensions is necessary because there are many devices that can be connected to the openHAB platform. For this reason it is necessary to install an add-on in each case to make this communication possible with all the different types of devices possible.

Following the idea of what each of the concepts refers to, it is clear the process to follow to configure the functionality of a certain device within the server. In this case, the Philips Hue configuration is shown as an example, but the process is equivalent for all the different devices that you want to control together from the openHAB server. To carry out this step of the configuration it considers that the devices that are connected to the openHAB server are what can later be managed using the voice assistant.

First it is necessary to install the necessary plug-in for each specific type of device to work. In the section of add-ons there is the option to add Bindings, which are the add-ons dedicated to make this communication possible. In the specific case of the Philips Hue, the corresponding add-on has been found and installed (see Figure 5).

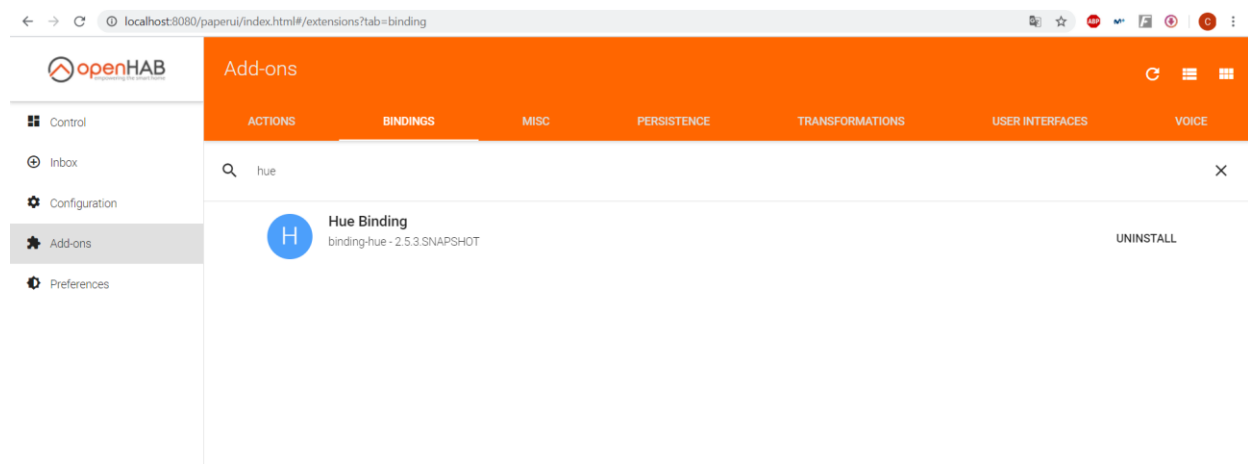


Figure 5: OpenHAB website for system configuration (Add-ons)

Secondly, the next phase of the process is based on adding the different devices available in the network to the system taking advantage of the plug-in installed in the previous part. It is simply a matter of searching in the Inbox section for the different elements available and adding them to the platform.

After adding these devices, in the configuration section, the different devices that have been added must appear in the "Things" section. In this case, as seen in Figure 6, the bridge necessary for the different lights to work, as well as the lights themselves, appears. It is also possible to see another example of a connected device, such as the IRTrans³² device based on infrared control of different devices.

³² <https://www.irtrans.de/en/>

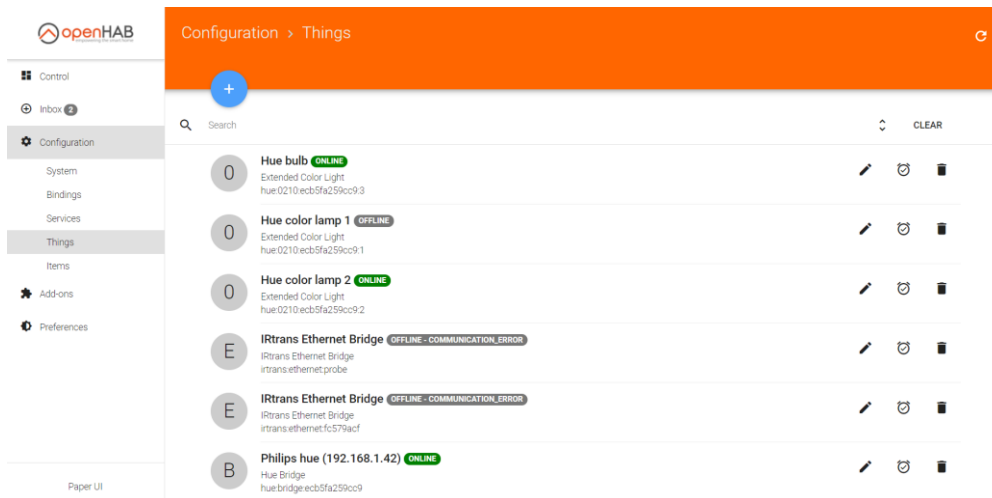


Figure 6: OpenHAB website for system configuration (Things)

This section also shows information about which devices are online at any given time and which are not.

Finally, the last point in this part is to configure the different devices separately. At this point, one can change, for example, the reference name with which each device will be addressed, or the different functionalities that one wishes to control in each device (Items). To make this individual configuration, it is necessary to select each device and create the desired items within them.

It is possible to order all these features in different ways, being able to create profiles and add different labels to each one of them to use them as a reference to be able to organize them in the best possible way. Inside each Thing, the different functionalities assigned to each one appear. In the example of the Philips Hue (see Figure 7), there are three different functionalities: on the one hand, the possibility to turn the device on and off, on the other hand, the option to regulate the intensity and finally the colour change.

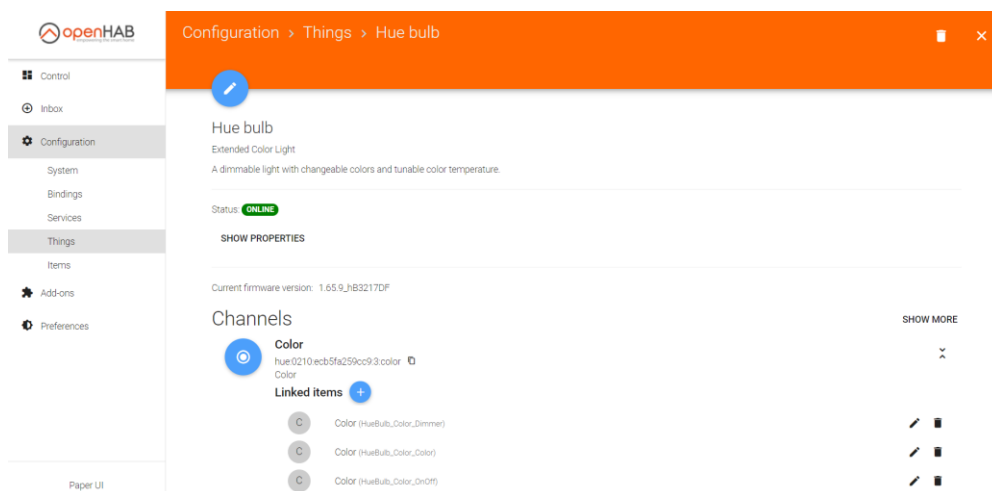


Figure 7: OpenHAB website for system configuration (One thing, the Hue bulb)

Similarly, in the Items section, all the functionalities created appear independently of the device to which they are linked.

After following this process with all the different devices, the configuration of the openHAB platform is completed. Based on this configuration, the openHAB server can be used as an intermediate point between the IoT devices and the voice assistant. From this point onwards, the process is focused on the configuration of the voice assistants and on getting both points together.

As an additional comment, there is a predefined skill for the use of Mycroft's voice assistant with the openHAB server. It has not been used in this case because it limits the system's possibilities. A clear example is that it does not allow to change the color of the lights and limits the variability of possible commands for the user to perform a certain command.

3.4 Mycroft

Mycroft's voice assistant is currently the best known Open Source alternative on the market. This is why many large companies have chosen to collaborate with them. Jaguar, to create a personalized voice control system for their vehicles [45], or Mozilla that has opted to collaborate in the development of this technology offering assistance in the project [46].

The general idea of this analysis will focus on configuring the Mycroft system as optimally as possible. Similarly, the aim is to create a certain skill to control the different IoT devices available using an openHAB server as an intermediate platform. The voice assistant uses the REST interface of the openHAB server to remotely control the intelligent home devices [44]. In this way, the control of all the desired devices can be easily included so that any possible parameter can be controlled.

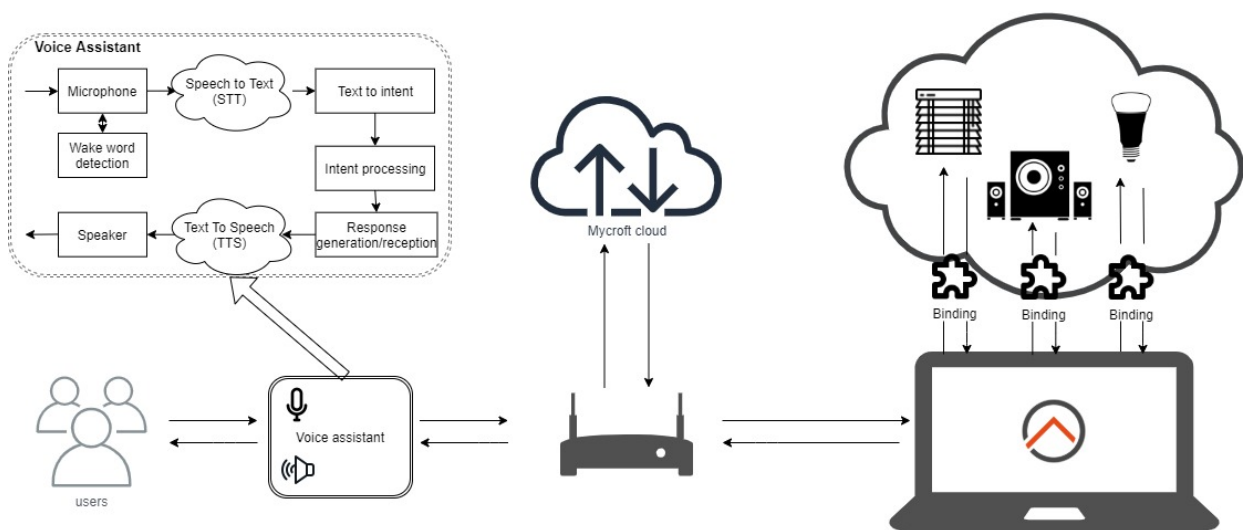


Figure 8: Schematic of a Smart Appliances control system with an openHAB server and Mycroft voice assistant

Figure 8 shows the general scheme that converts the set of elements into a voice device control system using the Mycroft's assistant. First, it is possible to see how users interact with the voice assistant where different processes are included. In this case, as Mycroft's system works partially in the cloud, the voice assistant itself sends data to the cloud in order to carry out the different processes. Finally, the voice assistant sends commands to the openHAB server, which acts as an intermediate point to send the different commands to the different connected devices.

In this case, communication with the cloud involves 4 fundamental steps:

- Device identification (account authentication)
- Configuration of different parameters of the device and the skills from the website
- STT: Sends up to 10 second audio file - returns transcription
- TTS (using Mimic2): Sends text, returns audio file

In relation to the voice assistant part different options are allowed for its implementation. In the official Mycroft documentation appears five different ways to integrate this part of the system. This tests carried out in this case were based on the installation of Picroft's system³³ on a Raspberry Pi 3+. This idea was chosen because the Raspberry Pi was intended to be used exclusively for the voice assistant.

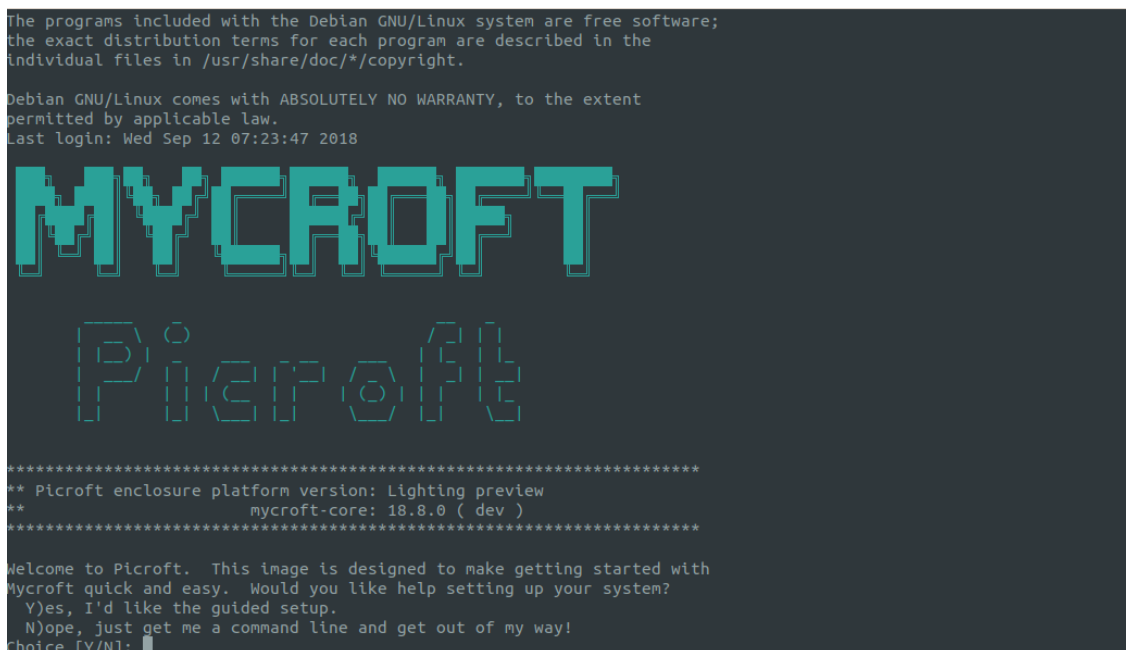


Figure 9: Picroft's interface when starting the system on Raspberry

Figure 9 shows what appears on the screen when the Raspberry Pi is started with Picroft's system installed. This operating system is pre-designed for easy use of Mycroft's voice assistant on a Raspberry Pi 3, 3B+ or 4. This option allows for the use of a disc image on a Micro SD card to burn this system.

³³ <https://mycroft-ai.gitbook.io/docs/using-mycroft-ai/get-mycroft/picroft>

The other alternatives offer the possibility of installing this system on typical operating systems such as Windows, Linux, Mac OSx or Android for mobiles. There is also the option of installing it in a Docker container.

In order to use the voice assistant independently on the Raspberry it is necessary to connect a speaker and a microphone to make possible the interaction between the user and the machine. Consequently, a PlayStation Eye³⁴ has been used as a microphone and the speakers of a monitor to carry out the tests. Using a monitor in this case facilitates the system setup process. In reality this type of system is designed to be used with a microphone and speakers integrated into a single device and without any screen. The fact that this study is based on an analysis of the system does not need to meet these requirements.

The first step for the configuration of the voice assistant installed on the Raspberry Pi must be to connect to the internet. For this type of system it is essential that all devices are connected to the same network, in this case, they are all connected to the same WiFi network (2.4GHz WiFi connection, does not support 5GHz networks).

After connecting the device to the network it is necessary to follow the steps of the initial configuration in which you must indicate what will be the type of input and output audio. With this configuration, the audio input is selected via USB because the connection between the PS Eye and Raspberry is made via a USB port. The type of microphone is also indicated, the PS Eye appears among the options, so it was selected. And the audio output selected is via HDMI by the fact that the sound comes out through the monitor which is connected via an HDMI cable.

The first thing that happens is that the voice assistant starts to repeat continuously a registration code. At the same time, in the screen, a kind of console appears where it is shown what is happening inside the assistant. In the later starts, this console can be accessed by executing the command *mycroft-cli-client* in the system. This code is necessary to be able to pair the Mycroft system with an account (in *home.mycroft.ai*) that allows the user to select some configuration options. This configuration is necessary to be able to start working with the voice assistant.

When accessing this website from any device, it is possible to configure different parameters of the device: the name, voice, location, even the wake word to be used. The option to use a custom wake word appears.

To check that everything is working properly and that the parameters set on the website have been configured, just say one of the following commands:

"Hey Mycroft, what time is it?" or "Hey Mycroft, what's the weather like?"

³⁴https://www.playstation.com/en-ae/content/dam/support/manuals/scee/web-manuals/peripherals/ps3/ps3-eye/SCEH-00448_PS%20Eye%20Web_GB.pdf/

The system should generate a response which would indicate that the basic configuration has been correctly installed, as in this case.

From the device configuration website it is possible to configure and install various skills that have been shared in the marketplace. This is a great advantage of being an Open Source system with such a big community behind. In the case of this particular system, it has been chosen to download and test most of these skills to see how they really work.

Setting up

The next step of the process setting up a skill for the voice assistant to connect to the openHAB platform.

Subsequently, a skill of his own was designed. In order to create a Mycroft skill, a number of requirements must be followed for it to work properly. A folder must be created with the name of the skill, which in turn must have two other folders inside (dialog and vocab), these folders contain the information of the input commands and the answers. There is also a file with the name "`__init__.py`" which contains the operation of the skill as such³⁵.

These two folder have to have different subfolder inside them depending on the languages that should be supported. In this case it has been thought to work in two languages, in English (subfolder called *en-us* due to the configuration languages), as it is the basic language in which all the skills available in the marketplace work, and, in this case, also German (*de-de*).

Inside these subfolders there should be different files in *.dialog* format in which the different options of answers with which the system can respond to certain commands are collected. They are divided into different files with different replies depending on the types of commands marked. Each line of the code identifies each of these possibilities that the system will randomly pass through the TTS process to generate a response to the user.

As an example, it is possible to use the dialog file represented in the Listing 1 that collects the different options of answers for the case of the adjustment of blinds for the English language. The concrete use depends on how these elements are inserted in the general Python code.

```
It has been adjusted to your liking  
Adjusted to your liking  
Done
```

Listing 1: Commands save in BlindsAdjust.dialog

The vocab folder follows the same structure, it has different subfolders depending on the different system languages. Within these subfolders there are files containing the different keywords that make it execute or not a part of the skill (a certain part of the Python code).

³⁵ https://github.com/sesma24/mycroft_openhab_skill (Skill repository)

These files, whose format is .voc, allow the code to be fragmented, which makes its handling more efficient. If the system receives a certain voice command, it only takes into account a part of the written code and does not have to go through everything. An example of this can be the *OnOff.voc* file represented in the Listing 2 that collects the different keywords that will be mapped to the function executing the respective skill.

```
Turn
Switch
Change
```

Listing 2: Keywords save in OnOff.voc

The key file that marks the operation of the skill as such is collected within the `__init__.py` file. In this skill it has been based on the sending of different POST requests from the voice assistant to the openHAB platform.

```
def initialize(self):

    self.log.info(self.lang)
    if self.lang == 'en-us':
        self.speak('I am an english gentleman')
        switch = IntentBuilder("ControlIntent").require("OnOff").build()
        self.register_intent(switch, self.things_onoff)

        percentage = IntentBuilder("ControlIntent1").require("Percentage").build()
        self.register_intent(percentage, self.things_control)

        music = IntentBuilder("ControlIntent2").require("MusicCommands").build()
        self.register_intent(music, self.music_control)

    elif self.lang == 'de-de':
        self.speak('Ich bin Deutscher')

        switchG = IntentBuilder("ControlIntentG").require("OnOffG").build()
        self.register_intent(switchG, self.things_onoffG)

        percentageG = IntentBuilder("ControlIntentG").require("PercentageG").build()
        self.register_intent(percentageG, self.things_controlG)

        musicG = IntentBuilder("ControlIntentG").require("MusicCommandsG").build()
        self.register_intent(musicG, self.music_controlG)
```

Listing 3: Initializing and detecting language in Python code

The general structure that the code must follow in order to work correctly must be to create a certain class that extends the MycroftSkill class. Within this, the `__init__()` method must be defined so that it acts as a constructor. The `initialize()` part is executed just after the skill is built

and registered, as a previous point to the execution of the different functions (see Listing 3). This section collects the initial configuration and the part to register the different attempts. At the end of the code, the `stop()` method must be introduced and a final code block called `create_skill()` that returns the new skill.

A function (`send_post_openhab`) was defined for sending an http POST request to the openHAB server. The function expects 2 parameters: `item` (item name) and `signal` (item value) (see Listing 4).

```
def send_post_openhab(self,item, signal):
    URL= 'http://openHAB_server_ip:8080/rest/items/'+item
    headers = {
        'Content-Type': 'text/plain',
        'Accept': 'application/json',
    }

    data = signal
    url_archivo_salida = 'response.xml'

    try:
        resp = requests.post( URL, headers = headers, data = data )

    except Exception as e:
        print( 'The exception >> ' + type(e).__name__ )
        raise e

    else:
        #requests.codes.ok = 200 => OK
        if( resp.status_code == requests.codes.ok ):
            with open( url_archivo_salida, 'w' ) as f:
                f.write( resp.text )
                f.close()
        else:
            out = 'resp.status_code >> ' + str(resp.status_code) + ' != '
            + str(requests.codes.ok)
```

Listing 4: Function definition `send_post_openhab`

It is also necessary to access that URL from a browser to see the exact name of the different elements. This information is necessary so that the data can be entered correctly in the different POST requests that will appear throughout the code.

One advantage of using an openHAB hub is that it allows information on all possible actions that can be performed with all devices to be collected in one place. This unification makes the task of introducing new devices into the system much easier.

The first part of the code (see Listing 3) sets the language id. Following a structure like that of the Listing 5 for the different types of commands, the execution of the different parts of the code is controlled according to the different keywords identified in each command.

```
switch = IntentBuilder("ControlIntent").require("OnOff").build()
self.register_intent(switch, self.things_onoff)
```

Listing 5: Register and identification of On/Off intents

The rest of the code is based on different functions that are executed depending on the appearance of different keywords. These functions are based on sending different POST requests using the function created above. At the same time, after each command is sent to the server, a response is generated for the user. To do this, the *speak_dialog* function is used, which is included in the Mycroft library.

For all this to work it is necessary to import the Mycroft libraries as well as the one dedicated to the identification of the attempts.

Finally, after finishing the development of the skill, there is the part of entering it into the voice assistant system. It is necessary to finish by copying the general folder, which contains all the elements necessary for the operation of the skill, into the appropriate directory within the system (/opt/mycroft/skills) where all the skills are collected. Mycroft's system has its own tools to create the skill directly on the device but in this case it has been chosen to develop it on an external computer, for convenience.

If nothing is changed from the initial configuration, the system will default to English. In order for the system to work in another language, in this case German, must be changed the system configuration. When a certain language is selected, only the part of the different Skills dedicated to each specific language will work.

A very important detail to bear in mind is that, nowadays, if you want to use an alternative language to English in Mycroft, you need to introduce tools from the big companies into the process, such as Google's ASR. In that case, the advantages in relation to privacy that these systems present are lost because the data end up going through these companies.

To check the correct functioning of the part of the system developed in German, the following parameters must be configured to make the whole system work in an alternative language, with the inconveniences mentioned above.

First, it is necessary to change the default language of the system, this can be done simply by executing the following command Listing 6:

```
mycroft-config set lang "de-de"
```

Listing 6: Command to set german language in Mycroft

Similarly, to be able to use the system again in English, you only have to execute the command indicating that the language becomes "en-us".

Using the basic configuration the system is not capable of operating correctly in a language that is not English. It is necessary to change the system configuration by adding some specific variables to the system to make it work in German, as seen in Listing 7 .

```
{
  "Lang": "de-de",
  "ipc_path": "/ramdisk/mycroft/ipc/",

  "stt": {
    "module": "mycroft",
    "mycroft": {
      "Lang": "de-de"
    }
  },
  "tts": {
    "module": "google",
    "google": {
      "Lang": "de"
    }
  },
  "play_wav_cmdline": "aplay -Dhw:0,0 %1",
  "play_mp3_cmdline": "mpg123 -a hw:0,0 %1",
  "enclosure": {
    "platform": "picroft"
  },
  "max_allowed_core_version": 20.3,
  "skills": {
    "auto_update": true
  }
}
```

Listing 7: Configuring Mycroft's system parameters to work in German (JSON format)

As can be clearly seen in this part of the configuration, it is necessary to introduce elements from the large companies in the market. In this case the Google module for the TTS process has been introduced, so that the system is able to work under these conditions. In these cases all the advantages in relation to privacy that this system presents are lost.

3.5 Rhasspy

The Rhasspy system is an alternative that is clearly identified by being a system that works completely locally, this, as mentioned above, has its good and bad points. The fact that the processing is done locally and without sending any information to the cloud ensures that it is the best solution in relation to the privacy. It is clear that no information ends up reaching external companies, so you do not lose control of the information.

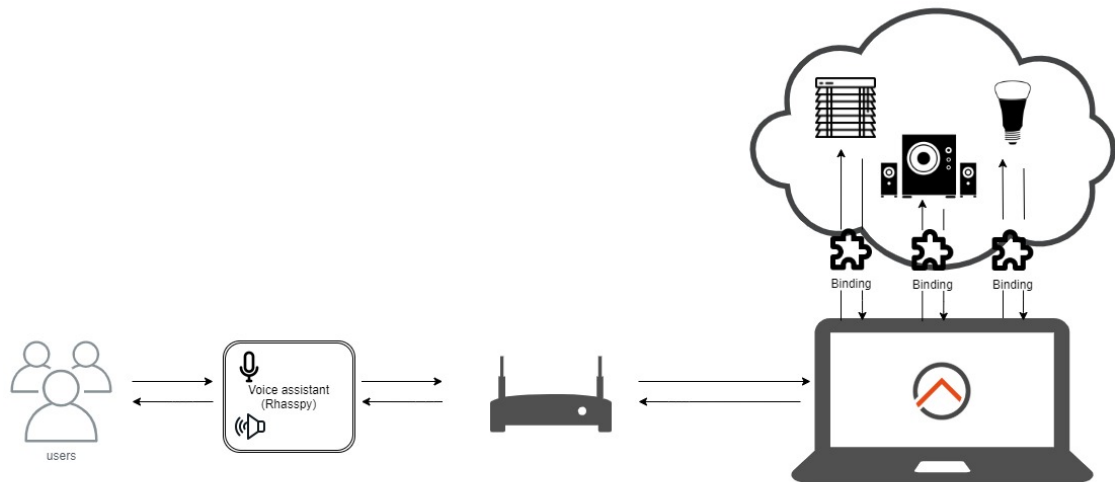


Figure 10: Schematic of a Smart Appliances control system with an openHAB server and Rhasspy voice assistant

Figure 10 shows that the whole system works locally. In this case, unlike Mycroft, there is no data sent to the cloud. The whole process of the voice assistant is carried out within the device itself, and in turn, it also sends the different commands to the openHAB server to be sent to the different devices.

The negative side of a system with these features is that some additional functionality that other systems like Mycroft may have are lost. In the case of Rhasspy, for example, there is no marketplace in which users can download skills created by other users. Another negative point is that it is very difficult to configure certain skills, such as some weather-related skills, which require an Internet connection to obtain the necessary information.

As with the Mycroft system, the next step is to explain how the voice assistant has been installed on a Raspberry Pi. It also indicates how a skill has been developed to control the different IoT devices and how it has been integrated into the whole system.

Setting up

In relation to the voice assistant part, different options are allowed for its implementation. In the official Rhasspy documentation appears four different ways to integrate this part of the system, the installation based on the use of a Docker has been used.

To install the Raspbian operating system on the Raspberry, there are a number of steps to follow. First, it is necessary to download the official image (NOOBS³⁶) to a computer, connect an SD card and format it. Secondly, the downloaded zip file is unzipped on the SD card. After that, the card is connected to the Raspberry, with all the accessories attached (including a

³⁶ <https://www.raspberrypi.org/downloads/noobs/>

wired Internet connection, and a screen with keyboard and mouse, at first). To finish the installation it is only necessary to follow the process indicated by the system.

Once the installation is complete, it is essential to connect the device to a WiFi network if it is to be placed in a location where there is no network point. Whichever option is chosen, it must be configured in the same network where the openHAB server, and the different IoT devices, are located.

The next step, if the option of using Rhasspy with Docker is chosen, is to install the Docker in the system and make sure that the user of the operative system is part of the Docker group. This process is done very easily by executing two simple commands in the system console as seen in Listing 8:

```
curl -sSL https://get.docker.com | sh
sudo usermod -a -G docker $USER
```

Listing 8: Commands to install Docker in Raspbian

At this point the system is ready to launch the image of Rhasspy in the Docker. To select the language simply is necessary to change the last line of the following command (see Listing 9). Depending on which profile is chosen, "en" or "de", the system will start and therefore also work in English or German respectively. It is possible to select other languages but in this case these two have been selected. In this case as in most of the Open Source systems, the system works better in English than in the rest of the languages.

```
docker run -d -p 12101:12101 \
  --restart unless-stopped \
  -v "$HOME/.config/rhasspy/profiles:/profiles" \
  --device /dev/snd:/dev/snd \
  synesthesiam/rhasspy-server:latest \
  --user-profiles /profiles \
  --profile en
```

Listing 9: Command to run Rhasspy system

In the command itself there are different key parameters. The "-d" indicates that it is executed in the background, the number behind "-p" indicates the port on which the system will work, the "--device" indicates that Rhasspy is allowed access to the microphone, and, as previously mentioned, the profile section reflects the language in which the system will be started.

For the configuration part of the Rhasspy system as such, it should be noted that this system offers a wide range of possibilities for selecting the different tools with which the voice assistant set will work. It is possible to choose both the wake word and the intent handing, going through the process part of STT and TTS, and not forgetting the configuration of the connected devices (microphone and speakers) necessary for the system to work.

To make the different changes to configure the voice assistant the system provides a website that can be accessed from the browser directly (*http://localhost:12101*). From this website it is possible to configure all the system parameters in a very simple way (see Figure 11). The tab referred to these issues within the website is called "Settings".

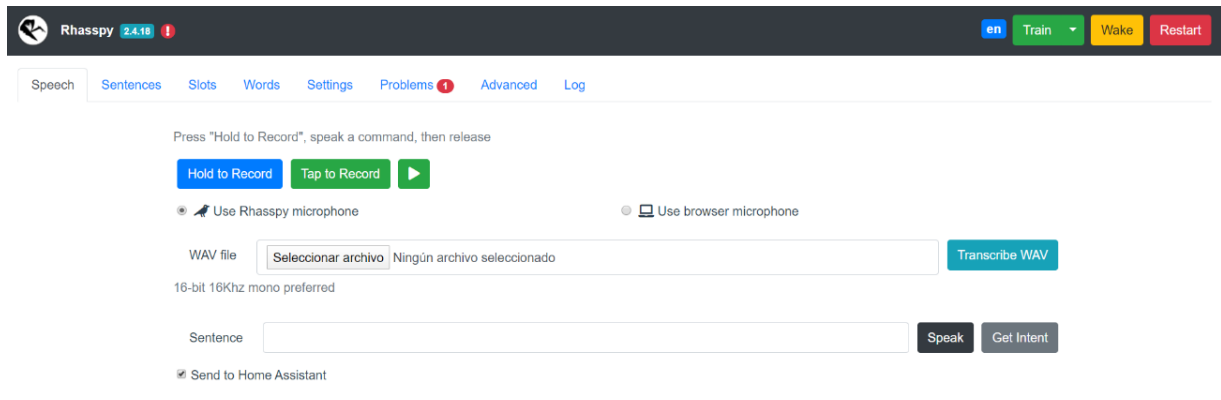


Figure 11: Rhaspy website for system configuration

In this case, after analysing the different options, the following configuration has been chosen:

- Wake word: Snowboy with sensitivity of 0.6
- STT: Kaldi instead of Pocketsphinx
- Microphone: Pyaudio

For the rest of the configurable parameters, it has been decided to leave the default values. This configuration has been considered as the most optimal for the system. After making any changes to the configuration it is necessary to restart the system.

It should be noted that in the case of Mycroft, Kaldi can also be used for the STT process. However, the default setting of Mycroft is very well optimised, so it is not necessary to make this change.

To check that all parameters have been set as indicated, it is possible to access to the "Advanced" tab. An automatically created JSON appears containing all the changes that have been made to the default configuration of the system. In this case, with the indicated configuration, it would be as shown in Listing 10.

```

{
  "handle": {
    "command": {
      "program": "$RHASSPY_PROFILE_DIR/ex12.py"
    },
    "forward_to_hass": true,
    "system": "command"
  },
  "microphone": {
    "pyaudio": {
      "device": "5"
    }
  },
  "speech_to_text": {
    "system": "kaldi"
  },
  "wake": {
    "system": "snowboy"
  }
}

```

Listing 10: JSON that picks up the advanced configuration of the Rhasspy

Regarding the problem shown above in the Rhasspy website home page in the Figure 11, it is due to the fact that the system has been configured to run a certain skill. This skill is a certain Python code to control the IoT devices as it has been done in Mycroft. The error appears because the pre-installed part to control a Home Assistant server is not used.

As it is reflected in the previous JSON, Listing 10, some lines has been added to make this configuration possible. It is indicated that for the intent handler the request is going to be derived to a certain Python file. The file called *ex12.py* is stored in the */home/pi/.config/rhasspy/profiles/en* to be able to access easily taking advantage of the variable *\$RHASSPY_PROFILE_DIR* generated by the own system of the assistant³⁷.

For the skill to be introduced also in the German, as they are independent files, the equivalent *ex12g.py* file must be placed in */home/pi/.config/rhasspy/profiles/de*. In this profile, system information is collected when the profile is set to German.

In order for the skill to work also in German, the equivalent *ex12g.py* file must be placed in */home/pi/.config/rhasspy/profiles/de*. This folder contains all the information in the system regarding the German profile.

The fundamental operation of this system is described in the lines of Python code reflected in Listing 11. Depending on which commands are detected (according to the configuration made

³⁷ https://github.com/sesma24/rhasspy_openhab_skill

on the website), a certain function will be executed. This type of configuration allows the system to be fragmented and optimised.

Furthermore, in the case of Rhasspy, the different possible responses to each command are indicated depending on which group of commands they belong to. The speech function is used to convert the different responses into sound. It is clear that the communication between the assistant as such and the code is done by sending JSON's.

Depending on the different commands identified, a certain function is executed according to the group it is in. Inside the different functions, different POST requests are executed depending on the keywords detected. The same `send_post_openhab` function used in the case of Mycroft is used (see Listing 4).

It is also important to mention that in this case the system works with python3 so this has to be indicated in the first line of the code. In this case this fact makes it impossible to use `"word" in msg.data.get('utterance')` to detect if a certain word appears in the message said by the user, but rather it is used the alternative of `msg.find("word")>=0` which is the alternative that works in python3.

```
o = json.loads(sys.stdin.read())

intent = o["intent"]["name"]
msg = o["text"]

if intent == "GetTime":
    now = datetime.datetime.now()
    speech("Charles It's %s %d %s." % (now.strftime('%H'), now.minute, now.strftime('%p')))
    send_post_openhab("HueBulb_Color_Color", "27,100,100")

elif intent == "Hello":
    replies = ['Hi!', 'Hello!', 'Hey there!', 'Greetings.']
    speech(random.choice(replies))

elif intent == "PercentageControl":
    things_control(msg)
    replies = ['Okey', 'Done!', 'Adjusted as your like']
    speech(random.choice(replies))
    speech(msg)

elif intent == "ChangeState":
    things_onoff(msg)
    replies = ['Okey', 'Done!', 'Adjusted as your like']
    speech(random.choice(replies))

elif intent == "MusicControl":
    music_control(msg)
    replies = ['Okey', 'Done!', 'Adjusted as your like']
    speech(random.choice(replies))

# convert dict to json and print to stdout
print(json.dumps(o))
```

Listing 11: Identification of intents and call to the different functions in the Rhasspy code

In order to make the Python part of the code work in Rhasspy one more step has to be added in the system configuration. It is necessary to define in the "Sentences" section of the website the different possibilities that a user has to say a certain command. In this point is where it is indicated which commands belong to each group. Some examples of this are shown below in the Listing 12 (Appendix A complete):

```
[PercentageControl]
blind_name = ((blind | blinds | shutters) {name})
place_name = ((kitchen | living room | all | house | home | room | bedroom )
{name})7
blind_state = (up | down) {state}

regulate to (0..100) percent <place_name> <blind_name>
adjust <place_name> <blind_name> to (0..100) [percent]
roll <blind_state> [the] <blind_name> [of the] <place_name>
```

Listing 12: Examples of Rhasspy configuration Sentences

The definition of the different sentences follows a certain structure. The name of the group appears at the top. The definition of certain parameters that are useful for the definition of the commands follows. Finally all the different possible commands that the assistant must support are defined.

This process must be repeated as many times as group of commands wanted to be created. After that the system has to be trained to be able to identify these messages before using it again. This is one of the main disadvantages of the system working locally and that the processing is not done in the cloud.

3.6 Testing and calculation of statistics

After the configuration of the systems and the integration of the skills created, a series of tests were carried out to check the effectiveness of both systems. To do this, 20 probands were asked to record different audios of how they would tell a voice assistant a certain command. The goal of this is to be able to analyse in which cases the system correctly detected what the user was saying and in which cases the command that the user wanted to occur was executed.

1. Raise the blinds in the kitchen/living room.
2. Lower the blinds in the kitchen/living room.
3. Adjust the kitchen/living room blinds to x %.
4. Turn on/off the lights in the kitchen/living room/bedroom.
5. Change the lights from (x place) to color ...
6. Switch on/off the hob.
7. Switch on/off the oven.
8. Regulates the lights from (x place) to x %.
9. Turns on/off all house lights.
10. Turn on/off the ceiling/center/kitchen light.
11. Switch on/off the stereo.
12. Play/pause/... music

13. Change the music equipment to Bluetooth/tuner/cd/... mode
14. Mute the stereo
15. Set an alarm...
16. What's the weather gonna be like?/ it's gonna rain?/ something like that
17. Who is ... (famous)?
18. What time is it
19. Extra 1
20. Extra 2

Listing 13: Set of commands

Each of the probands was asked to record 20 different commands (see Listing 13). The first 14 were intended for the skill that had been created. The commands 15-18 were used to check the operation of other skills downloaded from the official marketplace. Two extra commands are included where the probands had to record two commands that they considered a system like these should be able to process correctly. The Appendix B contains the list of the exact sentences that appear in the recordings.

To obtain more realistic samples by adding more variability to the test data, the command set was passed to the users in Spanish but the recordings had to be made in English.

All the probands, except the 16th, made their recordings using a smartphone. The formats in which the recordings are made on each device are not always the same. The recordings of the different users are divided into 4 different audio formats as reflected in the Table 2.

Format	Probands
.mp4	1,5,8,14
.ogg	2,4,6,11,12,17
.m4a	3,7,9,10,15
.mp3	16

Table 2: Format of the audios of the different probands

The proband 16 made the recordings with an iMac, therefore the audio files are in .acc format.

In Mycroft's case the tests of 17 of the 20 people were conducted with background noise (music at medium volume included) using the speaker of a Xiaomi mi 9t smartphone as a player for the recordings. The device was placed at a distance of about 50 cm and at a volume of 60% of the PS Eye. The other three people did the physical tests in the room where the system was located without using recordings.

The same audios were used to analyse the Rhasspy system. In this case for each of the probands the test was only carried out with 17 of the commands recorded instead of the 20 that were used in the other system. In this assistant there is no option to download different

skills from one marketplace as in Mycroft. Commands 15, 16 and 17 in the list have not been used in this case.

The tests as such have been done under the same conditions as in the case of Mycroft. The recordings of the 17 people reproduced on a Xiaomi mi 9t were used and 3 other people carried out the tests directly in person.

Each test was performed three times, if 2 or more correct results were obtained it was identified as correct and if not as error.

The results are represented in percentages to make them clearer. The percentages of correct cases and erroneous cases are represented. The following formulas, reflected in Table 3, are used to obtain these values:

$RR = \frac{C_{rec}}{N_{cmds}} \times 100$	$ER = \frac{C_{exec}}{N_{cmds}} \times 100$	<i>RR</i> = Recognition Rate <i>ER</i> = Execution Rate <i>RER</i> = Recognition Error Rate <i>EER</i> = Execution Error Rate <i>Crec</i> = Commands recognized <i>Cnorec</i> = Commands no recognized <i>Cexec</i> = Commands no executed <i>Cnoexec</i> = Commands no executed <i>Ncmds</i> = total number of commands
$RER = \frac{C_{norec}}{N_{cmds}} \times 100 = 100 - RR$	$EER = \frac{C_{noexec}}{N_{cmds}} \times 100 = 100 - ER$	

Table 3: Equations to obtain genral statistics

The above formulas are used to obtain the recognition and execution rates for both voice assistants. Depending on the system analysed in each case, the values entered in the formula change.

In the case of Mycroft $N_{cmds} = 400$, including the extra commands. For Rhasspy $N_{cmds} = 340$. For statistics where the extra commands are not taken into account the total number of cases decreases, $N_{cmds} = 360$ and $N_{cmds} = 300$, respectively.

The values of C_{rec} and C_{norec} are complementary, the sum of both values must always be equal to N_{cmds} in all cases. The same happens with C_{exec} and C_{noexec} .

It is also of special interest to fragment the percentages according to each specific command. In all this cases $N_{cmds} = 20$. This number comes from the fact that each command has been tested by 20 different people. Table 4 shows the formulas for this case:

$RR_i = \frac{C_{rec_i}}{20} \times 100$	$ER_i = \frac{C_{exec_i}}{20} \times 100$
$RER_i = \frac{C_{norec_i}}{20} \times 100 = 100 - RR_i$	$EER_i = \frac{C_{noexec_i}}{20} \times 100 = 100 - ER_i$

Table 4: Equations to obtain statistics for each command

4 Results

This section shows the results obtained on the analysis of the functioning of the Mycroft and Rhasspy system. Below are some graphs that summarize the results, in Appendix C are the tables with all the raw data.

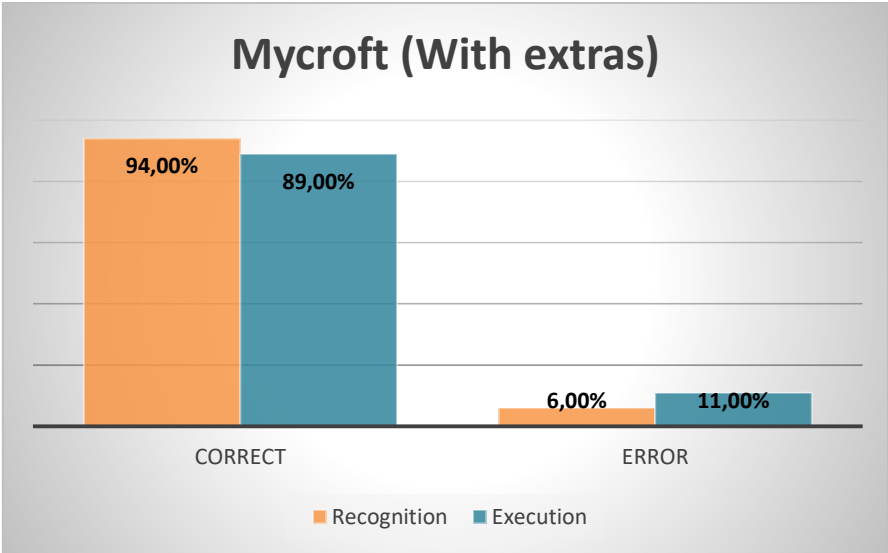


Figure 12: Mycroft recognition and execution rate with extra commands

Figure 12 shows the results of speech recognition (yellow) and command / intent execution (blue). It is clear that the number of correct cases is much higher than the number of wrong cases, which is a good indicator of the correct functioning of the system.

The Figure 13 also reflects the results of the same data but without including the two commands in which all users were allowed to add two additional ideas without any special requirements.

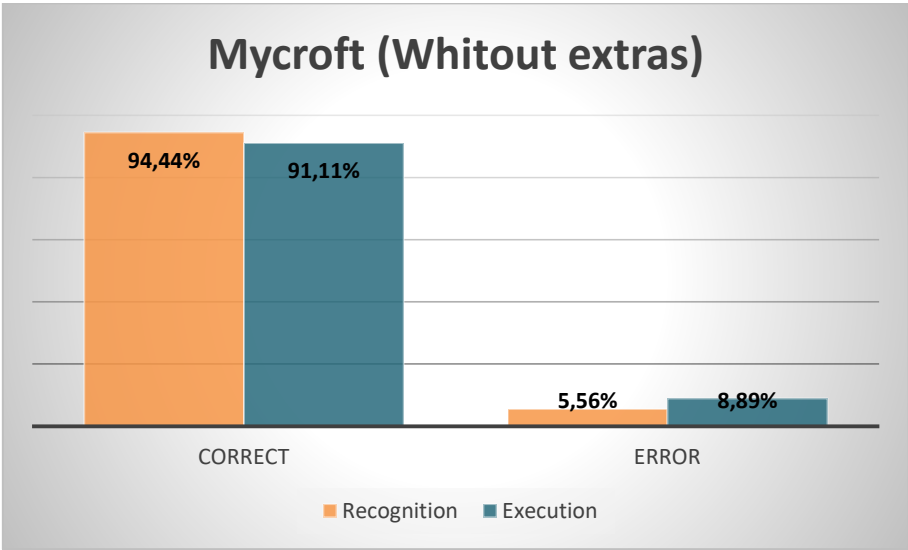


Figure 13: Mycroft recognition and execution rate without extra commands

Looking at the graph it can be seen that in the case of the recognition of the commands (without counting the extras) a 94.44% of success has been obtained against a 5.56% of errors, which is a quite high percentage. And in the case of those who have executed the indicated command correctly, it is obtained a 91.11% success rate, and therefore a 8.89% error rate. This reflects that there are some cases in which people say a certain voice command to execute a certain action that the system is not able to interpret correctly.

The results obtained for Rhasspy are reflected in the following graphs.

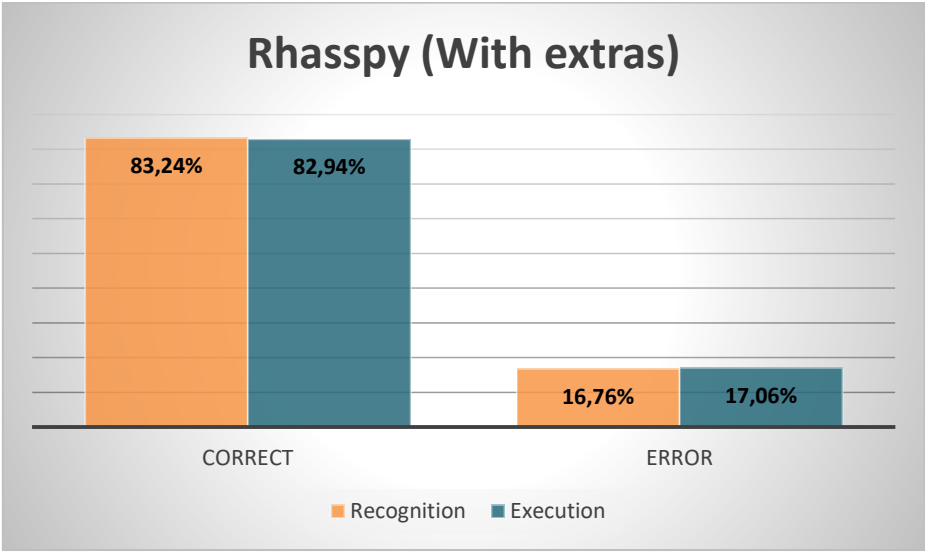


Figure 14: Rhasspy recognition and execution rate with extra commands

In this case too, two separate graphs have been made, one including the results of the extra sentences that have been asked of users and the other not.

It can be seen in the Figure 14 that the relationship between the percentage of commands that have been detected correctly and those that have been executed correctly is much more similar than in the case of Mycroft.

The characteristics of the Rhasspy system make it have a less wide range of possibilities and functionalities. Nevertheless this approach generates other advantages, and within this, the percentage of correct data is very high.

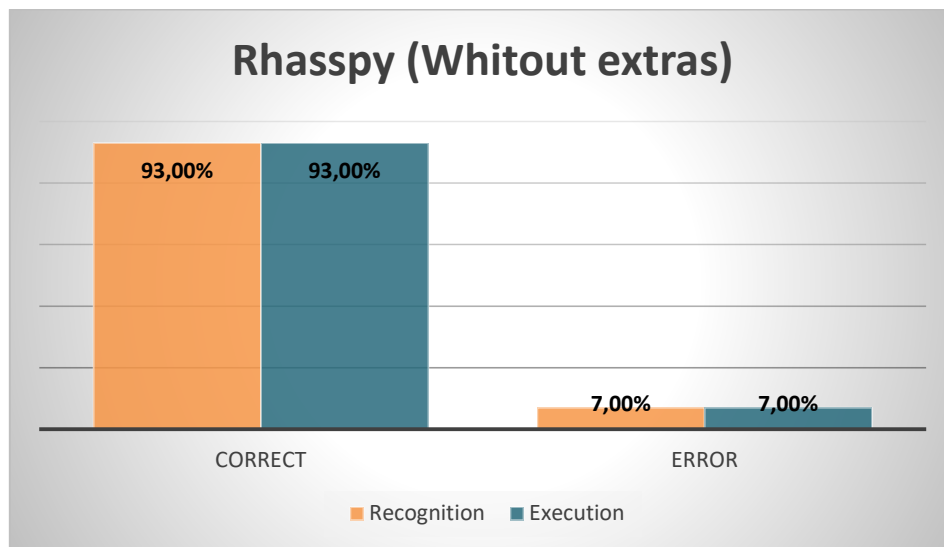


Figure 15: Rhasspy recognition and execution rate without extra commands

Comparing the Figure 15 with the Figure 14, whose only difference is the fact that the data of the extra sentences is not included, it can be seen how the number of correct cases with respect to errors is higher. While in the case where the extra sentences are considered there are correct values of around 83% in both the case of recognition and execution, in the case of not taking them into account the correct values are as high as 93% in both cases. This fact clearly indicates how this system is a system that works very well within established parameters (sentences that have been indicated to train the system), but that outside these limits, it is not capable of working properly.

This makes it clear that further configuration is necessary in the case of using an assistant that works completely locally and does not use the cloud in any process of its operation.

A detail that has been identified with respect to Mycroft's system is that Rhasspy takes a little longer to carry out the action as such indicated by the user on the different IoT devices. However, it must be said that the percentage of successes, both in the case of detection and in the execution of the commands, is very similar in both systems. In this particular case a success rate of 93% is obtained (without extras), which is a very high percentage for a system that also works completely locally.

Based on the more general data shown in the previous results, a more detailed analysis is proposed according to the different commands. The percentages of recognition and execution success for each command are considered separately.

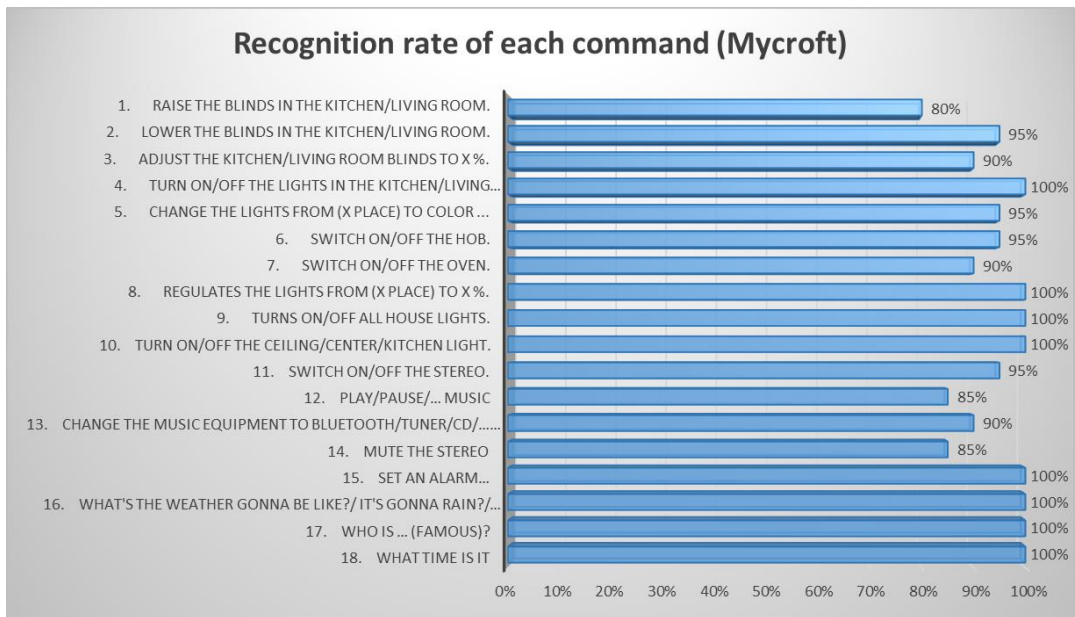


Figure 16: Mycroft recognition rates for each command

In the Figure 16 it is possible to see how the percentage of recognition of all the commands is equal or higher than 80%, even having 8 commands (4,8,9,10,15,16,17,18) in which 100% of the cases are correct.

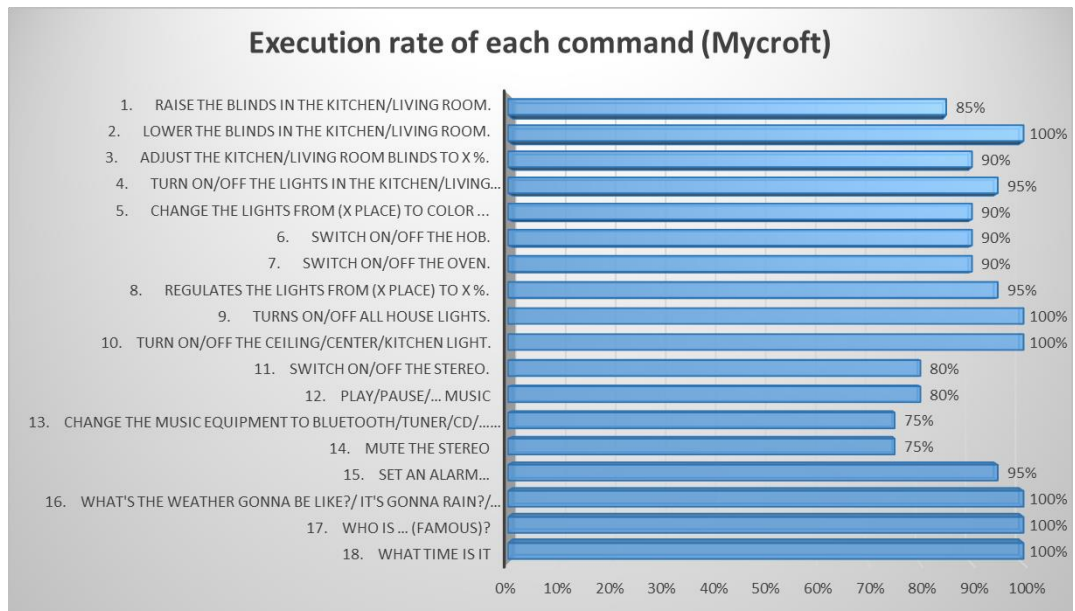


Figure 17: Mycroft execution rates for each command

Continuing with what has been observed regarding the recognition rate, in the case of the percentage of hits in the execution of the commands, very similar values appear as reflected in the Figure 17. In this case, the minimum execution rate of a command drops to 75% in two commands (13,14). The number of commands with 100% correct cases is 6 (commands 2,9,10,16,17,18).

The values obtained in the case of Rhasspy are shown in the same way below.

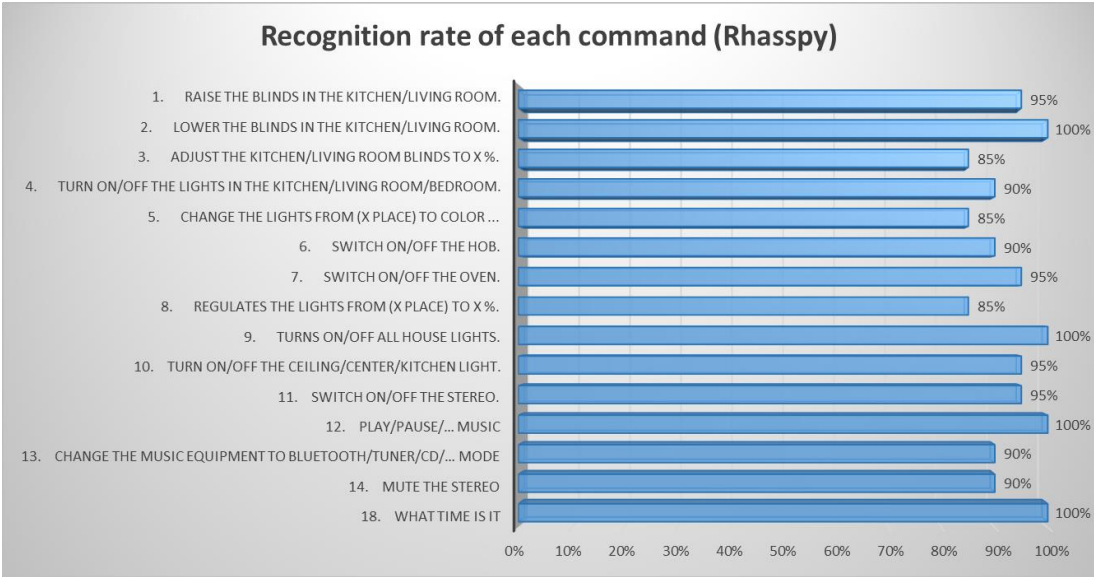


Figure 18: Rhasspy recognition rates for each command

Figure 18 shows the recognition rate results of the Rhasspy case. As it has been named before, it is a locally operated system so there are some commands that have not been included in this analysis. In terms of the results obtained, it should be noted that no command has a hit rate below 85%.

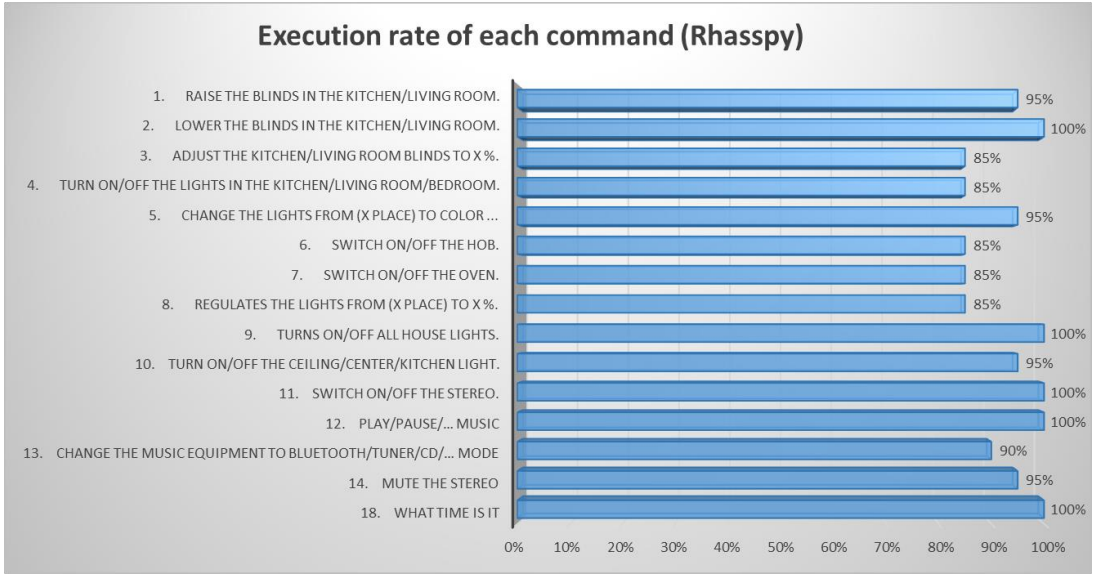


Figure 19: Rhasspy execution rates for each command

To finish with the results section, Figure 19 shows the values that indicate the percentage of correct cases in the case of the execution of the commands in the Rhasspy system. It can be seen that the values obtained are very similar to those of recognition of the commands, and, likewise, the lowest execution rate of a given command is 85%.

5 Discussion

Regarding the results obtained, it is concluded that, making a proper configuration of the systems, both the openHAB server and the voice assistants, a system with an optimal performance can be achieved. It works really well because have values higher than 90% both in the recognition of the commands and in the execution of the commands.

The most famous voice assistants, such as Amazon Alexa or Google's assistant, present recognition rates close to 100%, as seen in *Effectiveness of assistants*. However there are many more factors that must be taken into consideration when choosing different systems.

The success rates present lower values because the probands were not told the exact commands to record. They were only informed that they had to record a command to execute a certain order. This factor adds greater variability to the experiment but results in lower accuracy values.

It is also noteworthy that in the Mycroft (online) option, more differences are perceived between the results obtained in recognition and those obtained in execution compared to the results of Rhasspy (local). This is due to the fact that in the case of Rhasspy it is necessary to configure all the different possibilities of commands that can be identified by the user, which further limits the range of possibilities. The drawbacks are that fewer commands are identified and that a more extensive configuration process is required. The main advantage is that it is more difficult to identify strange things that do not make sense.

It is also important to take into account that the more different ways of saying a command, the more errors appear. For example, in the case of controlling the stereo there is a wide range of options to name the music equipment. The same applies to the blinds, where it also exists a broad spectrum of different commands to indicate that they should be raised.

Although it is not reflected in the results, but it has been possible to perceive at the time of making the analysis, the system that works locally takes longer to provide a response to the user. Although for Mycroft it is necessary to communicate with the cloud its processing time and delivery time are shorter than Rhasspy's times.

Besides, the fact that one system is connected to the cloud and another is not, means that different skills created by other users can be downloaded. This idea is clearly reflected in the percentage of accuracy in relation to the extra sentences. Adding these two commands to the list for each user reflects the advantage of being able to access a marketplace, with the added functionality this can provide to the system.

This added difficulty that the same keyword can be used for different types of commands or different skills must be taken into account. In this case, for example, it has been considered that it would be possible to use the word "turn" to change the colour of the lights, but this

possibility has not been integrated into the system. The possibility of fragmenting the system which makes the operation of the system more optimal conflicts with this added difficulty.

Last but not least, not being able to face such a big investment makes Open Source systems use great companies' components to implement other languages. However, in the specific case of Mycroft, the system is being developed by the users themselves so that it can work correctly in more languages without the need to resort to the tools of large companies. At the same time, this demonstrates the problem that an Open Source system poses and the advantage that means a large community behind the development of this technology.

6 Conclusion

Human-machine interaction has been, and continues to be, one of the most important issues in the technological field. The objective has always been to try to make this process as humane as possible, for which the development of voice assistants has been key.

Voice control had always been considered as a desired or utopian solution. The technological development over the last few years has allowed it to become a reality for the ordinary user.

Likewise, with the automation of the home something similar has happened. The large number of companies' commitment to these technologies has allowed many users to have the chance to use these technologies in their houses, thanks to their utility as for their price.

There are countless options on the market to create a Smart Home environment, covering from a huge number of smart devices from different brands to a wide range of voice assistants. Using voice assistants to control different intelligent devices around the home has become an ideal combination.

In many technological areas there are usually different alternatives from different companies, in this case something similar happens. It has been chosen to analyse the advantages and disadvantages of Open Source voice assistants, such as Mycroft and Rhasspy systems.

The obtained results indicate how these options present clear advantages in relation to privacy, personalization and control of what happens from two very different approaches, online and locally, respectively.

Similarly, it has been possible to verify the ease of integration of an own skill in a system to control different devices in the home using a voice assistant. The optimal functioning of these systems using both Mycroft's and Rhasspy's voice assistants has also been seen, as well as the differences observed between the two approaches.

Both approaches have proved to be acceptable. The Rhasspy system, which works entirely locally, is more secure in relation to privacy. In contrast, Mycroft offers the clear advantages of having a marketplace and being easier to configure.

7 Future Work

These systems are some of the Open Source options that have been most developed in recent years thanks to the large community behind them. However, as their access to large amounts of data is limited, their evolution is not as fast as other commercial systems on the market.

For this reason, it is essential that different skills continue to be developed and shared. The aim is that these systems can continue to exist and compete with the systems of large companies. This allows the presence in the market of quality options, such as Mycroft and Rhasspy, with their respective advantages.

In the specific case of the skill created, the next step would be to introduce a hearing into the system. In this case there are extra difficulties, such as the need to send accurate information in both directions of the communication channel to work with the established parameters.

Another fundamental aspect for the evolution of these voice assistants is to try to collaborate with the community that develop them, so that the systems can work in more languages. This is fundamental for the purpose of being used by more people.

With respect to the integration of voice assistants for the control of different devices in the home, the most important idea that there will be more and more devices on the market. The main important difficulty will be to add the use of new devices without overlapping the functionalities, so it will be essential to improve the current system in order to facilitate these process.

Bibliography

- [1] J. Smith, "Outerbox," 2 January 2020. [Online]. Available: <https://www.outerboxdesign.com/web-design-articles/mobile-ecommerce-statistics>.
- [2] „Language Technology Group,“ [Online]. Available: <https://www.language-technology.com/twin#:~:text='This%20Week%20in%20NLP'%20is,the%20afternoon%20for%20European%20readers..>
- [3] R. Pieraccini and J. Huerta, "Where Do We Go from Here? Research and Commercial Spoken Dialog Systems," January 2005.
- [4] E. Merdivan, A. Vafeiadis, D. Kalatzis, S. Hanke, J. Kropf, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, R. Hamzaoui and M. Geist, "Image-based Natural Language Understanding Using 2D Convolutional Neural Networks," *ACROSSING*, October 2018.
- [5] B. A. Desai and B. N. Veerappa, "Smart Voice Assistant for IOT," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 7, no. VI, June 2019.
- [6] T. Huxohl, M. Pohling, B. Carlmeyer, B. Wrede and T. Hermann, "Interaction Guidelines for Personal Voice Assistants in Smart Homes," in *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Timisoara, Romania, 2019.
- [7] J. Lau, B. Zimmerman and F. Schaub, "Alexa, Are You Listening?: Privacy Perceptions, Concerns and Privacy-seeking Behaviors with Smart Speakers," November 2018.
- [8] R. Dale, "Voice assistance in 2019," in *Natural Language Engineering*, vol. 26, January 2020, pp. 129-136.
- [9] E. Brown, "Smart speaker voice platforms compared," 21 November 2018. [Online]. Available: <http://linuxgizmos.com/smart-speaker-voice-platforms-compared/>.
- [10] "Canalys, Amazon reclaims top spot in smart speaker market in Q3 2018," 15 Thursday November 2018. [Online]. Available: <https://www.canalys.com/newsroom/amazon-reclaims-top-spot-in-smart-speaker-market-in-q3-2018>.
- [11] W. T. Gene Munster, "Annual Digital Assistant IQ Test – Siri, Google Assistant, Alexa, Cortana," LOUPVENTURES, 25 July 2018. [Online]. Available: <https://loupventures.com/annual-digital-assistant-iq-test-siri-google-assistant-alexa-cortana/>.
- [12] W. T. Gene Munster, "Annual Digital Assistant IQ Test," LOUPVENTURES, 15 August 2019. [Online]. Available: <https://loupventures.com/annual-digital-assistant-iq-test/>.

- [13] B. . X. Chen and C. Metz, "Google's Duplex Uses A.I. to Mimic Humans (Sometimes)," *The New York Times*, 22 May 2019.
- [14] S. Kunthara, "California law takes aim at chatbots posing as humans," *San Francisco Chronicle*, 13 October 2018. [Online]. Available: <https://www.sfchronicle.com/business/article/California-law-takes-aim-at-chatbots-posing-as-13304005.php>.
- [15] J. Cambre, Y. Liu, R. E. Taylor and C. Kilkarni, "Vitro: Designing a Voice Assistant for the Scientific Lab Workplace," in *the 2019*, June 2019.
- [16] S. Perez, "Voice-enabled smart speakers to reach 55% of U.S. households by 2022, says report," *TechCrunch*, 8 November 2017. [Online]. Available: <https://techcrunch.com/2017/11/08/voice-enabled-smart-speakers-to-reach-55-of-u-s-households-by-2022-says-report/>.
- [17] K. Wiggers, „How Amazon, Apple, Google, Microsoft, and Samsung treat your voice data,“ *VentureBeat*, 15 April 2019. [Online]. Available: <https://venturebeat.com/2019/04/15/how-amazon-apple-google-microsoft-and-samsung-treat-your-voice-data/>.
- [18] T. Ammari, J. Kaye, J. Y. Tsai and F. Bentley, "Music, Search, and IoT: How People (Really) Use Voice Assistants," *ACM Transactions on Computer-Human Interaction*, vol. 26, no. 3, pp. 1-28, April 2019.
- [19] R. Aloufi, H. Haddadi and D. Boyle, "Emotionless: Privacy-Preserving Speech Analysis for Voice Assistants," August 2019.
- [20] I. Oomen and R. E. Leenes, "Privacy Risk Perceptions and Privacy Protection Strategies," in *Policies and Research in Identity Management*, May 2008, pp. 121-138.
- [21] S. Barth and M. De Jong, "The Privacy Paradox – Investigating Discrepancies between Expressed Privacy Concerns and Actual Online Behavior – A Systematic Literature Review," vol. 34, no. 7, pp. 1038-1058, April 2017.
- [22] A. Acquisti and J. Grossklags, "Privacy and rationality in individual decision making," *IEEE Security and Privacy Magazine*, vol. 3, no. 1, pp. 26-33, February 2005.
- [23] M. B. Hoy, "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants," *Medical Reference Services Quarterly*, vol. 37, no. 1, pp. 81-88, January 2018.
- [24] D. o. H. a. S. Care, "NHS health information available through Amazon's Alexa," 10 July 2019. [Online]. Available: <https://www.gov.uk/government/news/nhs-health-information-available-through-amazon-s-alexa>.
- [25] M. Hadian, T. Altuwaiyan, X. Liang and W. Li, "Efficient and Privacy-Preserving Voice-Based Search over mHealth Data," in *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, July 2017.

- [26] B. Masters, „The AI doctor will see you now,“ *Financial Times*, 26 February 2020. [Online]. Available: <https://www.ft.com/content/d0aeec8-5703-11ea-abe5-8e03987b7b20>.
- [27] M. F. Schober, F. G. Conrad, C. Antoun, P. Ehlen, S. Fail, A. L. Hupp, M. Johnston, L. Vickers, H. Y. Yan and C. Zhang, “Precision and Disclosure in Text and Voice Interviews on Smartphones,” vol. 10, no. 6, 10 June 2015.
- [28] A. Liptak, “The Verge, Amazon’s Alexa started ordering people dollhouses after hearing its name on TV,” 7 January 2017. [Online]. Available: <https://www.theverge.com/2017/1/7/14200210/amazon-alexa-tech-news-anchor-order-dollhouse>.
- [29] D. Singh, I. Psychoula, E. Merdivan, J. Kropf, S. Hanke, E. Sandner, A. Holzinger and L. Chen, “Privacy-Enabled Smart Home Framework with Voice Assistant,” in *Smart Assisted Living*, January 2020, pp. 321-339.
- [30] G. Bell and J. Kaye, “Designing Technology for Domestic Spaces: A Kitchen Manifesto,” *Gastronomica*, vol. 2, no. 2, pp. 46-62, Spring 2002.
- [31] S. Uma, R. Eswari, R. Bhuvanya and G. S. Kumar, “IoT based Voice/Text Controlled Home Appliances,” *Procedia Computer Science*, vol. 165, pp. 232-238, 2019.
- [32] M.-M. Moazzami, D. Mashima, U. Herberg, W.-P. Chen and G. Xing, “SPOT: a smartphone-based control app with a device-agnostic and adaptive user-interface for IoT devices,” *UbiComp '16: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pp. 670-673, September 2016.
- [33] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson and P. Dutta, “The Internet of Things Has a Gateway Problem,” *HotMobile '15: Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pp. 27-32, February 2015.
- [34] B. Romano, “Managing the Internet of Things,” in *the 2017 ACM SIGCSE Technical Symposium*, March 2017.
- [35] C. Olson and K. Kemery, “The 2019 Voice report,” April 2019. [Online]. Available: <https://about.ads.microsoft.com/en-us/insights/2019-voice-report>.
- [36] R. A. Brooks, D. Dang, J. De Bonet, J. Kramer, J. P. Mellor, P. Pook, C. Stauffer, L. Stein, M. Torrance and M. Wessler, “The Intelligent Room Project,” *CiteSeer*, June 1999.
- [37] J. Newman, “Can Mycroft’s Privacy-Centric Voice Assistant Take On Alexa And Google?,” *Fast Company*, 29 January 2018. [Online]. Available: <https://www.fastcompany.com/40522226/can-mycrofts-privacy-centric-voice-assistant-take-on-alexa-and-google>.

- [38] B. Popović, E. Pakoci, N. Jakovljevic, G. Kočiš and D. Pekar, "Voice Assistant Application for the Serbian Language," in *23. Telekomunikacioni forum TELFOR*, Belgrade, Serbia, November 2015.
- [39] W. Seymour, M. Van Kleek, R. Binns and N. Shadbolt, "Aretha: A respectful voice assistant for the smart home," in *Living in the Internet of Things (IoT 2019)*, January 2019.
- [40] M. V. Kleek, W. Seymour, R. Binns, J. Zhao, D. Karandikar and N. ShadBolt, "IoT Refine: Making Smart Home Devices Accountable for Their Data Harvesting Practices," in *Living in the Internet of Things (IoT 2019)*, January 2019.
- [41] K. Reid, „Private By Design: Free and Private Voice Assistants,“ Makezine, 17 March 2020. [Online]. Available: <https://makezine.com/2020/03/17/private-by-design-free-and-private-voice-assistants/>.
- [42] K. Kreuzer, 12 April 2014. [Online]. Available: <https://raw.githubusercontent.com/wiki/openhab/openhab1-addons/images/architecture.png>.
- [43] „The official site of Philips Hue,“ [Online]. Available: <https://www2.meethue.com/en-us>.
- [44] "openHAB official website," [Online]. Available: <https://www.openhab.org/docs/>.
- [45] Land Rover, "Jaguar Land Rover invests in artificial intelligence start-up Mycroft," [Online]. Available: <https://www.wildelandroversarasota.com/jaguar-land-rover-invests-in-artificial-intelligence-start-up-mycroft/>.
- [46] Mycroft, „Mozilla Archives - Mycroft,“ [Online]. Available: <https://mycroft.ai/blog/tag/mozilla/>.

List of Figures

<i>Figure 1: Summary table of different Open Source tools that make up the voice assistants (Source: [41])</i>	24
<i>Figure 2: OpenHAB Architecture Overview (Source: [42])</i>	26
<i>Figure 3: Philips Hue devices (Source: [43])</i>	27
<i>Figure 4: OpenHAB console using PuTTY</i>	29
<i>Figure 5: OpenHAB website for system configuration (Add-ons)</i>	30
<i>Figure 6: OpenHAB website for system configuration (Things)</i>	31
<i>Figure 7: OpenHAB website for system configuration (One thing, the Hue bulb)</i>	31
<i>Figure 8: Schematic of a Smart Appliances control system with an openHAB server and Mycroft voice assistant</i>	32
<i>Figure 9: Picroft's interface when starting the system on Raspberry</i>	33
<i>Figure 10: Schematic of a Smart Appliances control system with an openHAB server and Rhasspy voice assistant</i>	40
<i>Figure 11: Rhasspy website for system configuration</i>	42
<i>Figure 12: Mycroft recognition and execution rate with extra commands</i>	48
<i>Figure 13: Mycroft recognition and execution rate without extra commands</i>	48
<i>Figure 14: Rhasspy recognition and execution rate with extra commands</i>	49
<i>Figure 15: Rhasspy recognition and execution rate without extra commands</i>	50
<i>Figure 16: Mycroft recognition rates for each command</i>	51
<i>Figure 17: Mycroft execution rates for each command</i>	51
<i>Figure 18: Rhasspy recognition rates for each command</i>	52
<i>Figure 19: Rhasspy execution rates for each command</i>	52

List of Listings

<i>Listing 1: Commands save in BlindsAdjust.dialog.....</i>	<i>35</i>
<i>Listing 2: Keywords save in OnOff.voc.....</i>	<i>36</i>
<i>Listing 3: Initializing and detecting language in Python code.....</i>	<i>36</i>
<i>Listing 4: Function definition send_post_openhab.....</i>	<i>37</i>
<i>Listing 5: Register and identification of On/Off intents.....</i>	<i>38</i>
<i>Listing 6: Command to set german language in Mycroft.....</i>	<i>38</i>
<i>Listing 7: Configuring Mycroft's system parameters to work in German (JSON format).....</i>	<i>39</i>
<i>Listing 8: Commands to install Docker in Raspbian.....</i>	<i>41</i>
<i>Listing 9: Command to run Rhasspy system.....</i>	<i>41</i>
<i>Listing 10: JSON that picks up the advanced configuration of the Rhasspy.....</i>	<i>43</i>
<i>Listing 11: Identification of intents and call to the different functions in the Rhasspy code.....</i>	<i>44</i>
<i>Listing 12: Examples of Rhasspy configuration Sentences.....</i>	<i>45</i>
<i>Listing 13: Set of commands.....</i>	<i>46</i>

List of Tables

Table 1: Advantages of local systems vs advantages of cloud systems 8

Table 2: Format of the audios of the different probands 46

Table 3: Equations to obtain genral statistics..... 47

Table 4: Equations to obtain statistics for each command..... 47

List of Abbreviations

STT	Speech To Text
TTS	Text To Speech
JSON	JavaScript Object Notation
UK	United Kingdom
NHS	National Health Service
US	United States
IoT	Internet Of Things
CEO	Chief Executive Officer
ASR	Automatic Speech Recognition
NLU	Natural Language Understanding („Text To Meaning“)
MIT	Massachusetts Institute of Technology
GPL	General Public License
OSGI	Open Services Gateway Initiative
LED	Light Emitting Diode
PS	Play Station
URL	Uniform Resource Locator

Appendix A: Rhasspy sentence configuration

```
[GetTime]
what time is it
tell me the time

[GetTemperature]
whats the temperature
how (hot | cold) is it

[PercentageControl]
device_name = ((light | lights) {name})
blind_name = ((blind | blinds | shutters) {name})
music_name = ((music | stereo | sound system | yamaha) {name})
place_name = ((kitchen | living room | all | house | home | room | bedroom ) {
name})
color_state = (green | red | blue | yellow | orange| white | pink | purple) {c
olor}
source_state = (radio | tuner | usb | net radio | auxiliar | aux | bluetooth |
cd) {source}
blind_state = (up | down) {state}

turn on [the] <place_name> <device_name> [to] <color_state> [color]
turn on [the] <place_name> <device_name> [in] <color_state> [color]
put [the] <place_name> <device_name> [to] <color_state>
set <place_name> <device_name> [to] <color_state>
adjust <place_name> <device_name> [to] <color_state>
regulate <place_name> <device_name> [to] <color_state>
change <place_name> <device_name> [to] <color_state>
adjust <color_state> <place_name> <device_name>
set [to] <color_state> <place_name> <device_name>
adjust [to] <color_state> <place_name> <device_name>
regulate [to] <color_state> <place_name> <device_name>
change [to] <color_state> <place_name> <device_name>
tune <place_name> <device_name> [to] (0..100) [percent]
regulate <place_name> <device_name> [to] (0..100) [percent]
adjust to (0..100) percent <place_name> <device_name>
tune to (0..100) percent <place_name> <device_name>
regulate to (0..100) percent <place_name> <device_name>
adjust <place_name> <device_name> to (0..100) [percent]
tune <place_name> <device_name> to (0..100) [percent]
regulate <place_name> <device_name> to (0..100) [percent]
tune <place_name> <blind_name> [to] (0..100) [percent]
regulate <place_name> <blind_name> [to] (0..100) [percent]
adjust to (0..100) percent <place_name> <blind_name>
tune to (0..100) percent <place_name> <blind_name>
regulate to (0..100) percent <place_name> <blind_name>
```

```

adjust <place_name> <blind_name> to (0..100) [percent]
tune <place_name> <blind_name> to (0..100) [percent]
regulate <place_name> <blind_name> to (0..100) [percent]
roll <blind_state> [the] <place_name> <blind_name>
set <blind_state> [the] <place_name> <blind_name>
draw <blind_state> [the] <place_name> <blind_name>
pull <blind_state> [the] <place_name> <blind_name>
raise [the] <place_name> <blind_name>
lower [the] <place_name> <blind_name>
roll <blind_state> [the] <blind_name> [of the] <place_name>
set <blind_state> [the] <blind_name> [of the] <place_name>
draw <blind_state> [the] <blind_name> [of the] <place_name>
pull <blind_state> [the] <blind_name> [of the] <place_name>
raise [the] <blind_name> [of the] <place_name>
lower [the] <blind_name> [of the] <place_name>
change to <source_state> mode [the] <music_name>
adjust to <source_state> mode [the] <music_name>
set to <source_state> mode [the] <music_name>
change to <source_state> function [the] <music_name>
adjust to <source_state> function [the] <music_name>
set to <source_state> function [the] <music_name>
put [the] <music_name> in <source_state> mode
change [the] <music_name> to <source_state> mode
adjust [the] <music_name> to <source_state> mode
set [the] <music_name> to <source_state> mode
put [the] <music_name> in <source_state> function
change [the] <music_name> to <source_state> function
adjust [the] <music_name> to <source_state> function
set [the] <music_name> to <source_state> function
change [the] <music_name> volume to (0..100) [percent]
adjust [the] <music_name> volume to (0..100) [percent]
set [the] <music_name> volume to (0..100) [percent]
regulate to (0..100) [percent] [the] <music_name> volume
change to (0..100) [percent] [the] <music_name> volume
adjust to (0..100) [percent] [the] <music_name> volume
set to (0..100) [percent] [the] <music_name> volume
regulate to (0..100) [percent] [the] <music_name> volume
change [the] <music_name> to scene (0..10)
adjust [the] <music_name> to scene (0..10)
set [the] <music_name> to scene (0..10)

```

```
[ChangeState]
```

```

light_name = ((light | lights) {name})
space_name = ((ceiling | center | table | work) {name})
thing_name = ((music | stereo | yamaha | sound system | hob | oven | radio) {name})

```

```

place_name = ((kitchen | living room | all | house | home | room | bedroom ) {
name})
the_state = (on | off) {state}

turn <the_state> [the] <place_name> <light_name>
turn [the] <place_name> <the_state>
switch <the_state> [the] <place_name> <light_name>
switch [the] <place_name> <light_name> <the_state>
turn <the_state> [the] <thing_name>
turn [the] <thing_name> <the_state>
switch <the_state> [the] <thing_name>
switch [the] <thing_name> <the_state>
turn <the_state> [the] <space_name> <place_name> <light_name>
turn [the] <space_name> <place_name> <light_name> <the_state>
switch <the_state> [the] <space_name> <place_name> <light_name>
switch [the] <space_name> <place_name> <light_name> <the_state>
turn <the_state> [the] <place_name> <space_name> <light_name>
turn [the] <place_name> <space_name> <light_name> <the_state>
switch <the_state> [the] <place_name> <space_name> <light_name>
switch [the] <place_name> <space_name> <light_name> <the_state>
turn <the_state> [the] <light_name> [of the] <place_name>
turn [the] <light_name> [of the] <place_name> <the_state>
switch <the_state> [the] <light_name> [of the] <place_name>
switch [the] <light_name> [of the] <place_name> <the_state>

[MusicControl]
music_name = ((music | stereo | sound system | device | song | equipment | rad
io) {name})

play [the] <music_name>
pause [the] <music_name>
next <music_name>
previous <music_name>
mute [the] <music_name>
mute [the] volume
mute off [the] <music_name>
active [the] sound [again]
active [the] volume [again]

```

Appendix B: List of extra commands

Person	Extra 1:	Extra 2:
1	open garage door	tell me the most important news
2	make a song play of estopa	lower volume to 50 percent
3	what is pamplona city what will be the lottery winning number?	what is club atletico osasuna?
4	play pop music	how is the traffic on my way to work?
5	what day is today?	do a phone call to mum
6	send a message to Peter	what month are we in?
7	check my email	regulate house temperature to 20 degrees
8	tell me if I have messages in my email	how cold is it?
9	make me laugh	tell me the news of today
10	remind me to drink water in an houer	i want lo learn something
11	add oranges to shopping list	tell a fairy tale
12	play top trending music	regulate heating to 23 degrees
13	roll a dice	place a remaining of buying milk
14	play highway to hell	who is going to win this night, osasuna or real madrid?
15	roll a dice	tell me a joke
16	how tall is Everest?	what is the movie 1917 about
17	could you search on internet for sesma	stock price off facebook
18	tell me the news	where is Paris?
19	spell carlos	can you sing
20		say "I love you"

