# Analysis of voice assistants in eHealth

**Thesis** · July 2019

DOI: 10.13140/RG.2.2.14691.37924



**3 authors:**

Mathias Wolfgang Jesse
Alpen-Adria-Universität Klagenfurt
**6** PUBLICATIONS **8** CITATIONS
SEE PROFILE

Claudia Steinberger
Alpen-Adria-Universität Klagenfurt
**54** PUBLICATIONS **235** CITATIONS
SEE PROFILE

Peter Schartner
Alpen-Adria-Universität Klagenfurt
**90** PUBLICATIONS **407** CITATIONS
SEE PROFILE

Mathias Wolfgang Jesse

# Analysis of voice assistants in eHealth

MASTER THESIS

submitted in fulfilment of the requirements for the degree of

Master of Science

Programme: Master's programme Information Management

Alpen-Adria-Universität Klagenfurt

**Co-Supervisor**
Mag. Dr. Claudia Steinberger
Alpen-Adria-Universität Klagenfurt
Institut für Angewandte Informatik

**Evaluator**
Assoc. Prof. Dipl.-Ing. Dr. Peter Schartner
Alpen-Adria-Universität Klagenfurt
Institut für Angewandte Informatik

Klagenfurt, July 2019

# Affidavit

I hereby declare in lieu of an oath that

- the submitted academic paper is entirely my own work and that no auxiliary materials have been used other than those indicated,
- I have fully disclosed all assistance received from third parties during the process of writing the paper, including any significant advice from supervisors,
- any contents taken from the works of third parties or my own works that have been included either literally or in spirit have been appropriately marked and the respective source of the information has been clearly identified with precise bibliographical references (e.g. in footnotes),
- to date, I have not submitted this paper to an examining authority either in Austria or abroad and that
- when passing on copies of the academic thesis (e.g. in bound, printed or digital form), I will ensure that each copy is fully consistent with the submitted digital version.

I understand that the digital version of the academic thesis submitted will be used for the purpose of conducting a plagiarism assessment.

I am aware that a declaration contrary to the facts will have legal consequences.

Mathias Wolfgang Jesse m.p. Klagenfurt, 29.07.2019

# Acknowledgements

Throughout the writing of this thesis I have received a great deal of support and assistance. I would first like to thank my two supervisors, Assoc. Prof. Dipl.-Ing. Dr. Peter Schartner and Mag. Dr. Claudia Steinberger, whose expertise was invaluable in the formulating of the research topic and methodology in particular. Their door was always open whenever I ran into a trouble spot or had a question about my research or writing.

In addition, I must express my very profound gratitude to my parents, my siblings, and to my partner for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. They are always there for me and this accomplishment would not have been possible without them.

Finally, there are my friends, who were of great support in deliberating over our problems and findings, as well as providing happy distraction to rest my mind outside of my research.

# Abstract

A voice assistant is software that can analyse and understand what humans say. To do so, it must interpret the sounds made and semantically evaluate the meaning of the now-understood phrase. After a corresponding answer has been generated, it can respond with a fitting response, which can be textually, vocally, or both. Therefore, a voice assistant can be used as a form of interaction and control for different tools. Thus, it is the most natural method of interaction we currently know. For this reason, this master thesis concentrates on the usage of voice assistants in the field of eHealth.

Accordingly, the project 'MyELGA' connects with the existing electronic health records (ELGA). The general idea is that everybody should have access to their medical information when and where they need it. The application 'e-Medikation' holds onto all information regarding a patient's medication and transmits it between different health service providers (e.g., hospitals or doctors). Additionally, there is no need for the patient to handle the paperwork. This project has the advantage of being a central point of information and can be used by everybody with access. MyELGA adds a step to this concept with a voice assistant. Asking for information with the application is integrated into the everyday routines of users, reducing the barriers to use in patients' daily lives, which they would otherwise experience. The use of a website or other generic virtual assistants can be especially more complicated.

As MyELGA is already realized with the voice assistant Alexa, it is used in this study to demonstrate how the application could be realized with a better assistant. To understand the most fitting for use in MyELGA a set of criteria has been designed to answer crucial questions regarding ease of use, security, and extendibility. The criteria are divided into either general or eHealth specific features. A special focus placed on security aspects of the different assistants, is a necessity because of the nature of personal data. Having private medical information leaked or simply exposed to the public is not in the interest of a patient. Therefore, not only are the criteria examined, but also current state of each product. As there are so many products offered, it is necessary to further narrow the observation.

In the end, an explanation of the benefits of all assistants is presented. MyELGA can then be implemented using the voice assistant that best meets the necessary criteria for the project. The result encompasses the main functionality of MyELGA which can be used to evaluate the criteria and ensure the promised characteristics align with reality.

# Table of Contents

# 1 Introduction

Conversing with a computer has always been a futuristic idea, thought to only be possible in science fiction films. However, recent changes in technology allowed for the creation of voice interfaces that are actually able to make this dream come true. Devices that facilitate this approach are called voice assistants [HoyM18, p. 81-83]. By developing an application that runs on such an environment one can harness the benefits of this natural way of interacting. Many people are not capable of using other types of interaction (e.g., the elderly) or simply want to use a hands-free form of communication with their devices.

Such people have special needs, which must be treated accordingly. One such need is active participation in the care-taking process. Patients need the means to stay more autonomous, informed and self-reliant when it comes to their health. In Austria the electronic health record (ELGA) is used to provide users with ways of controlling their own medical information. Such an approach can be classified in the field of eHealth. Through methods like digitalization and centralization, tasks revolving around health care are improved and made available for the vast majority [ELGA19e] [Ande16, p. 25-26]. Here, the MyELGA application features. MyELGA combines the benefits of a voice interface with the possibility of controlling one's medical information through ELGA. A previous work by Klade [Klad19] created a prototype using Amazon's Alexa as the base voice assistant. After building a functional assistant doubt arouse if Alexa was the best solution for a voice assistant. The thesis and prototype created by Klade form one of the cornerstones of this work.

The goal of this thesis is to analyse the current state of voice assistants in the eHealth sector. Therefore, the Alexa in MyELGA is used as one of the foundations for determining the criteria of comparison. Furthermore, a target group is defined, manifesting further aspects and demonstrating how to interpret the uncovered solutions. Security is also an important aspect that must be considered when working with personal data. The scope of this thesis only observes the security standards created or imposed by the voice assistant. Regulations and security standards offered by the applications and infrastructure surrounding the voice assistant must be upheld, not mitigated. These criteria are then used to find the best voice assistant at this time. As there are always certain criteria more important for a variety of applications, an emphasis is placed to the needs of the MyELGA project. The next step is to create a prototype that demonstrates the first steps for implementing a voice assistant that works in a similar fashion to the existing MyELGA prototype. Evaluations of this prototype are used to reinforce the findings and show relevance for future works.

The thesis is structured as follows. The beginning of Chapter 2 provides a general introduction to voice assistants. This introduction features the basic concepts of these assistants and how

they compare to other virtual assistants. Furthermore, the benefits they provide to society and especially the elderly is observed. To complete this introduction, the topic of security is covered, including a short excursion on the General Data Protection Law.

In Chapter 3, the focus is turned to ELGA, explaining how it is structured and devoting some attention to the security aspects of the ELGA infrastructure, such as how these risks are managed. The view shifts from the underlying ELGA application to the MyELGA project. With this project as the focal point, the architecture and performance are documented. Future desired functionality is extracted from the previous work of Klade [Klad19] and discussed at the end.

As all necessary information is provided beforehand, a target group can be designed and Chapter 4 can present a criteria catalogue based on said target group, the MyELGA application, and security measures. These criteria are grouped under these three characteristics and presented with a short description, including how they are evaluated. This chapter closely accompanies the next one, Chapter 5, which uses the criteria catalogue as a basis to compare relevant voice assistants. To do so, actual statistics, trends, and other information are observed to make sound assumptions for the act of choosing actual assistants. With the voice assistants selected and the criteria evaluated, a forerunner can be found and used for the next chapters.

Chapter 6 uses the best assistant from the criteria catalogue and demonstrates how an implementation of the MyELGA application would look. Different scenarios are used to depict the process and evaluate relevancy for future works. The fulfilment of the most important criteria is re-evaluated with the actual prototype to show if the promised characteristics align with reality.

Finally, Chapter 7 collects the gathered findings and discusses if the found assistant is feasible. Furthermore, an outlook for future implementations and other considerations are provided.

# 2 Introduction to the concept of voice assistants

This chapter gives an overview of what a voice assistant is and how it functions. Most voice assistants consist of the same natural language components, which is why these will be discussed briefly. Furthermore, the difference between voice assistants and virtual assistant is discussed, in order to elucidate the circumstances under which an assistant with the capabilities of understanding and producing voice is to be preferred. Afterwards, the user group of the elderly is considered. Aside from their needs and problems, this part of the paper addresses how elderly people benefit from using voice assistants and what is necessary to improve adoption. Finally, an overview of security standards and related issues is given.

## 2.1 Voice assistants in general

Every consumer is able to talk to what is known as a voice assistant. Voice assistants are software agents that can be integrated in computers, mobile phones, smart TVs and other platforms. They are able to understand commands issued by voice and to work on tasks autonomously. It does not matter if it is a simple question or a more complicated task, like playing music, the voice assistant helps accomplish it to the best of its capabilities. To do so, a voice assistant usually needs a microphone, GPS-data and access to the internet. Additionally, most assistants use artificial intelligence technology in order to learn constantly. This way devices are able to memorize the needs and practices of their users and create a personalized experience [HoyM18, p. 81-83] [Conr17, p. 740-741].

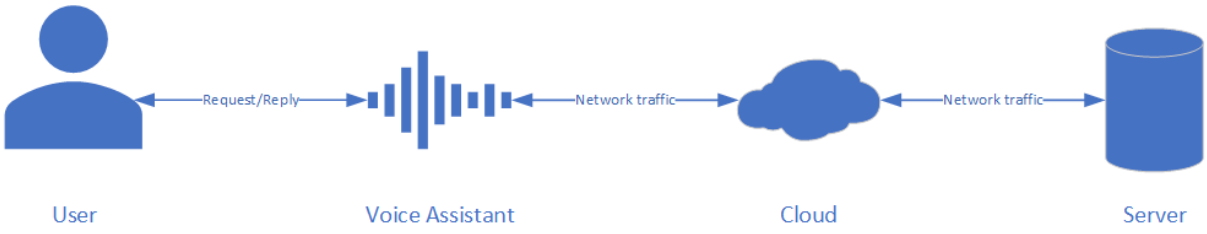


**Fig. 2.1:** Generalized process of a voice assistant

Concretely, the voice assistant constantly listens for a key word. If said prompt is recognized, the following utterance will be recorded. The voice assistant sends an understood query to a cloud which then uses natural language technology on the text and finds the corresponding application which can be located on a cloud or on a server. The voice assistant's reaction depends on its programming and on the type of task understood. In most cases, the application creates a reply in text form. The natural language components of the voice assistant transform the text into a

voice sample and play it in response to the users aloud.
Despite their name, voice assistants they not restricted to auditive input. They can interpret written commands and texts, usually in the form of online chats (chatbot) or direct input via a touchscreen, as well. In addition, some assistants are also able to comprehend pictures, which are uploaded or directly taken via a camera. The same applies to the data returned to the user which can either contain information in form of text, images, and music or initiate the start of other applications.
These functionalities were not there from the start and developed over time. They evolved from preceding models that, in comparison, had significantly fewer features. The first voice-activated systems were more restricted in the range of commands they could understand and did not have a connection to the internet. Newer assistants have a much better understanding of what a user is saying. This is facilitated by the server and services used by nowadays voice assistant providers. An in-depth analysis of those will be presented later on in Chapter 5. Another factor that improved the abilities of voice assistants is the advancements in the field of natural language processing. Current computers and mobile devices are much more powerful and, at the same time, cheaper than before, which also adds to the possibilities of assistants [HoyM18, p. 82-83] [SSDK18, p. 1641] [LRKR$^+$18, p. 3].

Voice assistants appear in many shapes, but a trend that evolved in recent years are smart speaker-based voice assistants. These are devices which are placed at home instead of being carried around. They consist of a base, which includes microphones and speakers. When installed and configured, the assistant can assist its user with basic tasks. As such it can answer questions, play radio, write messages and notes, call people, and depending on the provider, complete some other tasks. If required, it is possible to install additional software that helps with different activities like manoeuvring through a city or ordering food at a local restaurant [Blas18, p. 42-43].

### 2.1.1 Natural language

Due to the ongoing process of digitalization and a rather fast development of technology, everyday lives are changing very quickly. Thus, many people are not yet capable of using input devices like smart phones or computers. This is where voice assistants come into play. With their help, a person can use their mother tongue to control a voice-driven system [JuMa14, p. 422] [ReBB19, p. 77].

Considering the structure of natural language in voice assistants there are recurring elements that build this component. One term that is used for these elements as a whole is 'dialogue system'. Different companies pursue different approaches with their voice-driven assistants, like Amazon's Alexa, Google's Assistant or Apple's Siri, each concentrate on different tasks. While each company has their unique assistant, there are many interaction models depicting the general structure of voice assistants. For further explanations the model of Bellegarda [Bell13] is being referred to [KBo18, p. 99-100].

#### Speech recognition

The first step, called speech recognition or automatic speech recognition (ASR), is activated when the users start their request with a wake word. After processing that the wake word was issued, the following prompts are recorded and transcribed into actual words. These transcrip-

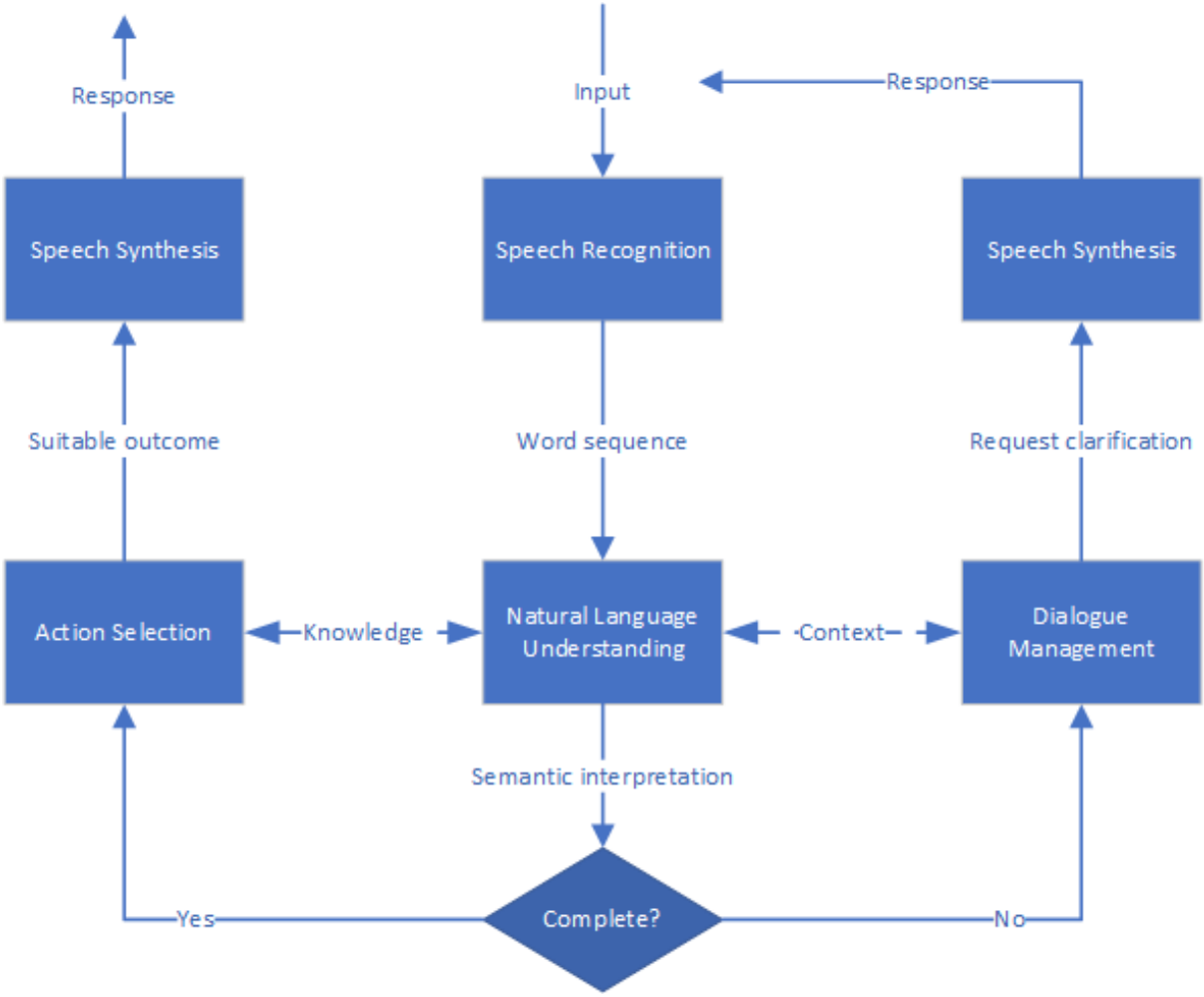


**Fig. 2.2:** Dialogue system [Bell13]

tions are used by later components [Bell13, p. 2029]. At first, these requests are only mere digital signals that are carrying the information which then have to be extracted.[HuAH01, p. 201] In the beginning, this was done by using acoustic feature-modelling but this was later replaced by deep-learning models. To achieve the best possible performance these models, have to be trained with thousands of hours of acoustic data. Afterwards, these models directly map the found data to actual human sounds [ACJB+14, p. 1-2].

Depending on the particular voice assistant, the actions triggered by speech recognition can differ. Because of the continuous advancements in this area of technology, it is now possible to identify the person speaking and to verify if whether they are authorized. To increase the accuracy of speech recognition, the system is trained with voice samples of a particular person. This, however, is not always of advantage. Sometimes applications are meant to be used by a variety of people and do not need this feature. Not only can the assistant identify the person talking, but it can also recognize the language spoken. With all this information search operations can be conducted on the transcribed words. Usually this process is necessary when searching for the wake word or other words with special meaning [ReBB19, p. 78-81].

### Natural language understanding

After sounds have been translated into actual words, these are linked to their meaning and understood. There are many things that need to be considered in regard to understanding single words. This has especially to do with the fact that making an open-domain system that is able to understand everything is likely to be prone to inaccuracy. Ambiguity is one of the problems arising. Words might be used in contrasting scenarios in which they have completely different meanings. Therefore, a domain is defined to reduce such ambiguity. Also, variability and the context are hard to figure out, when no restrictions are given. Thus, making a closed-domain system is far more reliable and accurate [HiMa15, p. 261-262].

When actual words are understood correctly, the next step is to semantically interpret the sentence. This is due to the fact that, at this stage in the process, the system is unaware of the utterance's meaning. There is often no direct connection between each sequence of words and the meaning [HuAH01, p. 835]. To understand a sentence, first and foremost the domain has to be classified. The assistant needs to know what the circumstances of this request are. These could be a hotel, an alarm or anything else.
The second task is to identify the intention of the users. Therefore, the device needs to figure out what their goals are. To establish this association, context and action have to be considered. For example, this could be making a reservation for a hotel room or setting an alarm.
In the end, the act of 'slot filling' is carried out. Its idea is to interpret what a user might want to tell the system without explicitly stating it. As an example, a user wanting to set an alarm might provide an input that could look like: 'Wake me tomorrow at 6'. Slot filling is necessary to understand that, although it is not directly stated, an alarm has to be set for the next day at 6 in the morning [JuMa14, p. 434].

### Dialogue management

Now that the system is sure what its users want, it needs to be able to conduct a full conversation. The purpose of a dialogue manager is to track where in the conversation a user and system are. This does not only involve the conversation's state but also the next assistant's actions. Jurafsky and Martin calls these two parts the 'dialogue state tracker' and the 'dialogue policy'. All necessary possibilities of how the conversation can go, should be covered by the manager. An optimal scenario would be that no situation exists, were the system does not know how to react or respond [JuMa14, p 446-454] [Bell13].

When observing most discussions, voice assistants need more than one prompt to completely understand their users. That is why follow-up queries are needed. Even if a sentence is unambiguous it might be incomplete, inconsistent or imprecise and the system needs to actively help with completing and correcting the query. Otherwise, it will not know what the actual task at hand is about. Doing so not only helps to understand the task, but also improves the efficiency of users progressing. For example, if the user provides sufficient information about a task the dialogue should progress and answer with further instructions to continue the conversation. If insufficient information was given the device's response needs to indicate this and help to acquire the needed input. This is usually done by asking to repeat the question or by providing additional data [HuAH01, p. 867-868].

#### Speech synthesis

Once all information is acquired and the sentences are understood, the gathered knowledge is used to infer what action has to be initiated. Since the users need some form of feedback, the result is then presented to them verbally. This process consists of two steps. The first one is to translate written into speakable text and the second is to actually generate phrases that sound as close to human speech as possible. Both steps in itself are complicated enough and a text-to-speech system that conveys text like a real human has yet to be invented. This means that Turing tests have not been completed with an optimal result [Bell13, p. 2029] [HuAH01, p.679-688]. This is due to the problems that arise from the human way of having conversations. Humans recognize subtle hints in the way things are said or what can be gathered by one's body language. A basic text-to-speech system is not capable of producing such kind of information and gesture. Therefore, it feels unsatisfying to talk to it [HiMa15, p. 263].

### 2.1.2 Comparing virtual assistants to voice assistants

As Huang [HuAH01] mentions, without regard to the actual type, first and foremost an interface has to work. There is no use in improving a form of interaction if it is not functioning properly. But not only must it be able to fulfil the task, it has to benefit the productivity of the assistant's owner in doing so. If an interface breaks the workflow or is more time-consuming than an alternative, people will not use it. Thus, it should eventually be tailored to user's need to maximize the satisfaction [HuAH01, p. 911]. This means the goals of a virtual and a voice assistant are the same, since the only difference is the way interaction is offered to the user. As voice assistants focus on the usage of voice, the way they are situated differs from any non-voice centred approach. Usually when a mistake happens in a generic graphical user interface (GUI) it is thought of as an error caused by the human using it. Where voice input is concerned, the problem that every time a mistake happens, it is thought of as a system error. To prevent such situations a lot of development and time has to be invested in effectively managing how to act when an error occurs. The strengths of using voice as an input, corresponds to the weakness of using pen to interact and vice versa. For example, to use a pen it is necessary to have a visual representation which voice does not need. However, speech is greatly impacted by the surrounding and background noises, whereas pen or touch are not [HuAH01, p. 911-914].

Situations in which a free-handed interaction is necessary or beneficial are where the strengths of voice interfaces lie. A fitting example is driving a car. As both hands are supposed to be on the steering wheel, having the ability to control a device by voice is experienced positively. The user's focus can remain on the traffic which minimizes the likelihood of an accident. Another example, that is also well-known, the use of small screen sizes that cause difficulties to the elderly generation or visually impaired people. As digitalization is evolving faster and touch is not that intuitive, having to use nothing but one's own voice to finish tasks is inherently more natural. Typing or pointing is no longer necessary, and the use of voice can ease certain tasks [Bedf17, p. 488]. Additionally, the use of machine learning allows a more precise addressing of the users. Their usage of a voice interface can be analysed with the help of tools to give even better recommendations and improve usability. Also, there is no limitation in terms of time and space. It is possible to use a voice assistant on the go and in a big variety of situations. However, using speech, as a form of input, has its downsides too. As already mentioned, an internet connection has to be established. Otherwise, the assistant can often be rendered useless.

When looking at what companies offer with their assistants, they usually only go for proprietary devices. Therefore, a customer is limited in the supported options. Data-security is another potential drawback which will be explained in the designated chapter 2.3. Another issue that often arises is the constant recording of voice and its possible infringement on privacy. Because of the novelty of voice assistants people are yet to understand and actively use this form of interaction.[KrWo18, p. 277-278]

Hirschberg [HiMa15] defined additional problems, which only occur when speech synthesis is used. Creating an answer is a difficult task on its own, but there are many nuances which have to be kept in mind. For example, a certain delay when answering is desirable but timing it is a significant problem. Anything unnatural is considered burdensome and lowers the overall experience concerning the conversation. Words that are filling pauses, like 'uh' or 'um', are needed as they make the conversation more natural. Hence, a lot of thought has to be put into how an answer is conceived rather than simply writing it on screen.[HiMa15, p. 263]

After having explained the benefits and drawbacks of voice assistants, an evaluation of the synergies of multiple forms of interaction will follow. Using only one mode to represent both input and output, is called unimodal. In contrast, if speech is combined with other modalities it is recognized as multimodal. These other types can be gesture, pen, touch, gaze and head, and body movement. These different options enable developers to utilize each option to their fullest and benefits the overall usability of a service. For example, the Ford Model U uses both touch and voice as a form of input. Similar vehicles focus on using voice-only, command-and-control interfaces. There are other projects, like Semio, which combine body-gesture and voice, or a neural network, which allows the integration of audio, visuals and motion. Studies show that people are not fond of the idea of using voice and touch simultaneously. However, they do like to use each interface for a different task. This could be entering data by voice and correcting errors by hand. For contrasting activities, a multimodal approach is most suitable. [KBo18, p. 99-100] [HuAH01, p. 913-914]

## 2.2 Focus on the elderly

With the ongoing demographic changes the average lifespan of humans increases and with it the number of elderly people in the population. By 2030, the estimated lifespan of a West European will be around 85 years by 2030. Due to the decline in the birthrate, more and more people are going to need caretakers and access to health services. This creates scenarios in which elderly people which are still residing on their own have the desire to live autonomously and self-controlled. These people are informed and connected which means an individual approach incorporating these characteristics is needed to meet all the needs of this generation. Although this trend does not appear grave yet, it could evolve into a much bigger issue. Many positions in health care are currently unoccupied and underpaid considering the toll it has physically and mentally on the caretakers. At the same time there is going to be a rise in the number of people that have age related diseases. Expenses in health care will increase and a further shortage of caregivers is the result. This means if there will be no appropriate response to these changes the impact on society and life of elderly people will be severe. Another issue is the difference between public and private health care. Currently each has a task of its own purpose. Public institutions have to be prepared to take in emergencies and care for those that already are

sick. In comparison, private clinics can focus on more lucrative treatments. The competition between these two institutions could lead to a downfall of quality in the public healthcare sector. [Ande16, p. 25] [HDKB+14, p. 26] [RaMi13, p. 579]
In order to help patients to stay autonomous, they need to be given more control over their health care. This is done by integrating people into the whole process of prevention and taking care of their health. Having easy access to their records and the medication that is being prescribed, is one of the first steps towards a more transparent and thought-out approach. Electronic Health (eHealth) and Ambient Assisted Living (AAL) are two possible ways to provide patients a way of controlling their health data and assistance in everyday life. [Ande16, p. 26] [HDKB+14, p. 26]

### 2.2.1 Assisting the elderly

Although there are many definitions of eHealth, there is one that is being widely used and will be referred to in this paper.

> "eHealth is an emerging field in the intersection of medical informatics, public health and business, referring to health services and information delivered or enhanced through the Internet and related technologies. In a broader sense, the term characterizes not only a technical development, but also a state-of-mind, a way of thinking, an attitude, and a commitment for networked, global thinking, to improve health care locally, regionally, and worldwide by using information and communication technology". [Eyse01]

Therefore, eHealth as a term sums up many concepts surrounding the current care-taking process. Because of digitalization and the connection of health-related fields, e.g. medicine, health care, sport and diet, it has been become possible to improve and organize ways in sustaining people and their health. Furthermore, it encompasses institutions that focus on health-care and working on a more centralized view on patients and possible improvements [Enge19, p. 1-4] [Ande16, p. 27].
The core features, introduced by eHealth, benefit the health care system. These features, as stated by Andelfinger [Ande16, p. 25-26], are helping in various ways. First of all, they support the prevention of illnesses. Next, they centralize electronic health data created by all instances working with the patient and thus reducing duplicate examinations. This information is saved into every patient's electronic health record. Moreover, the centralization of information helps to improve the overall quality of medication. This is accomplished by managing the corresponding data faster and making it more reliable at the same time. Overall this helps to reduce costs in the processes surrounding health care. With people being more autonomous and taking care of themselves, caretakers are relieved of some of their workload and thus are able to take care of more patients with an even higher degree of quality. [Ande16, p. 27-28]

Another approach to assisting elderly people is AAL. As the name suggests, Ambient Assisted Living is a concept that provides situational and unobtrusive support in the patient's home. In order to keep up with the elderlies' rising tendency of staying in their home while at the same time trying to reduce costs caused by private caretakers, one solution is to empower people at home with the use of ambient intelligence. In this case, sensors and machine-to-machine technology are installed in the patients' surroundings. This establishes a safer environment for the elderly and thus enhances the quality and duration of the users' lives. In comparison to eHealth, AAL

differs as it focuses more on the integration of devices into everyday life. These offer different services that ensure the patient to stay autonomous and independent, while still receiving the needed attention and care. In addition, the devices can record patient data and increase the quality of treatment. For example, an AAL solution could be a reminder to take medication, a video surveillance for assistance or an emergency application to notify caretakers in case of falling. Also, the tracking of daily activities is used to help with tasks and to manage social interactions with family, friends and caregivers [HDKB+14, p. 26] [HVNC09, p. 1-2] [RaMi13, p. 579].
The ways these devices can be integrated into the life of patients vary. One typical solution is a Smart Home. This is a home which is set up with different sensors and actuators. With access to the gathered data a system can perform tasks and make living more comfortable. Another solution is to integrate intelligence in wearable or mobile devices. These help with monitoring the mobility and location of a user. As an addition, it is possible to track blood pressure, blood glucose or other health data via sensors integrated into wearables. These use technology akin to infrared sensors, optical sensing and oscillometric. The already mentioned integrations are more passive than the use of robotics. Robots find their purpose in scenarios where the patient needs assistance in physical form. Typically, such situations are centred around self-maintenance. Feeding, grooming and dressing are just a few to give a general idea. [RaMi13, p. 579-581] [HVNC09, 3-4]

For eHealth and AAL to work properly in our society, a general understanding of the kind of information that can be documented on people has to be established. Health care was initially recorded in written form, but these notes have evolved into a connected and digital medium. Today, the way we create such records on patients differs. The question of what kind of data is collected arises. This can differ between eHealth applications. Regardless, it should be in line with the judicial, medical and economical parameters. Even further, a patient should always be the focal point of such applications. Either through decreasing information asymmetries', creating transparency, increasing quality, or reducing costs. The patient should never be at loss because of such applications. Another problem with eHealth, AAL and other systems is the level of trust and willingness the patient has in regard to the application. In order for them to unfold their full potential, users must want to use these systems. But trust is not the only important thing to keep in mind. The frustration occurring when using such a system has to be acknowledged. Be it psychologically or technologically, the overall experience of using a system intending to help is impaired either way. Such frustration can occur because people are losing their strength and autonomy as they grow older, which is the psychological frustration. On the other hand, the operation of user-unfriendly interfaces can lead to another kind of frustration which the user is not at fault for. Thinking even further, as such applications are very dynamic, it can occur that they are wrongly mapped or do not fulfil the tasks they are designed for, which only adds to the frustration and uneasiness of users and patients [Ande16, p. 28-29] [Enge19, p. 7-8] [HVNC09, 3-4].

### 2.2.2 Application examples of voice assistants

The way voice assistants are employed, changes drastically, based on activities, environment and brand. Although most assistants encompass the same basic functionality, each has its own selling propositions and offers unique features. The basic functionality ranges from searching

the web over playing media to setting reminders and alarms. [HoyM18, p. 84-85] The highest requested task issued by users is to ask a question. To be exact, a research done by Microsoft and Bing [OlKe, p. 20-21] found that the most frequently used 'productivity' tasks were asking for quick facts, listening to music, checking directions and getting information about weather and news. In comparison, a different study by Voicebot.ai [KiMu19, p. 15-16] came to a similar result. Asking a question, listening to a streaming service and checking the weather were their top three results. The different results can be seen in the latter found tasks, as setting an alarm and a timer can be found in this study.
When focusing on smart home management, it is apparent that more than half of the owners of assistants (around 54%) manage their home with their help. Especially smart TVs and smart lights are popular. These are followed by game consoles or smart media controllers, which are controlled by voice. Lastly, thermostats and video doorbells are slowly appearing in smart homes as well. The most frequent tasks used in the field of smart homes are playing music and to changing the lighting settings [KiMu19, p. 17-19] [OlKe, p. 20].
Although the already mentioned categories make up the highest usage, the focal point in this thesis is set on 'Use a favourite Alexa skill/Google Action'. In this field we can find most applications which do not fit into the scheme and still make up about 18,3% of daily use. As such, this is the category in which eHealth or AAL skills are located. On the topic of health, questions about 'health and wellness' amount to 16% of all queries used to search with the help of voice assistants [KiMu19, p.16-18].

When observing the current state, eHealth is already available on smart speakers and the like. The project HealthPal is an example of a fusion between eHealth and voice assistants. Being a mobile health monitoring device that aims to assist elderly people in assessing their medical data without them having to worry about IT related skills. This is emphasized through the introduction of a voice interface. System, voice assistant and medical data are all combined in a PDA [KoSt19, p. 1]. Another project called CADENCE focuses on helping people with dementia. The assistant works as an encouraging guide that assists in everyday tasks. Such systems are called intelligent cognitive assistants (ICA). This does not limit them to only helping people with cognitive impairment but in this specific case they also assist people without impairment. The basic idea is to boost the users' capabilities with the use of the system [WoKK15, p. 1] [SSDK18, p. 1641].

## 2.3 Security and privacy issues with voice assistants

The possible threats for the users of voice assistants are numerous and diverse. Having a voice interface and being connected to the internet poses a unique problem. Anybody who has access to the assistant, be it physical or digital, has the power to gather information about the owner. That could either be personal data saved in calendars, emails or other connected devices (e.g. smart home). A lack of authentication is the main concern actions have to be taken against [HoyM18, p. 84-85].

Additionally, these devices have the capability of recording anything at all times. Although the assistants are supposed to only transmit data containing the wake word, people are concerned. To counter this distrust, companies make the process as transparent as possible [Cand17]. That

being said, each assistant has its own strengths and weaknesses, but these will be discussed in more detail 5 later on. The next section discusses risks all voice assistants have in common.

### 2.3.1 Risks of using voice assistants

Voice assistants, smart TVs, webcams and other smart home gadgets are considered Internet of Things (IoT) devices. With the trend continuing, by 2020 there are going to be around 12.2 billion devices connected to the internet. However, these are prone to cyberattacks because of their poor security measures. It has been shown that by observing the data flow and replaying certain packets it is possible to monitor and change the status of lightbulbs and power switches. This poses a threat to personal safety if such data is compromised and used against the owner of a smart home. Furthermore, there have been recorded DDoS attacks with the exploitation of vulnerable IoT devices [AFHV17].

What is even worse for voice assistants is that there are attacks utilizing the voice interface. As it is the main form of interaction with voice assistants, the easiest way to attack such a device is by voice. Simply speaking, any person that has direct contact with the assistant is capable of issuing commands. If the device is able to distinguish the different people that are talking, it can use unique permissions for each person. However, the accuracy of matching a voice to its owner is not perfect yet. As such, it is necessary to be cautious when other people are unsupervised with a voice assistant. These can be either complete strangers, friends, or even just kids. Especially with the ease of ordering items via an assistant, children could order almost anything they see on television. Besides using a PIN code there is currently no viable protection against a person that is in hearing-range of the device. [Cand17, p. 12-14] [HSWW17, p. 5-7]

Televisions also pose a certain threat. It is possible that there is an advertisement or show that essentially asks the voice assistant to add products to the shopping cart or to turn off certain devices. The same concept applies to radio shows. Particularly listeners of analogue radios are at risk, because broadcasting your own programme on popular frequencies is possible. Another way of infiltrating a voice assistant is by connecting to a Bluetooth speaker or a smart TV. Either of those are able to play a sound file or video and can command a voice assistant to do something eventually. [Cand17, p. 12-14] [LTXL+18, p. 3]
Even with the user authenticating and training his voice on the device, there is still a way to get commands recognized. Recordings of the authenticated user can be arranged to create a different outcome and replayed to the voice assistant. [GoPo18, p. 2]

If a person makes commands audibly, it is easy to hear and comprehend what somebody else is trying to accomplish. But if an undecipherable or inaudible sentence is recorded by the assistant it is harder to recognize the malevolent intent. By distorting a recording of the wake word or a command, an audio sample is created that is unrecognizable for human ears but understood by voice assistants. This is done by keeping key vocal features of the human language. Depending on the attentiveness of a person and the amount of distortion that happens to the sample, there is still a chance for the word or phrase to be understood by humans. [HSWW17, p. 7]
A study conducted by Zhang et al. [ZYJZ+17] shows that attacks exist which are completely inaudible to the human ear. These hidden commands are created by taking an ultrasonic carrier and modulating an audio sample on top. Because of the underlying weak spots and behaviour of the hardware, these attacks are feasible and can be carried out. This leads to a new perspective as commands are now able to sneakily invade the devices. They can be used to open malicious

websites, spy on the owner, inject fake information, deny the service or conceal an attack. With the former problem of televisions at hand, it is possible to connect inaudible commands and an advertisement. Without actively recognizing the prompt, products can be added to the users' shopping cart, or even worse. [ZYJZ+17, p. 1-3] [HoyM18, p. 85]

Anything a voice assistant is not able to do out-of-the-box, can be installed on some voice assistant devices as an application. Which introduces the next risk: malicious applications. These can imitate the behaviour of an already existing skill and thus gather personal information about the user. Without having unique invocation names for applications, it is easy to develop a skill using the same name or wake phrase. As some stores for applications do not allow duplicate invocation names there are workarounds by simply creating applications with similarly sounding names. [ZMFW+, p. 1-4]

### 2.3.2 Privacy issues

One major concern voice assistants have to face is the preservation of the user's privacy. By design these devices are constructed to listen to everything that is happening around them. Even though the companies reassure that only when the wake word is said a recording is actually transmitted, there have been malfunctions. Users get anxious when, without their knowledge, requests are sent to the server because of a phrase that was misinterpreted. The main problem users see is the possibility that such recordings might be used against them. This could be done by stealing or leaking personal data. As most requests are saved in data centres to assist the understanding of nuances of the users' language and accent, an attack on such a centre could cause serious damage. Additional concerns come up concerning children, especially with current approaches to developing voice assistants tailored towards kids. What many people seem to overlook though is that all modern smart phones have such a voice assistant already installed. Furthermore, there should be precautions taken to secure such devices, as there already exists malware to infiltrate them. [HoyM18, p. 85] [Cand17, p. 17]

Using mobile voice assistants in public is a unique problem of its own. Whilst being surrounded by other people there is a tendency to avoid drawing any unwanted attention to oneself. The voice interface on the other hand does not really support such a behaviour. Commands have to be spoken out loud and thus create a situation in which users get stared at. This can also disturb the environment. As an example, a library is not an appropriate place to ask a voice assistant what the current score of a football game is. If another person is sitting next to the user, it can also intrude their privacy. Also, telling the voice assistant a credit card number or any other personal information might be inappropriate and avoided by the user. Depending on the amount of people in the surrounding this behaviour changes. The more humans are in the users' vicinity the less likely a voice interface is used. At this point the usage of multimodal interfaces comes into play. Users are more likely to use gesture-based interaction in crowded places. It is deemed more socially acceptable and less embarrassing than using voice to interact. [MoVu15, p. 307-312]

With the development of General Data Protection Regulation (GDPR) privacy concerns regarding the processing of personal data are minimized. Voice assistants and the installed applications have to align with those regulations. These encompass the means of the data that is processed [Euro16, § 5], lawfulness of processing [Euro16, § 6], condition of consent [Euro16, § 7] or the right to erasure [Euro16, § 17], to name but a few. Users of the application have to be informed

what personal data and for what reason said data is being processed and saved. This basis can never be changed without further consent of the affected user, meaning that no software or system is allowed to use the data in any other way. An example would be to use the recorded voice samples in order to gather biometrical information about people in the future. Which, if the user did not specifically accept those terms, is against the law. According to the GDPR [Euro16, § 4(14)]: "biometric data means personal data resulting from specific technical processing relating to the physical, physiological or behavioural characteristics of a natural person, which allow or confirm the unique identification of that natural person, such as facial images or dactyloscopic data". This means that, if voice is used to uniquely identify a person it falls under even stricter regulations because it is considered sensitive personal information [Euro16, § 9] [Conr17, p. 742-745].

# 3 ELGA and MyELGA

This chapter focuses on two topics. First of all, ELGA, the electronic health record in Austria is addressed. Special attention is paid to giving a general overview of what tasks ELGA can accomplish and its benefits for patients and other involved parties. Afterwards, an explanation about ELGA's architecture and the application "e-Medikation" is being given in order to be able to address the second topic, namely the Alexa skill MyELGA. As the security measures of ELGA are the basis for everything that is built on top, it is important to briefly explain what is done to secure data. When explaining MyELGA the focal point will be its architecture and the state of development. To finish the chapter, future desired functionality of the MyELGA application is gathered and discussed.

## 3.1 ELGA

The overall goal of ELGA is to give all patients the needed autonomy and access on their medical data. In order to do so it digitalizes and decentralizes the corresponding clinical reports and findings. That way, a patient can easily access their personal health information without restriction to space or time. This is because ELGA was constructed as an information system which is always available and accessible. Results that health service providers generate are immediately updated and available for the patient reducing the need for paperwork. Such providers can be hospitals, doctors, pharmacies, dentists or health care professionals, that can also access already gathered health data. This helps them to work with patients and improves the overall quality [ELGA19e].

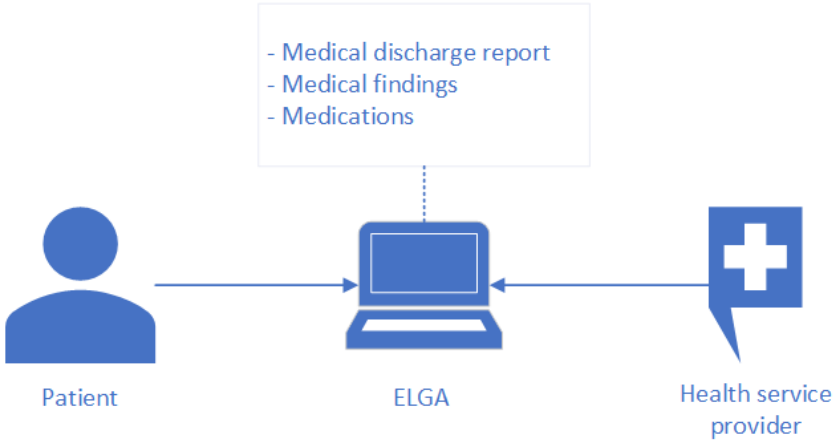


**Fig. 3.1:** Simplified ELGA structure

The use of IT as a tool in health services began around the year 1990. First, it was primarily used to document, order, manage and administrate processes centred around patients and their therapies. The use of IT has become a standard nowadays and is present in laptops, smart phones and tablets. These also offer access to information for consultation purposes with a patient [Amme18, p. 20]. In December 2015, ELGA was introduced to different health service providers in Vienna and Styria. Since then, it is being gradually implemented throughout Austria. Simultaneously the access portal was opened to all citizens [ELGA19e].

Although the service is still evolving the online access offers a patient more control and ancillary data on his medical information. There are two ways to authorize the access and later log in. This is accomplished either via a mobile phone signature or the citizens card issued by the Austrian government. After successfully entering the required access-data a patient can see who and when somebody looked at their medical records. Also, the portal shows which health service provider has access to the data which is currently available on the patients. If there are findings or data on the medication intake that they do not or especially want to share with certain providers, they can lock or unlock them. The portal offers additional functionality to change the duration of access for every active provider [ELGA19c].

Another part of ELGA are the access rights which can be transferred to another person. It allows patients to be represented in ELGA. This must happen via a full power of attorney given by the patient. With this the representative is able to see the same as the patient would usually. Since obtaining such admission is powerful, additional security measures must be taken. When the representatives are accessing the system, they have to tick a checkbox signalizing that they are not the patient. Now there must be a valid full power of attorney saved digitally in the service of the data security commission. This is checked and if deemed correct the representative is able to see the medical information of the patient. [ELGA19d]

### 3.1.1 Architecture of ELGA

The core components of ELGA are diverse. Currently there are two active applications in ELGA: "e-Befund" and e-Medikation. They cover different needs of the patients and are all electronically updated on the ELGA access portal. On the one hand, components manage information about discharge from a hospital, laboratory findings, and radiology evidence which can all be grouped together to form the e-Befund application. On the other hand, there is e-Medikation which saves information on prescribed medical treatment and their side effects. They all have in common that they use the same restriction of access which is predefined by ELGA. This means not every user can view data of other participants [StHo11, p. 343]. In future versions of the access portal additional data provision is planned. As stated in ELGA's own educational material these are supposed to be findings of medical specialists, pictures, patient decrees, health care proxies and legal medical registers [Sabu19, p. 7]. Furthermore, the application "e-Impfpass" is a pilot project that has started in 2018. It saves all data about vaccinations a patient receives and helps with reducing redundant therapies. As an additional plus it gets rid of the analogue form of the existing immunization card [BBHL$^+$18, p. 125].

Overall ELGA is a decentralized information system with centralized parts. Some segments have an interface which is used by health service providers to transmit the personal medical information about each patient. Such a segment consists of a document-registry which is an index that points to the location of saved data such as documents. This information is decentralized

and usually saved on the server of a health service provider. Another component are the indexes for providers and patients. Both exist to identify either a provider or patient. As every entry is distinct, every document can be linked to the patient and health service provider. Overall this helps with managing and documenting what happens in ELGA. Another part is the authorization system. It defines who, at specific time periods, is allowed to view the data [Klos19, p. 8-9] [StHo11].

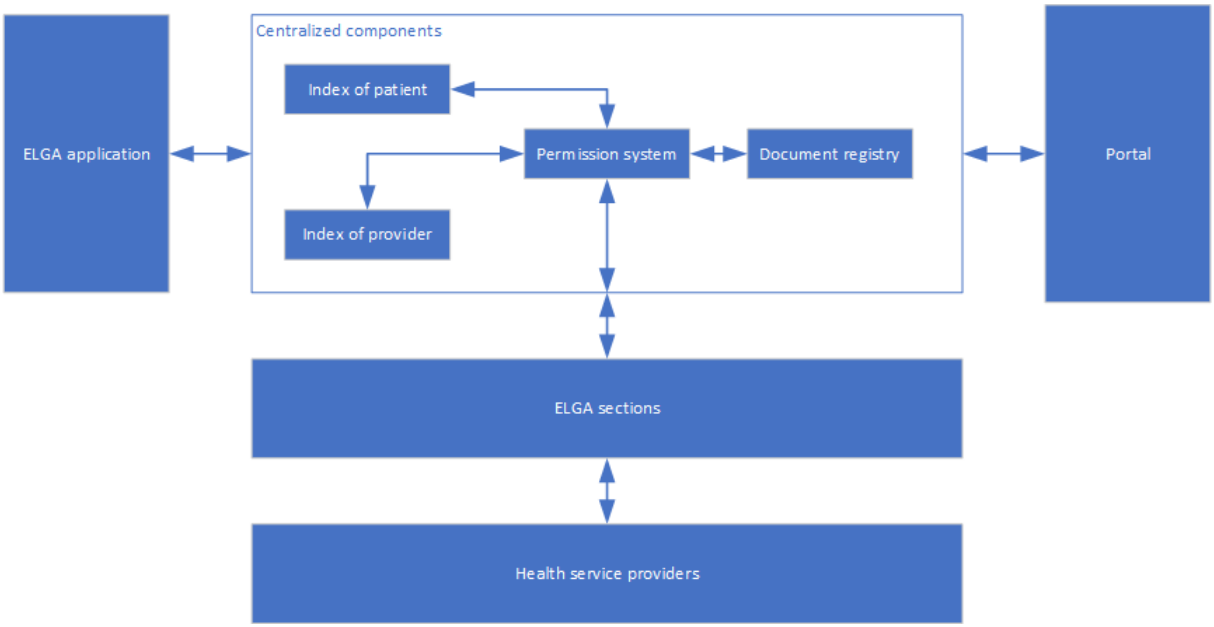


**Fig. 3.2:** Architecture of ELGA [Klos19, p. 9]

Special focus is set on the application e-Medikation which manages all information that is saved regarding the medication of a patient. This functionality was proposed in the year 2010 and is currently available in ELGA. A big roll out of the functionality started in early 2018 and should end by late 2019[Sozi19]. The core of e-Medikation is a database in which all prescribed medication is entered. Additionally, all non-prescribed medicinal products the patient buys that may have interactions with other drugs, can be saved in the database too. This helps doctors to reduce the interactions between different medications. Also, the problem of prescribing the same drug twice is solved this way [ELGA19c] [Sabu19, p. 8].

With data being usable in different IT systems there is a necessity to uphold interoperability. Therefore, all documents processed in ELGA have to comply with the "Health Level Seven Clinical Document Architecture, Release 2.0" (HL7 CDA) standard, which has been defined by different representatives of health, software and research sectors. With the standard in mind, implementation guides for ELGA were developed. The CDA standard structures header and the overall layout of the documents [Klos19, p. 88-90].

Contained in that header is information about the document itself: type, creation date, confidentiality, language and other metadata. The patient and every other involved party or organization is also mentioned. Medical information and documents are to be found in the body. It can be structured in form of sections, which either are readable by humans or machines. The baseline is that any important medical data must be in the part that is readable by humans, meaning

that the primarily used source is the narrative section. Depending on the content the structure of bodies can vary and is not always the same. Further information regarding this issue can be looked up in the implementation guide [ELGA19b, p. 35-81] [Klos19, p. 88-90].

### 3.1.2 Security aspects of ELGA

The exact measures that have to be taken can be found in the Health Telematics Act. This means that by law ELGA is enforced to secure data and must use predefined means to do so. One such thing, which is stated in the Health Telematics Act, is that there has to exist a system for documentation that complies with the General Data Protection Regulation. What kind of information is allowed to be documented has been decided on beforehand [Bund12, §22(2)]. Furthermore, the Act defines how access should be granted [Bund12, §21], data must be saved [Bund12, §20] and the identity of users has to be validated. [Bund12, §18]

Every finding that is created by health service providers they are liable for. Saving and securing such information is in the interest of said provider and the patients. They either use their own server infrastructure or a third-party service. Additionally, accessing and saving the data is only done if an actual therapy has been established. In order to establish one, a doctor has to use a card to verify his access and the patient his e-Card, for example. This way, a therapy is officially started and ELGA can check the data that is available to the doctor. Encryption is used for information, created and saved by the services of ELGA. To ensure a secure transportation of all medical data leaving and entering ELGA, the transmission is also encrypted. Furthermore, all involved parties using the system oblige to implement the security guidelines created by ELGA. These are oriented on international standards (e.g. ISO 27000) [ELGA19a] [Klos19, p. 252-258].

But even though these laws and restrictions exist, there are still ways to attack the system and therefore two possible scenarios which could happen to break the data protection are listed. The corresponding defence mechanisms implemented by ELGA are added [StHo11, p. 344-345].

**Attack from the outside:** This could happen if an external person tries to forcefully get access to document-registry. For this to happen, they would have to pass by the permission system of ELGA. Due to state-of-the-art security measures like data encryption and the decentralized approach, it is made more complicated to achieve that. Even if the person's attack is successful, they are left with the document-registry which only contains an index about the location of documents. This means the attacker does not have actual access to the medical information immediately.
Another scenario would be the incorrect protection of credentials the patient has for the ELGA portal. This problem can only be solved through a care- and mindful approach by the user and has to do less with ELGA itself [StHo11, p. 344-345].

**Attack from the inside:** It is possible that a person working as or for a health service provider uses their access rights to view data unrightfully. Counter-measures to this problem are roles with set rights. These are distributed to the right applicants and every access is documented. Additionally, there are laws prohibiting and fining such behaviour [StHo11, p. 345].

### 3.1.3 Benefits and drawbacks of using ELGA

The benefits of ELGA can be seen in the introduction of ELGA 3. In many situations, the two parties, patient and health service provider, profit from using it. One such benefit is an

increase in the quality of therapy. The workflow spanning around a patient is also simplified and becomes more coherent. Different applications of ELGA help with the process. This means the communication is significantly easier and more consistent. Therefore, no medication is prescribed multiple times anymore and there is less of a risk causing side effects with other treatments. Additionally, there is the constant access to all needed information on a patient. The problem of a client forgetting or losing medical data has almost disappeared completely [HEHS+12, p. 155-156] [Sabu19, p. 5].

One advantage, from which only the patients benefit, is the added empowerment they receive concerning their data. As already mentioned, when using ELGA, a person is always in charge of their own medical information. Not only can a patient be certain that nobody unprivileged accesses their data, but they can also authorize particular people to see their data for a set period. As for health service providers using ELGA they are now able to acquire information about the patient for further inspection. The time limit is set to 28 days as a standard but may be changed via the access portal. This time span can be shortened or lengthened but the overall maximum period is a year [HEHS+12, p. 155-156] [Sabu19, p. 5-6]. Security-wise, ELGA is a well-thought-out system which utilizes various methods to ensure a high level of protection and control. This ensures additional trust in the system on the part of the patient [StHo11, p. 345].

Although there are a lot of benefits, the drawbacks of ELGA should not be ignored. Especially when looking at what can interfere with ELGA in its entirety. One such factor is that some health service providers do not use ELGA. The problem at hand is that some professionals do not want to provide information about their practices. Moreover, the patient also needs to display active interest in controlling and managing their medical information. Both sides need to work together and form a partnership so each one can really benefit from the system. Although the mentioned drawbacks have shown to exist there can still evolve other problems. ELGA still is not active in all of Austria and therefore over the course of time different problems might appear [Amme18, p. 22].

## 3.2 MyELGA

MyELGA combines the convenience and interactivity of a voice assistant with the e-Health environment of ELGA. The prototype is an Alexa skill whose functionality is mainly centred around the application "e-Medikation". There are plans to extend MyELGA and to adapt ELGA applications and functions. When launching the skill, it asks for basic information about the user. By either using voice commands or a touch-display all necessary data can be entered. Information gathered this way is divided into three parts: first, information used to identify the person, with their name and social security number. Second, information to further describe the patient, namely age, height, and weight. Third comes all necessary medical data. These contain information about allergies or illnesses [Klad19, p. 17-19].

The application's initial scope was to cover the basic medical features that ELGA offers. With MyELGA being an all-in-one solution, some additional features were planned but not yet implemented. These range from a reminder to take medication or a measurement to informing a person of trust in case of a critical situation. In addition, an implementation to the e-Impfpass application was planned. It was supposed to remind the patient of visiting an institution in order to renew a vaccination [Klad19, p. 19-24].

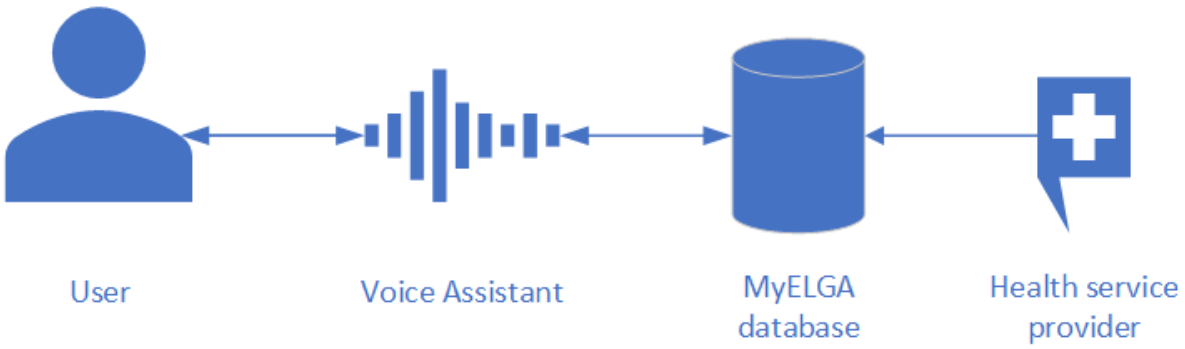


**Fig. 3.3:** Simplified structure of MyELGA [Klad19]

Currently the prototype is able to assist with many tasks regarding the e-Medikation application and even beyond. As already mentioned, MyELGA helps with saving initial information about the patients. They are able to enter and change their personal data. Variables like weight, height, or age can change over time and must be adapted accordingly. If the users want to learn a certain detail on the patient, they can inquire about it in MyELGA [Klad19, p. 25-28].
Another possible application is the management of medication. In addition to the functionality of ELGA, a database is connected to the assistant with information about all the prescribed medication. Simply inquiring in MyELGA about the effects of a certain drug provides the asker with an answer. The patients can see and expand the list of medicine requiring intake and let the skill know that they have taken a specific drug on a certain day. Every time this happens MyELGA documents the intake and notifies the user if restocking of the medication is needed. Asking the skill what medicine they have to take will return a list of all medications to consume that day [Klad19, p. 25-28].
The third and last setting MyELGA currently is used in, are measurements of blood sugar and blood pressure. Overall the core function is similar to the one before. When measuring blood sugar or blood pressure the patient can save those values by using voice commands. Additionally, the patient is able to ask the skill if everything is okay. If said levels are out of the ordinary, a warning is replied to the patient. [Klad19, p. 25-28]

### 3.2.1 Architecture of MyELGA

The basics regarding voice assistants were already covered in the beginning of Chapter 2. Now the focus is on the specific structure of MyELGA. For illustration purposes, a few terms, particularly used by Amazon, have to be introduced first. When developing a skill for Alexa, one has to keep in mind that the voice assistant communicates with the Amazon Alexa Service which then transmits data to the service endpoint. The Amazon Alexa Service is where the processing of voice requests happens. This is done by creating an interaction model which encompasses utterances and intents. It designates the corresponding command (utterance) to the correct service (intent). The actual functionality of a skill is saved in the service endpoint. This can either be a web service hosted by any cloud hosting provider and connected via HTTPS, or the Amazon Web Service Lambda (AWS Lambda). For the MyELGA skill AWS Lambda was used, as it offered an easier and modernized way to develop the prototype. No server needs to be

managed and everything is hosted by Amazon itself. In practice such functionality is called an AWS Lambda function [Klad19, p. 14-16] [Amaz19h].

As Amazon defines the overall structure of a skill's development process, they offer additional tools to help with that. In this case the AWS Lambda Tool was used. The tool supports different programming languages of which Java was used for MyELGA. The selected option for the integrated development environment (IDE) was Eclipse. It offered access to an open-source plug-in called AWS-Toolkit for Eclipse, which allowed the direct debugging of Java applications in the IDE and additional API for developing AWS services. Information about the patient is saved in the Amazon Relational Database Service (Amazon RDS) and will be referred to as the MyELGA database from here on [Klad19, p. 32-35].

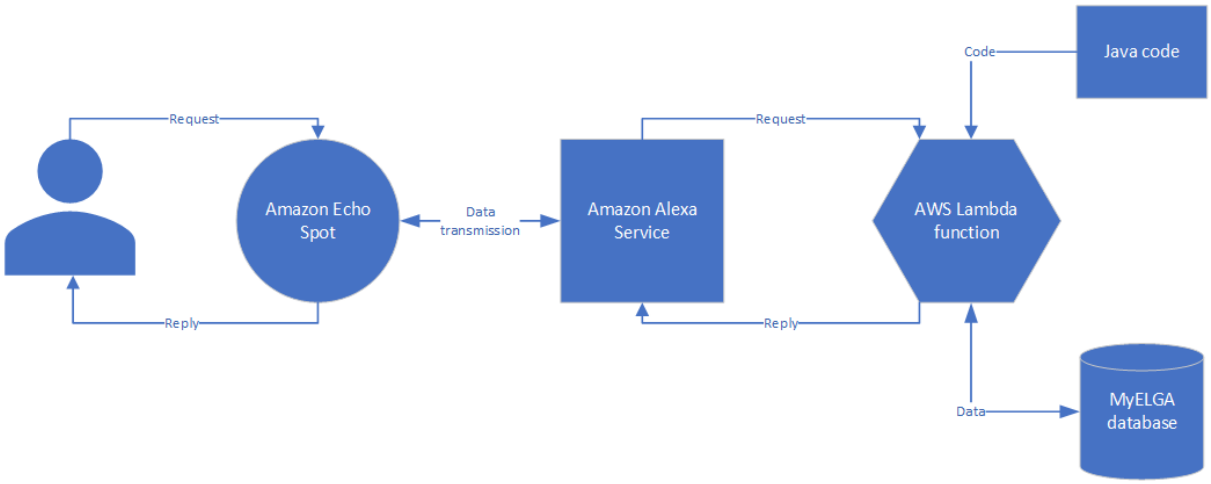


**Fig. 3.4:** MyELGA architecture

When a user utters a request, which is recorded by the Echo Spot, it gets transmitted to the Amazon Alexa Service. The interaction model then transforms the audio into an actual request, a so-called intent. This is then sent to the AWS Lambda function which imports the Java code. It has access to the MyELGA database and depending on the requests issued by a user, data is added, changed or read from the database. Finally, a response is created and returned to the user, who can then act on the received information. This response could consist of a simple answer to a question or follow-up questions to reassure information [Klad19, p. 32-35].

### 3.2.2 Desired functionality

For future research and development of the prototype, necessary improvements must be considered. According to a study by Klade [Klad19, p. 83-84] the following changes should be considered when searching and evaluating voice assistants.

At the time the skill was developed, the range of functions available was not big enough to create all scenarios desired. Especially the functionality of reminders had to be put on hold because there was no way of implementing it with Alexa. Tests showed that if a user has to initiate this activity it is not deemed user-friendly. Therefore, the aim is to achieve a bigger range of functions.

Other complaints were directed at Alexa's accuracy. This could be improved by changing the interaction model, but it is also promising to examine the accuracy of other voice assistants

and compare it to Alexa's. If another assistant would able to have better accuracy, it could be beneficial to future developments of the application.
Another idea is to add other ways of interaction to the skill which could help with entering data in the beginning. This means that multimodal and multi-device support could be an interesting functionality to look into [Klad19, p. 83-84].
As of now, MyELGA uses the offered database by Amazon (Amazon RDS). The free-tier has restrictions to the security which are a downside. With the intent of creating an active system this is not desired. Other ways of implementing their own infrastructure or using an existing database with higher security-standards have to be investigated and evaluated [Klad19, p. 38].

# 4 Set of criteria

After presenting all the necessary information and an outline on the topic, this chapter focuses on the process of defining the criteria catalogue for later comparison. This is done by creating a target group that represents customers of MyELGA. With the persona, desired future functionality and the security aspects of voice assistants, criteria are generated and explained in detail. Each criterion is defined through a name, description and a method of measurement. Finally, a finalized set of criteria is presented and used for further comparison.

## 4.1 Target group

Chapter "Focus on the elderly" 2.2 introduces the foundation for creating the target group. A more detailed depiction is generated to investigate criteria concerning the elderly. As the comparison focuses on a possible alternative of a voice assistant for MyELGA, the target group is essentially, the expected customer. Although caretakers might use the system too, the focus has to be set on the actual patients. Overall the persona of a typical expected user is represented by these four categories:

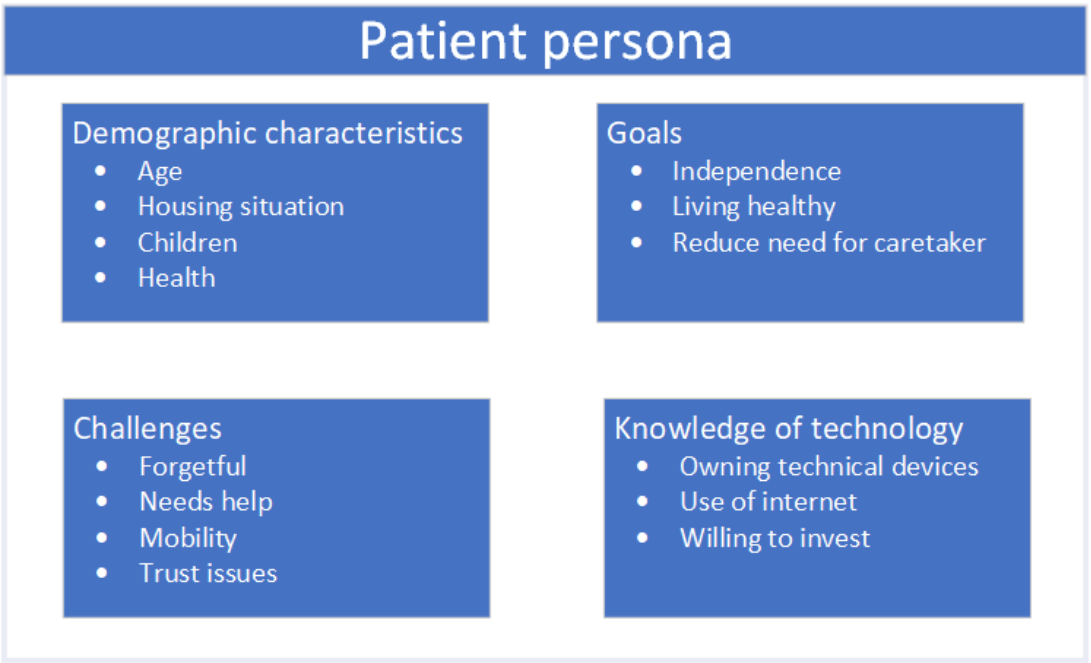


**Fig. 4.1:** Patient persona

Beginning with the **demographic characteristics** of the target group, a typical user of this application is above the age of 65 and speaks German. The gender does not play a significant role in the target group and although the marital status is irrelevant as well, the patients are expected to live with another person, this could be a partner or a housemate. The users have at least one child, but it does not live at their place any longer. Regarding the health of typical patients, they are supposedly taking multiple medications and have to visit the hospital around two times a year. Because of a higher risk of obtaining age-related diseases this means they either have reoccurring check-ups or actual therapy to go through.

Furthermore, the created persona has set itself certain **goals**. Due to the already mentioned trend of wanting to stay independent and autonomous this characteristic is added to the target group. This means that for them, anything benefiting this goal is more than welcome. Examples are devices or applications that directly add beneficial value to this process and give the patients the ability to do more of their daily activities on their own. Additionally, the target group is trying to actively follow healthy guidelines. They are improving their lifestyle to stay fit. Although the patients might have a caretaker, it is in their best interest to reduce the usage of these caretakers. This does not necessarily mean to remove the caretaker, but the goal is to rely as little on others as possible.

One of the many **challenges** that a member of the target group has to face are age-related symptoms, dementia being one of them [WoKK15, p. 1]. Therefore, forgetting about medication, dates, and check-ups at the doctor's office are just a few examples of the type of information which might be lost. Especially when a caretaker asks a patient for their medication intake, and they cannot provide any details, this poses a problem. If questions or situations arise where the patients need help and no person is in their direct, they have to wait for someone to come by who they can ask for help. Such situations occur because of the elderly's lack of agility. According to Shrestha [SMHM17, p. 1-2] they tend to lose their mobility over time and thus become dependent on public transport. This means any application must try to reduce the users' mental load instead of adding to it. They need to be easy to use and maintain. Another issue is the level of trust users put in the companies selling those devices and providing services. Fear of the misuse of their personal data and of being monitored are the users' main reservations regarding those companies.

In the last section **knowledge of technology**, the requirements on the user side for dealing with new technology are described. A patient is expected to at least own one technical device (e.g. computer, laptop, tablet, smart TV, or smart phone). This is the foundation for attitude and skill of the expected users. They do not have an aversion against technology but are rather open about learning. Overall the users have fundamental knowledge on the operation of technology. Thus, they are able to use the internet and do so regularly. Furthermore, the expected patients are willing to pay a reasonable price for their device(s) and see the added value in doing so. However, if costs of purchase and facility get too expensive, the target group is not interested in overspending.

## 4.2 Defining the set of criteria

The next step, which is split into three different categories, is to define the various criteria. First, the criteria are derived from the user and target group's preferences. They revolve around

independence, usability and trust. The criteria are inferred from chapter "Focus the elderly" 2.2 and the earlier defined target group 4.1. After that, criteria related to the MyELGA application is defined and illustrated. Especially the future desired functionality will be of great relevance for section 3.2.2. The third group of criteria will be centred around security and attack scenarios of voice assistants, which are addressed in the introduction on voice assistants, Chapter 2. This is not an exhaustive list of criteria as they are inferred from the gathered information in this thesis. They are merely supposed to give an impression of voice assistants' status quo and their unique features.

To describe each criterion in exactly the same way, a basic scheme is used. Every entry consists of the name, the reason for choosing it and the characteristics.

### 4.2.1 Users and target group

#### Ease of installation

This criterion describes how easy it is for the patient to install the device, focusing on the first time using and configuring the voice assistant. Further use and comfort of dialogue are not covered by this.

With the criterion being neither numerical nor binary, its characteristics require explaining in more detail. If the process takes only little to no effort at all, the criterion is satisfied. Most users have only fundamental technological skills, which is why for them, the best option is a plug-and-play installation. Anything else, such as the need for additional software, having to read a complicated manual or the help of a technician is undesirable and results in the criterion not being satisfied.

#### Adding functionality

Most users wish for an easy operation after the device is installed. Tasks like installing additional applications to add to the voice assistant's functionality or using the device for everyday activities need to be achievable without excessive external assistance.

The sheer amount of interpretation and range this criterion encompasses makes it hard to assess. Although it may be assumed that every user has some basic knowledge, the question is which tasks lie outside of that comfort zone. Therefore, the idea is to look how convenient adding functionality, namely harnessing the array of functions offered (weather, time, news), is for the user. If they can accomplish this without needing external help, the criterion is deemed fulfilled.

#### Cost

The criterion cost consists of two aspects. First of all, the acquisition expenses for the voice assistant and, second, the facility costs. To ensure the comparability, they will be observed over a timespan of one year.

The result of this observation is a ranked list of the costs in Euro, starting with the smallest price and enabling comparison. The cheapest causing the highest level of user satisfaction and thus being the best option for this criterion. Additional attention is paid to reasonable pricing, taking this aspect into consideration where the results are concerned.

#### Comfort of dialogue

Having a conversation with the voice assistant is a highly desired feature. For the dialogue to feel as natural and consistent as possible, different features have to be considered. The assistant

needs to be capable of connecting sentences using already stated information and the context are examples of this. Other ways, that add comfort to a conversation are included in this criterion.

The information gathered on the comfort of dialogue will be compared. If voice assistants offer different implementations and features, they will be analysed and ranked depending on the amount of comfort they add.

#### Support of German

As the focus is on German-speaker, it is vital that the devices and features used support the German language, especially if one thinks of this paper's target group, for which it is necessary to use their mother tongue in order to operate easily and naturally.

Users need to be able to accomplish all operations involving the voice assistant, such as the setup or everyday operations, while using the German language. Thus, any kind of discontinuity concerning the monolingual operationality are looked for. If the assistant offers continuous supports for the German language, it is assumed that the criterion is fulfilled.

#### The companies' trustworthiness

Another important factor is the trust users have concerning a company. If trust is lacking this diminishes the user's happiness with the assistant. Some might even be discouraged from purchasing a device because of their reservations.

Trust itself is a criterion which is difficult to measure. For a comprehensible comparison all the observed companies' actions taken in order to establish a higher level of user trust are gathered. Such actions can increase transparency concerning the underlying system or simply consist of third-party reviews.

#### Smart home integration

As an additional feature, users might want to integrate their voice assistant into a future or already existing smart home. Thinking of the elderly, the integration can be used for AAL approaches to further assist them.

The criterion is deemed satisfied, if there are ways of integrating the voice assistant into a smart home. Additionally, the complexity of the process is taken into consideration. Not only is easiness of use important, but the amount of supported smart home applications and devices should also be taken into consideration. At the end the chosen voice assistants are ranked based on these findings.

#### Number of applications

The term 'applications' in this criterion refers to the extra software that can be installed or enabled on the device. Usually, a market of some sorts is used to download additional features for the voice assistant.

To determine which assistant is equipped with the most applications, the corresponding market is observed, and the number of available software applications compared.

### 4.2.2 MyELGA

#### Own infrastructure

The use of personal infrastructure can add a layer of security and furthermore trust. This is especially true for the act of saving data on a server or database with the users being able to set permissions and access that data themselves.

Thus, this criterion considers whether interfaces for additional infrastructure are provided. The characteristic is fulfilled if the users can choose to use their infrastructure. If further information is presented in the needed settings, these are presented as well.

#### Restrictions in development

This criterion describes the overall flexibility in the stage of development a voice assistant offers. As many functionalities can be implemented by the developers themselves, focus is set on the restrictions that have been identified in a previous study conducted by Klade [Klad19]. Among these are the fact that conversations are only allowed to start, if a user initiates them, and that the server needs to be set to be publicly visible in order to work with Alexa.

Although the desired functionality can vary depending on the planned project and its features, these restrictions existed at that time of the first prototype developed by Klade and reduced the developers' freedom and application's security. These specific cases are looked into and compared with others.

#### Accuracy of queries understood

The elderly, like all other users of voice assistants, have to rely on the assistants' understanding of their commands. Without the device correctly figuring out what the users want, frustration will arise and, in the end, lead to dissatisfaction. In this context, accuracy describes how accurately the assistant understands what a person says. It does not matter if there exists a way of improving this aspect (e.g. learning a user's voice), as the criterion only measures the correctness of the query understood under normal circumstances.

The accuracy is measured, and the highest value meets the criterion the best. As there might not be a study covering all voice assistants, the one with the highest coverage will be used.

#### Support for other devices

In contrast to the former criterion of "smart home integration", this one covers all supported devices available to the voice assistant. The different components the assistant can be deployed on and can works with, such as smart TVs, smartwatches and smartphones are of interest.

The diversity and sheer number of possibly supported devices are compared and ranked. Unintentional ways of using other devices will not be considered in this step, as it is more of a workaround and cannot be directly related to the assistant. Hence, its functionality cannot be guaranteed, neither now nor in the future.

#### Multimodality

This criterion observes if the voice assistant offers a way of using more than one interaction and/or representation method. This comes in handy when a lot of data has to be entered or a visual aid is given.

As already mentioned, the criterion is fulfilled if it offers multimodality. It does not matter which combination it is provided as long as it is intentional. The same rule as in the crite-

rion of "support for other devices" applies here. If there is no intended way, guaranteeing its functionality with the next update, for example.

### 4.2.3 Security

#### Location of voice commands

Concentrating on security related criteria, this criterion focuses on the queries' location of storage. In this context queries are commands understood and processed by the voice assistant. The location can differ from either being saved locally, on a server, or not at all.

For this study, it is preferred if queries are not saved at all, or second best, only maintained locally. The worst case, in this scenario, is the use of a foreign server, as it grants the least amount of control.

#### Location of intelligence

Similar to the criterion before, the focus lies on the intelligence's location for either the step of understanding or processing. With regard to security any transmission involves the risk of an additional attack. The basic structure is shown in the figure 2.1. Possible locations of intelligence are either the voice assistant, the cloud or the server.

Therefore, this criterion is met if the intelligence is as close as possible to the initial input (the user).

#### Security of data transmission

This criterion observes what happens when a request is created and sent to a remote destination (e.g. server). Of particular interest is the way the data is encrypted. Encryption is needed to lower the potential damage an attack could cause on the transmission channel.

If no data is transmitted or the transmission is secured adequately the criterion is deemed satisfied. In this context "adequately" means that a state-of-the-art encryption is used.

#### Access to saved information

Now, the focus is the kind of information an intruder would have access to if they physically seize the device, be it either the reading of existing data on the system or the implantation of malicious software onto the assistant.

The best case would be that no data is stored locally, and that no information can be accessed, if this access had not been granted explicitly beforehand. Also, the possibility of overwriting the existing operating system has to be investigated. This criterion is also interested in the possibility of an overwriting of the existing operating system. In terms of security, an immutable and indelible software is the best choice.

#### Authentication

Not every person should have access to the voice assistant and the saved information. Usually authentication is done by a password but as mentioned in the introduction, there are ways of distinguishing the voices of different people. Thus, there could be more possibilities to authenticate than just a mere password.

Different authentication methods of the voice assistants are set side by side to discover possible differences. If no basic authentication is offered, this criterion is not met.

#### Other possibilities of attack

Any attack scenarios that have not yet been discussed are listed here. These cannot be aggregated and have to be treated separately.

The number of scenarios found, and their impact is used to assign a particular level of threat to each voice assistant. Here, the focus is on current and major problems that are still relevant. If there is no way to check for its relevancy, it will not be listed.

#### Compliance to GDPR

Due to the current legal situation systems like voice assistant have to align with the GDPR. This concerns the information of whether and how their data is used. The sane guidelines also apply to the applications running on these devices, but these are not of interest in this study.

For this criterion to be met the deciding factor is whether a voice assistant fulfils the legal standards.

## 4.3 Finalized set of criteria

After describing each criterion in detail and the way these results are fulfilled, a set of 20 criteria can be presented and will used to compare a selection of different voice assistants in the next chapter.

| Category | Name of criterion | Characteristics of results |
|---|---|---|
| **Users and target group** | | |
| | Ease of installation | Binary |
| | Adding functionality | Text |
| | Cost | Numerical |
| | Comfort of dialogue | Text |
| | Support of German | Binary |
| | The companies' trustworthiness | Text |
| | Smart home integration | Text |
| | Number of applications | Numerical |
| **MyELGA** | | |
| | Own infrastructure | Binary |
| | Restrictions in development | Binary |
| | Accuracy of queries understood | Numerical |
| | Support for other devices | Text |
| | Multimodality | Binary |
| **Security** | | |
| | Location of voice commands | Text |
| | Location of intelligence | Text |
| | Security of data transmission | Binary |
| | Access to saved information | Text |
| | Authentication | Text |
| | Other possibilities of attacks | Text |
| | Compliance to GDPR | Binary |

**Tab. 4.1:** Set of criteria

# 5 Current developments in the field of voice assistants

This chapter covers the current development and statistics of voice assistants. The market situation and trends expected in this field are emphasized. With the latest information, the core of this chapter is the comparison of selected voice assistants. In this section, selections are taken from the currently available assistants, then compared to the previously defined criteria catalogue. With this analysis, a frontrunner is chosen with a focus on eHealth, the target group and security aspects.

## 5.1 Statistics and trends

Overall, there are more voice assistants in smart phones than there are smart speakers supporting voice interfaces. With almost 78% of people owning a smart phone and only 35% owning a smart speaker, it comes as no surprise that the smart phone is considered the launchpad for using a voice assistant. Currently, twice as many consumers in the United States use a smart phone to interact with their voice compared to a smart speaker. The remaining consumers interacting with their voice use other devices like TVs, laptops, watches, and other appliances [SUMO19] [SKIM18].

Studies that focus on the voice assistant market make a distinction when reporting on their findings. They either group all forms of voice assistants together, or they separate them depending on the devices used. The research by Microsoft and Bing [OlKe, p. 9] does not distinguish, instead they examine the overall usage in the United States. People were asked what assistant they had previously used. The results showed that Apple's Siri and Google Assistant were the most used digital assistants with 36% of usage each. They were followed by Amazon Alexa (25%) and Microsoft Cortana (19%). A small percentage was found to use other voice assistants.

A study done by Skim [SKIM18, p. 5] demonstrates that the most used voice assistants on a smartphone in Germany are Google Assistant (68%) and Siri (27%). Comparatively, the same study shows that the in the United States, the most used Siri (52%) is followed by Google Assistant (48%). When looking at smart speakers, the whole matter changes. In Germany, as in the United States, the most popular smart speaker is Amazon Echo, followed by the Google Home solution. The whitepaper from Sumoheavy [SUMO19] states that about 85% of all smart speaker owners have an Amazon Echo or Google Home device. It is therefore understood that these two companies are the market leaders in this sector.

With the focus on the smart speakers, it can be shown that the highest market share is occupied by the Amazon devices: Echo Dot, Echo and Plus. They are followed by Google with their own line of smart speakers: Home and Mini. The dominance of Amazon and Google is undoubtedly immense, but it does not mean that there are not any other competitors. Although Apple and

Sonos occupy the rest of the market, approximately 10% of the market share is covered by all the other smart speakers. By adding visual aid as a functionality to the device, it becomes a smart display. The market for these devices is mainly divided into the same two brands, Amazon Alexa and Google Assistant, which are currently the only options being commercially sold [KiMu19, p. 8-9].

One of the most notable trends is the increase in online shopping completed via voice assistants. All observed studies together see potential in voice assistants being used for transactions. Typically, consumers use this voice assistants to re-order products they already own, or they do not care about the brand of product and do not need to compare them. Such purchases could be household products, experiences (film tickets, flights, hotels) and groceries. Another trend is the rise of complex tasks solved with these assistants. For example, such actions can include cancelling orders, accessing banking and managing credit cards. Tackling and improving such scenarios can make the voice assistant more relevant for eCommerce and retailers. The only problem that prevails is the trust issues users have towards the companies [OlKe] [SUMO19] [SKIM18]

## 5.2 Comparison of selected assistants

The first question, that must be answered, is what voice assistants should be chosen for this comparison. To begin, MyELGA was observed. As the current implementation uses Amazon's Alexa, it must be used in this study. It is of interest if an improvement to the current solution can be achieved; therefore, the status quo must be established. Making use of the statistics provided in the previous chapter, Google Assistant forms the second market leader in smart speakers. Thus, considering its market volume, Google Assistant was added to the analysis.

The other large commercial assistants Apple's Siri and Microsoft Cortana were not used despite their volume. Siri unfortunately forces the developer to use an Apple device to create an application [Appb19]. Additionally, the costs of a HomePod (329€), Apple's smart speaker, are too expensive for further use [Appa19]. The target group would not consider buying this device; therefore, Siri is not discussed any further.
Cortana has a different drawback, as at the beginning of 2019, announcements were made stating that Cortana would not compete with another consumer hardware any longer [Dani19]. Later that year the application for Alexa which was not running on all devices operating under Microsoft was removed of said restriction. Now, every device running the Windows operating system can install and use the Alexa application [KuWe19]. Thus, Cortana is not viable for MyELGA or this study.

The other voice assistants added to the comparison were Mycroft, CMUSphinx and MaryTTS, and Snips. Mycroft is one of the few open-source solutions that offers a smart speaker to go with their software. Furthermore, it allows for different natural language components to be changed independently of one another. Another approach is the speech recognition toolkit CMUSphinx and the text-to-speech application MaryTTS. These are of interest to the MyELGA and, therefore, should be investigated further. CMUSphinx in combination with MaryTTS are two components that can be used to develop an application that works with voice input, with CMUSphinx being the core for processing voice and MaryTTS overseeing the text-to-speech transformation. As they are not stand-alone systems, it was difficult to compare them to existing

voice assistants, as many criteria depend on the implementation and characteristics of the final product. It depends on how the final application is developed; therefore, many criteria were not applicable [CMUS19d] [DFKI19] [Mycr19l].

To generate a basis for comparison, a fictional application facilitating these two libraries was introduced. The implementation should have a similar structure; therefore, a smart speaker was conceptualized with the following specifications:
CMUSphinx was deployed on a Raspberry Pi and MaryTTS on a server running Linux. The Raspberry Pi was equipped with a microphone, speaker, display and all necessary components to connect them. After an initial training of the libraries with open-source voice data, no additional voice commands issued by the users must be saved. Furthermore, transmission to the server and back are encrypted, as a defence mechanism against attacks from the outside. With no intention of adding functionality through the users, the whole application can be deployed through the developer and stay the same throughout use (except updates). Consumers would not be able to alter the system on their own. Criteria are either associated with the two components on their own or the fictional application. To distinguish the different scenarios, an additional tag ('application') is provided in the results, if it is only applicable with the fictional application. Otherwise, it is left blank with only the corresponding result [CMUS19c] [open19b] [DFKI19].

Another voice assistant used for this study is Snips. It is an open-source software that can build voice assistants. It mainly stands out because of its attention to privacy. It was proposed as an alternative for Alexa through the Austrian Research Promotion Agency (FFG), and thus, it was evaluated for the next step.

In the end five voice assistants were analysed and compared to each other: Alexa, Google Assistant, Mycroft, Snips, and the combination of CMUSphinx and MaryTTS. As different brands offer different ways of using their assistants, a common basis had to be defined. Therefore, the overall scenario is that a smart speaker is set up in the living room of a consumer. The criteria for comparison were taken from the earlier developed catalogue and grouped the same way.

### 5.2.1 Users and target group

#### Ease of installation

Amazon's smart speaker is most often bought in the form of an Amazon Echo device. The installation asks for a smartphone, tablet, or Wi-Fi enabled computer and a wireless network. The first step is to plug the smart speaker into an electric outlet. The users are then instructed to set up the device with the Alexa app or a computer. To do so, an account must be created. The next step connects the voice assistant to the Wi-Fi network and basic personalization is completed. Afterwards, the Amazon Echo is installed, and the device is ready to work. If the users can use a smart phone, the process is easy and can be done with little to no help. Even if they are not able to complete the installation themselves, a more tech-savvy person can assist. There is no need for a technician or company to be called [Amaz19s].

A similar process can be found in using the Google smart speaker. Google Home has comparable requirements for installation: a smart phone or tablet with the up-to-date Google and Google Home apps, an account for Google, and a Wi-Fi connection. The smart speaker must be plugged in, and the mobile device must be connected to the wireless network. Then, the next step is

to start the Google Home application; the presented instructions must be followed. As already mentioned, this process is analogous to Alexa's and the installation is just as easy [Goog19v].

The open-source solution Mycroft also offers a smart speaker with a screen, the Mycroft Mark 1. The device must first be connected to a power source and then either configured through the Wi-Fi network or by connecting to the assistant with an ethernet cable. Next, the instructions must be followed to set up the device. In the process a user account has to be created on the Mycroft website. This account creation ends the installation and makes it as easy as the aforementioned procedures [Mycr19g].

Comparatively, the voice assistant Snips works differently. As there exists no finished smart speaker device sold by the company, the consumers must build their own. For this approach either the Snips Maker Kit [Snip19m] or a Raspberry Pi with other hardware has to be used. Both approaches work the same way, as they must be assembled and programmed by the consumer. For the voice assistant to run on the device, operating software, network settings, and the Snips platform and hardware must be configured manually. Overall, the process is much more time-consuming and demands that users know computers very well. Even if the Raspberry Pi was preconfigured, it would have to be adjusted to work in a user's network, which still poses a difficult task. Thus, the process is not doable without the further help of a technician [Snip19k].

As previously mentioned in the short introduction to CMUSphinx and MaryTTS, these two components do not have a procedure remotely similar the other voice assistants. CMUSphinx is a toolkit for speech recognition, and MaryTTS is used for speech synthesis. If they are implemented in any application, the act of installing that program establishes the ease of installation. Both components therefore could not be compared to the other assistants and even the fictional application was still too loosely defined as to have a suitable answer.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Fulfilled? | ✓ | ✓ | ✓ | × | — |

**Tab. 5.1:** Criteria - Ease of installation

#### Adding functionality

Alexa can install additional applications with the help of the voice interface, the Alexa app, or the website. If the users know the exact name of an application, they can use it by telling Alexa to enable it. Otherwise, the app and website offer skills for the device, which can be viewed and selected through them. Configuration and connecting additional services can also be accomplished via this hub. Applications thus added can be used immediately after enabling them, which makes it a very user-friendly process [Amaz19n].

Google Assistant facilitates their Google Home app similarly to Alexa and their Alexa app. Applications can be used by stating their name, or by browsing the Google Home app. The moment these are activated, they are ready to go. If it is necessary, the application can be linked or unlinked. This manages the access to personal data and accounts. The configuration of applications is also done through the Google Home app. Other software that runs on android also has the possibility to utilize the voice interface, if developers planned for it to work. This voice assistant is also fulfilling the criterion as it works similar to the one before [Goog19u].

Mycroft does not offer an app, but it does use a website. When users want to add a new skill to their devices, they must log into their account and browse the Mycroft marketplace. Upon finding an application that fits their desired functionality, adding it to the voice assistant only requires a click. Furthermore, the use of voice is enough to directly install applications without the web interface. After downloading the skill, it is ready and can be activated by the users, which also fulfils the criterion [Mycr19c].

Snips is stiffer in adding functionality later. The voice assistant, when first developed, has all the necessary components added and is then ready to use. If additional services must be added, the whole voice assistant must be reconfigured and installed again. This reinstallation cannot be done by a user without in-depth knowledge of the platform and development process. Thus, Snips does not meet the requirements of this criterion [Snip19j].

CMUSphinx and MaryTTS cannot define, how functionality is added. Again, it is defined by the application being developed with these two components. With regard to the fictional example, the idea is to create a finished product that cannot simply add functionality later. The application must be completely established in the beginning, so it can be deployed on the Raspberry Pi.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Fulfilled? | ✓ | ✓ | ✓ | × | × (application) |

**Tab. 5.2:** Criterion - Adding functionality

#### Cost

To compare the costs of acquisition for each voice assistant, the basic model must be observed. Alexa and Google Assistant especially offer a wide variety of different devices. The prices were taken from the respective shops.

The Amazon Echo is priced at 94.99€ and is overall the cheapest solution over the other voice assistants. The range of prices on the Amazon smart speakers ranges from 42€(the Echo Dot), to 229.99€ (the Echo Show). In terms of performance, the Echo has the only real drawback in that it does not offer a screen like the Echo Show does. Otherwise, it has all the functionality of the other devices except with a less powerful speaker [Amaz19f].

Looking at the Google Home device its price is set at 99€. With the Google Home Max at 299€ and the Google Home Mini at 59€, they are marginally more expensive than the Amazon solution, but still very affordable. The Google Nest Hub could be of special interest, as it is priced at 129€ and offers a screen. The only problem is that it cannot be purchased in Austria yet [Goog19o].

As an open-source solution, the Mycroft 1 Mark is one of the few voice assistants that comes with a prebuilt hardware. It is sold for 131€, not accounting for additional shipping to Austria. Thus, this assistant is quite expensive in comparison to the others. Moreover, the screen used by the Mycroft consists of a small grid. The use of the screen is different and will be discussed later on in the criterion 'Multimodality' [Mycr19i].

In contrast, Snips has one major difference that particularly draws attention. There is no prebuilt smart speaker sold. There are suggestions on what to buy, but users are free to do as

they want. For this study to have a basis for comparison, the Voice Interaction Base Kit was used. This hardware is designed for developers to work and tinker. It is priced at roughly 101€ must be assembled by the users themselves [Snip19r].

No solution offered by any of these companies has additional costs after purchasing the voice assistant. The acquisition costs constitute the full final costs and can be directly compared to one another.

To use CMUSphinx and MaryTTS no money must be spent, as both are open-source. Depending on the device the application later is deployed, the costs are thus defined. As it is not recommended to use a Raspberry Pi for deploying MaryTTS, an additional server can be used. Therefore, the fictional application is a combination of a Raspberry Pi and a server. The actual price for this implementation is hard to define. With an additional server and a display in contrast to the other open-source solutions, the fictional example should be the most expensive. Additionally, facility costs occur because of the server infrastructure needed [open19b].

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Price in € | 94.99 | 99 | 131 | 101 | >131 (application) |
| Ranking | 1 | 2 | 4 | 3 | 5 |

**Tab. 5.3:** Criterion - Cost

#### Comfort of dialogue

This criterion measures the contribution in terms of adding comfort to the users experience of a conversation with the voice assistant. Alexa offers an entire set of functionalities to help improve this experience. One such concept is that the context for a previous conversation is saved. Thus, the state between the assistant and user is remembered to understand follow-up questions. Furthermore, Alexa offers follow-up mode, which listens for requests, after a voice command is issued. This function removes the need to use the wake word again and makes the conversation feel more natural. Another functionality supported by Alexa is the ability to ask for required information, after a request is made by the user. Thus, not only the missed detail and not the whole phrase must be repeated [Amaz19j] [Amaz19v] [Amaz19m].

The Google Assistant also offers the same features shown in Alexa. Therefore, context, follow-up questions, follow-up mode and the selective asking of information are all implemented in the Google Assistant. Both voice assistants are very comfortable for actual conversations [Goog19f] [Goog19x] [Goog19q].

However, Mycroft can only work within the context of a conversation. This functionality at least allows for follow-up questions. But with no follow-up mode, the wake word must be repeated for every question, even if it is in the same context. Furthermore, the assistant cannot ask for selective information by itself. This lack does not mean there is no way to establish these functions, just that there is no implementation yet [Mycr19a].

As the next voice assistant, Snips can save the context of a conversation and compared to Mycroft ask for specific intents, thus allowing a conversation without constantly repeating the wake word. The follow-up mode is different from the commercial solutions, but nonetheless, it offers a more fluent dialogue. Therefore, Snips is more comfortable for having conversations than Mycroft [Snip19l] [Dure18].

Dialogue management is a complex matter, making this comparison hard to conduct as there is no way to know what functionality is integrated and benefits comfort. From the researched information Alexa and Google Assistant offer the most comfortable form of dialogue. They are followed closely by Snips which can still have a convenient conversation. Last is Mycroft, because of the small number of features that benefit comfort. CMUSphinx and MaryTTS do not contribute to the comfort of dialogue as it is a task that must be accomplished by the overarching application encompassing them. The created fictional example cannot define how dialogue is managed and what kind of comfort is added, which would require a more concrete example.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Ranking | 1 | 1 | 3 | 2 | — |

**Tab. 5.4:** Criterion - Comfort of dialogue

#### Support of German

Alexa, Google Assistant, CMUSphinx, MaryTTS and Snips all allow for the support of German. Thus, means natural language understanding as well as text to speech is provided in German. Despite Mycroft not officially supporting it, the company has taken steps to support it in the future. For now, to change the language, the user must manually adjust the natural language understanding, speech synthesis and all the parts surrounding those components. Therefore, this criterion is deemed not satisfied, as there is no easy way to use the voice assistant in the German language [Amaz19u] [Goog19d] [Snip19d] [Mycr19e] [CMUS19e] [Mary19].

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Fulfilled? | ✓ | ✓ | × | ✓ | ✓ |

**Tab. 5.5:** Criterion - Support of German

#### The companies' trustworthiness

Alexa saves voice data to personalize conversations and is willing to give it to third-party applications actively used by the users. How data is collected and what happens after an interaction with the voice assistant is explained understandably on the web page concerning Alexa's privacy. The voice assistant can be muted, and recordings of the users can be manually deleted and managed through the website. Because of recent uprising in the community of voice assistants regarding the deletion of voice recordings, Alexa has introduced a new feature, that allows the user to directly delete recordings through the voice interface [Amaz19e] [Hanb19].

Google Assistant mentions data collection to improve the overall usability of the applications. There are different settings the users can change concerning their data that alter the experience of the voice assistant. Personalization, training of the assistant and third-party applications depend on the information gathered through Google. If users do not allow any data to be saved, the assistant would only be able to use a small portion of its functional range. The users are said to be in control of their data. Alexa and Google have this kind of user empowerment in common. Current development of the Google Assistant reduced the size of natural language models, allowing it to run faster, locally and, more importantly, without an internet connection [Goog19i] [Ludl19a] [Goog19e].

Mycroft states it collects data only if the users allow for this collection to happen. Therefore, an opt-in policy is implemented, which lets users decide on the use of their data. Nonetheless, if voice commands are transmitted to the Mycroft server, metadata on those queries may be gathered. The users are able to consent to Mycroft making their recordings public or saving them at all for testing, developing, and improving their technology. The greatest benefit of Mycroft is that it is entirely open-source and the code can be viewed on their page. This availability improves the trust people place in the company, contrary to Alexa and Google, where the user is only able to check the information, provided [Mycr19d] [Mycr19l] [Mycr19h].

Snips does not need any personal data to work. The basic idea behind this voice assistant is to be private by design. Deploying it to a device and not needing any internet connection is one important feature to the developers of Snips. Additionally, the source-code used can be directly viewed. Thus, this company is one of the most trusted of all the examined voice assistants [Snip19f] [Snip19s].

CMUSphinx and MaryTTS do not need any personal data either, and they are open-source too. The developers of CMUSphinx were located at Carnegie Mellon University, whereas MaryTTS is currently maintained by a research group formed through the Cluster of Excellence MMCI and the German Research Center for Artificial Intelligence (DFKI). In this context, these developers have been deemed very trustworthy because of their closeness to research institutions [DFKI19] [GLPP+14].

In conclusion, the greatest transparency was provided by the open-source solutions, as the users can view the code of the voice assistant and see what is happening behind the scenes. Alexa and Google also offer a variety of information on their voice assistants, but they cannot match the level of trust open-source solutions offer. All voice assistants offered at least one source of information concerning their security guidelines and what they do with the gathered data. Although it cannot be ruled out that there exist further actions to increase and endorse trustworthiness, the found approaches were enough to provide a basic understanding.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Ranking | 3 | 3 | 2 | 1 | 1 |

**Tab. 5.6:** Criterion - Trustworthiness of company

#### Smart home integration

There are two ways to integrate smart home devices in Alexa. Either approach requires the appliance to be supported by Alexa, although most commercially offered devices are. The first method is to use guided discovery, which connects to smart gear already in the same network. Then, the installation is completed through the Alexa app, which is explained with instructions. The other method is that the users install the needed Alexa skill for the device to be discovered and configured. After enabling the skill, the procedure is the same as above; Alexa or the Alexa app create a connection to the devices [Amaz19i].

Similarly, the integration of smart devices into the Google Assistant only works if they are supported. If they are, users can find the appliance through the Google Home app. For the app to discover them, the devices must be in the same network. Thereafter, an application for the

selected smart home gear can be downloaded and voice commands are immediately activated [Goog19g].

When observing the open-source solutions Mycroft, CMUSphinx and MaryTTS, and Snips the most significant difference from commercial voice assistants is their strength and simultaneously their weakness. These voice assistants have no direct support for numerous smart home appliances. To use these devices, additional software must be employed. One example of this is openHab [open19a], which allows the connected voice assistants to communicate with all supported appliances. Therefore, an application is offered and makes implementation a lot easier. Although this boosts the capabilities of these assistants, the downside is that users must do more of the work. They need to integrate software that works as a middleman and configure it, which makes this solution a lot harder to realize for users without the necessary technical knowledge [Mycr19n] [Alph19] [open19b].

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Ranking | 1 | 1 | 2 | 2 | 2 |

**Tab. 5.7:** Criterion - Smart home integration

#### Number of applications

Counting the offered applications in the markets for Mycroft and Snips is an easy task, as there are only a handful of applications. In comparison, it is much more tedious to count the apps listed Alexa's and Google's markets. The sheer amount is one hindrance, but duplicates are where the real problem lies. With many applications listed in different categories across the market multiple times, no automatic view can be created without additional manual work. The numbers for Google Assistant and Alexa are not exact values, but more general guidelines. With the gap being so tremendous, the impression can be conveyed on how far the numbers of applications are apart. In contrast to all other assistants, CMUSphinx and MaryTTS do not offer any kind of market or extra applications that can be installed or enabled [Kins19] [Mycr19j] [Snip19n].

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Applications | ∼56.000 | ∼4.200 | 75 | 161 | — |
| Ranking | 1 | 2 | 3 | 4 | — |

**Tab. 5.8:** Criterion - Amount of applications

### 5.2.2 MyELGA

#### Own infrastructure

As already mentioned in the chapter of MyELGA 3.2, Alexa offers the developers the use of their own database for the deployment of a skill. Furthermore, many types of databases can be connected to the RDS of Amazon and used for further development. Also, skills can be hosted as a web service, thus capable of completely removing any cloud provider [Amaz19t] [Amaz19o].

Firebase is a service for Google Assistant that hosts the fulfilment of actions as an HTTP web service but allows for any other HTTP web service to be used to achieve this task. Furthermore, with webhooks, the transmission of fulfilments is established via HTTP/JSON, which

can be interpreted by other services or applications. Therefore, data can be taken from external databases or web services. This makes Google Assistant extremely cooperative facilitating infrastructure [Goog19l] [Goog19c].

The open-source solutions were developed with the goal of creating software that could be used in most environments. Mycroft is a modular system that immediately supports different natural language components, but it can also support any component installed, if the developers tend to it themselves. Snips has a similar concept, as it communicates over the message queuing telemetry transport protocol (MQTT). By listening to this traffic, any action can be programmed. Thus, it is again in the hands of the developer to make an application that does the job. CMUSphinx and MaryTTS do not impose any restrictions on the use of other infrastructure. However, this capability can change in the actual application developed but both components satisfy this criterion [Mycr19l] [Snip19h] [Snip19a] [CMUS19b] [DFKI19].

Overall, all the voice assistants allow the integration of infrastructure. The open-source solutions pose a more difficult task to accomplish in comparison to the other two solutions. On However, they allow for a much wider implementation of infrastructure. Therefore, all the voice assistants meet the criterion, but each has their own advantage.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Fulfilled? | ✓ | ✓ | ✓ | ✓ | ✓ |

**Tab. 5.9:** Criterion - Own infrastructure

#### Restrictions in development

The two restrictions that need special attention are, if a method exists for the voice assistant to initiate a conversation and if a personal database must allow all incoming traffic.

**Scenario 1: Initiating a conversation** Alexa still does not allow for conversations to be started by the assistant. But, in contrast to the former observations by Klade [Klad19], Alexa allows notifications. These are in no way a replacement for the actual start of a conversation, but they are a first step in the right direction. They allow an application to leave short messages in a queue, which are signaled to the users with an audio and visual cue. They can then ask for new notifications, which are read to them by the voice assistant [Amaz19q].

The Google Assistant does not allow for a conversation to be started by anyone other than the user. However, daily reminders and notifications exist, allowing an application to announce information to the owner of a smart phone, and these notifications can start an intent and initiate conversations thusly. This functionality is not available on smart speakers though [Goog19h] [Goog19s].

Neither Mycroft nor Snips demonstrated any way to manage this problem and had no functionality similar to that of notifications. CMUSphinx and MaryTTS cannot do any of this without anyone implementing it. Even with the designed fictional application, there is too little information on this functionality. Therefore, no voice assistant was able to solve this scenario as required.

**Scenario 2: Visibility of an additional database** All incoming traffic from any IP-address must be allowed, for Alexa at the time of this study. Thus, implementing a database in the Alexa

environment like Klade's prototype did, forces this practice. There are approaches, however, that can bypass this need. Specifically, when the developer uses a virtual private cloud (VPC) provided by Amazon. The idea is that by using a VPC, a configuration on the network can happen. Thus, not all IP's must be allowed. As a downside, the developer must use another service provided by Amazon, but overall, this method allows for better management and security of the database [Amaz19w].

In the scope of this research, the other voice assistants were all able to connect to a database without their interface having to accept all IP's. The Google Assistant, for example uses web-hooks, which allows for communication between fulfilments and the action. Mycroft, Snips, and CMUSphinx and MaryTTS are open-source and do not have any restrictions on the use of databases, if the developers are able to create the connection themselves. Although this is not stated explicitly in any documentation, examples found on their official site show, that it is possible [Goog19c] [Gree19] [Fors19] [Snip19h].

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Initiation | × | × | × | × | — |
| Database | ✓ | ✓ | ✓ | ✓ | ✓ |
| Ranking | 1 | 1 | 1 | 1 | 2 |

**Tab. 5.10:** Criterion - Restrictions in development

#### Accuracy of queries understood

Finding a comprehensive study of all the selected voice assistants that covers their accuracy with natural language understanding poses a problem. Different studies with distinct sets of test data and approaches are difficult to combine. A study conducted by Coucke et al. [CSBB+18] compared Snips, Google Assistant, Alexa, and other NLU engines. The test was conducted by taking multiple training sets for a given intent and then feeding them to the engines. The results were documented for precision and recall, which can be combined to form the F1-score. As this value allows for a better basis, it is the critical factor for comparison. With a training set of 70 data sets, the results are as follows:

| NLU provider | Precision | Recall | F1-score |
|---|---|---|---|
| Alexa | 0.680 | 0.495 | 0.564 |
| Google Assistant | 0.770 | 0.654 | 0.704 |
| Snips | 0.795 | 0.769 | 0.790 |

**Tab. 5.11:** Average precision, recall and F1-score by Coucke et al. [CSBB+18]

Mycroft in comparison does not have a complete NLU engine of its own but rather uses some preselected engines for speech-to-text and intent parsing. Thus, it is not a viable mean of comparison and is excluded from this criterion. The only studies found containing CMUSphinx, compared it to other open-source solutions in terms of error rate. But as it is not comparable to the other assistants, it is also not of relevance to this criterion [Mycr19l] [GLPP+14] [LKKT+18].

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| F1-score | 0.564 | 0.704 | — | 0.790 | — |
| Ranking | 3 | 2 | — | 1 | — |

**Tab. 5.12:** Criterion - Accuracy of understood queries

#### Support for other devices

Despite the already large quantity of devices sold by Amazon directly, there are even more that come with Alexa included. To motivate developers and makers, Amazon supplies many tutorials, guidelines, and learning materials. In the category of 'Alexa Built-in Devices' on their website, a wide variety of products are shown that work with Alexa, including headphones, thermostats, smart phones, tablets, and even devices for vehicles. There is significant involvement encouraging people to make appliances with Alexa, as these devices must go through a certification process from Amazon [Amaz19d] [Amaz19c].

Google also offers great diversity in their products which can all utilize Google Assistant. They also sell a wide range of self-promoted products but with other companies facilitating the assistant. This approach appears in multiple fields like smart displays, smart phones, smart watches, smart TVs, and cars. The Google Assistant is available freely as a software development kit with many tutorials and guides. Significant support is offered for users interested in working with the Google Assistant [Goog19m] [Goog19n].

Smaller companies are not capable of producing a wide variety of their own products to be sold. Mycroft has, at least, their smart speaker, but beyond this device, there are only a few possibilities for deploying the Mycroft assistant, currently limited to the Raspberry Pi, Linux, and Android [Mycr19c].

With Snips being a smaller company as well, they have a similar but slightly larger range of supported devices as Mycroft. Raspberry Pi, Android, iOS, macOS and Linux can run the voice assistant. In contrast to Mycroft's smart speaker, they have a Maker Kit, which is designed for testing, creating and developing [Snip19e] [Snip19g].

Depending on the version used, CMUSphinx can be deployed on Linux, Raspberry Pi, Windows, MacOS, Android and iOS. MaryTTS is implemented in Java and can run on any environment supporting Java. As mentioned earlier, MaryTTS is not light-weight and therefore, deployment on a Raspberry Pi is not recommended. The two libraries themselves have high versatility in terms of supported devices [CMUS19b] [CMUS19a] [DFKI19].

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Ranking | 1 | 1 | 3 | 2 | 2 |

**Tab. 5.13:** Criterion - Support of other devices

#### Multimodality

Many devices built for Amazon Alexa come equipped with a screen and support multimodality. Developers can use these displays, to visualize prompts or information and address the need for a diverse way of interaction. In a skill, both voice and screen interaction can be realized to work together. Alexa can show images and videos through the installed displays [Amaz19k].

The same applies for Google Assistant as it also uses displays as an additional method of interaction. The different input modalities are for example screen, keyboard, touch, and voice. The support of these varies depending on the observed device. Displays can show images and animated pictures. To play videos, a Chromecast or similar device supporting it must be used however [Goog19w] [Goog19t].

Although the overall look would not suggest it, the Mycroft display is also customizable. With a face as the standard setting, it is also possible to customize certain parts of it. With a grid 32 pixels wide and 8 pixels high, the 'mouth' of the smart speaker is relatively small. If a displayed text should be too big, the first part is shown and then it scrolls to show the remaining text. Another way of using this mouth area is to draw a picture on it. Again, the restrictions created through the small grid only allow small images to be displayed. Despite the installed screen on the Mycroft Mark 1, there is also the Mycroft GUI, which was added recently. This GUI allows the developer to create a graphical response that can be used on other devices with a screen [Mycr19b] [Penr19].

In contrast, Snips does not specifically state any use of a display in their documentation, but in their forum, users have demonstrated how they use the voice assistant simultaneously with a screen. They do so largely with the interfaces the system has and the MQTT protocol [Snip19q].

The ability to create multimodality is not imposed nor promoted by the components CMUSphinx and MaryTTS. It is in the hands of the developer to create multiple interfaces that work with the overarching application. However, the fictional smart speaker is equipped with a display. By developing a skill that can facilitate the screen it is also possible for CMUSphinx and MaryTTS to use it.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Fulfilled? | ✓ | ✓ | ✓ | ✓ | ✓(application) |

**Tab. 5.14:** Criterion - Multimodality

### 5.2.3 Security

#### Location of voice commands

Amazon states that they collect, process, and save the voice commands issued by the user. In return, the queries are saved on their servers. Although doing so provides the least amount of control, Amazon tries to improve this flaw by making it possible to manage and delete the voice commands saved on their servers. Recently, this deletion was only possible via their website, but this function has now been added to the voice interface of all assistants. To do so, the requests are directly linked to an Amazon account. There is no intended way to stop the service from saving voice commands [Amaz19b].

Observing the Google Assistant highlights an interesting approach. Essentially, Google allows the user to change their preferences for usage of their voice commands. Not only must the users be signed in to their account but they also must enable the feature 'Voice and Audio Activity'. If both factors apply, then the queries are transmitted to servers, but the moment the feature is disabled, no voice commands are saved. Google also allows the user to manage the Voice and Audio activity. They can see and delete entries but must be logged in, as the commands are linked to their account [Goog19r] [Goog19j].

As mentioned in the criterion of 'Trustworthiness of company', Mycroft follows an opt-in privacy policy. Thus, if the user does not consent to their data being saved, the data are only processed on a server and deleted afterwards. If the user approves of his or her voice commands being used for training, development, and testing, they are saved as open data on the servers [Mycr19l] [Mycr19h].

Snips is the only voice assistant that completely works offline and on device, so it has a huge advantage. With no voice command ever being transmitted to a server it offers the highest security. All queries stay on the device and are private. [Snip19f].

Again, it is not possible to compare the components CMUSphinx and MaryTTS to the other voice assistants. There is no specific standard that defines where voice commands are saved. The fictional application does state to not save any queries after having processed them, so Mycroft's approach is the closest, and therefore, they both were treated the same way.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Ranking | 4 | 3 | 2 | 1 | 2 (application) |

**Tab. 5.15:** Criterion - Support of other devices

#### Location of intelligence

The process of how Alexa understands, and processes voice commands was explained briefly in Chapter 3.2. As it elucidated, most Alexa devices have their natural language components in the cloud. So, to operate, the voice assistant must be connected to the Internet. There is an exception, however, in the local voice control. Its function is to support basic voice commands, ranging from asking for the time to controlling smart home devices. To work properly, a device with the built-in smart must be used, other appliances do not have this capability. The exception has limited understanding and processing components directly on device [Amaz19x] [LRKR+18, p. 3].

Currently the Google Assistant also does all the tasks that need natural language intelligence in the cloud. Thus, the assistant relies on an internet connection to function. Current research at Google is reducing the size of their voice assistant, so it runs completely on device and offline. This feature has not yet been implemented, so the criterion must be rated on its current status. In the future, this rating may change. [Ludl19b] [Goog19j].

The documentation on Mycroft does not explicitly state, where components are located; therefore, an expert interview was held over the Internet. A representative of Mycroft stated that the wake word detection and intent parsing occur on the device itself. Speech-to-text is located in their server structure and the text-to-speech functionality depends on the used engine, either on their server or the device [Mycr19l].

Snips again has everything on the device the application is deployed on. All the intelligence for understanding and processing is locally implemented, and it has no need for an internet connection [Snip19f].

CMUSphinx and MaryTTS can both work on servers and on local devices. The final implementation decides where the intelligence is located. Observing the fictional example shows that the component CMUSphinx is located on device and MaryTTS on a server [CMUS19b] [DFKI19].

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Ranking | 3 | 4 | 2 | 1 | 2 (application) |

**Tab. 5.16:** Criterion - Location of intelligence

### Security of data transmission

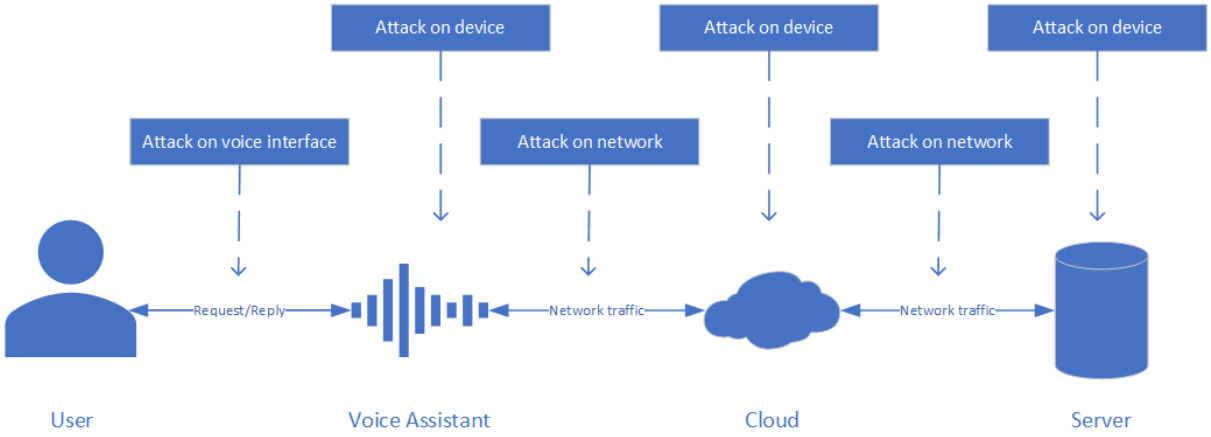


**Fig. 5.1:** Attack scenarios on the overarching process of voice assistants

With the process depicted in the graphic above, the attack scenarios currently of interest now are those on the network, specifically, how the different companies manage the transmission of data sent to their servers. Alexa requires the developers of skills and devices to implement all their security measures. In the scenario of transmission, developers must thus implement transport layer security (TLS) for all communication except the initial setup. Additionally, they must validate certificates for all communication done over TLS. On their website AWS mentions the security measures they use to secure data of their customers. They talk especially about encryption for data in transition and saved on their databases [Amaz19r] [Amaz19l].

The Google Assistant divides encryption into three different categories: data in rest, data in transit and encryption in use. For this criterion, the encryption of data in transit is of interest. Google had stated the use of TLS as an example of securing data that is transmitted. Although this page on security concerns all applications of Google and not specifically the voice assistant, it should still be applicable, demonstrating that data is encrypted and secured in this specific case [Goog19k].

Mycroft again does not specifically state, that they use any form of encryption in their transmission. But when looking at the source code provided, it is apparent, that they use a TLS protocol for encryption. After asking a representative of Mycroft, this information was found to be accurate. Furthermore, any data on their servers transported to the speech-to-text service must be encrypted (e.g. Google STT) as well [Mycr19l] [Mycr19k].

Like the two criteria above, Snips does not need any encryption or other security measures, as there is no data transmission [Snip19f].

As CMUSphinx and MaryTTS do not influence what kind of security for transmissions must be used, no useful finding can be presented with only the two components. In the end, the final application affects security, and as such the fictional example must be observed. Although it was

stated in the beginning that a form of encryption would be used to secure data transmission, it has not been specified. As all the other voice assistants use a TLS protocol for encryption it can safely be assumed that an application built with CMUSphinx and MaryTTS would do so as well.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Ranking | 2 | 2 | 2 | 1 | 2 (application) |

**Tab. 5.17:** Criterion - Security of data transmission

#### Access to saved data

When using Alexa, any application must be manually enabled by the users and the permissions to that skill must be set additionally. With no restrictions or security enabled any person using the voice assistant can access all enabled skills and the information they hold. Therefore, the owners of a smart speaker oversee securing their data. The more applications linked to the device, the more information an intruder can access. On the upside, after searching through the companies offered materials and forum, there was no documented method for changing or deleting the voice assistant on a smart speaker [Amaz19b] [Amaz19n].

Google Assistant links applications to Google Home devices and the user's profile. This process only occurs if the user permits an application to be connected; otherwise, such information is not retrievable. The problem at hand, however, is that without authentication, any person near a Google Home device can request information on these applications and personal data. If for example, the calendar, email, or contacts are provided to the assistant, anyone that has local access can ask for this information. The level of risk is again set by the permissions granted by the user. A benefit found after searching through the company's site and forum, is that there is no way the operating system can be deleted or changed on a Google Home device [Goog19u] [Goog19j].

Mycroft Mark 1 as open-design and open-source allows many changes to the hardware and software. The foundation of the device is a Raspberry Pi, which can be programmed to do anything the developer wants. Therefore, any allowed access to data is completely in the hands of the developer and owner of the Mark 1. In contrast to afore mentioned assistants, the operating system can be modified [Mycr19m] [Mycr19f].

Snips practices the same concept of data access. It can be deployed on different devices but as the focus is on smart speakers, the observed device is the Raspberry Pi. In return, it allows for the same rules of access as seen before. To what extent a voice assistant has access to local or other data is decided by the developer. Furthermore, the assistant is an application that can easily be removed and changed [Snip19k].

With no actual implementation it cannot be determined how CMUSphinx and MaryTTS regulate access to other data. The components only explain the specifics to how they access their needed information. With the fictional application being deployed on a Raspberry Pi, the same access regulation applies [CMUS19b] [DFKI19].

In the case of this criterion the possibility of changing the operating system is deemed more of a security issue than the access to other applications and personal data, as accessing data

through the voice assistant is time-consuming, if the attacker does not know where to start, and it only happens once. Changing the whole operating system can make the voice assistant into a listening device for a longer period for example.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Ranking | 1 | 1 | 2 | 2 | 2 (application) |

**Tab. 5.18:** Criterion - Access to saved data

#### Authentication

Different methods of authentication promoted by the assistants are here explained briefly and then compared against each other. The concepts of voice matching, guest mode, a code and Auth0 could be found. This list of methods for authenticating users is not exhaustive.

**Method 1: Voice matching** Voice matching is the act of training the voice assistant to recognize a person by their voice. Alexa and Google can both create voice profiles and link them to permissions and information. The main reason for using this method is to create a personalized experience when using the voice assistant. Alexa only shows messages sent to the recognized user or shows customized music playlists. The other observed voice assistants do not offer this kind of functionality [Amaz19a] [Goog19a].

**Method 2: Guest mode** The guest mode is a way of allowing visitors to use a limited amount of commands, if they are in the same room with the Google Home device. With their smart phones they can cast media onto the assistant without entering the private Wi-Fi of the owner. This concept only available on smart speakers sold by Google [Goog19p].

**Method 3: Code** If users want to purchase a product via the voice assistant, they must insert a 4-digit code to verify that they are eligible to do so. Only Alexa offers this functionality, which is optional and must be activated by the user. After enabling it, the code must be defined and used thereafter [Amaz19p].

**Method 4: Auth0** With all the different ways of authentication, none really provide much security. Voice matching is flawed, as recordings or similar voices can access the same things, as the original speaker could. Guest mode is nice if visitors are around, but it has no further use. The code serves as a little hurdle, but as it does not change, a person could hear or read it. In contrast, the use of Auth0 can provide a better method of authentication. Auth0 sends a code to another device or service so the user can verify their person. It is supported by all voice assistants and other services, as it was not solely designed for a single implementation. Alexa and Google can directly work with Auth0 and do need significantly less preparation than the open-source solutions. The users and developers of Mycroft, Snips, and CMUSphinx and MaryTTS need their own application to work with Auth0 and the API. Such an approach is possible but time-consuming as a developer needs to implement it [Amaz19g] [Goog19b] [Auth19].

#### Other possibilities of attacks

There are still two possible attack scenarios that must be investigated that the whole process in picture 5.1 is covered. These remaining scenarios are the attack on the voice interface and the attack on devices like a cloud or server. By researching the first scenario it was apparent that attacks on the voice interface had universally the same goal. They attempt to trick the

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Voice matching | ✓ | ✓ | × | × | — |
| Guest mode | × | ✓ | × | × | — |
| Code | ✓ | × | × | × | — |
| Auth0 | ✓ | ✓ | ✓ | ✓ | ✓ |
| Ranking | 1 | 1 | 2 | 2 | 2 |

**Tab. 5.19:** Criterion - Authentication

voice assistant to understand things, that no human can hear, either by replaying at times when nobody else was listening or by masking the malicious intent behind neutral phrases [GoPo18]. Some of these attacks were covered in more detail in the chapter on risks of voice assistants, Chapter 2.3.1. There are exceptions and differences between assistants, when the observed device is a smart phone. These differences are beyond the scope of the criteria, though [ZCLW+18].

Attacks on servers and clouds formed the second scenario, which needed to be explored. It was not possible to find any relevant details on specific attacks on the different server infrastructures and clouds. Although some attacks do happen, none of these were specifically directed towards the services surrounding voice assistants. This lack, in return is a good sign.

To conclude, no specific attacks that deviate from the norm could be found, and therefore, this criterion is no longer necessary.

#### Compliance to GDPR

On their blog, AWS states that all their services are GDPR ready. The process is presented in a rather transparent way, and an 'GDPR Center' was established on their website. It features information on the means currently used to uphold compliance. Furthermore, they explain their services and how they help their customers to comply with the GDPR [Wool18].

With recent fines issued against Google, they needed to update and change their data consent policy, as there was not enough information provided. An explanation on how and why data are processed can be seen on their policies page. It is depicted as transparent and links are provided to further information and a settings page, where the users can change their permissions and what data they want Google to use [Port19] [Goog19i].

Mycroft does not specifically discuss how they comply with the GDPR, but they do offer enough information on how and what they process on their policy site [Mycr19h].

Snips however, promotes their alignment to the regulation on their front page. They make their processes as transparent and understandable as possible, which is easily done, as everything is processed on device [Snip19s] [Snip19f].

Compliance to the GDPR fulfilled by CMUSphinx and MaryTTS can change depending on the observed implementation but as far as the components themselves, there is no indication of noncompliance with the regulation. No personal data is processed from the beginning.

| | Alexa | Google Assistant | Mycroft | Snips | CMUSphinx and MaryTTS |
|---|---|---|---|---|---|
| Fulfilled? | ✓ | ✓ | ✓ | ✓ | ✓ |

**Tab. 5.20:** Criterion - Compliance to GDPR

## 5.3 Winner of the analysis

With all the gathered information on the observed voice assistants, the results can be aggregated into a table. Every criterion fulfilled is highlighted with a green hue. The second-best option is also displayed with a yellow hue. This colouring emphasizes the characteristics of each assistant. The last two rows present the accumulated number of a voice assistant as the best and next best option. CMUSphinx and MaryTTS are referred to in this table as the 'combination'. If a criterion was only rateable because of the fictional example created in the beginning, the tag 'application' is added.

As can be immediately seen in Table 5.21, the voice assistants Alexa and Google Assistant meet the most criteria. Overall, they are very similar in the ways they function and what they offer. The only difference that can be seen is in the amount of applications and costs. The greatest drawback to these two assistants is where voice commands are saved and where the intelligence is located. In comparison, the open-source-solutions are much more diverse. Although Mycroft offers a very comfortable installation and further ways of adding functionality later, it lacks comfort in holding a conversation. Snips, alternatively, is designed around being private and secure, which is its biggest strength. The downside of this assistant is its intricate installation and stiff design, as an application must be extended through the developer. The last observed assistant formed of the elements CMUSphinx and MaryTTS has the most elaborate evaluation. With the fictional application some criteria can be measured and compared. In the end, the final product and implementation of these two components establishes most of these results and can change. Therefore, if further use of these is desired, a study on the specifically used implementation must be conducted.

If no specific emphasis were given to the different criteria Alexa and Google assistant would be the frontrunners in this study. Regarding the future of MyELGA it became apparent that the focus needed to be on the security and privacy of the voice assistant, as these two items impact the basis of data processed by the MyELGA application. Most of the data are personal and private information, which is handled in a health context; therefore, the commercial voice assistants must be excluded because of how they save and process data. Information should not be automatically saved on servers in the first place. Even if there are options to delete it these approaches are insufficient for the requirements set by this specific application. Therefore, the pool of viable winners is limited to Mycroft, Snips, and CMUSphinx with MaryTTS. Amongst these options, Snips fulfils the most criteria, particularly those that centre around security and data processing. Mycroft and the combination of CMUSphinx and MaryTTS did meet some necessary requirements, but they are finally inferior to Snips. Thus, Snips is the most promising voice assistant for MyELGA and the best option in this evaluation.

| Criterion | Alexa | Google Assistant | Mycroft | Snips | Combination |
|---|---|---|---|---|---|
| Ease of installation | ✓ | ✓ | ✓ | × | — |
| Adding functionality | ✓ | ✓ | ✓ | × | ×(application) |
| Cost | 1 | 2 | 4 | 3 | 5 (application) |
| Comfort of dialogue | 1 | 1 | 3 | 2 | — |
| Support of German | ✓ | ✓ | × | ✓ | ✓ |
| The companies' trustworthiness | 3 | 3 | 2 | 1 | 1 |
| Smart home integration | 1 | 1 | 2 | 2 | 2 |
| Number of applications | 1 | 2 | 3 | 4 | — |
| Own infrastructure | ✓ | ✓ | ✓ | ✓ | ✓ |
| Restrictions in development | 1 | 1 | 1 | 1 | 2 |
| Accuracy of queries understood | 2 | 2 | — | 1 | — |
| Support for other devices | 1 | 1 | 3 | 2 | 2 |
| Multimodality | ✓ | ✓ | ✓ | ✓ | ✓(application) |
| Location of voice commands | 4 | 3 | 2 | 1 | 2 (application) |
| Location of intelligence | 3 | 4 | 2 | 1 | 2 (application) |
| Security of data transmission | 2 | 2 | 2 | 1 | 2 (application) |
| Access to saved information | 1 | 1 | 2 | 2 | 2 |
| Authentication | 1 | 1 | 2 | 2 | 2 |
| Other possibilities of attack | — | — | — | — | — |
| Compliance to GDPR | ✓ | ✓ | ✓ | ✓ | ✓ |
| Fulfilled criteria (1) | 14 | 12 | 6 | 9 | 5 |
| Fulfilled criteria (1 and 2) | 16 | 16 | 13 | 14 | 13 |

**Tab. 5.21:** Evaluated criteria catalogue

# 6 Practical elaboration

In order to establish an understanding of how Snips can be used in the MyELGA project, this chapter provides a basic overview of the voice assistant. Furthermore, the specifics for developing an application with this assistant are presented. Focus is placed on the implementation of an Android application with Snips. For a sound comparison of the existing prototype with the newly developed one, the same basic scenarios and functionality are implemented. Finally, a comparison to the promised characteristics is conducted to show the relevancy of Snips for the MyELGA application. An evaluation of the prototype is thus conducted to show the accuracy of the assistant.

## 6.1 Introduction to Snips

As mentioned in the criteria catalogue, Snips is a voice assistant focused on user privacy. The assistant enforces security around sensitive data with a private-by-design approach. No voice commands are sent to a cloud for processing; thus, everything stays on the device by default. Recognizing the wake word, mapping speech to the intent, and managing a dialogue are all integrated locally into the assistant and can be freely used and adjusted by the developer. Therefore, the assistant works completely offline after the initial creation. Additionally, the wake word can be changed to any other word or sequence. Snips provides state-of-the-art deep-learning methods and data generation to help with the creation and matching of utterances to intents. The voice assistant can have multiple devices set up in the same home and can be used simultaneously on these devices. Also, these assistants can have dialogues through more than just a simple command instead, a connected conversation.

With the assistant not transmitting any data to a server, the training of the assistant must be completed beforehand. When a user creates an assistant with the Snips platform, the training is undertaken before the assistant is downloaded and used. After deploying the final application, a basic interaction with the voice assistant can be represented in these few steps:

1. User formulates a command like 'Turn the lights off'.
2. The deployed assistant turns the command into a structured representation (intent and slots).
3. An action is performed (e.g. turning the lights off).

Snips stands out even more in comparison to other voice assistants because of the size of the application. As the main goal is to be portable and relatively light while still offering security and privacy, significant effort was put into creating a voice assistant that can run completely on smart phones and other IoT devices. The need for a cloud or server to process speech is

removed and even with a small set of utterances the accuracy of the final implementation is high (F1-score of 93%) [Snip19p] [CSBB+18, p. 1-3].

## 6.2 Implementing MyELGA with Snips

From the perspective of a developer, the procedure to create a voice assistant with Snips can be summarized in a few steps. First an account on the Snips console website is needed [Snip19b]. Then the user logs in, and a voice assistant can be created. For the assistant to deal with different tasks and commands, predefined applications can be added, such as asking for the weather, adding groceries, and listening to music. If additional scenarios need to be developed, they can be implemented directly through the website. After having created the desired voice assistant it can be downloaded and immediately deployed in the used environment with all the combined functionality. These steps are explained with an example of the MyELGA prototype [Snip19b] [Snip19o].

### 6.2.1 Creating the voice assistant

To create a voice assistant a user needs to login to the website of Snips. There an assistant can simply be created through clicking on the button 'Create a New Assistant'. The user is prompted to choose a name for the assistant and the language to be used. Available languages currently are English, French, German, Italian, Japanese, and Spanish. For the prototype, the desired language was German. After setting up the assistant, the wake word can be chosen, and different applications can be added to the assistant. These applications can be either chosen directly out of the already existing ones or designed by the users themselves. As the MyELGA application did not exist beforehand a new one was created. In the process of creating a new application, the icon, name, and description can be changed. If the users want to publish their application on the Snips platform, changing these options is necessary. After the application is built, it can be immediately tested. Snips offers a test environment on the right-hand side. Tests can be executed by either using a microphone and saying the commands or by entering them into the textbox. As a result, the understood input, confidence, intent, and slots are presented. These can be filtered to simulate a scenario in which only a few intents are available. Intents can be either be enabled by default or must be enabled through the developer manually, such as if a multi-turn dialogue is intended for use.

The next step is to define the necessary intents. There are two tabs one is called 'Intents' and one is 'Actions'. Intents are created by using the web-interface by simply clicking on 'Create New Intent' or, if an already existing intent exists, by importing it. The actions-tab allows Python code to be entered to program the assistant's behaviour after understanding the intents. Actions can be either coded directly in web-interface, by importing Git repositories or by using predefined Home Assistant components. For this specific case, these were not investigated further, as working with Android requires a different approach [Snip19j] [Snip19b].

Now, the different intents must be created. A name and a description can be entered before the intent editor appears. Here, the users are asked to enter utterances that trigger the intent. Again, these examples can be imported and exported. As simple sentences usually are not enough to cover all needed functionality, slots can be added. These slots represent variables that can be extracted in the utterance. For the utterances to use these, a section must be

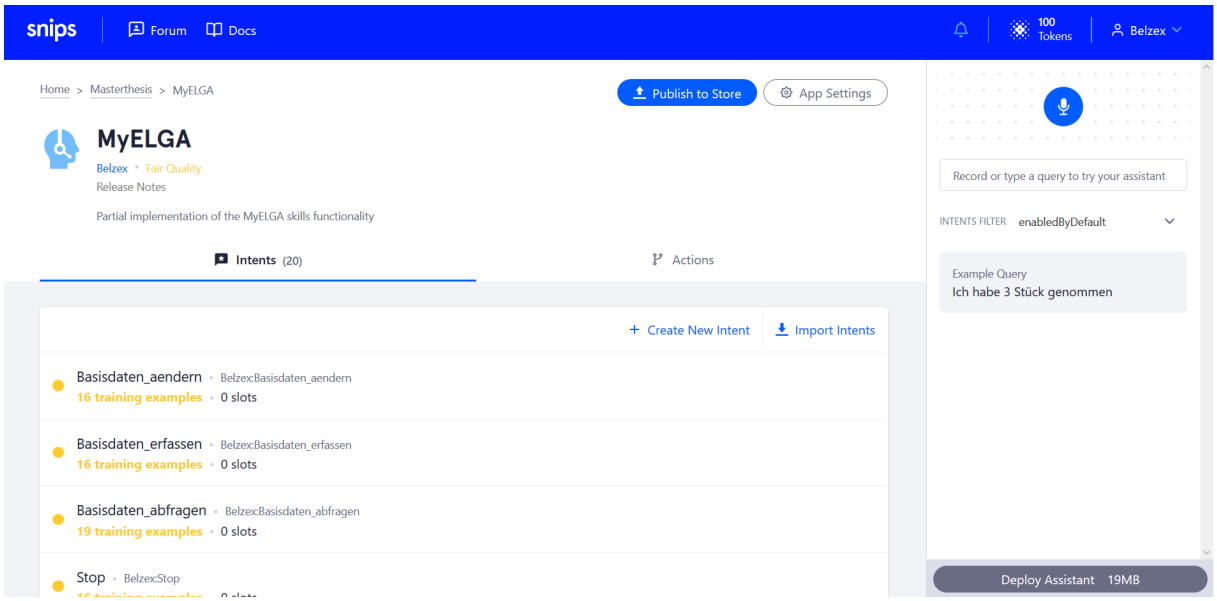


**Fig. 6.1:** Example of the console [Snip19b]

highlighted and the corresponding type selected. There are built-in slots like numbers, date and time, duration, and temperature. The benefit of using them is that no additional training must be done for the slot values to be recognized. In the case of MyELGA these are too few and do not cover all the needed characteristics of the values. Specific slot values like the brand of a medication or the form of said drug are necessary information to add a medication to the list of a user. These slots must be created and trained with values. This works similarly to the way intents are trained. Entries are added to the slot to help Snips understand the defined value. If problems arise when Snips tries to filter the slots, the strictness of the found value can be adjusted. Reducing this strictness results in likely values to be found fitting. This process can be undertaken when all the entries are distinct. However, if they are too similar, a stricter slot type is more fitting. When creating a slot, the developer can decide if the it is required. Sentences are searched for the different values and Snips only lets the user progress if all required information is provided. Should a user forget to enter necessary words or sequences the assistant can ask specifically for that value. The developer only needs to decide on a fitting prompt for when this case occurs and no further changes to the implementation must be done. Again, how well Snips recognizes these slots can be tested through the right-hand side of the console. It is important to check if the intent (with the desired slot) is enabled by default. Otherwise, a filter must be created for testing purposes. If the confidence is too low, more utterances should be provided for both slot and intent. The quality level can be seen next to the name of the intent. Red stands for poor, yellow for fair, and green indicates the best quality. The prototype of MyELGA implemented in Snips had a fair quality as only a small number of utterances was entered. For future tests and development, a higher amount of entries should be considered [Snip19b].

Thus, all necessary intents with their slots were created for the prototype. Once an intent is saved, the whole assistant is automatically trained with the new phrases. The next steps are to write code for what happens when certain intents are triggered and then download the assistant. If it was meant to be deployed on a Raspberry Pi, this step would be done through the actions-

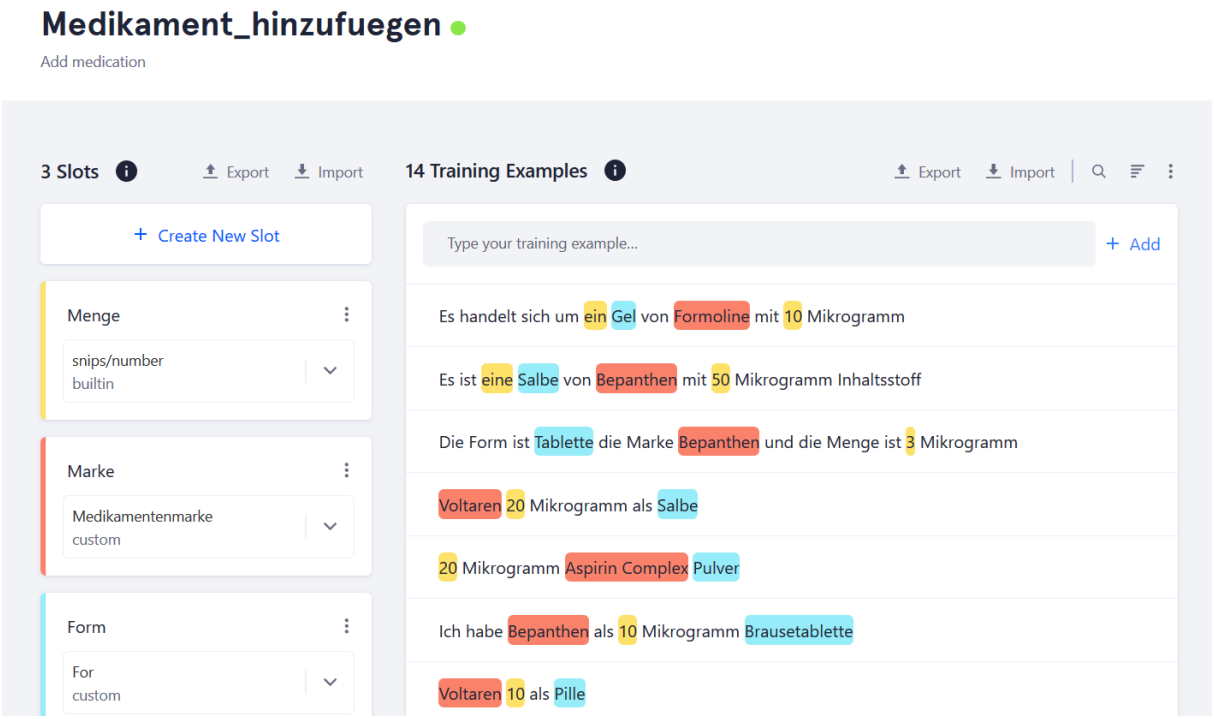


**Fig. 6.2:** Example of an intent (intent editor) [Snip19b]

tab on the website. For this specific case, there was no code in the web-interface as it would be directly written in the Android application. Therefore, this step was skipped, and the assistant was downloaded.

### 6.2.2 Deploying the voice assistant

Snips can be deployed on Raspberry Pi, Android, iOS, macOS, and Linux devices. As mentioned, depending on the used environment for the final implementation, the steps to realize such an application vary. Therefore, the researched scenario only covers an Android application. When comparing the different end devices, no restrictions in their functionality appeared.

For an Android application to be developed a few requirements must first be met: Android Studio is needed as the integrated developer environment, and the version required must be at least 3.0 or higher. The second necessity is the Java development kit with a version of 7.0 or higher and the Android API Level of 21 or higher. After ensuring that these conditions are fulfilled, the process of developing an Android application facilitating the earlier created Snips voice assistant is ready to begin [Snip19i].

When depicting the whole process to create the prototype as an Android application, not all the code snippets are presented. The basic idea and the most important parts are added to this thesis to help understand the concepts. Further instructions can be found in the guide provided by Snips [Snip19i]. First, a new Android Studio project is created to implement the voice assistant. The used API level must be at least 21 for all necessary functionality to work. After setting up the project, the library of Snips should be included in the structure. Therefore, the build.gradle (Module: app) file adds the Snips Nexus repository. Further, the Snips library should also be mentioned in the dependencies. The newest version of the library should be used.

```
repositories {
  maven {
    url "https://nexus-repository.snips.ai/repository/snips-maven-releases/"
  }
}
dependencies {
  // ...
    implementation('ai.snips:snips-platform-android:0.62.3@aar') {
  transitive = true
  }
}
```

With the basic settings configured, the voice assistant can be moved to the directory. When downloading the assistant, a zipped file is created. Snips recommends using a folder which located in the directory of the AndroidManifest.xml file. Further microphone access must be granted in the AndroidManifest file.

```
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

By using two helper methods the assistant is unzipped directly in the application. These methods are implemented in the **MainActivity** class and launched in the **onCreate()** method. Next the permission to record audio is required. The users are asked if they grant permission to the application to record audio. This asking can be done through a method called **ensurePermissions()**, which checks the permissions set, and if the users have not allowed for voice to be recorded yet, it prompts them to grant this permission. A second method is used to start the Snips components if all the needed permissions have been received.

With the necessary rights the Snips client can be created. This process consists of an instance of the Snips platform (client) and methods that work with the understood speech input. Different characteristics and settings can be changed when initiating the assistant. These revolve around enabling functionality (e.g. dialogue, hotword, logging and streaming). As these functions form the core and all later implemented functionality relies on them, they must be changed to fit the required assumptions.

The next snippet exhibits how the listener for the keyword is created. Every time the assistant recognizes the wake word, a session is started with the method **client.startSession()**. The first parameter is used by the TTS service and always said in response to the keyword being recognized. To define what intents are allowed to be recognized after hearing the wake word an ArrayList can be forwarded. The values added to the list must be strings structured like 'username:intentname'. This list is also called the intent filter. The third parameter defines the **canBeEnqueued** value. If it is set to true, the session starts when there is no pending session; if false, the session is dropped if there is already a running one. Last, a **customData** field is provided, which can be used to enter data not intended to be saved otherwise. As long as the session continues, the customData field is forwarded and can be extracted.

```
client.setOnHotwordDetectedListener(new Function0<Unit>() {
  @Override
  public Unit invoke() {
    // Wake word detected, start a dialog session
    Log.d(TAG, "Wake word detected!");
    client.startSession(null, new ArrayList<String>(),
      false, null);
    return null;
  }
});
```

The most crucial component for handling the actions is presented in the next snippet. The listener waits for intents to be understood and can initiate different tasks depending on the value received. By taking the name of the intent, the developer can distinguish between different scenarios by using a switch-case structure or a simple if-else. The conversations must be managed by the developer. A possible way to do so is with the customData field or variables set in the application. The **client.endSession()** needs the current ID of the session to terminate it and again has a field for the TTS service. The text entered as this parameter is provided to the user if the session is terminated.

```
client.setOnIntentDetectedListener(new Function1<IntentMessage, Unit>() {
  @Override
  public Unit invoke(final IntentMessage intentMessage) {
    // Intent detected, so the dialog session ends here
    client.endSession(intentMessage.getSessionId(), null);
    Log.d(TAG, "Intent detected: " +
      intentMessage.getIntent().getIntentName());
    return null;
  }
});
```

As an example of how a dialogue can be managed, if the intent 'Basisdaten_erfassen' is recognized in the MyELGA prototype, the session is continued, and an answer is presented to the user by the **client.continueSession()** method. The last value is again the ArrayList which holds the value for the intent that recognizes different names. Therefore, it is only possible to have answers recognized in that context. If no valid value is retrieved, the assistant repeats the prompt. However, if an answer is given without a valid slot, the assistant takes the defined question from the slot to ask specifically for the value. It is also possible to set the action type to a notification. The only difference is that no reply is expected of the user. This action can be used to inform the person using the device or provide additional feedback at certain steps in a conversation.

```
 case "Belzex:Basisdaten_erfassen":
  client.continueSession(intentMessage.getSessionId(), "Welcome to the Snips voice assistant. Could
break;
```

Additionally, there are other methods that provide information like when the assistant is started, an error occurs, and a simple logging method. These are not as interesting as the other occurrences and are more straight forward, but they are still mentioned here for completeness [Snip19i] [Snip19c] [Snip19h].

### 6.2.3 Database connection

With the important data in the MyELGA database, it was necessary to show that it is possible to retrieve data from said database with an implementation of Snips. Special focus was given to create a connection to the database without allowing all traffic directed towards it.

The approach selected to achieve this goal is a combination of a PHP script located on an Apache web server and a MySQL database. To establish this system, XAMP was used to locally create the server and database. The greatest benefit is that the credentials for accessing the database are located on a server and all access can be managed directly there. If a person was to reverse-engineer the application and the credentials were directly in there (e.g. JDBC-connector), it would pose a larger threat than having the data on a secured server. It is still necessary to establish a high-level of security on the server as there are other attack scenarios that break into the web server, but for the Android application the weak spots have been reduced.

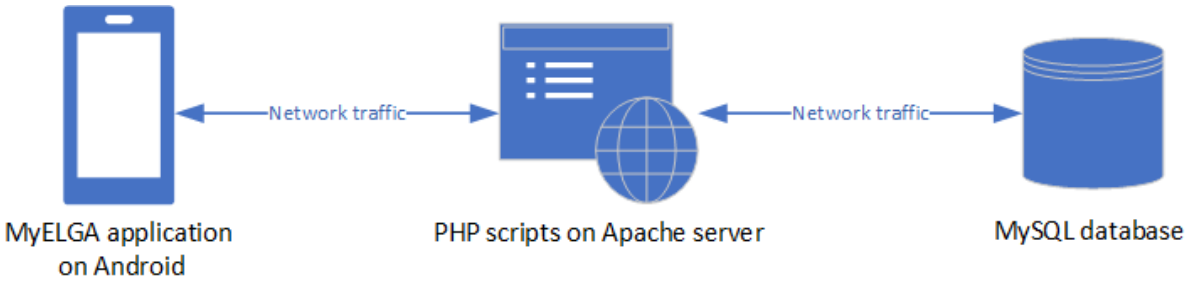


**Fig. 6.3:** Accessing the database with the Android prototype

Starting with the Android application - the first step was to add permission to use the Internet and implement the OkHttp client. Internet-permission is needed to access the Apache web server and receive the JSON-reply. OkHttp serves as the supporting service between the server and application. It offers the functionality to establish the connection and manage the packages being delivered. The support of different TLS versions is especially beneficial for future development. Accessing the web server is achieved through an asynchronous task that builds a request directed towards the URL of the PHP script on the server. A reply is expected in the form of JSON, which encompasses all the information saved on the database.

On the web server PHP is enabled, and a script establishing a connection to the database is created there. All the credentials for the connection are in the PHP file; thus, access to the web server must be restricted and controlled, or any person accessing this file would have direct access to the database. The TLS is one way of doing so, but other approaches can be pursued with the server directly managed by the developer. The only action directed towards the MySQL database is to retrieve all the information in the table and return the data as a JSON-reply.

The database is then filled with a table and generic data to simulate a user and their medication. As the goal concentrates on only showing that access was possible, the data have no relation to real people and did not resemble any implemented structure. The used values were name, weight, height, and medication. These items were simple integers and strings. The database was configured to only allow access initiated by the web server [Apaa19] [Squa19].

## 6.3 Interaction models

To show that Snips can process the same scenarios as Alexa, three functionalities were selected and implemented. The task was as close to the initial interaction model as possible to provide a foundation for comparison. These models were taken from the work of Klade [Klad19, p. 53-66] and altered slightly. One such alteration is the translation to English. Although the initial Alexa skill and Android prototype are both implemented in German, English is used for the evaluation and presentation in this thesis. Furthermore, the Android application used a textbox to give additional information on the status of the conversation, which is a slightly compromised text of the used voice response. This aspect is also not included in the interaction model as it is not available in the Alexa skill. Overall, 20 intents were used across three different scenarios. Each interaction model is described briefly, and the used intents and slots are mentioned.

Before examining the different models, a few things must be clarified. Snips does not offer a predefined intent for the answers 'Yes', 'No' and 'Cancel'. One way to solve this problem is by importing intents from a different application accessible through Git or the Snips website. For this project, these answers were self-made. The cancel-intent was added to all intents so the user could abort any action. This interaction is not explicitly depicted in the models to present a clearer picture of the dialogues.

### 6.3.1 Acquisition of personal data

The first functionality focuses on the patients entering their data for the first time. This process is split into several parts and managed as a continuous dialogue which can be seen in Figure 6.4. By triggering the 'Basisdaten_erfassen' intent, the process begins. Phrases that cause the

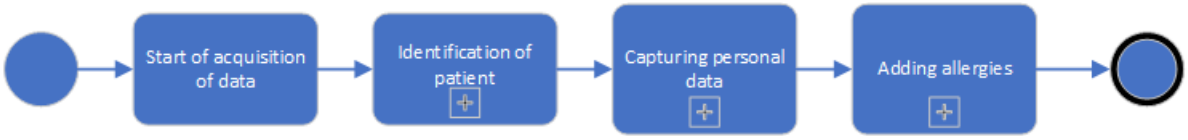


**Fig. 6.4:** Overview of the acquisition process [Klad19, p. 56]

intent to be executed are, for example 'Start the setup' or 'I want to change my data'. After the initial start, the user is required to identify their person through their social security number and birthdate. If this step is finished, the next steps ask the patients to provide personal data on their height, weight, and allergies. By completing all steps, the process is finished, and the data are saved [Klad19, p. 55-61].

In more detail the first subprocess asks the users for their name or nickname. This name is later used to refer to the patient and creates a more personalized interaction. Then, the users are asked to state the first four numbers of their social security number. If a valid value is detected, the conversation is continued, and the assistant asks for their birthday. The users then reply with the date. Again, the value is checked for correctness and the assistant repeats the captured social security number and birthday to the user. The assistant then asks if the understood data is correct. The user can answer with 'Yes' which triggers the next process and saves the currently collected data. If information was interpreted incorrectly or something is amiss, the user can say 'No' which repeats the whole process by prompting the user to enter the four digits of their social security number. Thus, the user can correct the different values.

Something to be considered at this point is that if a slot is not recognized or completely missed, the specific message that was created in the web-interface is prompted. Therefore, this interaction needs not be directly implemented in the source-code. Additionally, Snips is configured to return an error message if no valid message is understood after several attempts. As this error rarely occurs, it is not further implemented in this prototype. When an error occurs, the dialogue is reset, and the user can initiate another conversation.

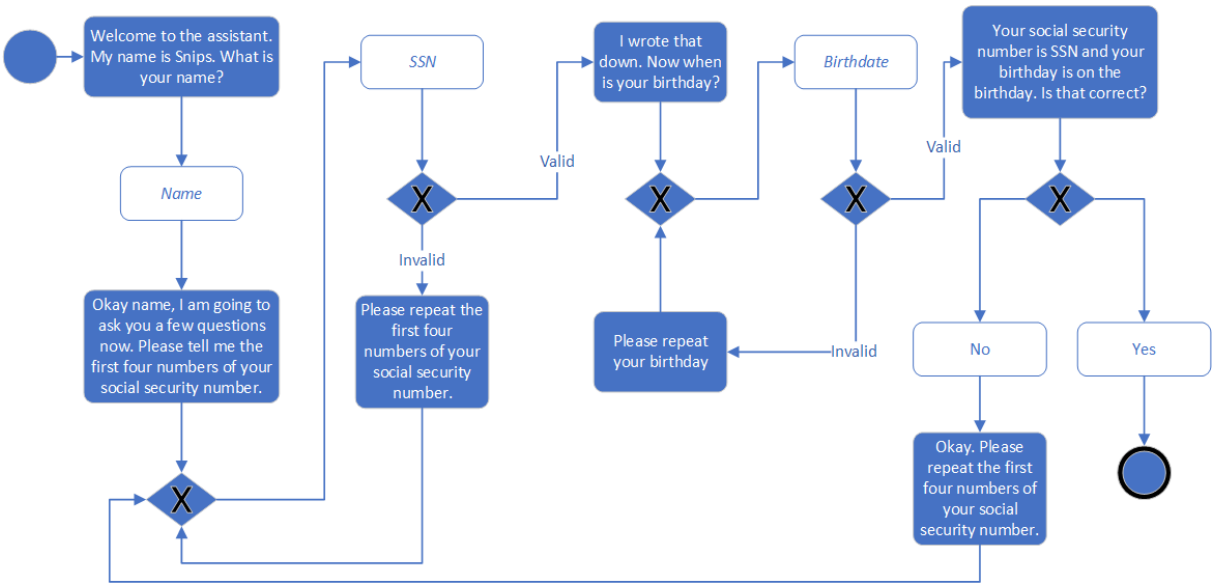


**Fig. 6.5:** Identification of patient [Klad19, p. 56]

To implement the identification process, the following intents with the corresponding slots were used:

- Starting the process: Basisdaten_erfassen
- Name: Namen_anpassen (Custom slot type: name)
- SSN: SVNR_anpassen (Built-in slot type: number)
- Birthdate: Geburtsdatum_anpassen (Built-in slot type: datetime)
- Yes: Ok
- No: NOk

Answers like 'Yes' and 'No' change the results and how they continue or end the session by checking the intent that was used before. Although, the customData field could be used for management, simple variables of the Boolean type are used to save the current state of the conversation and the intents that have already been executed.

The next subprocess aims to capture the personal data of the patient and is structured shown in Figure 6.6. Overall, the procedure works similarly to the one above. After completing the process of identification, the user is asked to state their height. By entering a valid value for height, the assistant then asks the user to state their weight. As seen in the subprocess earlier, the patients are asked again if the captured personal data is correct. By declining, the process

is repeated, and the values can be changed. Otherwise, the data is saved, and the process continues.

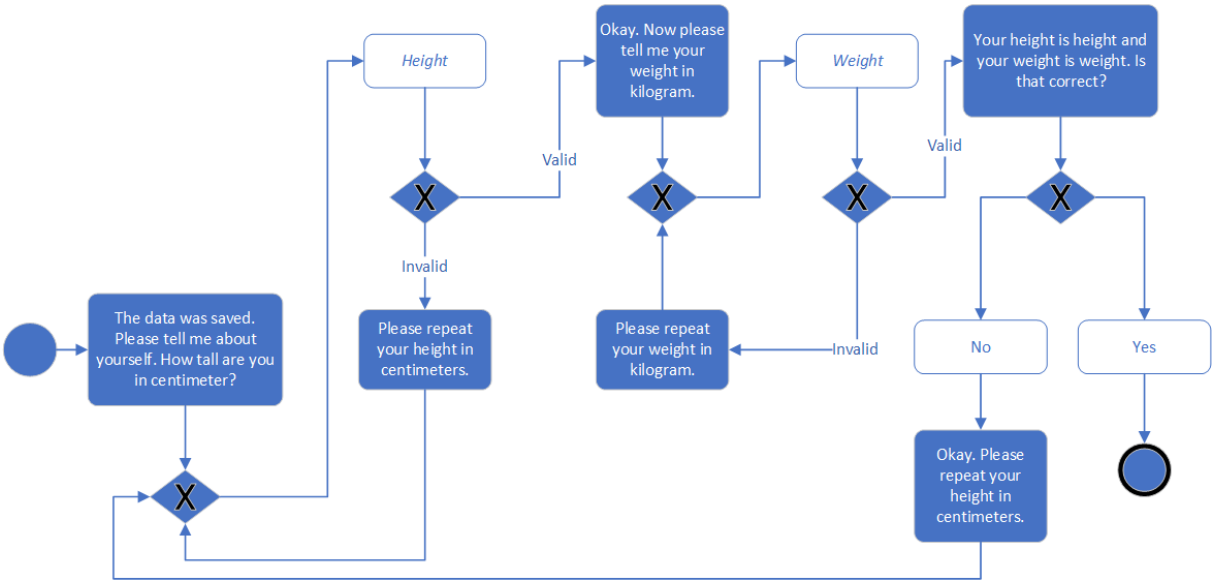


**Fig. 6.6:** Capturing personal data [Klad19, p. 57]

The implementation is accomplished through two new intents that change the height and weight. The slot type for height and weight is the built-in number from Snips; thus, the measurement must be defined. For height, the chosen dimension is centimetre, and for the weight it is kilogram. Up until now the process has been identical to the prototype from Klade [Klad19].

- Height: Groesse_anpassen (Built-in slot type: number)
- Weight: Gewicht_anpassen (Built-in slot type: number)
- Yes: Ok
- No: NOk

When all the previously mentioned subprocesses have executed correctly the last step is to add allergies. The patients are asked if they have an allergy and if yes, they are instructed to name the allergy, which is then saved corresponding to the user. As it is possible to have multiple allergies, the patient can also name an additional intolerance or allergy. This process is different to the original model because of the removal of a selection for the type of allergy. Doing so would have simply split the procedure into doing the same thing twice.

With only one type of allergy, the number of intents is smaller as food and drug allergies are connected in one intent. The used intents are as followed:

- Allergy: Allergie_hinzufuegen (Custom slot type: allergy)
- Yes: Ok
- No: NOk

As the allergies and intolerances are now in one single slot, it must be adjusted. The custom slot is trained to recognize some examples. Here the use of synonyms is useful as many allergies do come with a similar expression. One example is lactose intolerance, whose equivalent expression

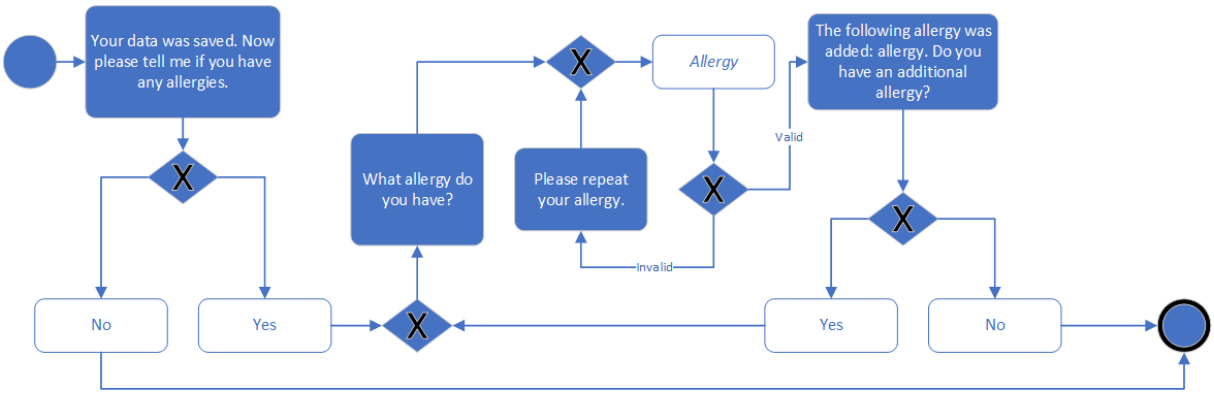


**Fig. 6.7:** Adding allergies [Klad19, p. 58]

would be milk intolerance. Both terms should be mapped to the same slot value which is done with the help of synonyms in the web-interface.

### 6.3.2 Adding medication

The second scenario implemented centres around finding and adding medication. This action is a subprocess of the original process 'Management of medication' [Klad19, p. 61-63]. The basic concept of this interaction model is for patients to provide form, brand, dosage, and amount of a medication they want added to their profile. Furthermore, this process is used for the intake of medication and retrieving information on the drug. To show that more than one slot can be effectively used in one intent this scenario was chosen for implementation in the Android prototype.
The users provide the different characteristics of a drug in one single phrase. Any slots that are missing the assistant directly asks for. Therefore, if the necessary information is understood by the application, the assistant asks if the result is correct. Assuming the right medication was found the next step is to state the amount. The only difference to the original application is that there must always be a value for the amount. Thus, if no data should be saved as the stock, the user would have to say zero.

The medication added through this process is not checked in the background for correctness. Any valid data understood by the assistant is saved locally for later use. The prototype designed by Klade adds original data (e.g., descriptions and side-effects) if the medication is recognized. This check can easily be implemented as an additional method that uses the medication provided by the user. If the result is valid, the interaction created by the no-intent could be used.

Otherwise, this process is almost identical to the original interaction model. The use of multiple slots is solved rather easily through Snips and offers enough comfort for the users, who do not have to repeat the characteristics of their medication.

- Starting the process: Starte_Medikament_hinzufuegen
- Medication: Medikament_hinzufuegen (Custom slot type: form, brand and dosage)
- Amount: Bestand_festhalten (Built-in slot type: number)
- Yes: Ok
- No: NOk

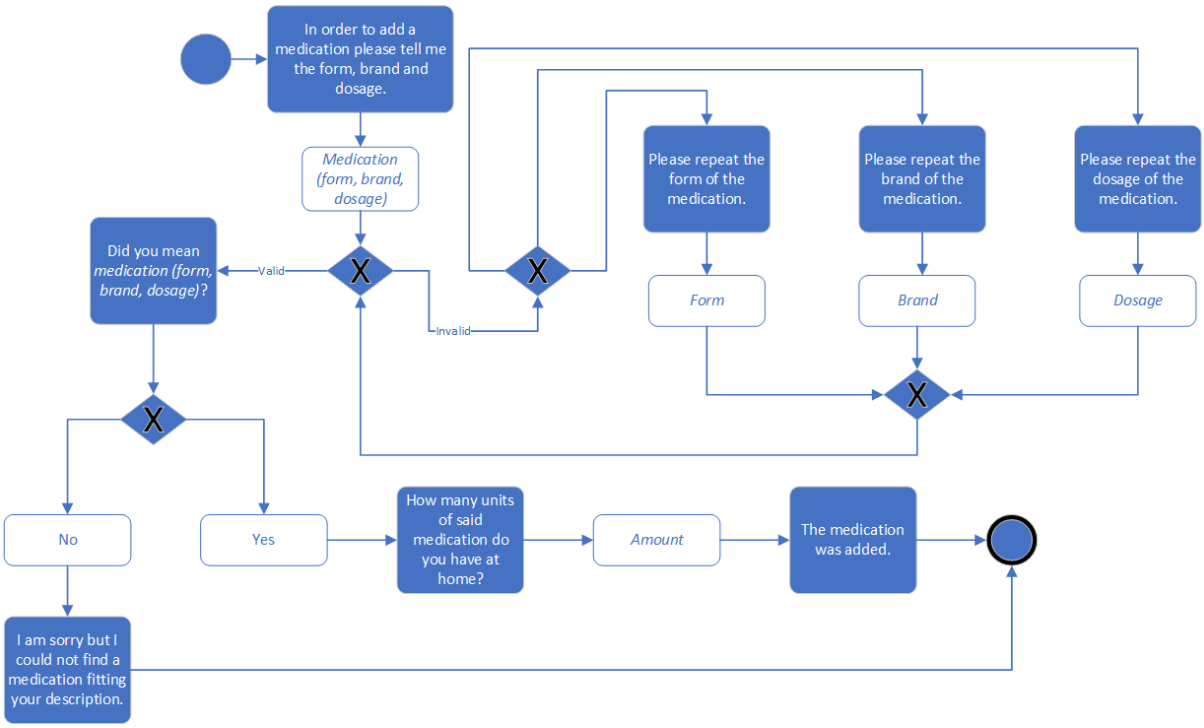


**Fig. 6.8:** Adding medication [Klad19, p. 66]

### 6.3.3 Taking medication

Lastly the scenario of taking a certain medication was created. It is also part of the 'Management of medication' process. After the patients state that they took a medication, they are asked for the exact details of it. This process is identical to the one before. Form, brand, and dosage are needed to identify the drug. The users can then confirm if the found medication is the one, they consumed. After the details are provided the locally saved medications linked to the patient are compared to the already identified drug. If they are the same, the process is continued, and the intake can be stated. The amount of the medication is reduced by the intake of the patient and saved. However, if the user did not add the medication to their profile, no intake can be saved. The process ends with the assistant telling the patient that no such drug was found.

In order to implement this scenario, almost all the same intents were used as seen before. The only differences are that the amount works differently now. Earlier, the stock was defined in the scenario of adding medication which is now reduced by the amount. Also, the start of the conversation must be adjusted to fit the context.

- Starting the process: Bestand_aendern_starten
- Medication: Medikament_hinzufuegen (Custom slot type: form, brand, and dosage)
- Amount: Bestand_aendern (Built-in slot type: number)
- Yes: Ok
- No: NOk

Overall, these scenarios fulfilled their intended tasks and demonstrated that the actual implementation of the defined prototype by Klade is possible and feasible with Snips. Further works

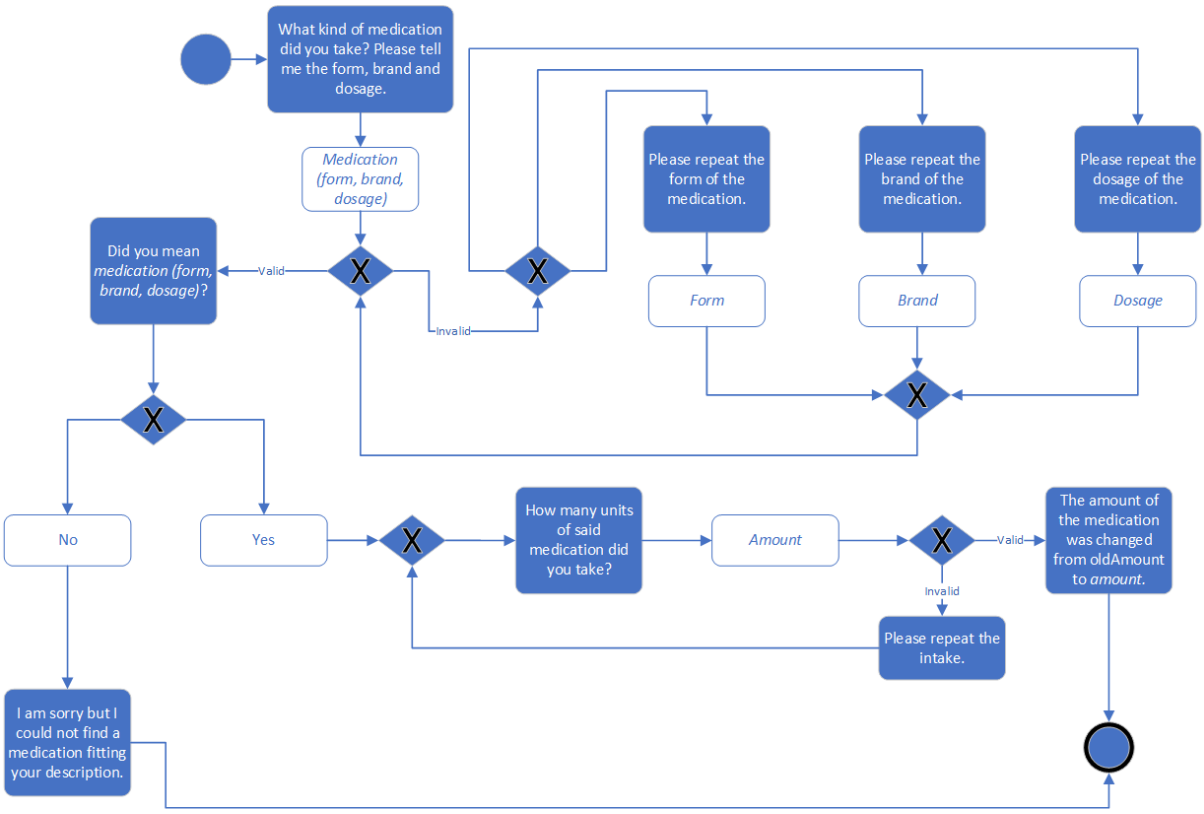


**Fig. 6.9:** Taking medication [Klad19, p. 63]

should focus on recreating all missing scenarios and functionalities and integrate the complete structure of the data. The data model that encompasses all the patient's information and the medication especially need to be realized.

## 6.4 Evaluation of the prototype

Snips has a 'Unit Test' tool that allows testing the robustness of the voice assistant. These unit tests are recordings of users that test certain intents. Audio files can either be imported or directly recorded with the web-interface. Each voice sample has its own entry and is checked separately. The developer chooses the intent that should be triggered by the entry. If a result is expected, it can be written into the entry, and slots can be tagged [Snip19t].

For testing the prototype eight people were asked to trigger five different intents. The test was held in a room with little to no background noise, and the phrases to trigger each intent were shown to every participant. They all thus possessed the same basis for creating valid sentences. The only exceptions were the slots as these were up to the individual person to decide upon. Phrases were recorded through the web-interface of the Snips platform which was opened on a PC. Of the eight people five were male and three were female. The selected intents were responsible for capturing the name (Namen_anpassen), starting the acquisition of data (Basisdaten_erfassen), documenting the intake of medication (Bestand_aendern_starten), setting height/weight (Groesse_anpassen/Gewicht_anpassen) and approving (Ok) or disapproving(NOk).

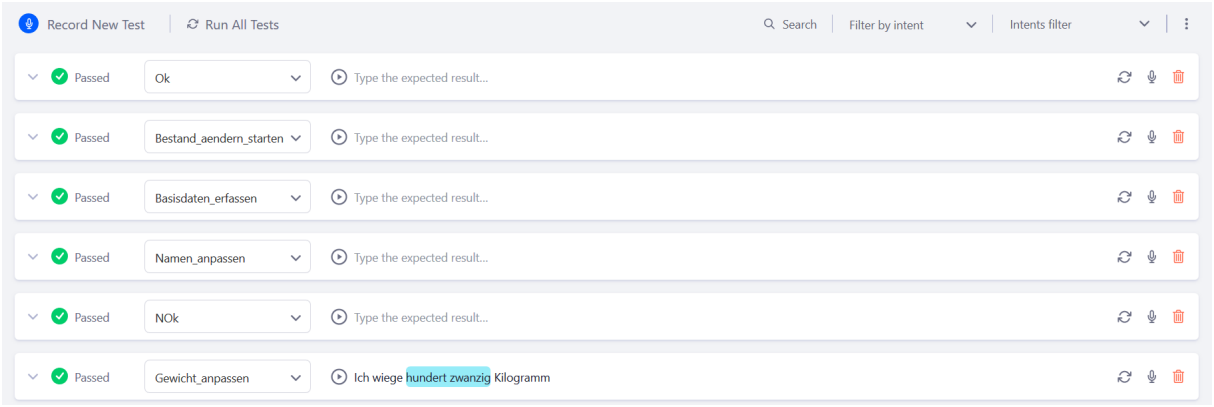


**Fig. 6.10:** Unit test [Snip19t]

Of 40 tests, 37 total were passed and triggered the right intent and slots. The other three tests each had a distinct error. The first was as a user was trying to change their height. Snips recognized nothing in the voice file and returned 'unknownword' as the response, possibly due to too much background noise or sloppy pronunciation of the sentence. It may also be that the person was too far away from the microphone. Another error that occurred, was when a user tried to start the 'Basisdaten_erfassen' intent. The assistant understood 'anpassen', which is not used in this intent. Thus, the assistant triggered a different intent which had the word as an utterance. The last problem was as a user tried to enter their name, and the assistant recognized the value 'null'. With no additional information understood, Snips recognized the value as a number and interpreted that the birthday should be set. Thus, the 'Geburtsdatum_anpassen' intent was triggered, which resulted in a failed test.

In the end, two utterances were misunderstood by the assistant and could not be further checked for correctness. The other phrase provided a clue for improving the robustness of the assistant. Adding more utterances and making them more concise is important for always triggering the proper intent. For future improvements of the model, it is recommended to increase the number of utterances, but otherwise, the evaluation demonstrated that the assistant was able to understand and process most phrases. There does not exist a necessity to make major changes to the assistant as the functionality currently implemented is interpreted correctly. Slot values as well as the intents are accurate enough although only a small set of training data was provided.

## 6.5 Comparison to promised characteristics and Alexa

The relevance of Snips for the MyELGA project is ascertained by the actual abilities of the assistant. As it is possible for a company to describe their product in many ways, a thorough evaluation of the functionalities had to be conducted first. This evaluation was executed by recreating the scenarios and integrating certain necessary characteristics that were desired in the final product. The set of criteria provided a first impression for what needed to be examined. Special focus was given to the characteristics crucial for choosing Snips over the other assistants. Those characteristics were the support of German, the accuracy of the queries, the access to the database and most importantly the location of the commands and intelligence. With the prototype as an example, these criteria are evaluated again to verify their correctness.

**Support of German** After configuring the assistant to work with German no further settings had to be adjusted. The text-to-speech functionality as well as the ASR worked with German and understood most words entered. The availability of built-in slots, like numbers and dates, offered in various languages makes it much easier to develop a voice assistant with Snips. This is a necessary feature that any voice assistant had to fulfil to be feasible for this project.

**Accuracy of queries** The only information on the accuracy of the voice assistants was the study conducted by Coucke et al. [CSBB+18]. To test the ASR and NLU of Snips in this thesis, the previously mentioned test was issued in the Section 6.4. With eight users testing the robustness of the model, it could be shown that the Snips assistant was accurate with the interaction models used. No indication was found that the criterion was not sufficiently fulfilled.

**Access to database** Accessing the database without allowing all access was a problem the previous prototype experienced. By implementing a connection that did not impose any threats to the security of application, web-server, or database, this criterion could be fulfilled. Snips does allow the integration of an infrastructure so that security can be retained and managed by the developer.

**Location of commands and intelligence** As Snips claims to work completely offline, the created prototype was also tested without the permission set to use the Internet and with the phone being disconnected from any network. This test worked without problem if the connection to the MySQL database was excluded. In hindsight, no commands would thus be transmitted to an external service, and everything would indeed be located on device, opposed to Alexa, which saves all valid voice commands and uses them to train the assistant.

The prototype was able to fulfil all the criteria established beforehand and did not falsify any findings for the other criteria.

# 7 Conclusion

The results of this study have shown that although not all voice assistants are ideally for the use in a medical environment, many do provide a vast amount of functionality that can help develop applications directly used by users with their voice. Multimodality was also featured in many devices, and it can help improve usability and open the devices to other uses. With the trend of voice-purchasing and more complicated tasks it is necessary for voice assistants to constantly evolve and offer new functionality. One notable characteristic is the way many assistants process their data. Amazon's Alexa and Google Assistant, especially are known for sending all their voice data to an external server. Although these are supposedly secure, the question arises if assistants that work completely on device are more trustworthy and reliable. With the focus on eHealth and medical environments, this external server does pose a problem. Therefore, the use of assistants like Snips, Mycroft, and a self-made voice assistant are better suited.

The research demonstrated that the most significant drawback for voice assistants is the offered comfort, increase in costs, and lack of functionality, overall. Developers must place significantly more effort into creating an assistant than they would if they used a more well-known assistant. The created criteria catalogue was evaluated, and it highlighted that Alexa meets the highest number of criteria. Overall, Alexa is the most polished assistant, closely followed by the Google Assistant. With only small differences, both voice assistants fulfil the same criteria. However, it should be further investigated if Google implements their assistant to work completely offline. Such a feature could change the evaluation and outcome, with Google Assistant becoming a viable option. As private medical data on patients is processed by the application, an emphasis had to be placed on the criteria that centred on security. To this end, Snips was the best voice assistant of the study and used for the practical elaboration.

With Snips in place, a sound prototype could be created that showed the necessary functionality required for further use. The application was deployed on an Android device and worked immediately. Scenarios that established the core of the MyELGA functionality were taken from the previously created prototype by Klade [Klad19] and used as test examples for the new prototype. These scenarios were able to work in a similar matter, and although only dummy data was used, the previous criteria could be evaluated for correctness. With the real implementation and findings in the criteria catalogue aligning, the findings were deemed correct. If the project is continued, the use of more test data for the slots and utterances is recommended. Although the tests did show promising results, the last few inaccuracies of the voice assistant should be terminated this way. Finally, Snips is a feasible and viable solution for MyELGA and should be considered for future development.

# List of Figures

# List of Tables

# Table of Abbreviations

| | |
|---|---|
| AAL | Ambient Assisted Living |
| API | Application programming interface |
| ASR | Automatic Speech Recognition |
| AWS | Amazon Web Service |
| CDA | Clinical Document Architecture |
| DDoS | Distributed Denial of Service |
| eHealth | Electronic Health |
| ELGA | Austrian Health Record |
| GDPR | General Data Protection Regulation |
| GUI | Graphical User Interface |
| HL7 | Health Level Seven |
| IoT | Internet of Things |
| IT | Information Technology |
| JDK | Java Development Kit |
| NLU | Natural Language Understanding |
| PDA | Personal digital assistant |
| RDS | Relational Database Service |
| TLS | Transport Layer Security |

# Bibliography


[ACJB+14] Awni Y. Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng: Deep Speech: Scaling up end-to-end speech recognition. In: *CoRR*, abs/1412.5567 (2014).

[AFHV17] A. Sivanathan, F. Loi, H. H. Gharakheili, V. Sivaraman (Hrsg.): Experimental evaluation of cybersecurity threats to the smart-home: 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) (2017).

[Alph19] Alpha200: Skill für Snips.ai zur Ansteuerung von Geräten mit OpenHAB. `https://github.com/Alpha200/snips-openhab` (visited on 25.07.2019).

[Amaz19a] Amazon.com: About Alexa Voice Profiles. `https://www.amazon.com/gp/help/customer/display.html?nodeId=202199440` (visited on 25.07.2019).

[Amaz19b] Amazon.com: Alexa and Alexa Device FAQs. `https://www.amazon.com/gp/help/customer/display.html?nodeId=201602230` (visited on 25.07.2019).

[Amaz19c] Amazon.com: Alexa Built-in Devices. `https://www.amazon.com/b?ie=UTF8&node=15443147011` (visited on 25.07.2019).

[Amaz19d] Amazon.com: Alexa Connected Devices. `https://developer.amazon.com/alexa/connected-devices` (visited on 25.07.2019).

[Amaz19e] Amazon.com: Alexa, Echo-Geräte und Ihre Privatsphäre. `https://www.amazon.de/gp/help/customer/display.html?nodeId=GA7E98TJFEJLYSFR` (visited on 25.07.2019).

[Amaz19f] Amazon.com: Amazon Echo (2. Gen.). `https://www.amazon.de/gp/product/B06ZXQV6P8?ref=ODS_v2_FS_AUCC_rd&th=1` (visited on 25.07.2019).

[Amaz19g] Amazon.com: AVS UX Setup and Authentication. `https://developer.amazon.com/docs/alexa-voice-service/setup-authentication.html` (visited on 25.07.2019).

[Amaz19h] Amazon.com: Build Skills with the Alexa Skills Kit. `https://developer.amazon.com/docs/ask-overviews/build-skills-with-the-alexa-skills-kit.html` (visited on 25.07.2019).

[Amaz19i] Amazon.com: Connect Smart Home Devices to Alexa. `https://www.amazon.com/gp/help/customer/display.html?nodeId=201749240` (visited on 25.07.2019).

[Amaz19j] Amazon.com: Context. `https://developer.amazon.com/docs/alexa-voice-service/context.html` (visited on 25.07.2019).

[Amaz19k] Amazon.com: Create Skills for Alexa-Enabled Devices With a Screen. https://developer.amazon.com/de/docs/custom-skills/create-skills-for-alexa-enabled-devices-with-a-screen.html (visited on 25.07.2019).

[Amaz19l] Amazon.com: Data Privacy. https://aws.amazon.com/en/compliance/data-privacy-faq/ (visited on 25.07.2019).

[Amaz19m] Amazon.com: Dialog Interface Reference. https://developer.amazon.com/docs/custom-skills/dialog-interface-reference.html (visited on 25.07.2019).

[Amaz19n] Amazon.com: Enable Alexa Skills. https://www.amazon.com/gp/help/customer/display.html?nodeId=201848700 (visited on 25.07.2019).

[Amaz19o] Amazon.com: Getting Started with Amazon RDS. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.html (visited on 25.07.2019).

[Amaz19p] Amazon.com: Manage Voice Purchasing Settings. https://www.amazon.com/gp/help/customer/display.html?nodeId=201952610 (visited on 25.07.2019).

[Amaz19q] Amazon.com: Notifications Overview. https://developer.amazon.com/docs/alexa-voice-service/notifications-overview.html (visited on 25.07.2019).

[Amaz19r] Amazon.com: Security Requirements. https://developer.amazon.com/docs/alexa-voice-service/security-best-practices.html (visited on 25.07.2019).

[Amaz19s] Amazon.com: Set Up Your Echo (2nd Generation). https://www.amazon.com/gp/help/customer/display.html?nodeId=202189140 (visited on 25.07.2019).

[Amaz19t] Amazon.com: Steps to Build a Custom Skill. https://developer.amazon.com/docs/custom-skills/steps-to-build-a-custom-skill.html (visited on 25.07.2019) (visited on 25.07.2019).

[Amaz19u] Amazon.com: Supported Alexa Features by Country for International Version Echo Devices. https://www.amazon.com/gp/help/customer/display.html?nodeId=202207000 (visited on 25.07.2019).

[Amaz19v] Amazon.com: Turn on Follow-Up Mode. https://www.amazon.com/gp/help/customer/display.html?nodeId=202201630 (visited on 25.07.2019).

[Amaz19w] Amazon.com: Tutorial: Konfigurieren einer Lambda-Funktion für den Amazon RDS-Zugriff in einer Amazon VPC. https://docs.aws.amazon.com/de_de/lambda/latest/dg/vpc-rds.html (visited on 25.07.2019).

[Amaz19x] I. Amazon.com: Use Local Voice Control with Offline Echo Devices. https://www.amazon.com/gp/help/customer/display.html?nodeId=GCC6XV9DX58VW5YW (visited on 25.07.2019).

[Amme18] E. Ammenwerth: From eHealth to ePatient: The Role of Patient Portals in Fostering Patient Empowerment. In: *An Official Journal of the European Federation for Medical Informatics*, 14, 2 (2018), 20–23.

[Ande16] V. P. Andelfinger: eHealth: Grundlagen und Bedeutung für die Gesundheitssysteme heute und morgen. *In: V. P. Andelfinger, T. Hänisch (Hrsg.), eHealth : Wie Smartphones, Apps und Wearables die Gesundheitsversorgung verändern werden*,

Springer Fachmedien Wiesbaden, Wiesbaden (2016), 25–29, https://doi.org/10.1007/978-3-658-12239-3_5.

[Apaa19] Apache Friends: XAMPP Apache + MariaDB + PHP + Perl. https://www.apachefriends.org/de/index.html (visited on 25.07.2019).

[Appa19] Apple Inc.: Apple HomePod. https://www.apple.com/de/shop/buy-homepod/homepod (visited on 25.07.2019).

[Appb19] Apple Inc.: Xcode 11. https://developer.apple.com/xcode/ (visited on 25.07.2019).

[Auth19] Auth0: Auth0 Overview. https://auth0.com/docs/getting-started/overview (visited on 25.07.2019).

[BBHL+18] F. Bachner, J. Bobek, K. Habimana, J. Ladurner, L. Lepuschütz, H. Ostermann, L. Rainer, A. E. Schmidt, M. Zuba, W. Quentin: Austria: Health system review 2018. Health Systems in Transition. In: (2018).

[Bedf17] J. Bedford-Strohm: Voice First? Eine Analyse des Potentials von intelligenten Sprachassistenten am Beispiel Amazon Alexa. In: *ComSoc Communicatio Socialis*, 50, 4 (2017), 485–494.

[Bell13] J. R. Bellegarda: Large-scale personal assistant technology deployment: The siri experience. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* (2013).

[Blas18] B. Blass: Wie Alexa & Co. den Alltag verändern. In: *Der Freie Zahnarzt*, 62, 11 (2018), 42–44, https://doi.org/10.1007/s12614-018-7534-0.

[Bund12] Bundesministerium für Digitalisierung und Wirtschaftsstandort: Gesamte Rechtsvorschrift für Gesundheitstelematikgesetz 2012, Fassung vom 04.04.2019. https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20008120 (visited on 25.07.2019).

[Cand17] Candid Wueest: A guide to the security of voice-activated smart speakers: An ISTR Special Report. https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/istr-security-voice-activated-smart-speakers-en.pdf (visited on 25.07.2019).

[CMUS19a] CMUSphinx: Building an application with PocketSphinx. https://cmusphinx.github.io/wiki/tutorialpocketsphinx/ (visited on 25.07.2019).

[CMUS19b] CMUSphinx: Building an application with sphinx4. https://cmusphinx.github.io/wiki/tutorialsphinx4/ (visited on 25.07.2019).

[CMUS19c] CMUSphinx: CMUSphinx: Installing on Raspberry Pi. https://cmusphinx.github.io/wiki/raspberrypi/ (visited on 25.07.2019).

[CMUS19d] CMUSphinx: CMUSphinx Tutorial For Developers. https://cmusphinx.github.io/wiki/tutorial/ (visited on 25.07.2019).

[CMUS19e] CMUSphinx: Frequently Asked Questions (FAQ). https://cmusphinx.github.io/wiki/faq/ (visited on 25.07.2019).

[Conr17] C. S. Conrad: Künstliche Intelligenz — Die Risiken für den Datenschutz. In: *Datenschutz und Datensicherheit - DuD*, 41, 12 (2017), 740–744.

[CSBB+18] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, J. Dureau: Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces (2018).

[Dani19] Daniel Berger: Microsoft-Chef: Cortana ist keine Konkurrenz mehr für Alexa und Google. https://www.heise.de/newsticker/meldung/Microsoft-Chef-Cortana-ist-keine-Konkurrenz-mehr-fuer-Alexa-und-Google-4282240.html (visited on 25.07.2019).

[DFKI19] DFKI GmbH: Mary Text To Speech Overview. http://mary.dfki.de/documentation/overview.html (visited on 25.07.2019).

[Dure18] J. Dureau: Private & Context-Aware Speech Recognition with Snips. https://medium.com/snips-ai/coming-soon-private-context-aware-speech-recognition-with-snips-8985624a5fb7 (visited on 25.07.2019).

[ELGA19a] ELGA GmbH: Datenschutz und Datensicherheit. https://www.elga.gv.at/faq/datenschutz-und-datensicherheit/index.html (visited on 25.07.2019).

[ELGA19b] ELGA GmbH: ELGA CDA Implementierungsleitfäden: Allgemeiner Implementierungsleitfaden für ELGA CDA Dokumente. https://www.gesundheit.gv.at/r/service/Allgemeiner_CDA-Implementierungsleitfaden_%28Version_2_06%29.pdf?pamm85 (visited on 25.07.2019).

[ELGA19c] ELGA GmbH: ELGA-Portal. https://www.elga.gv.at/faq/elga-portal/index.html (visited on 25.07.2019).

[ELGA19d] ELGA GmbH: Teilnahme an ELGA. https://www.elga.gv.at/faq/teilnahme-an-elga/index.html (visited on 25.07.2019) (visited on 25.07.2019).

[ELGA19e] ELGA GmbH: Wissenswertes zu Elga. https://www.elga.gv.at/faq/wissenswertes-zu-elga/index.html (visited on 25.07.2019).

[Enge19] C. Engemann: eHealth. *In: D. Kasprowicz, S. Rieger (Hrsg.), Handbuch Virtualität*, Springer Fachmedien Wiesbaden, Wiesbaden (2019), 1–13, https://doi.org/10.1007/978-3-658-16358-7_18-1.

[Euro16] European Parliament and of the Council: Regulation on the protection of natural persons with the regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46l/EC: General Data Protection Regulation (4.5.2016), https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[Eyse01] G. Eysenbach: What is e-health? In: *J Med Internet Res*, 3, 2 (2001), e20, http://www.jmir.org/2001/2/e20/.

[Fors19] Forslund: Mycroft skill for making drinks. https://github.com/forslund/skill-cocktail/tree/3486c112b74a1945bf3138e4e8562ad7e4483da5 (visited on 25.07.2019).

[GLPP+14] C. Gaida, P. Lange, R. Petrick, P. Proba, A. Malatawy, D. Suendermann-Oeft: Comparing open-source speech recognition toolkits. *In: Technical Report of the Project OASIS* (2014).

[Goog19a] Google LLC: Access the Google Assistant with your voice. `https://support.google.com/assistant/answer/7394306?hl=en&ref_topic=7391797` (visited on 25.07.2019).

[Goog19b] Google LLC: Account linking. `https://developers.google.com/actions/identity/` (visited on 25.07.2019).

[Goog19c] Google LLC: Actions on Google Conversation HTTP/JSON Webhook API. `https://developers.google.com/actions/build/json/` (visited on 25.07.2019).

[Goog19d] Google LLC: Change your language or use multiple languages. `https://support.google.com/assistant/answer/7394513?hl=en&ref_topic=7391797` (visited on 25.07.2019).

[Goog19e] Google LLC: Choose what to share with your Google Assistant. `https://support.google.com/assistant/answer/7126196?hl=en&ref_topic=7110546` (visited on 25.07.2019).

[Goog19f] Google LLC: Contexts overview. `https://dialogflow.com/docs/contexts` (visited on 25.07.2019).

[Goog19g] Google LLC: Control smart home devices using Google Home. `https://support.google.com/googlenest/answer/7073578?hl=en` (visited on 25.07.2019).

[Goog19h] Google LLC: Daily Updates. `https://developers.google.com/actions/assistant/updates/daily` (visited on 25.07.2019).

[Goog19i] Google LLC: Data protection by Google. `https://policies.google.com/privacy` (visited on 25.07.2019).

[Goog19j] Google LLC: Data security & privacy on Google Home. `https://support.google.com/googlenest/answer/7072285?hl=en` (visited on 25.07.2019).

[Goog19k] Google LLC: Encryption in Transit in Google Cloud. `https://cloud.google.com/security/encryption-in-transit/?hl=en` (visited on 25.07.2019).

[Goog19l] Google LLC: Firebase Services. `https://developers.google.com/actions/tools/assistant-firebase-services` (visited on 25.07.2019).

[Goog19m] Google LLC: Get the Google Assistant. `https://assistant.google.com/platforms/phones/` (visited on 25.07.2019).

[Goog19n] Google LLC: Google Assistant SDK. `https://developers.google.com/assistant/sdk/` (visited on 25.07.2019).

[Goog19o] Google LLC: Google Store. `https://store.google.com/` (visited on 25.07.2019).

[Goog19p] Google LLC: Guest mode & Google Home. `https://support.google.com/googlenest/answer/7182412?hl=en` (visited on 25.07.2019).

[Goog19q] Google LLC: Have a conversation with Google Home. https://support.google.com/googlenest/answer/7685981?hl=en (visited on 25.07.2019).

[Goog19r] Google LLC: Manage Google Voice & Audio Activity. https://support.google.com/websearch/answer/6030020?hl=en&ref_topic=6032684 (visited on 25.07.2019).

[Goog19s] Google LLC: Push notifications. https://developers.google.com/actions/assistant/updates/notifications (visited on 25.07.2019).

[Goog19t] Google LLC: Responses. https://developers.google.com/actions/assistant/responses (visited on 25.07.2019).

[Goog19u] Google LLC: Set up & manage apps for the Google Assistant (formerly Services). https://support.google.com/googlenest/answer/7126338?hl=en&ref_topic=7128170 (visited on 25.07.2019).

[Goog19v] Google LLC: Set up your Google Home speaker or Google Nest display. https://support.google.com/googlenest/answer/7029485?hl=en&ref_topic=7196250 (visited on 25.07.2019).

[Goog19w] Google LLC: Smart displays. https://developers.google.com/actions/surfaces/displays (visited on 25.07.2019).

[Goog19x] Google LLC: Webhook for slot filling. https://dialogflow.com/docs/fulfillment/webhook-slot-filling (visited on 25.07.2019).

[GoPo18] Y. Gong, C. Poellabauer: An Overview of Vulnerabilities of Voice Controlled Systems. http://arxiv.org/pdf/1803.09156v1.

[Gree19] Greenido: App for the google assistant that give you information on bitcoin (e.g. price, market cap etc). https://github.com/greenido/bitcoin-info-action/ (visited on 25.07.2019).

[Hanb19] M. Hanbury: Alexa can now delete your recorded voice commands, but Amazon hasn't made it easy. https://www.businessinsider.de/amazon-has-a-new-feature-to-delete-alexa-recordings-2019-5?r=US&IR=T (visited on 25.07.2019).

[HDKB+14] A. Hochgatterer, M. Drobics, J. Kropf, M. Bammer, P. Kastner, M. Fritz: Ambient Assisted Living – Intelligente Assistenz durch M2M-Technologien. In: *e & i Elektrotechnik und Informationstechnik*, 131, 1 (2014), 26–32, https://doi.org/10.1007/s00502-013-0188-3.

[HEHS+12] S. Herbek, H. A. Eisl, M. Hurch, A. Schator, St. Sabutsch, G. Rauchegger, A. Kollmann, T. Philippi, P. Dragon, E. Seitz, St. Repas: The Electronic Health Record in Austria: a strong network between health care and patients. In: *European Surgery*, 44, 3 (2012), 155–163, https://doi.org/10.1007/s10353-012-0092-9.

[HiMa15] J. Hirschberg, C. D. Manning: Advances in natural language processing. In: *Science*, 349, 6245 (2015).

[HoyM18] M. Hoy: Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. In: *Medical Reference Services Quarterly*, 37 (2018).

[HSWW17] W. Haack, M. Severance, M. Wallace, J. Wohlwend: Security analysis of the Amazon Echo. In: (2017).

[HuAH01] X. Huang, A. Acero, H.-W. Hon: Spoken Language Processing: A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st Aufl. (2001).

[HVNC09] H. Sun, V. D. Florio, N. Gui, C. Blondia (Hrsg.): Promises and Challenges of Ambient Assisted Living Systems: 2009 Sixth International Conference on Information Technology: New Generations (2009).

[Klad19] Julia Carina Klade BSc: Sprachassistenz zur PatientenunterstÃ¼tzung (2019).

[JuMa14] D. Jurafsky, J. H. Martin: Speech and language processing. Pearson London (2014).

[KBo18] V. Këpuska, G. Bohouta (Hrsg.): Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home): 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC) (2018).

[KiMu19] B. Kinsella, A. Mutchle: Smart Speaker Consumer Adoption Report. `https://voicebot.ai/wp-content/uploads/2019/03/smart_speaker_consumer_adoption_report_2019.pdf` (visited on 25.07.2019).

[Kins19] B. Kinsella: Google Assistant Actions Total 4,253 in January 2019, Up 2.5x in Past Year but 7.5% the Total Number Alexa Skills in U.S. `https://voicebot.ai/2019/02/15/google-assistant-actions-total-4253-in-january-2019-up-2-5x-in-past-year-but-7-5-the-total-number-alexa-skills-in-u-s/` (visited on 25.07.2019).

[Klos19] A. Klostermann: Gesamtarchitektur. `https://www.elga.gv.at/fileadmin/user_upload/Dokumente_PDF_MP4/Technisches/ELGA_Gesamtarchitektur_2.30a.pdf` (visited on 25.07.2019).

[KoSt19] A. Komninos, S. Stamou: HealthPal: an intelligent personal medical assistant for supporting the self-monitoring of healthcare in the ageing society (2019).

[KuWe19] M. Kutter, M. Westenthanner: Cortana vor dem Aus? Windows 10 bekommt neue Alexa-App. `https://www.chip.de/news/Cortana-vor-dem-Aus-Windows-10-bekommt-neue-Alexa-App_152544499.html` (visited on 25.07.2019).

[KrWo18] T. Kruse Brandão, G. Wolfram: Digital Connection: Die bessere Customer Journey mit smarten Technologien – Strategie und Praxisbeispiele (2018), 91–107.

[LKKT+18] B. Lakdawala, F. Khan, A. Khan, Y. Tomar, R. Gupta, A. Shaikh: Voice to Text transcription using CMU Sphinx A mobile application for healthcare organization (2018).

[LRKR+18] I. Lopatovska, K. Rink, I. Knight, K. Raines, K. Cosenza, H. Williams, P. Sorsche, D. Hirsch, Q. Li, A. Martinez: Talk to me: Exploring user interactions with the Amazon Alexa. In: *Journal of Librarianship and Information Science* (2018).

[LTXL+18] X. Lei, G.-H. Tu, A. X. Liu, C. Li, T. Xie: The Insecurity of Home Digital Voice Assistants - Vulnerabilities, Attacks and Countermeasures (2018).

[Ludl19a] D. Ludlow: Google Assistant update brings offline support, driving mode and better conversations. https://www.trustedreviews.com/news/google-assistant-update-brings-offline-support-driving-mode-and-better-conversations-3714218 (visited on 25.07.2019).

[Ludl19b] D. Ludlow: Google Assistant update brings offline support, driving mode and better conversations. https://www.trustedreviews.com/news/google-assistant-update-brings-offline-support-driving-mode-and-better-conversations-3714218 (visited on 25.07.2019).

[Mary19] MaryTTS: Github MaryTTS. https://github.com/marytts/marytts/tree/master/marytts-languages (visited on 25.07.2019).

[MoVu15] A. E. Moorthy, K.-P. L. Vu: Privacy Concerns for Use of Voice Activated Personal Assistant in the Public Space. In: *International Journal of Human–Computer Interaction*, 31, 4 (2015), 307–335.

[Mycr19a] Mycroft AI: Conversational Context. https://mycroft.ai/documentation/skills/conversational-context/ (visited on 25.07.2019).

[Mycr19b] Mycroft AI: Enclosure display control. https://mycroft.ai/documentation/skills/display-control/ (visited on 25.07.2019).

[Mycr19c] Mycroft AI: Get started. https://Mycroft.ai/get-started/ (visited on 25.07.2019).

[Mycr19d] Mycroft AI: Github Mycroft. https://github.com/MycroftAI (visited on 25.07.2019).

[Mycr19e] Mycroft AI: Languages are Hard. https://Mycroft.ai/blog/languages-are-hard/ (visited on 25.07.2019).

[Mycr19f] Mycroft AI: Mark 1. https://Mycroft.ai/documentation/mark-1/ (visited on 25.07.2019).

[Mycr19g] Mycroft AI: Mark 1 User Guide. https://Mycroft.ai/wp-content/uploads/2017/06/Mark_1_User_Guide.pdf (visited on 25.07.2019).

[Mycr19h] Mycroft AI: Mycroft AI Privacy Policy. https://Mycroft.ai/embed-privacy-policy/ (visited on 25.07.2019).

[Mycr19i] Mycroft AI: Mycroft Mark 1. https://Mycroft.ai/product/Mycroft-mark-1/ (visited on 25.07.2019).

[Mycr19j] Mycroft AI: Mycroft Market. https://market.Mycroft.ai/skills (visited on 25.07.2019).

[Mycr19k] Mycroft AI: Mycroft Single Sign-On: Balancing Convenience and Privacy. https://medium.com/@Mycroftai/Mycroft-single-sign-on-balancing-convenience-and-privacy-769ef0143268 (visited on 25.07.2019).

[Mycr19l] Mycroft AI: Mycroft Software and Hardware. https://mycroft.ai/documentation/Mycroft-software-hardware/ (visited on 25.07.2019).

[Mycr19m] Mycroft AI: Open-sourcing our mechanical, electrical and industrial designs. https://github.com/MycroftAI/hardware-Mycroft-mark-1 (visited on 25.07.2019).

[Mycr19n] Mycroft AI: openHAB skill in the Mycroft market. https://market.mycroft.ai/skills/fe2fc8e2-4435-4c04-a40b-47b557ce0ee5 (visited on 25.07.2019).

[OlKe] C. Olson, K. Kemery: Voice report: From answers to action: customer adoption of voice technology and digital assistants. https://about.ads.microsoft.com/en-us/insights/2019-voice-report (visited on 25.07.2019).

[open19a] openHAB: Landing page. https://www.openhab.org/ (visited on 25.07.2019).

[open19b] openHAB: Mary Text-to-Speech. https://www.openhab.org/addons/voice/marytts/ (visited on 25.07.2019).

[Penr19] S. Penrod: The Mycroft GUI – The Screen is Dead. Long Live the Screen! https://mycroft.ai/blog/the-Mycroft-gui-the-screen-is-dead-long-live-the-screen/ (visited on 25.07.2019).

[Port19] J. Porter: Google fined 50 million Euro for GDPR violation in France. https://www.theverge.com/2019/1/21/18191591/google-gdpr-fine-50-million-euros-data-consent-cnil (visited on 25.07.2019).

[RaMi13] P. Rashidi, A. Mihailidis: A Survey on Ambient-Assisted Living Tools for Older Adults. In: *IEEE Journal of Biomedical and Health Informatics*, 17, 3 (2013), 579–590.

[ReBB19] M. Reichel, L. Baum, P. Buxmann: Anwendung eines sprachbasierten KI-Dienstes in der Gesundheitsbranche am Beispiel der Entwicklung eines Alexa-Skills: Mit Algorithmen zum wirtschaftlichen Erfolg (2019), 77–93.

[Sabu19] S. Sabutsch: ELGA Schulungsunterlagen. https://www.elga.gv.at/fileadmin/user_upload/Dokumente_PDF_MP4/Technisches/ELGA_Basis_fuer_Schulungsunterlagen_V2.0.pdf (visited on 25.07.2019).

[SKIM18] SKIM: Voice Tech Trends 2018: Consumer Behavior & Brand Implications in the US, UK and Germany: Exploring Voice and Digital Assistants Attitudes and Adoption Around the World. https://info.skimgroup.com/hubfs/Marketing%20Files%20(not%20Google%20indexed)/Campaign%20Specific/CPG%20TST%20Digital%20Q3%20Q4%202018/SKIM%20Voice%20Tech%20Trends%202018_Consumer%20Behavior%20and%20Brand%20Implications%20in%20US,%20UK,%20Germany.pdf (visited on 25.07.2019).

[SMHM17] B. P. Shrestha, A. Millonig, N. B. Hounsell, M. McDonald: Review of Public Transport Needs of Older People in European Context. In: *Journal of Population Ageing*, 10, 4 (2017), 343–361, https://doi.org/10.1007/s12062-016-9168-9.

[Snip19a] Snips: Connection to MQTT. https://docs.snips.ai/articles/console/actions/actions/code-your-action/manual-action (visited on 25.07.2019).

[Snip19b] Snips: Console. https://console.snips.ai (visited on 25.07.2019).

[Snip19c] Snips: Dialogue API Reference. https://docs.snips.ai/reference/dialogue (visited on 25.07.2019).

[Snip19d] Snips: General FAQ. https://docs.snips.ai/additional-resources/faq/general-faq/ (visited on 25.07.2019).

[Snip19e] Snips: Getting Started. https://docs.snips.ai/getting-started (visited on 25.07.2019).

[Snip19f] Snips: Landing page of Snips. https://snips.ai/ (visited on 25.07.2019).

[Snip19g] Snips: Linux amd64. https://docs.snips.ai/articles/other-platforms/linux-amd64 (visited on 25.07.2019).

[Snip19h] Snips: Platform Configuration. https://docs.snips.ai/articles/platform/platform-configuration (visited on 25.07.2019).

[Snip19i] Snips: Quick Start Android. https://docs.snips.ai/getting-started/quick-start-android (visited on 25.07.2019).

[Snip19j] Snips: Quick Start Console. https://docs.snips.ai/getting-started/quick-start-console (visited on 25.07.2019).

[Snip19k] Snips: Quick Start Raspberry Pi. https://docs.snips.ai/getting-started/quick-start-raspberry-pi (visited on 25.07.2019).

[Snip19l] Snips: Slot Types. https://docs.snips.ai/reference/slot_types (visited on 25.07.2019).

[Snip19m] Snips: Snips Maker Kits. https://docs.snips.ai/the-maker-kit (visited on 25.07.2019).

[Snip19n] Snips: Snips Market. https://console.snips.ai/store/de/ (visited on 25.07.2019).

[Snip19o] Snips: Snips Platform. https://docs.snips.ai/ (visited on 25.07.2019).

[Snip19p] Snips: Technology. https://snips.ai/technology/ (visited on 25.07.2019).

[Snip19q] Snips: Voice enabled SmartMirror$^2$ - snips inside. https://forum.snips.ai/t/voice-enabled-smartmirror-snips-inside/1784 (visited on 25.07.2019).

[Snip19r] Snips: Voice Interaction Development Kits. https://www.seeedstudio.com/snips.html (visited on 25.07.2019).

[Snip19s] Snips: Voice, Privacy, and the "No Compromise" Model. https://snips.ai/blog/voice-privacy-and-the-no-compromise-model/ (visited on 25.07.2019).

[Snip19t] Snips: Creating Unit Tests for your assistant in the Console. https://docs.snips.ai/articles/console/unit-tests (visited on 25.07.2019).

[Sozi19] Sozialversicherungs-Chipkarten Betriebs- und Errichtungsgesellschaft m.b.H. - SVC: e-Medikation. https://www.chipkarte.at/cdscontent/?contentid=10007.767252&action=2&viewmode=content (visited on 25.07.2019).

[Squa19] I. Square: OkHttp with HTTPS. https://square.github.io/okhttp/https/ (visited on 25.07.2019).

[SSDK18] M. A. Siddike, J. Spohrer, H. Demirkan, Y. Kohda: People's Interactions with Cognitive Assistants for Enhanced Performances (2018).

[StHo11] A. Ströher, W. Honekamp: ELGA – die elektronische Gesundheitsakte vor dem Hintergrund von Datenschutz und Datensicherheit. In: *Wiener Medizinische Wochenschrift*, 161, 13 (2011), 341–346, https://doi.org/10.1007/s10354-011-0011-x.

[SUMO19] SUMO Heavy Industries: 2019 Voice Commerce Survey: The Current State The Current State and Future of Voice-Assisted Shopping. https://gallery.mailchimp.com/d449b0fd51c384d5fbcc900f0/files/eb03fba0-3bbc-4a98-8980-b17bcf7ec6c9/2019_Voice_Commerce_Survey.pdf (visited on 25.07.2019).

[WoKK15] M. K. Wolters, F. Kelly, J. Kilgour: Designing a spoken dialogue interface to an intelligent cognitive assistant for people with dementia. In: *Health Informatics Journal*, 22, 4 (2015), 854–866.

[Wool18] C. Woolf: All AWS Services GDPR ready. https://aws.amazon.com/blogs/security/all-aws-services-gdpr-ready/ (visited on 25.07.2019).

[ZCLW+18] R. Zhang, X. Chen, J. Lu, S. Wen, S. Nepal, Y. Xiang: Using AI to Hack IA: A New Stealthy Spyware Against Voice Assistance Functions in Smart Phones. In: *CoRR*, abs/1805.06187 (2018).

[ZMFW+] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, F. Qian: Understanding and Mitigating the Security Risks of Voice-Controlled Third-Party Skills on Amazon Alexa and Google Home. http://arxiv.org/pdf/1805.01525v2.

[ZYJZ+17] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, W. Xu: DolphinAtack: Inaudible Voice Commands. In: *CoRR*, abs/1708.09537 (2017).