

École polytechnique de Louvain

Controlling security rules using natural dialogue: An application to smart home care

Authors: **Stanley GOFFINET, Donatien SCHMITZ**

Supervisors: **Etienne RIVIERE, Axel LEGAY**

Readers: **Lionel METONGNON, Annanda RATH, Igor ZAVALYSHYN**

Academic year 2020–2021

Master [120] in Computer Science

Abstract

The constant evolution of information and communication technologies in all areas of society implies the development of many defense mechanisms by researchers to ensure the protection of the data we share as well as our privacy. The excitement of smart systems is accompanied by an increase in the number of end-user concerns about personal data management and security struggle in the IoT domain. This fight is very abstract for end-users, making it very difficult for them to take control of their privacy as those security mechanisms can become very complicated to use, especially for elderly user.

The motivation behind our thesis is a project from the Brussels region called "Smart and Social Home Care" that aims to offer a smart home solution specifically designed for older users. The current system where the elderly live in retirement homes or assisted living will no longer be viable in this aging context. The nature of support for the elderly will change in the coming years, with a major shift towards home care (daily life assistance) as opposed to nursing homes.

To cope with the issue of elderly people defining access control rules, we propose an innovative voice assistant which efficiently allows to create new security rules and update existing ones in a simple and intelligible way based on a series of questions and examples of information that should or should not be sent to the social circles associated with the elderly person's social network.

Our solution has been tested and widely approved as an efficient and relevant alternative to a more usual web interface.

Acknowledgements

We would like to give special thanks to our supervisors, Etienne Riviere and Axel Legay, for their availability, constructive criticism, and enthusiasm. Doing weekly meetings was certainly something that helped this thesis end well and we thank them for that.

Another great source of feedback and positivism was Igor Zavalysyn who was also present in every meeting we had with Etienne and Axel. We cannot thank him enough for all the help he has given us and all the things we have learned from him.

We also thank Guillaume Ortegat for helping us with the evaluation process.

Last, but not least, we would like to thank our family for supporting us throughout this thesis. We also thank them for taking the time to participate in this project through the experiments we conducted.

Contents

1	Introduction	4
2	Background and related work	7
2.1	The breakthrough of home automation	7
2.2	Privacy and Security risks	8
2.2.1	IoT threats	8
2.2.2	User preferences and concerns	9
2.3	Difficulty of setting access control rules	10
2.3.1	Global ICT privacy tools problem	10
2.3.2	IoT privacy tools problem	11
2.3.3	IoT privacy tools problem for elderlies	12
2.3.4	Survey and findings	13
2.4	IoT Systems for aging people	14
2.5	Towards home cares instead of care homes	15
2.5.1	Description of the global project	15
2.5.2	Market research among elderlies	17
2.6	Monitoring devices and sensors	17
2.7	Protection-aware smart hubs	20
2.7.1	Privacy-aware smart hubs	20
2.7.2	Safety-and-security-aware smart hubs	21
2.8	Dialogue-based policy manager	23
2.9	Acceptance of a voice interface by elderlies	24
3	Methodology	26
3.1	System overview	26
3.2	Dialogue system	27
3.2.1	Question semantic	27
3.2.2	Expected answers	29
3.2.3	Full dialogue semantic	30
3.3	Policy format	31
3.3.1	Event description	31

3.3.2	Taking "actions"	32
3.4	Policy heuristics	33
3.4.1	Classification	33
3.4.2	Policy creation	35
3.4.3	Policy monitoring	36
3.5	Conclusion	37
4	Implementation	38
4.1	Voice-assistant services	38
4.1.1	Speech-to-text	39
4.1.2	Text-to-speech	40
4.1.3	Intent recognition	40
4.1.4	Dialogue manager	41
4.1.5	Speech-to-Intent (Bonus)	41
4.2	Orchestrator: Laravel Framework	42
4.2.1	Policy heuristics and question generator	42
4.2.2	API	42
4.2.3	Web interface	43
4.3	Databases	43
4.3.1	MySQL	43
4.3.2	Redis	44
4.4	Deployment	45
4.4.1	Docker	45
4.4.2	Configuration	46
4.4.3	Rhasspy services	47
5	Evaluation	49
5.1	User tests	49
5.1.1	Demographic form	50
5.1.2	Voice recognition	51
5.1.3	Screen-based interface	52
5.1.4	Dialogue-based interface	53
5.2	Results	54
5.2.1	Demographic form	55
5.2.2	Voice recognition	56
5.2.3	Screen-based interface	57
5.2.4	Dialogue-based interface	58
5.3	Further discussion	61
6	Conclusion	63

Appendices	67
A Classification	68
B Implementation	71
C Evaluation	74
D Web interface	78

Chapter 1

Introduction

Home automation is still young but has some bright days ahead of it. Whether it is Amazon Echo, Google Dot, Apple HomePod, or even open-sourced solutions such as Home Assistant, they are present in our homes to help us ease our daily lives. Modernizing your home to better control your comfort and save money is a possible goal to achieve with home automation. The assistance of home automation allows to remotely-control devices, allows to improve the comfort of the users by automating certain tasks, allows to increase the efficiency of management of resources such as lights, and, also, it allows to reinforce security with the arrangement of locks or intelligent doors. The Thunderbird International Business Review [7] states that some home automation solutions such as smart locks or smart thermostats were the first adopted by homeowners, where smart TVs and decorations were more belated. The latest one they show in Figure 1.1 is assistance to the elderlies.

Home automation can be orchestrated in several ways, cloud for solutions such as Amazon Alexa, local for solutions like HomeAssitant IO¹. On the other hand, interactions with the user might be done with a graphical interface (e.g. dashboard), a voice assistant, or a combination of both. New technologies mean new threats to data privacy which have been addressed by many papers [9, 18, 20], each discussing the security and privacy concerns of the Internet of Things in different ways. From these threats come many public concerns about the use of IoT technologies. A lot of work [1, 5, 6, 12] has been done to study user preferences and concerns about the privacy implications of IoT.

We have highlighted the risks associated with the use of such technologies as well as user behavior and perceptions. To solve these problems, it is necessary to establish rules of confidentiality. In general, these methods for countering IoT

¹<https://www.home-assistant.io/>

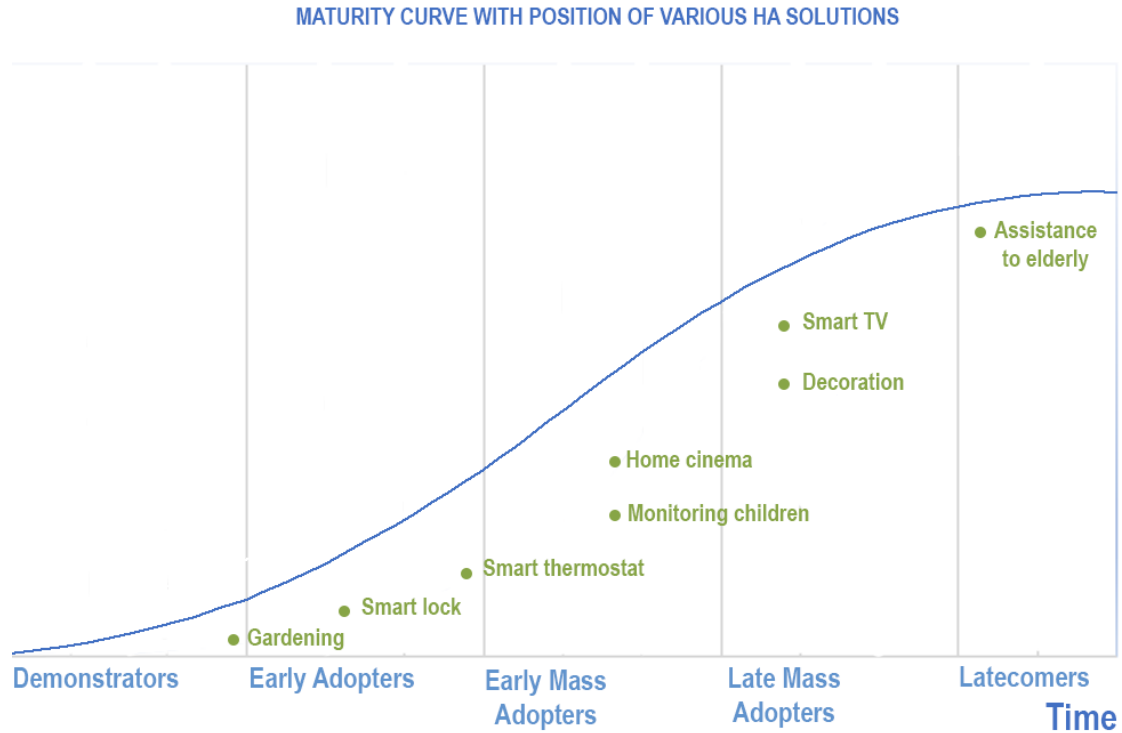


Figure 1.1: Maturity curve of home automation solutions. Reprinted from "Home Automation — A Fast - Expanding Market", by Darshan Chandrashekar Khedekar, Amelia Carrera Truco, Diego A. Oteyza, Guillermo Florez, Huertas, *Thunderbird International Business Review*, Vol. 59, No. 1, January/February 2017

threats are not easy to implement for someone who does not have the necessary knowledge in this field and the global information and communication technologies field (ICT). This difficulty in defining access control rules by using technologies is even more complicated for certain portions of the population, such as the elderly, who develop new and inappropriate behavior as a result of their inability.

In this thesis, we will focus on a very specific use-case of home automation: home care for seniors. The population of elderly is growing [4] and the number of seniors that prefer to adapt their home to ease their life instead of joining an adapted facility such as care homes [17] is also increasing. The adaptation of senior homes has already been studied [3], proposing their solutions in the shape of sensors, monitoring devices, smart objects, or even sometimes (in the case of Habitat[3]) smart homes. The motivation of our thesis is a project from Brussels project that aims to build a smart home care solution for the elderly. The project Vitadomi, funded by the Innoviris innovation fund of the Brussels region, is a collaboration

between Electronic Design International (EDI)², UCLouvain, and Sirris (Agoria)³. Market research has been done by this project team with different elderly people to find out their opinion on the system created.

To get an overview of the environment and this specific security context, we scoured the literature to see what was being done in protection-aware smart hubs. Many projects [11, 13, 19, 23] have been carried out by researchers and teams to create a system of privacy control through smart hubs. These projects can be divided into three types, those who want to increase security, those who want more safety, and those who fight for privacy. We, therefore, asked ourselves what would be the best way or tool to make these systems usable for older people.

In this paper, we present our voice-based interface shaped as a non-intrusive dialogue system that allows the user to stay close to its security rules (policies). This personal assistant tries to guess the security rules that might be of greater interest to the user. Considering that the user's preferences might change, the personal assistant also challenges the already defined security rules to ensure that its behavior is still the one expected by the user. To better understand how this works, we will use a top-down approach by starting with an overview of the overall system and after its design principles (dialogue system, policy format, and policy heuristic). We will present the conceptual contribution first and then the technical aspects.

To prove the usefulness and relevance of our system, we carried out experiments with users between 59 and 88 years old to assess its acceptance and relevance by older people. This evaluation consisted of a multi-part study. First, voice samples were collected from the participants and fed into a script that analyzed which speech recognition was most accurate for our scope. Subsequently, we conducted a situational role-play with the participants in which their role was that of a future user while our role was that of a future technician helping the user to configure this system. The participants were then asked to use our voice assistant to define security rules on their own.

The results show that all of our participants found the system easy to use, relevant, easy to learn, and intuitive. The SUS⁴ scores being spread between 82.5 and 92.5, we can emphasize the solution of a voice assistant being an encouraging alternative to a web interface in the context of elderly people.

²https://www.electronic-design.be/nl_BE/

³<https://www.sirris.be/>

⁴The SUS (System Usability Scale) score rates the usability of a service from 0 (mediocre) to 100 (excellent)

Chapter 2

Background and related work

Before describing our proposed solution, we must first describe the full scope of the environment surrounding our thesis topic as well as the existing literature and work in this domain. We will use a top-down approach that will present existing works and their limitation regarding our specific problem.

2.1 The breakthrough of home automation

One of the most promising markets of the 21st century is home automation. In fact, the global market for smart home devices is growing every year in 2019 to reach 815 million units shipped in 2020 and will continue to grow [22].

In general, the data collected by home automation technologies allows for the creation of even more value, which supports the energy and ecological transition. The digitalization of the sector through IoT and the use of data is essential, but new technologies imply new threats and therefore new ways of protection.

Category	2019 shipments	2019 market share	2023 shipments	2023 market share
Video entertainment	339,8	41,70%	417,7	29,90%
Home monitoring	156,6	19,20%	307,9	22,10%
/Security				
Smart Speaker	134,8	16,50%	200	14,30%
Others	183,6	22,50%	470,5	33,70%
Total	814,8	100%	1396	100%

Table 2.1: Smart home devices shipments in millions and market share

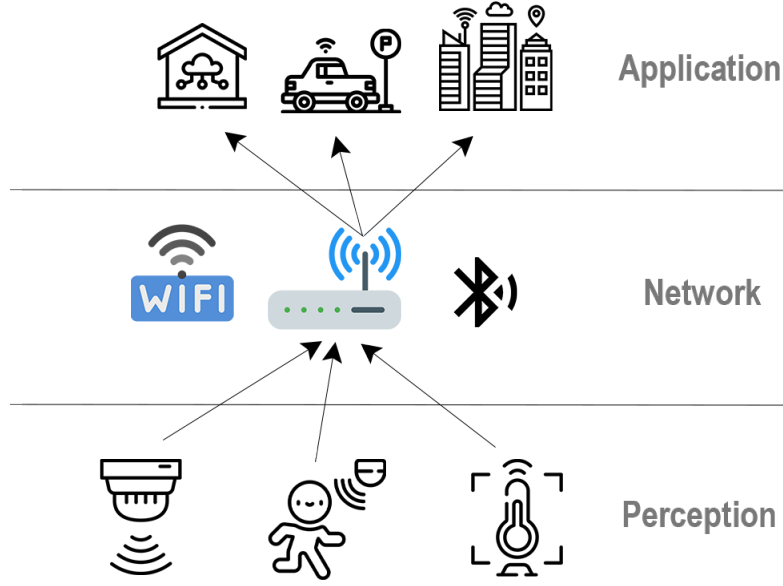


Figure 2.1: IoT layer system. Reprinted from *Internet of things (IoT) security: Current status, challenges and prospective measures* by Rwan Mahmoud, Tasneem Yousuf, Fadi Aloul, and Imran Zualkernan, 2015, 10th International Conference for Internet Technology and Secured Transactions (ICITST)

2.2 Privacy and Security risks

IoT technologies pose some challenges. Security experts have identified concerns and threats with smart devices, including privacy risks and vulnerable and unreliable devices. From these threats derive preferences and concerns of the users of such technologies.

2.2.1 IoT threats

Some works [9, 18, 20] have been done to study the security and privacy of end-users interacting with these systems. Each addresses the security and privacy concerns of the Internet of Things in a different way.

For some [18], the analysis of this system has to be done step by step. The IoT system operates mainly on three layers: perception, network, and application, we can see that in figure 2.1. Each of these has its associated security and privacy issues.

- (1) The perception layer, which includes all devices that collect data such as sensors, is very fragile in the event of an attack on wireless protocols. The strength of the wireless network must be maintained at all times. This layer is also very susceptible to physical attacks as it has a number of devices in publicly accessible spaces, and the power consumption and computational capabilities of its small devices are very limited, making the perception layer very insecure. A typical attack is the node capture attack. The attacker can add another node that sends malicious data into the system. This malicious data can be presented as a DoS attack if a lot of data is sent to consume a maximum of resources and make the system slow down.
- (2) The network layer has the role of transferring and managing the sending of device data to other devices or to the internet via Wifi, Bluetooth, or Network cable. It mediates between all IoT devices in the network. This layer, as with the perception layer, can also be attacked by a DoS. An analysis of unauthorized traffic can also be a threat as it would be akin to passive surveillance. The key and encryption system must be strong to ensure optimal protection. What is also impractical is that in such an IoT system absolutely everything is connected, which means that any subject belonging to this network must be autonomous in its analysis of the network and in its ability to defend itself.
- (3) The last layer is the application layer. It ensures the legitimacy, completeness, and confidentiality of the data. It is in this layer that the intelligent environment is complete. The IoT environment is a collection of hundreds of thousands of different devices and associated interfaces and software, there is no common foundation for initial security and there is a global lack of trust policies and standardized policies.

2.2.2 User preferences and concerns

A lot of work [1, 5, 6, 12] has focused on studying user preferences and concerns regarding the privacy implications of IoT.

Naeini et al., [1] conducted a study with 1007 participants to understand the general expectations and preferences of smart environment users. They looked at each factor of data collection to see its contribution to user comfort and preferences. For example, they deduced that their participants would be less comfortable if a statement in the system explained that the data collected was shared with others. Their analysis shows that many people want to retain control over their data and that future IoT systems should take this into account when designing privacy

notices. It even suggests the adoption of the idea of personalized privacy assistants to make it easier for users to manage their control. This is the direction in which our contribution is heading.

To further encourage our approach, Eric Zeng et al. [12] conducted semi-structured interviews with people who live in smart homes to learn about end-user daily routines, attitudes, expectations, and actions regarding security and privacy. They identified gaps resulting from a limited technological understanding of smart homes and low awareness of some security problems. They also studied the mitigation strategies that users use when they have security or privacy concerns. They investigate whether these strategies are effective. They found that mitigation strategies vary greatly, implying that security best practice has not become the standard. In terms of technological mitigation, some users have installed their IoT system on a distinct Wi-Fi system. In terms of non-technical mitigations, some of these involve changing the user’s behavior when dealing with these devices. For example, avoiding saying or even doing certain things around the Echo¹ or in front of a camera. In these instances, participants used mitigation strategies whose benefits are ambiguous, which suggests, according to the authors, limited underlying technical knowledge.

Having reviewed the range of work on privacy in smart systems, we have highlighted the risks associated with the use of such technologies as well as user behavior and perceptions. The range of proposed solutions for privacy protection in IoT will be discussed in Section 2.7. Jacobsson et al. [8] point out that it is essential to understand the users to create workable privacy mechanisms.

2.3 Difficulty of setting access control rules

We have seen that IoT often poses security and privacy risks in many scientific articles. We have also seen in other works the range of behaviors and concerns associated with its threats. In general, the methods for countering IoT threats are very theoretical and not easy to implement for someone who does not have the necessary knowledge in this field.

2.3.1 Global ICT privacy tools problem

This problem of using tools to manage security threats can be related not only to the case of intelligent systems but to the global information and communication technologies (ICT). Even if the tool has been well made, it will always remain

¹Amazon smart hub

difficult to use for a part of the users who do not have the necessary knowledge to use it.

Security mechanisms are effective only when used correctly. This is explained by Alma Whitten and J.D.Tygar in their paper [21]. According to them, user interfaces for security still tend to be messy, confusing, or even non-existent. Effective security requires a different standard of usability and one that will not be achieved by UI design techniques appropriate to other types of consumer software.

To test their hypothesis, they conducted a practical test on subjects. The purpose of this experiment was to confront novices in the field of cryptography and have them use an interface that was very well designed according to general standards for cryptographic processing of email. The experiment concluded with the majority of the subjects not being able to complete their work using the interface provided. This underpins the idea that a number of user interface design flaws can also contribute to security failures when the user is not very experienced or knowledgeable about the tool or interface used.

2.3.2 IoT privacy tools problem

In the case of IoT environments, this problem is also proliferating. In their paper, Norbert Nthala et al. [6] focused on the key role of social relationships in home security by conducting a two-part survey on 1,000 subjects to explore the issues that affect home security practices. Based on their findings they made some recommendations.

In their view, the existing practices have focused too much on improving user security awareness. Instead, they suggest focusing on developing skills, targeting tools, and promoting a sense of trust and recognition. This would capitalize on two characteristics of support currently desired in the home: duty of care and continuity of care. Their second recommendation is in line with Alma Whitten and J.D.Tygar's view [21] that there is a great need for tools for non-experts to manage the safety congestions of different devices and services in the home. The proliferation of networked devices and services in their home makes security management too complex for its users. The development of tools to manage the security configurations of several devices centrally would motivate home users and simplify this task. These tools will need to be developed to support the services that currently assist home users.

2.3.3 IoT privacy tools problem for elderlies

This concern for the elderly has also been addressed by Alisa Frik et al.[2]. The aim of their research was to inform the design of systems that enable older people to make better decisions and take control of their personal data. In their paper [2], they divide privacy and security risks into four main groups: information collection, information processing, information dissemination, and privacy invasion. We will now describe each of the identified groups and explain how our solution would provide for the needs of these older people while removing the risks and consequences associated with these groups.

- (1) The basic principle of a connected home is to collect information to automate tasks and lighten the user's daily life, so information collection is a dominant risk in this kind of environment. In our context of medical monitoring, this could be similar to surveillance and this could influence the feelings of anxiety and paranoia of the person being monitored. Our system will need to empower the user regarding this monitoring to create a sense of comfort for the older person so that the monitoring set up is their own.
- (2) The processing of information. Any processing involves a risk factor, as it can lead to unauthorized access to personal information, accidental shared access, or abuse of power by an unknown person. Moreover, its dangers could lead, in the case of an assisted living facility, to serious consequences such as physical safety or health damage. In fact, it will be very complicated to make a false manipulation and allow the processing of one's data by a third party, if both the medical staff and the user are aware of the policies associated with each data. In addition, participants generally regarded medical staff as trustworthy for personal information and welcomed the use of patient portals to manage and exchange medical information.
- (3) The information dissemination is the greatest fear for older people is that their data may be shared and sold for profit or for bad intentions which would result in a bad reputation for the person or unwanted social judgment. Furthermore, even if the initial intention is good, the disclosure of health information may endanger the benefits that older people could receive such as social insurance, incapacity benefit, insurance protection, or being eligible for assisted living. But again, in our case only the people voluntarily associated and verified by the medical service and the user of our service will be in possession of their data and only them unless it is distributed by the

competent medical authorities.

- (4) The most ambiguous group of risks according to the level of dependence of the elderly person on the medical monitoring service, the invasion of privacy. In their paper, it states that participants were particularly concerned about location data and data on their activities in the home which could be seen as sources of compromising information to facilitate physical attacks on them or their property. This is true, but unfortunately, if it turns out that GPS and activity tracking is a medical duty because the person has already made an attempt on their life or run away from home due to dementia or Alzheimer's, it is the medical duty to use this kind of data despite its intrusion into the privacy of the patient. In our solution, if this kind of data is not mandatory, of course, the user can disable himself the use and the taking of their data but it will not be always the case.

2.3.4 Survey and findings

Alisa Frik et al.[2], asked 46 elderlies selected to represent a wide range to find out the behaviors, techniques, and strategies raised by the elderlies. In general, they said that older adults may be particularly vulnerable to certain risks and have more difficulty mitigating them because their knowledge and experience with these technologies is relatively low. In addition, they noted that older people often face usability problems and uncertainties in mitigating these risks. That is why they value independence and security more than privacy. In the end, their managing privacy often leads them to limit or simply avoid using new technologies.

In their conclusion and suggestions [2], they said that the difficulty of using technology leads to a low self-efficacy in terms of privacy and security. According to older users, this is due to their lack of familiarity with technology and their own lack of skills and knowledge. Removing these barriers is an important basis for enabling older people to use technology in a safer and more comfortable way. One solution of the conclusion is to design interfaces that optimize the ability of older users to authenticate, configure settings, and perform other security tasks without error and within a reasonable time. The second solution would be for designers to force themselves to make information management (e.g. editing and deleting personal records) easier. To continue in the same direction, companies should involve older people in the development process through participatory design and usability testing.[2]

Improving system design is exactly the line and goal we have tried to achieve

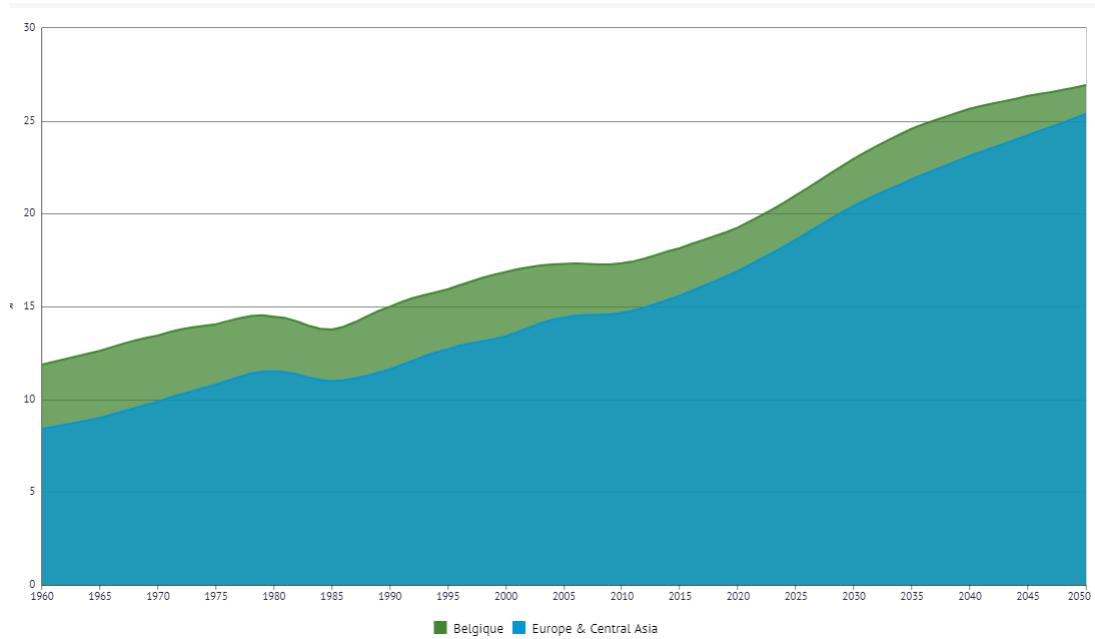


Figure 2.2: Belgian and European population aged 65 and over projection. Sources : Bureau fédéral du Plan (BFP), janvier 2021 ; Statbel ; Calculs : IWEPS

in this thesis, to make security and privacy management systems easier to use and more understandable by older people. To achieve this goal, instead of improving the knowledge of the users, we make this management super intuitive and fast by an adequate formulation of security configuration questions related to the use of the elderly and their preferences.

In that way, their research [2] will be very useful to check that our privacy and data management ergonomic solution does not produce unwanted behavior and strategies of end-users and that it makes it easy to learn and use for our older targeted audience.

2.4 IoT Systems for aging people

The health system must evolve to meet the new problems of the aging population, with the help of IoT systems. To mention our country, in Belgium there were more than two million people over the age of 65 [4], which corresponds to 19% of the population and this percentage will approach 26% in 2050. In addition, we can see that this evolution will also be reflected all over Europe in figure 2.2

These evolutions are expected to have profound individual consequences but also for the public sector, businesses, and civil society, including health care systems. New devices of assistance and accompaniment will have to be developed to facilitate the management of data and the medical monitoring of our population which is aging and will have more and more health problems. These new devices will have to be instantaneous and adaptable to each patient, which is why this evolution will be accompanied by the digitalization of the processes and data and the use of home automation.

2.5 Towards home cares instead of care homes

This aging of the population will lead to an increase in the number of seniors requiring long-term care at some point in their lives, either at home or in nursing homes. In recent years, the choice between staying at home or going to a care home has been discussed a lot, with the launch of new projects to make seniors' homes more optimal in order to help the occupant stay as long as possible. In Belgium, the Brussels region is increasingly funding research and development in this area of home care.

2.5.1 Description of the global project

The motivation of our thesis is a project from Brussels project that aims to build a smart home care solution for the elderly. The project Vitadomi, funded by the Innoviris innovation fund of the Brussels region, is a collaboration between Electronic Design International (EDI)², UCLouvain, and Sirris (Agoria)³. In fact, our work is only a part of the global project in which we are evolving. To better understand, we will present the general idea of the system.

The general principle of the project is to develop a service to help elderly people to make their homes more elaborate so that they can stay at home as long as possible. This service would be accompanied by a smart hub and a set of data collection devices, such as sensors or home automation instruments, to manage all the risks associated with the aging of the user. This system will react according to events ranging from smoke detection, through a disturbance in their habits, to a fall of the person being accompanied.

²https://www.electronic-design.be/nl_BE/

³<https://www.sirris.be/>

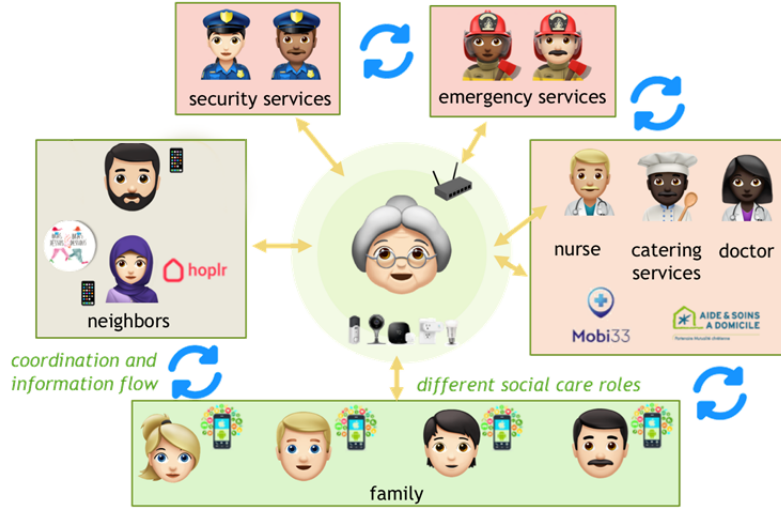


Figure 2.3: Social circle principle

Depending on the importance and risks associated with the event observed by the home automation system, a response action will be performed by the system to address the particular situation encountered. This action will either take the form of an alert, a call, a message, or a notification to the appropriate social circle. As we can see in figure 2.3, the user will define their social circle and will attribute it to each event perceived by the system action and a set of individuals belonging to their social circle. To give an example, the user could define that an alert is sent to their carer in case of a fall or an unusual heartbeat, or that a notification is sent to their close family, such as their children when, the system has detected suspicious activity in the house.

For the proper implementation of this kind of security system, an initialization with a security manager is essential to build a solid base without errors. This initialization will be done through a dialogue between the manager and the user. Afterward, the manager will introduce all the information collected during this conversation through an application or a companion web interface. Regarding the further refinement of the security information, it will be done by requesting a voice assistant to modify the old values set, or create brand new policies. It is in this last part that we add our ideas and it is from these ideas that this thesis is derived.

2.5.2 Market research among elderlies

The vast majority of elderly people prefer, when they have the choice, to adapt to their home so that they can remain in their environment, rather than having to move to a specialized institution. They are indeed rather reluctant to the idea of moving into a retirement home and having to leave their habits and their home.

Most seniors are aware that staying at home can present a risk to their safety. Indeed, advancing age requires the installation of specific equipment in their home, notably to limit the risk of falling, and this goes with expenses that vary greatly depending on the degree of autonomy and health of the elderly person.

These statements are based on the market research carried out by the Brussels project team. They conducted a series of interviews with different elderly people to find out their opinion on the system and to see if they would be inclined to use it if it existed in practice.

We have summarised the important information from these interviews in Figure 2.4. A series of questions were asked about the usefulness of smart devices ranging from emergency buttons to smart fridges, about their use and knowledge of IT technologies, and about their level of concern about the rules governing the protection and use of their data with such devices.

We can say several things from the results of the questionnaires. According to all participants in the study, they need to stay in their homes as long as possible. The emergency button and the fall detector are the most important things if they were to use a smart system. On the other hand, the smart fridge to anticipate future shopping and needs is the least useful device according to all respondents. The opinion on including an indoor camera in the system is mixed, some don't want it for privacy reasons while some might accept it if the camera respects their idea of privacy. Finally, despite the technologically connected profiles of the participants, some of them do not care so much about the security rules that govern information technology, either they don't mind or they say they have nothing to hide. On the other hand, others are very cautious and vigilant.

2.6 Monitoring devices and sensors

Senior monitoring is crucial in the context of home assistance. An automated and intelligent system in a home allows senior citizens to maintain their independence

Age	71	83	71	
Situation	single	single	engaged	
Technology	familiar (social networking, tablet, mobile phone)	familiar (network, tablet, mobile)	hyper connected (watch, mobile phone)	
Daily use of technology	yes	yes	yes	
Perceived loss of ability	yes	yes	yes	
first choice for security	daughter, family circle, child	eldest son, family circle	child, family circle	
Second choice for security	neighborhood	neighborhood	sister	
Importance of staying at home (no nursing homes)	very important	very important	very important	
				Total :
Device notice :				
Emergency button	interesting	interesting	5th	3
Connected watch	interesting	not interesting	interesting	2
Smart blind	not interesting	not interesting	interesting	1
Smart fridge	not interesting	not interesting	/	0
Fall detector	interesting	interesting	interesting	3
Smart lock	not interesting	interesting	/	1
Smart light bulbs	not interesting	not interesting	interesting	1
Electronic pillbox	not interesting	not interesting	interesting	1
Surveillance camera	not interesting	/	interesting	1
Presence detector	interesting	not interesting	interesting	2
Door opening and closing sensor	interesting	not interesting	/	1
Sleep detector, bed occupancy	interesting	not interesting	interesting	2
Smoke, fire, co2 detector	not interesting	interesting	interesting	2
Acceptance of an indoor camera	no desire need for privacy	/	no problem but respect for privacy	1
Knowledge of protection rules	very careful cautious	not careful nothing to hide	not careful, doesn't mind being on file at all	2

Figure 2.4: Survey summary

while ensuring continuous monitoring of the resident. This passive monitoring system for seniors allows the caregivers or the family to be automatically alerted in case of an unusual situation. Common monitoring devices include for example smartwatches, indicating blood pressure and heartbeat, or sleep trackers, analyzing sleep quality.

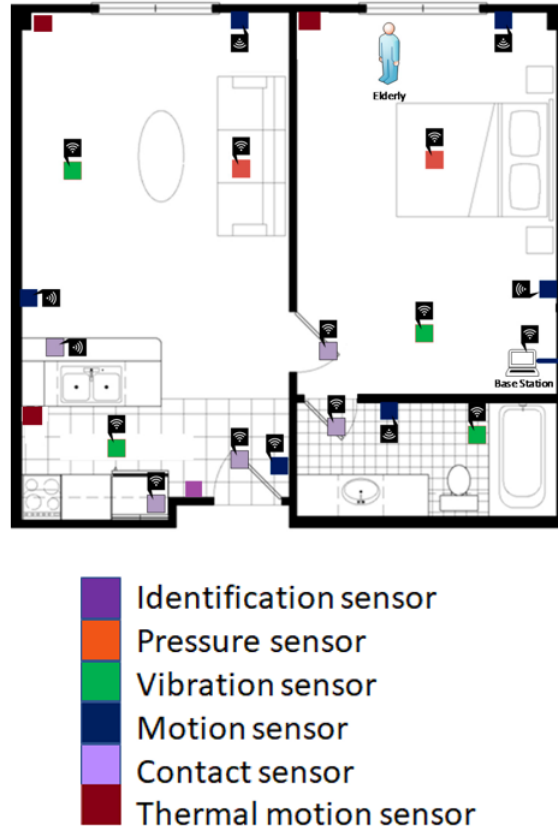


Figure 2.5: System sensors example

The role of the sensors is to capture physiological parameters and body activity as well as object state (door, window, pill dispenser). These devices then transmit all this information to a remote center. This information is processed by software that detect risk situations and inform the appropriate service (nurses in institutions, the doctor in the case of isolated patients). Figure 2.5 could be an example of an overview of a home automation system installed in an elderly person's home.

The Habitat framework [3] includes the following standalone Home Care devices: **wall light** for indoor localization that can for example indicate the presence of someone in a room; **Armchair** for Sitting Posture Monitoring indicating if the senior is not well sitted or sitted for a long period of time; **Belt** for Movement Information, used to detect falls; and many more less relevant to us. This paper helped us seeding our database of events that could be triggered in a senior's smart environment.

2.7 Protection-aware smart hubs

New privacy-aware smart hubs are emerging, allowing to prevent applications from sharing fetched private data without the user’s permission. The data is kept securely inside the hub and it only dispatches events such as a fall, smoke/fire detection, or the presence of another person in the room. Some events might be critical to the elderly’s health and need to be dispatched to a higher authority such as children, carers, or neighbors. But on the other hand, a user might find it very intrusive to share casual events with people he does not want to.

In recent years, many projects [11, 19, 23, 13] have been carried out by researchers and teams with the aim of creating a system of privacy control through smart hubs. What stands out most of the time is the principle of increasing the protection against external threats and therefore decreasing the risks. These projects can be divided into three types, those who want to increase security, those who want more safety, and those who fight for privacy. To get a global view of existing systems, we read and took ideas from papers in each of these types of protection.

2.7.1 Privacy-aware smart hubs

Regarding privacy, PFirewall by Haotian Chi et al., [11] was really useful for our solution design.

The principle of PFirewall is to control the data flow of a connected home or home automation system to improve the privacy of the users of this platform because in most cases the owners have no control once the home automation devices are linked to the platform. To do so, their platform automatically generates data minimization policies to disclose the strict minimum for the proper functioning of the devices or automation through a user-customizable interface. The user will be able to edit their preferences regarding the data they want to send to the smart hub. In general, PFirewall acts as an intermediary between devices such as sensors or connected objects and the platform that manages them in a transparent manner as we can see in Figure 2.6. This device retains almost 97% of the data without harming the proper functioning of the system by removing non-essential data. It’s not a question of not trusting their smart hub platform providers but the fact that they are a bit too curious in taking data, which impacts the privacy of the user.

We took inspiration from this system for the treatment of data sending limitations. Like PFirewall, at runtime, the system will check the policies associated with that particular run of an IoT device before sending new sensor data to the platform. In addition, it will also be possible as in their case that users can also define data

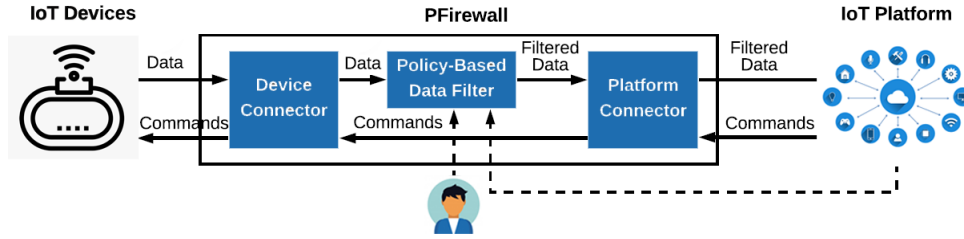


Figure 2.6: Architecture of PFirewall

policies. On the other hand, in relation to our particular user case in our thesis, there is a major limitation regarding the user effort. In fact, users must pair their IoT devices with the PFirewall mediator on a companion mobile application to connect PFirewall to the target platform. This mobile interface will also be used to edit policies, which is a huge effort and requires a lot of expertise for elderly people who in some cases are physically and mentally attenuated. Therefore, only the system for creating and editing data sharing limitation policies is considered in our paper as another interface system has been selected to address this problem.

2.7.2 Safety-and-security-aware smart hubs

Regarding safety and security, there is a lot of work that has been done over the last 10 years [13, 19, 23]. For the most part, it checks the safety and security status of the system through different mechanisms. The threat can come from either a bug, an application that creates a safety/security violation, malicious activity, or normal cross-app activity that would lead to an insecure or unsafe state.

To address these risks, several solutions have been proposed. For the case of Soteria [13], they made a statistical analysis to extract the state model which is then analyzed for compliance. But this type of analysis does not fit our thesis because defining fixed safety properties that would apply to all users is not an adequate solution because each user will have their own opinion on the dangerousness or not of the element for which the property applies. This can be seen in the paper [2] where some elderlies are reluctant in using devices with cameras while another percentage of the study have no problem with these devices.

With regard to HomeGuard [23], they aim to discover any threats and unsafe states related to inter-application interaction. This is done by running the installed application to check its behavior against the behavior of the applications already installed and trying to prove that no conflicts occur. They then rank the associated

risks and present the detailed results and solutions to the users in a user-friendly way through a front-end interface so that they can decide to remove the new application. It is also possible for the user to reconfigure it. In principle, their project [23] has the same goal of facilitating the security management of the environment by deploying an efficient and simplified interface. This kind of problem is very difficult to solve for someone who has no expertise in the field, which makes it an interesting article for our project. On the other hand, the field of study of problem-solving is not wide enough, indeed, it focuses only on problems related to inter-application interactions and only that. Indeed, a simple uninstallation of the software can solve its problems. So in general, their usability and procedural facilitation techniques are interesting but their application is not very useful in our case.

Another interesting system is IoTGuard [19]. This dynamic policy enforcement system on IoT devices protects users from unwanted device states by monitoring the behavior of applications. Its approach is similar to that of HomeGuard [23] and PFirewall [11] in some technical aspects. For example, they focus a lot of analysis on the interactions between devices like HomeGuard [23] because they believe that these interactions are the real causes of security and safety violations. Furthermore, in the operation of their system, they use the same logic of recognizing security and safety policies as PFirewall [11] and apply these policies in a pertinent and proactive manner for each interaction of sets of applications. On the other hand, as stated in their writings, one of the limitations of their system is the correct decision-making for blocked states. The simple blocking of a state by a user policy can have physical consequences like locking a door. If a not-complex enough policy locks a door when the user is on holiday and that policy blocks the door when maintenance or security would like to access the room through a time slot already specified by the system, this situation may or may not be ambiguous depending on the circumstances. To help and keep the IoT environment stable when an action is rejected, they want to create more sophisticated policies to better handle blocked states. This problem can also arise when a user changes policies via the IoTGuard platform. Indeed, when the system is very complex and the specification of a policy is incorrect, the system can prevent legitimate states from acting and even encourage its states to conflict with other policies. To give an example, one policy may allow an action when a specific event occurs, while a second policy may deny a set of actions in which the first one is a part of.

This is proof that despite making the use of its security management systems more and more ergonomic and intuitive, it is still possible for the user to make a mistake and even more in our case with elderly people who have no knowledge

of its interaction problems. This is why one of the main goals of our project is the fact that our system will only be a compliment and not a replacement of these automated systems while keeping in mind that a monitoring system of the management of this system will be put in place by the competent authorities such as the hospital services.

2.8 Dialogue-based policy manager

We have seen how difficult it is to manage a privacy manager, and several examples of privacy management systems. We have also seen the context of home assistance for the elderly that needs to evolve. Now we have to ask ourselves the question of finding the best way or the best tool to link all these issues. We have found it and it is a policy manager based on dialogue. The aim of this thesis is to develop a more ergonomic and convenient way, presented as a dialogue-based interface, to manage and monitor personal data in the context of medical monitoring of a senior person.

Juan Manuel Lucas Cuesta et. al. made a study [10] which presents an improved method for user feedback by allowing users to update their preferences based on data obtained from different sensors, they use a spoken dialogue system to manage interactions between users and the system.

The choice of a voice assistant as a user-system interaction tool was validated by their team on the basis that voice dialogue systems are very well suited to meet the requirements of IoT domains such as automotive, health, and others. They allow people to communicate with a system in a more natural way, without the need for technical training. According to them, this is a much more natural way of interacting than forcing users to use various interfaces or hardware such as a PDA, keyboard, touch screen, remote controls, or other physical devices. This kind of ambient intelligence that supports people in their daily lives in a proactive and discrete way offers computer technologies the possibility to improve the living conditions of certain segments of our society such as the elderly.

Our task will be to probe the user's concern about security and privacy by generating a set of questions with formal answers such as **yes**, **no**, or simple words such as **my son**, **my neighbor**. This will generate well-formatted policies that would be used both when filtering data flows and when reacting to a specific event that would require notifying one of the user's relatives.

We chose this type of questionnaire in relation to the result of Melinda Hamill et

al. [15]. In their case, it uses a simple two-word vocabulary of **yes** and **no** to greatly improve recognition. Overall recognition rates increased from about 50% to 90% in their project. This is due to the fact that they use only two words to considerably limit their misunderstanding which can be critical during an emergency or in our case for policy configuration affecting the management of confidential data. We have chosen to take this approach but also to allow a bit more freedom while maintaining low word errors by asking non-boolean questions that would require the user to answer with the name of a specific relative that would be enumerated by the voice assistant.

2.9 Acceptance of a voice interface by elderlies

Due to the limitations of the existing systems and solutions described in the previous section, we seek to find a simpler way to make our system practical and easy to use for older people. As mentioned before, it is often very difficult for the elderly to follow the drastic improvement in technologies nowadays.

To facilitate the use of this policy management device, our user platform will be a voice assistant allowing the elderly to set their own preferences without having to use a screen-based application that could be more complicated to use. The generation of our grandparents is the last technophobic generation that did not have experience with the latest technologies. This problem will disappear with time, until the moment when the majority of the world population will be born and immersed in this technological era. But during this lapse of time, it is our duty to minimize the user effort of these technologies for our grandparents. Moreover, voice assistants are very well accepted by the elderly because they allow them to improve their living conditions and to remain autonomous longer.

Studies [14] show that the majority of the elderly are willing to use a voice interface for home automation. Their results showed that voice technology has a great opportunity to facilitate the daily life of elderly people. Furthermore, their participants particularly appreciated the fact that voice technology could provide more safety, such as the fall management and help call system.

On the other hand, it appears that the improvement of comfort was only a secondary concern, moreover, it raised the concern of the elderly who feared that such a system would make them lazy and cause a faster deterioration of their health condition. To sum up, home automation associated with a voice assistant should not promote an unhealthy lifestyle, which implies that we should consider that

users are at the center and that technology should be adapted to their needs and not the opposite.

Chapter 3

Methodology

In this section, we present a dialogue system that allows users to manage their security rules in a simple and transparent manner. To better understand how this works, we will use a top-down approach by starting with an overview of the overall system and its design principles. We will present the conceptual contribution first and then the technical aspects in Chapter 4.

3.1 System overview

Our objective is to design, develop and evaluate a voice interface allowing users to control their privacy and security preferences. The developed voice interface should allow users to set their access control preferences in an intelligible way. To change their preferences, the voice assistant will ask a series of questions chosen according to an algorithm that takes into consideration all the useful data, such as the daily use of the policies or the people most solicited by the smart hub, based on examples of information that should or should not be sent to specified relatives.

The system consists of three main parts as described in Figure 3.1:

- **The privacy-aware smart hub** that will hold all the rules defined by the user, either with a screen-based interface on setup or with the voice assistant. The hub will send some data, that will be described later on, to the voice assistant and wait for a response from it in order to process the answer and create, or not, new rules.
- **The voice assistant** that will get relevant data from the hub to create new questions and will react accordingly to the user's answers. When the assistant has all the pieces of information needed for the creation of a new policy, it will send it back to the hub to be processed.



Figure 3.1: System actors

Remark: The voice assistant would be an integrated part of the hub and not a third physical component per say.

- **The user** the assistant will interact with in order to create new rules or update existing ones.

The part that concerns us in this paper is the one that acts as a link between the user and the smart hub for the management and editing of security and privacy policies, the voice assistant. This assistant should be able to manage full dialogue and react accordingly to the user’s answers. Pre-defined questions would be fed to the assistant which would then ask more precisions about the user’s concern on its security. In this section, we will discuss the semantic of those questions in the context of policy creation and monitoring, the format in which those policies would be store in order to be effectively used, and finally, some heuristics for improving the quality of the questions asked.

3.2 Dialogue system

To address the issue of screen-based interfaces for defining security policies, we present an easy-to-use dialogue-based interface. In this section, we will discuss the interaction between the assistant and the user along with some examples. We first describe the questions asked by the assistant to the user as well as the expected answers which will in the end create, or not depending on the user’s answers, new security policy rules. The policy rules format will be elaborated in Section 3.3.

3.2.1 Question semantic

We can define two kinds of questions regarding the user’s policies: questions that would ask for the creation of new policies in concern of some specific data flows, or questions about already existing policies in order to check if the preferences of the

user have not changed. We will discuss the semantic of both types in the following paragraphs. It is important to keep in mind that questions regarding the creation of new policies can only involve events that are not already defined in any policy rules. Similarly, when monitoring policies, only events being covered by one or more policies will be involved.

Policy creation

When creating new policies, the user will be asked a formal question describing precisely the kind of event involved in the policy as well as the actor the data needs to be transmitted or not. We will take the following generated question as an example.

Voulez-vous que je notifie votre voisin ou votre fils si vous oubliez de prendre vos médicaments?

In this case, the event being triggered has been sent by the smart pill dispenser that has detected that it has not been opened when it was supposed to. The assistant will then offer a choice of one or two actors, in this case, the neighbor or the son, along with an action to take, a notification.

In this context, we will talk about taking actions regarding specific actors but it is easy to think of this as an allowance or denial of data flow. Indeed, when no policies are defined we can think of it as the smart hub denying the sending of data to anyone. When new policies are being defined, this will perform two tasks: one regarding security by warning relatives of some severe or not events, and one regarding privacy by allowing a flow of data between the senior and the concerned actors regarding specific events.

Policies monitoring

It is sometimes more important to update deprecated policies for risky events than to create new policies for benign ones. Several scenarios make the monitoring of policies an important process.

- When an actor involved in a policy is being altered (e.g. dies or simply moves out), it might be wise to change this policy which might not be valid anymore. What if a severe accident is being detected, such as a fall, and that the only actor being warned cannot be reached anymore?

- Another important aspect is that the user might want to change and update existing policies that are often triggered. Take as an example a policy that warns the user's son each time there is an increased heart rate detection. If the user takes some medication that has the side effect of increasing the heartbeat, the policy might be triggered a lot when it is not so necessary.

Some ideas of solutions will be presented in Section 3.4 where we describe our heuristics to determine relevant questions for either the creation or update of policies.

3.2.2 Expected answers

As mentioned in Section 2.8, answers need to be short and intuitive in order to reduce the risk of word errors. This is why answers can only be of two kinds: boolean (*oui*, *non*) or relative names (*fil*, *fille*, *voisin*, *police*, ...). Below are example questions along with their expected answers.

- *Q: Voulez-vous que je notifie votre voisin ou votre fils si vous oubliez de prendre vos médicaments?*
Expected answers: {"Non", "Oui", "Mon voisin", "Mon fils", "Tous les deux"}
- *Q: Si vous chutez, dois-je appeler votre fille?*
Expected answers: {"Non", "Oui"}

One might notice that for the first question we allow *Oui* as an answer which might seem like an inappropriate reply. Those kinds of questions can be seen in two different ways:

- the question expects one of the enumerated actors as the answer
- or the assistant asks the user if it should warn ANY of the enumerated actors, in which case the user might reply with *Oui* or *Non*. The assistant will then ask for precisions regarding the actors, for example with the following question:
Q: Dois-je notifier votre voisin, votre fils, ou les deux?
Expected answers: {"Mon voisin", "Mon fils", "Tous les deux"}

Unintended answers

It can happen that an unintended answer is given to the assistant. In such a case, it will ask the user to repeat the answer until an expected answer is given back. If this fails three times in a row, the assistant will warn and repeat the question.

To avoid cases where the assistant would be stuck because of not receiving proper answers, the questions need to be scoped in such a way that would not leave the user with any hesitation when thinking about an answer.

Giving unintended answers multiple times might also be a sign of decline in the health of the user. Specifically, during the first few minutes of a stroke, people tend to have problems with speaking and, in such cases, the assistant should warn the relatives of this happening.

In addition to unintended answers, voice samples might not be well processed by the speech-to-text tool and a different answer would be understood by the assistant. To minimize this risk, we present in the Evaluations section a small test to determine which speech recognition tool is more adapted to the user. More details about this can be found in Section 5.1.2.

This risk is always present and one thing we can do is asking a confirmation for each answer given. This solution however presents some downsides such as user fatigue which is amplified with older users. The following small dialogue gives an example of how this could be used:

Q: Voulez-vous que je notifie votre fils si vous n'avez pas reçu de visite ces deux dernières semaines?

A: Oui

Q: Très bien, je vais notifier votre fils si cela arrive. Dites "Oui" pour confirmer.

A: Oui

3.2.3 Full dialogue semantic

The user might not be happy with the choices he has been presented, specifically the actors selected and the action taken. In the case of a negative answer from the user when being presented the question, the assistant will ask if the user is willing to change the action that was offered to him. If he does accept, the assistant will find the second most relevant action for the current event (based on some properties described in Section 3.4) and offer him this new choice. The user can refuse until he finds an appropriate action or until there are no more actions available. The same process will happen with the available actors until, eventually, the user creates or not the new policy rules.

3.3 Policy format

Asking questions about the user's concern on privacy and security is one step but without an appropriate format for those policies, it quickly becomes unusable and therefore useless. In this section, we present our policies defined in a format that is similar to the one used by Haotian Chi et al. [11] which was thought for the Samsung SmartThing smart home platform.

3.3.1 Event description

In their paper, they present their data-flow policy as $\mathbf{P}=(\mathbf{T}, \mathbf{C})$, $\mathbf{T}$ being the triggering event (TRIGGER) of the policy and $\mathbf{C}$ the list of constraints on this event. We can adapt the format of TRIGGER to our case and we get the following triggering event format:

Listing 3.1: Triggering event format

```
TRIGGER: {  
  match (:type).(:subject).(:attribute)  
  satisfy (:operator)->(:value)  
}
```

As described in the paper, **type** indicates if the event was fired by a device which will always be the case for our prototype for simplicity reasons. **subject** tells us which device the event was fired from and **attribute** which kind of data. For example, there are monitoring devices that can act both as a heart-rate monitor and an electrocardiograph. The **operator** and **value** fields indicate the condition for this policy to be triggered.

Listing 3.2 shows examples of events defined in the PFirewall policy format. The full list of events we used for our prototype can be found in Listing A.1.

Listing 3.2: Event example for low heart rate and anomalous sleep

```
TRIGGER: {  
  match (device).(monitoring).(heart-rate)  
  satisfy (<)->(60)  
}  
  
TRIGGER: {  
  match (device).(monitoring).(sleep)  
  satisfy (==)->(anomalous)  
}
```

Values are processed by monitoring or sensor devices which in turn send only relevant data to the data-filtering hub. For example, if we have a device monitoring the sleep quality, we are not interested in any *noise* data but rather in the conclusion the monitoring device made based on this data. The sensor would in that case only warn the assistant if it has detected that the user had an anomalous sleep which is why we are checking if the event value is **anomalous** in Listing 3.2.

3.3.2 Taking "actions"

Of course the final goal of having policies defined is to perform specific action(s) when those events are being triggered. Listing 3.3 presents the actions taken as a set of **action.actor** pairs where **action** defines a way of communicating (see Appendix 3.2 for available actions) and **actor** is a user's pre-defined relative (see Appendix 3.3 for available actors).

Listing 3.3: Event actions format

```
ACTION: [{
    perform (:action).(:actor)
}, ...]
```

ACTION can contain as many action-actor pairs as needed and thus does not restrict to perform the same action on multiple users as they can all have their own action. Listing 3.4 gives an example of a full TRIGGER-ACTION definition.

Listing 3.4: Event actions examples

```
TRIGGER: {
    match (device).(monitoring).(heart-rate)
    satisfy (<)->(60)
}

ACTION: [{
    perform (message).(son)
}, {
    perform (call).(carer)
}, {
    perform (notify).(neighbor)
}]
```

3.4 Policy heuristics

Asking lots of questions can be exhausting, especially when you are getting older. We will present in this section some ideas on how to prioritize some events and optimize the questions asked based on some properties.

We start by offering the reader a classification of the different events, actors, and actions that we thought relevant as well as how to use this classification to improve the proposed policy quality. Then we discuss the matter of monitoring already defined policies.

3.4.1 Classification

It can be trivial to find what kind of actors would suit best for a specific event, the same goes for an action to perform. For example, if we take the event of a fall, we should probably make a call which is instant instead of notifying which could take quite some time to be noticed. In the same idea, the neighbor might be quicker to react than the children or friends.

This is why we have given for each of the events, the actors, and the actions, specific properties that will help match the three together better.

Events classification

We defined four properties to represent an event, each on a scale of 0 to 3 where 0 means *of lesser importance* and 3 *of most importance*:

- **Physical health:** represents the severity of the event on the physical health;
- **Mental health:** represents the severity of the event on the mental health;
- **Reactivity:** represents the degree of reactivity needed by the event. 0 means that no immediate actions are needed while 3 means an immediate one;
- **Privacy:** represents the degree of privacy of the event

As an example, for the event of a fall, we gave the properties shown in Table 3.1. A fall is of course of greater threat to physical health than to mental health, is not considered to be particularly private and needs to be taken care of immediately. Each event classification can be found in Appendix A.1. This classification has been done subjectively and should thus be taken with a grain of salt. A better grading scale and more suited values would improve the quality of the classification of the events.

Physical health	3
Mental health	0
Reactivity	3
Privacy	0

Table 3.1: *fall* event classification

Actors classification

When it comes to actors, there are a few things that would make one actor a better choice for an event than another one. Some events treat sensitive data such as the sleep quality or the weight of the user that not everyone should know about. We can think of this as some actors being allowed to know more about the person than some others. The same idea goes with the reactivity needed by a triggered event as some actors such as a neighbor are closer and more capable of directly intervene. We thus came with the following properties:

- **Privacy:** represents the degree of privacy the actor is allowed regarding the user. 0 means that the lesser amount of data should be communicated and 3 means that there is basically no issue in the actor knowing private pieces of information about the user.
- **Reactivity:** represents the degree of reactivity. A reactivity of 3 means that the actor is able to help the user in a short amount of time while a reactivity of 1 means that the actor would take more time to come and help the user. The value we attributed can be changed to more relevant ones as we don't know if the children live close or not from the user. This is all very personal and needs to be adapted to each user.

Table 3.3 gives the full definition of the actors classification.

Actions classification

We have defined three actions that all differ in their degree of reactivity. A call is an instant action that ensures a better chance of being noticed than a message and a notification. We thus describe each action with the following property:

- **Reactivity:** represents the degree of reactivity of the action on a scale from 1 to 3 where 1 is the lesser reactive and 3 is the immediate action;

Table 3.2 gives the full definition of the actions classification.

Action	Reactivity
Notification	1
Text message	2
Call	3

Table 3.2: Action listing

Actor type	Privacy
Son	3
Daughter	3
Grandson	3
Granddaughter	3
Neighbour	1
Carer	3
Police	2
Ambulance	3
Firefighter	2

Table 3.3: Actor types listing

3.4.2 Policy creation

With those classifications in mind, we can start to think about how to use those properties to optimize policy creation.

Selecting an event

When it comes to selecting what event the assistant should create a policy for, we can rank the events based on a *score* that would make use of the different properties we defined above and select the one with the highest, meaning the more important.

For example a simple event picker that we used for our evaluation was based on the following heuristic:

- (1) Sum the **Physical health** and **Mental health** properties of each event.
- (2) Select the one with the highest value.
 - (a) In case of a tie, select the one with the highest reactivity value
 - (b) Tie-breaker rule: select at random

Another way of selection is by logging each time a specific device has triggered an event and keeping a counter of the amount of time it happened. If there are no policies covering this event, it might be useful to create one as this event is happening a lot. Thus, a heuristic could make use of this counter to better rank the events according to their occurrences.

Selecting an actor

It soon became logical that the more actors we allow the user to choose from, the better chance we have that it will match their preferences. For example, if we only ask if a specific actor should be warned for an event, the user can only accept or refuse while when giving two actors, he might choose one, or both.

To improve the selection of actors, we can use some precious pieces of information from the already defined policies such as the number of policies an actor is involved in. This criterion will lead to the selection of the first actor. For the second one, we use the **reactivity** and **privacy** properties defined in the event and try to find an available actor that matches those criteria the closest.

Selecting an action

When it comes to selecting an action we have thought of three ways to improve the quality and relevance:

- Select the one matching the **reactivity** property of the event
- Select the action that is more often used in the current policies
- Select the action that is more often used with the selected actors

For the evaluation process, we used the first method of selection actions because of the lack of existing policies when using the voice assistant for the first time.

3.4.3 Policy monitoring

We mentioned in Section 3.2.1 the reasons why monitoring was important and will now describe how to determine when asking questions regarding existing policies would be relevant.

If an actor has been altered (updated or deleted) then we should probably make sure that the policy is still valid. The concerned policy would thus have a higher priority than the creation of a new one and be asked first. This will allow the event to create brand new policies or have none at all. In the latter case, the event concerned would join the other uncovered events until it is picked by the heuristic.

To achieve this, we can add three timestamps fields, **created_at**, **updated_at**, **deleted_at**, to each actors. If **created_at** and **updated_at** differ, this means that the actor has been altered. If **deleted_at** is set to any date, it means that the actor has been soft deleted and is not available anymore.

In the case of an event being triggered a lot ¹, asking the user if the defined policies regarding this event are still the user's expectation would take precedence over the creation of new policies. Note that to prevent an overflow of policy monitoring, one should keep track of when was the last time a question regarding an event has been asked in order to not ask it for a certain amount of time.

3.5 Conclusion

In this chapter, we have defined the scope of our voice assistant, how it will interact with the user, and how to use the answers from the user to create or update policies. The choice of heuristics is of course entirely subjective and should be the subject of further studies that are beyond our scope. The ones we have defined in the above section seemed well suited for our evaluation process but probably not for a final product.

Defining policy heuristics is an important process to ease the use of the assistant by the elderly. There are few things that we did not take into account such as how should the events of a newly deployed device be defined. Should the event properties be attributed by the user or by the device manufacturers? Giving this freedom to the latter one might lead to over-estimated values because of a *competition* between them where they would like their own devices to have policy rules before the others.

¹We don't discuss in this thesis the numerical aspect when the event triggers are considered to be *a lot* as this threshold would be individual to each event and is beyond our scope.

Chapter 4

Implementation

In this chapter, we present how we implemented this voice assistant by describing each and every component needed in order to create policies, generate questions and interact with the user. The different components required for our voice assistant are as follows:

- A set of voice-assistant services for the dialogues. In order to be fully operational, the services needed are:
 - **speech-to-text**: to transcribe the user’s answer into text;
 - **text-to-speech**: to read text sentences out loud to the user;
 - **intent recognition**: to match the text transcribed with a set of pre-defined *keywords* or full *sentences*;
 - **dialogue manager**: to handle the full scope of the discussion, e.g. detecting when the user finishes a sentence ;
- A database, or in our case several databases, to store the user’s data;
- A REST API to create, read, update, delete policies and fetch relevant questions;
- A web interface for setting up the assistant and performing the evaluation;

4.1 Voice-assistant services

Using one of the leading voice assistants such as Amazon Alexa, Apple Siri, or Google Assistant was something against our goal of improving the user’s control of their data. On one hand, we want to restrict the data-flows but on the other, we would allow new ones. This is why open-sourced voice assistant is what we

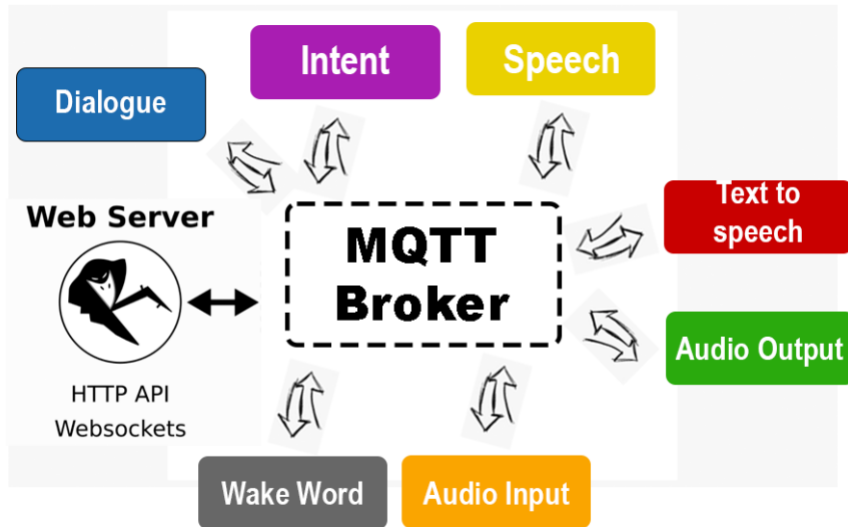


Figure 4.1: Services from Rhasspy

eventually decided to go with.

There are many open-source voice assistants available, each and every one of them having their own specs. Some notable ones that we considered were:

- **Almond:** developed by the Stanford University, this *Privacy-Preserving Virtual Assistant* allows you to define commands in a similar manner as you would when defining Amazon Alexa Skills. The issue here is that there is no real control of the services used and French is not an option.
- **Kaliope:** this lightweight voice-assistant targets specifically French speakers for home automation tasks but does not give enough freedom when it comes to customizing the services mentioned above.

The one that looked the more promising at first glance was **Rhasspy** and is the one we chose to use. Rhasspy is a fully customizable offline set of voice-assistant services that are easy to deploy and configure. Figure 4.1 describes the services communicating with each other via an MQTT broker.

4.1.1 Speech-to-text

When a user interacts with our voice assistant, an audio file of the said sentence is saved and sent to a speech-to-text software that will try and transcribe it into French text. Rhasspy includes several software solutions such as **Kaldi**, **Mozilla**

DeepSpeech, and **PocketSphinx**, each having their own pros and cons. To choose the most suitable tool, we performed some tests on the participants (See Section 5.2.2 for the comparison of the speech-to-text tools) and picked **Kaldi** as it was the most efficient one.

4.1.2 Text-to-speech

Once the voice command has been processed, it is common to produce speech as a response to the user. As we can see from Figure 4.1, Rhasspy comes with a text-to-speech service to transform a textual item into an auditory one. Rhasspy offers several text-to-speech systems, they are very similar so anyone will do as long as it supports French. The only differences are the voice grains and the more or less robotic pronunciation of each. For the evaluation purpose, we used the built-in text-to-speech tool from the Firefox engine which was easier to use although a bit robotic.

4.1.3 Intent recognition

Once the user's sentence has been transcribed into a text format by the speech-to-text service, the next step is to match it with a set of predefined intents. In our case, we don't handle the intent with specific actions, such as turning the lights on, but just use them to match the user's answers with a set of expected answers. This allows a simplification of the matching phase as we don't need to manually check each raw token/word to determine what the answer was. We can simply define all the expected answers in a single file and let Rhasspy decide with a confidence level what the spoken answer was.

The intent recognition service will try and find the most probable intent based on the user's input and return a JSON with plausible intents along with a confidence level. Listing 4.1 gives an example of a speech-to-intent request with the **Kaldi** speech recognition toolkit and the **Fsticuffs** intent recognition. In this case, the audio file was containing the *Tous les deux* sentence and the tool gives a confidence level of 1 for this exact intent.

Listing 4.1: JSON response from the intent recognition service

```
1 {  
2   "entities": [],  
3   "intent": {  
4     "confidence": 1,  
5     "name": "AnswerBoth"  
6   },
```

```

7   "raw_text": "tous les deux",
8   "raw_tokens": [
9       "tous",
10      "les",
11      "deux"
12  ],
13  "recognize_seconds": 0.05541273500011812,
14  "slots": {},
15  "speech_confidence": 1,
16  "text": "tous les deux",
17  "tokens": [
18      "tous",
19      "les",
20      "deux"
21  ],
22  "wakeword_id": null
23  }

```

4.1.4 Dialogue manager

When discussing with a voice assistant, it is almost a necessity that the voice should be the only part of the body doing the job and not for instance a finger clicking a **RECORD** button. There are many open-source tools that effectively detect when a voice sample starts and finishes by guessing what bits of the audio are silences. We can use such a tool to perform a full dialogue where the assistant will ask questions, wait for the user to finishes a sentence, remove silences from the audio, and transcribe it.

4.1.5 Speech-to-Intent (Bonus)

There are a few drawbacks when using two separate tools to first transcribe audio into text and then text into intents. If the speech-to-text tool has no idea of the available answers, it might wrongly transcribe some words that sound alike but have completely different meanings. Take the example of the word *mère* which also sounds like *mer* and *maire* in French.

To bypass this issue, Rhasspy makes available speech-to-intent tools that will make use of a dictionary of the available words defined in the list of intents when transcribing audio files to text.

4.2 Orchestrator: Laravel Framework

Laravel is a PHP-written web application framework with a large ecosystem and active open-source community. It particularly comes in handy for building web-app requiring authentication, job processing, event broadcasting, API, ... Additionally, Laravel very rich ecosystem can ease developers' life with numerous free packages such as debugging (Laravel Telescope), administration panel (Laravel Nova), deployment (Laravel Sail), token-based API (Laravel Sanctum), ..., all of which are free and easy to set up. Laravel also makes it easy to interact with any kind of database (SQL, NoSQL, Redis, GraphQL, ...).

We chose Laravel for our project based on personal preferences and any other popular frameworks (Django, Ruby on Rails, Express.js) would have been good alternatives.

4.2.1 Policy heuristics and question generator

We chose to implement the heuristics and question generator described in Section 3.4 in PHP directly within the Laravel project as it was the easiest way to fetch databases data, make complex queries, and work with collections.

When a request for the generation of a question regarding the creation of a new policy is made, the controller will call a service that first verifies if there are any uncovered events and if at least one actor is defined. The generator will then select the most relevant event and one or two actors along with an action.

To construct the spoken sentence, we created the following question template that will be fed the chosen event, actors and action:

- *Dois-je {{action}} {{actor}} si {{event}}?*
- *Si {{event}}, dois-je {{action}} {{actor_1}} ou {{actor_2}}?*

In order to feed the question with correct sentence syntax, we added a **sentence** column to the events, actions, and actors tables describing, in French, the item. For example the fall event would have the sentence *vous chutez*, the son actor *votre fils*, and the call action *appeler*. We can then get from the above template the following sentence: *Si **vous chutez**, dois-je **appeler** **votre fils** ou **votre voisin**?*

4.2.2 API

For the purpose of testing, we built a multi-tenant software where each tenant is a user of the supposed smart environment. When performing live tests on real

users, it soon became apparent that a single-tenancy system was not appropriate for the sake of keeping track of every tester's data. The API we built is therefore a RESTful API where the user is specified in the HTTP request URI when accessing resources.

Authentication

In the scenario of a multi-tenancy system, authentication is of course primordial to identify the tenant and forbid access of resources, such as policies or actors, to unauthorized users. The Laravel Sanctum package provides an easy way to authenticate users on a Single-Page-Application (SPA), such as ours, by issuing personal access tokens. It is designed, amongst other things, for applications that need to communicate with an internal API.

4.2.3 Web interface

While the goal of this thesis is to develop a non-screen-based interface, the web application is still useful for two things:

- **Setting up the assistant:** It is important to remember that the voice assistant should only be used by the user when the system has been deployed. This deployment needs to be supervised by a technician or a relative more at ease with technology to avoid any misconfiguration. In the setup process, actors such as children or neighbors will be defined, and doing so with a voice-based interface can quickly become tricky when giving a phone number for example.
- **Evaluation:** Comparing a screen-based interface with a voice-based one can be tricky if the user has only access to the latter one. Thus, we developed this web application also for evaluation purposes. Section 5.1.3

4.3 Databases

4.3.1 MySQL

Config: MySQL:8.0 ¹

¹<https://hub.docker.com/layers/mysql/library/mysql/8.0/images/sha256-68b207d01891915410db3b5bc1f69963e3dc8f23813fd01e61e6d7e7e3a46680?context=explore>

In order to store the policies defined as well as the events handled by the app and the user's relatives' personal pieces of information, such as a phone number or the relation between them, we chose the approach of a relational database.

The database is used to store global data such as the events or the actor types available, that are needed when installing the assistant, as well as user-specific data, policies, and actors information.

Amongst other things, Laravel makes it particularly easy to interact with a SQL database with the help of the object-relational mapper Eloquent. Listing 4.2 shows how easy it is to use Eloquent to fetch the covered and not covered events of a specific user. Line 5 retrieves the user based on the uniquely defined `name` column for the `User` model. The current policies of this user can be retrieved easily thanks to a `HasMany` Eloquent helper class. Note that, on Line 6, `$policies` is a query builder object and not the collection of defined policies. Line 7 retrieves the id of all the events the user already has a policy for and on Line 8 and 9 we can easily get the desired events.

Listing 4.2: Eloquent example

```
1 <?php
2 use App\Models\User;
3 use App\Models\Event;
4
5 $user = User::firstWhere('name', 'donatien');
6 $policies = $user->policies();
7 $eventIDs = $policies->pluck('event_id');
8 $coveredEvents = Event::whereIn('id', $eventIDs)->get();
9 $uncoveredEvents = Event::whereNotIn('id', $eventIDs)->get();
10 ?>
```

4.3.2 Redis

Config: Redis:alpine ²

In addition to the SQL database described above, the need for a faster database was soon apparent for the purpose of monitoring events. As explained in Section 3.4, the assistant makes use of numerous parameters, one of which is the number of times an event has been triggered. Those kinds of events might happen frequently and it is therefore important to be able to log them without slowing down the app.

²<https://hub.docker.com/layers/redis/library/redis/alpine/images/sha256-858da560de4602625d8c7f5a2cc80ff538fc63160cafe3b29e43c9a0b904acfa?context=explore>

Redis is an in-memory key-value data storage that has the benefit of being able to hold persistency which is something valuable in our case. It is in particular extremely fast for incrementing value which is exactly what we need.

We, therefore, use Redis to keep a counter of each event being dispatched. Each counter is identified in the database by the following key, `event:counter:{event_id}`, where `event_id` is the primary key of the event in the SQL database and where the value is the number of times this event has been dispatched to the app.

The app communicates smoothly with the Redis server, running in its own Docker container, via the Laravel built-in Redis Facade³. This Facade allows us to use the methods we are interested in: the getter, the incrementer, and the setter.

Listing 4.3 shows how easy it is to interact with Redis within a Laravel application.

Listing 4.3: Redis getter and incrementer

```
1 <?php
2 use Illuminate\Support\Facades\Redis;
3
4 Redis::incr('event:counter:1'); // -> 1
5 echo Redis::get('event:counter:1'); // print 1
6 ?>
```

4.4 Deployment

We made our voice assistant easily deployable and configurable with the help of Docker, which isolates each of the components needed. In this section, we will explain how to install and replicate our evaluation setup, and how to configure the different Rhasspy services.

4.4.1 Docker

Figure 4.2 shows the four containers with their respective exposed port. The containers can be built and set running easily with the Composer file shown in Listing B.1. There are two versions of the Laravel app available depending on the PHP version of the system. PHP 8.0 is still a recent release (November 2020) and PHP 7.4 is considered to be more stable. Nonetheless, both versions can be installed thanks to the two separate Dockerfiles present in the respective `docker/7.4` and

³<https://laravel.com/api/8.x/Illuminate/Support/Facades/Redis.html>

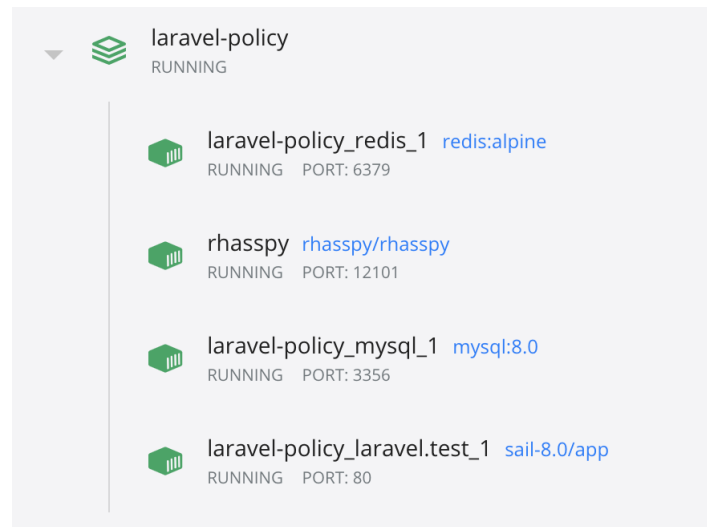


Figure 4.2: Docker containers

`docker/8.0` folders. The version can be easily specified in the `docker-compose.yml` file on line 5 in Listing B.1.

4.4.2 Configuration

Some configuration needs to be done when deploying the system for the first time. The migration of the database is handled by Laravel and all we need to do is to define the migration files along with some models, such as events or actor types, that we would like to be seeded when migrating a fresh database. For the web interface, assets need to be compiled with the help of the Laravel Mix library. The following steps will effectively set up the Laravel app:

- (1) `php composer.phar install` (or `composer install` if Composer is globally installed on the machine) will fetch and install all the PHP dependencies required for the app;
Remark: when building the image for the first time, the image `ubuntu:20.04` will be pulled and lots of PHP dependencies will be installed inside the container. Everything is handled in the Dockerfile and manually installing the app can quickly become very cumbersome. We thus **strongly** recommend following the Docker approach.
- (2) `php artisan key:generate` will set the application key to a random string;
- (3) `php artisan migrate:fresh --seed` will run all the migrations and seed all the models defined in the seeder classes found in the `database/seeder`s

folder;

Remark: The usage of `php artisan` commands will fail when working with containers and interacting with the databases. The commands need to be run from inside the container which is not user-friendly. We can use the following command instead: `docker-compose exec -u sail "${APP_SERVICE:-laravel.test}" php artisan migrate:fresh --seed;`

- (4) `npm run dev` or `npm run prod` depending on the environment to compile the assets;

We also need to configure and set up Rhasspy, specifically download the French acoustic models, define the intents, and train the models. This is done with the following steps:

- (1) Change the default profile with the one defining the different services used (e.g. Kaldi for speech-to-text, Fsticuffs for intent recognition, ...). The profile used for the evaluation can be found in the `rhasspy` directory. Rhasspy API allows us to POST this profile via the `/api/profile?layer=profile` route with the JSON-formatted profile as the request body. By default, Rhasspy exposes port 12101 on `localhost` but this can of course be changed to any preferred port.
- (2) The default language for Rhasspy is English but we specified in the `docker-compose.yml` file that our profile should be in French. The next thing to do is to download the different acoustic models and French-related packages for the speech-to-text tool. This is done via the POST `/api/download-profile` route.
- (3) Train the speech/intent recognizers for the current profile via the POST `/api/train` route.
- (4) Restart Rhasspy via the POST `/api/restart` route.

4.4.3 Rhasspy services

It might be relevant to change some software solutions for specific Rhasspy services. For example, a speech-to-text tool might be more adapted to the user's tone than another one. There are two ways of defining which services Rhasspy should use:

- (1) **Web interface:** Rhasspy comes with a web interface allowing to fully customize the software solutions for each service. Rhasspy will then ask to download optional files and restart itself;

- (2) **API:** As described above, we can make use of the API to specify in the profile file which tool should be used for the different services. Listing 4.4 shows the profile we used for the evaluation;

Listing 4.4: Profile file listing the current services

```
1 {
2   "intent": {
3     "system": "fsticuffs"
4   },
5   "microphone": {
6     "system": "pyaudio"
7   },
8   "sounds": {
9     "system": "aplay"
10  },
11  "speech_to_text": {
12    "system": "kaldi"
13  },
14  "text_to_speech": {
15    "system": "nanotts"
16  }
17 }
```

Chapter 5

Evaluation

To evaluate our system, we have decided to conduct two different kinds of tests. The first one is a voice recognition test where we evaluate each of the three (Kaldi, DeepSpeech, and PocketSphinx) speech-to-text tools available to see their correctness when transcribing speech to text but also their performance. It is important to keep in mind that a nice and fluid conversation comes from a low latency from the assistant. A fast and correct transcription is therefore crucial.

Finally, the more important step is to test this dialogue-based interface on real people, preferably on elderly people, to evaluate its acceptance and relevance, and have some further feedback that will guide further research.

This chapter will first present the experimental protocol for each test along with the relevant documents. We will next present the results along with some explanations and finally further discuss some of those results.

5.1 User tests

Evaluating a voice interface without prior knowledge of an alternative screen-based interface might be tricky. This section will explain how the experiments were conducted and how the participants were able to give their own impression on the usability, relevance, and pleasantness of our voice assistant.

The experiment consisted of eight parts, some being forms to fill, interactions with the assistant, or feedback sessions:

- (1) The participant filled a form with questions regarding their demography, technology usages, and concerns about privacy. Appendix C.1 shows the used form;

- (2) The participant was presented a web page with words or small sentences appearing on the screen. The participant was asked to read the sentences out loud while the microphone was capturing and saving the audio samples;
- (3) The experimenter described the situation where he was a technician installing new devices, such as sensors and monitors, in the participant's home to ease their life and improve their security. He explained how the system would be able to alert the participant's relatives when specific events have been detected.
- (4) The experimenter and the participant used the web application to define several actors with at least one child (if the participant has any) and one neighbor (Figures D.2 and D.3);
- (5) The experimenter then described the web interface for defining new policy rules and asked the participant to create two to three rules on their own (Figures 5.2, D.4, and D.5);
- (6) The participant would then answer four questions described in Figure C.2
- (7) The experimenter would next present to the participant the alternative of using a voice interface in place of the web one he used before. The experimenter would then ask the participant to use this new interface to create at least three new policy rules by answering the questions asked;
- (8) The participant would finally answer the last questions regarding this voice assistant (Figures C.2 and C.3);

5.1.1 Demographic form

The goal behind this form is to get some information about the participant such as their age, profession, and last degree of study, but also about their relationship with technology and with their data.

The first part of the form was regarding the frequency of usage of the following technologies: computer, smartphone, tablet, smart TV. The participant had to give a score from 1 to 7 where 1 means a rarely usage and 7 a frequent one. When evaluating a different kind of technology that they would usually use, such as a voice assistant, it is interesting to know their habits regarding some current technologies that might be widely deployed (computer, smartphone, tablet) or less frequent (smart TV).

```

Speech-to-text system: kaldi
Intent recognition system: fsticuffs
User tested: 5
File tested: AnswerFirefighter.wav - Intent recognized: AnswerFirefighter
File tested: AnswerBoth.wav - Intent recognized: AnswerBoth
File tested: AnswerSon.wav - Intent recognized: AnswerPolice
File tested: AnswerPolice.wav - Intent recognized: AnswerPolice
File tested: AnswerNeighbour.wav - Intent recognized: AnswerNo
File tested: AnswerNo.wav - Intent recognized: AnswerNo
File tested: AnswerCarer.wav - Intent recognized: AnswerCarer
File tested: AnswerDaughter.wav - Intent recognized: AnswerDaughter
File tested: AnswerYes.wav - Intent recognized: AnswerYes
Recognition rate: 7/9 77.777777777779%

```

Figure 5.1: Voice recognition test for user 5

The second part consisted of questions about the participant’s data as a whole. The goal was to know more about the participant’s concerns over their data but also their approach to keep their personal data secured. We used the example of website cookies that they all knew about to help them fill those questions.

5.1.2 Voice recognition

The next task was to collect voice samples of the participant in order to feed them later on to a script analyzing which speech recognition is more accurate. This was done on the web interface where the participant was asked to read sentences appearing on the screen out loud. Those very small sentences were the available answers that would be expected by the assistant when having a dialogue.

The voice samples are sent to the backend which stores them so that the script can select each sample, send it to Rhasspy via its API, process them by a voice recognition service, and then compare it with the expected sentence. The comparison was done by naming the audio file by its expected intent name. For example, the user **donatien** saying *Tous les deux* would have the file named **AnswerBoth.wav** stored in the subfolder **donatien/**. The script would just need to remove the substring *.wav* to compare it with the JSON response (See Listing 4.1) holding the processed intent.

Figure 5.1 gives an example of a real participant that shows that two sentences were not properly recognized by the speech-to-intent engine. For the sentence *mon fils*, Rhasspy recognized *la police* which in fact sounds alike as the end of the two sentences is quite similar. The same phenomenon is observed for the two sentences *non* and *mon voisin* but this time with the beginning of the sentences.

In addition to the recognition rate, we also timed each speech-to-text request to

ID	EVENT	ACTION(S)	
3	Si votre rythme cardiaque augmente anormalement	appeler Raphaëlle envoyer un sms à Bernard	
4	Si vous chutez	appeler Bernard	
5	Si vous n'avez pas reçu de visite depuis longtemps	notifier Raphaëlle	
6	Si votre rythme cardiaque diminue anormalement	envoyer un sms à Donatien	
7	Si votre pression artérielle est anormalement haute	envoyer un sms à Donatien	
8	Si vous prenez beaucoup de poids	notifier Donatien notifier Raphaëlle	

Figure 5.2: Interface for managing the user's policies

get an average transcription time for each of the software. This is one very important factor for the good sake of a nice and fluid conversation with the voice assistant.

Remark: This test was conducted first in order to avoid a break in the flow of the experiment. The voice recognition test was not part of a *scenario* that we presented to the participants and doing it at the very beginning seemed more appropriate and logical.

5.1.3 Screen-based interface

After being presented the roleplay scenario described in step (3), the participant was asked to use the web interface to manage their actors' list. The creation of new actors was a simple task for them to get accommodated to the computer but it was also because they needed at least one actor to be able to create new policy rules. When creating a new actor, they were presented a form where they could select the relationship between them along with some pieces of information such as a name and a phone number (Figure D.3).

The next step was to use the policy manager to create at least two new policies regarding an event that seemed fitting for them. First, they had to select from a panel of events (Figure D.4) the one they want to create a new policy rule for, then

create some action-actor pairs with the available actors they created prior to this (Figure D.5). Figure 5.2 shows how the participant has been presented their policy rules.

Evaluation

We then presented a very small questionnaire with only four questions to the participant. Those four yes-or-no questions were carefully chosen to get a global appreciation of the service they just used. We did not waste too much experiment time on this questionnaire as the web interface is not the final goal of this thesis, yet we needed some idea of the participant's thoughts on its usability and relevance.

The four questions were actually presented as affirmations such as *"I think that this interface is easy to use."* with two of them being positive aspects and the other two negative ones. Thus getting a *No* answer to a negative one is actually a good thing. By taking the amount of *Yes* answers for positive affirmations and *No* answers to negatives ones, we get a score from 0 to 4 indicating roughly the usability of this interface.

5.1.4 Dialogue-based interface

During this phase, the participant was presented with a simple interface where they could launch the voice assistant that would generate some questions in accordance with their current events and actors.

After the interaction with the voice assistant, the participant was presented with two forms.

SUS form

The SUS (System Usability Scale) form is a questionnaire of 10 affirmations allowing the participants to rate the usability of a service, in this case, the voice assistant and dialogue system. The 10 items present in the form consist of 5 positive and 5 negative aspects about the service such as its relevance, its usability, its ease of use, or its friendliness. Each item has a response range from 1 (strongly disagree) to 5 (strongly agree) which can be used with a small formula to get a score of 100.

The goal of this form is to prevent asking the participant only positive aspects about the service and be more critical about a wide range of characteristics. For example, if we ask the question *Was this service easy to use?*, we should also ask the opposite question *Would you need the help of a technician?* to ensure a well-thought answer regarding the ease-of-use. Additionally, items 4 and 10 do not

question the usability but the learnability which is something valuable in the case of elderly people. This data can thus be further studied on its own.

Once the questionnaire has been filled, we can use the following formula to get a score:

- (1) For each odd item (1,3,5,7,9), subtract 1 to the score. e.g. if the participant answered 5 (Strongly Agree) to the first question, we get $5 - 1 = 4$
- (2) For each even items (2,4,6,8,10), subtract the score to 5. e.g. if the score for question 2 is 2 (Disagree), we get $(5 - 2) = 3$
- (3) Sum up each score of the 10 questions
- (4) Multiply by 2.5 to get a score of 100

The final goal of a SUS form is not to give a *percentage of usability* of the service as it would be incorrect. The score can however be interpreted to get an appreciation of the service's usability. Aaron Bangor et. al [16] studied the meaning of the SUS score by asking an 11th question regarding the overall user-friendliness of the product on a scale from 1 (worst imaginable) to 7 (best imaginable). The results they got from a sample of 1000 SUS surveys allowed them to give an SUS survey a rank based on its score. Above 50.9 is considered OK, above 71.4 is Good, and above 85.5 is Excellent.

It is also important to question the reliability of using such a questionnaire with a small sample size such as ours. It has been shown that there is no correlation between the SUS score and the sample size so we should not worry about having only five participants.

Feedback form

The last part of the experiment was for the participant to give feedback on the dialogue system they just used. More precisely, they had to give one to three positive aspects about the assistant, such as the usability, the ease-of-use, the friendliness, ..., and one to three negative aspects about it. They could additionally put out remarks if they had any.

5.2 Results

The experiment was conducted on 5 participants from age 59 to 88 listed in Table 5.1. The experiments were done individually to prevent any interaction between the participants that could create some bias.

Participant	Age	Occupation	Activity sector	Studies
1	59	Consultant	Marketing	Computer sciences
2	88	Retired	Industrial	Industrial engineer
3	81	Retired	/	Psychology
4	68	Retired	Finance	Secondary
5	64	Retired	Administration	Secondary

Table 5.1: List of participants

Participant	F1	F2	F3	F4	Q1	Q2	Q3	Q4	Q5	Q6	Q7
1	7	7	7	2	Yes	Yes	No	No	Yes	Yes	3
2	5	2	1	3	Yes	Yes	No	Yes	Yes	Yes	7
3	7	7	7	1	Yes	No	No	Yes	Yes	Yes	1
4	7	7	1	1	Yes	Yes	No	Yes	Yes	Yes	7
5	7	7	2	1	Yes	Yes	No	Yes	Yes	Yes	7

Table 5.2: Participants' answers to the demographic form

5.2.1 Demographic form

Due to the Covid crisis, it was a bit complicated to perform the experiments on elderly people and on a lot of subjects. We nonetheless managed to perform the experiment on five adults having various profiles, four of them being retired. Table 5.2 shows the participants along with their answers to the following questions:

- **F1:** I use a computer frequently (1 Rarely - 7 Frequently)
- **F2:** I use a smartphone frequently (1 Rarely - 7 Frequently)
- **F3:** I use a tablet frequently (1 Rarely - 7 Frequently)
- **F4:** I use a smart TV frequently (1 Rarely - 7 Frequently)
- **Q1:** I am aware of the data collected by some applications (True - False)
- **Q2:** I have already updated data permissions for some applications (True - False)
- **Q3:** I always accept permissions asked by some applications (True - False)
- **Q4:** I always deny permissions asked by some applications (True - False)
- **Q5:** I think I have full control over my personal data (True - False)
- **Q6:** I would like more control over my personal data (True - False)

Participant	Kaldi		DeepSpeech		PocketSphinx	
	Rec rate (%)	Avg time (ms)	Rec rate (%)	Avg time (ms)	Rec rate (%)	Avg time (ms)
1	100	316	44.44	1382	55.55	1591
2	88.88	325	33.33	2272	11.11	1121
3	88.88	327	33.33	2260	0	1450
4	88.88	307	55.55	1387	44.44	813
5	77.77	308	33.33	1009	0	538

Table 5.3: Recognition rate and average time for the three speech-to-text tools

- **Q7:** I often worry about my data being collected by some applications (1 Rarely - 7 Frequently)

Overall, computers and smartphones (except for participant 2) are technologies that they are used to contrary to the smart TV which is way less frequently used. For the tablet, it is interesting that the participants either use them a lot or don't use them at all. We can see that 4 out of 5 participants always deny permissions requested by some application without taking a look at them and that only participant 1 takes the time to go through all of them without denying everything by default.

An interesting result is that all participants felt that they have control over their data but still would like more control which sounds a bit paradoxical. Wanting more control means that they probably don't have full control, or maybe they fear that in the future they will not have that same amount of control as they do now.

The results of the last question show that there is again a big gap between some participants frequently worrying about their data while some barely worry about it.

5.2.2 Voice recognition

Table 5.3 shows the results of the voice recognition rate and the average time for a speech-to-text transcription on the three available software, Kaldi, Mozilla DeepSpeech, and PocketSphinx.

Regarding the voice recognition rate, we can see that Kaldi shows the best results with the smallest value being 77.77% of voice commands being recognized. It is very interesting to notice that for participants 2, 3, and 4, the sentence *Oui* was recognized as *Non* which can be a catastrophic issue as they mean the exact

Participant	Q1	Q2	Q3	Q4	Score
1	Yes	No	Yes	No	4
2	Yes	No	Yes	No	4
3	Yes	No	Yes	No	4
4	Yes	No	Yes	No	4
5	Yes	No	Yes	No	4

Table 5.4: Caption

opposite. For participant 5, the answers *la police* and *mon fils* as well as *mon voisin* and *voisin* were mistaken.

The other two tools showed worse results as bad as a 0% recognition rate for PocketSphinx.

When it comes to the average time for the transcription of the voice command to an intent, we can see that Kaldi holds first place again by a large margin with an average of one-third of a second per command. The differences between the different participants come from the length of the audio samples which might slightly differ from one participant to another. As an example, we can see that participants 4 and 5 average time is lower for each software than the other four. Probably because the silences at the beginning and end of the sentences were shorter.

5.2.3 Screen-based interface

For the evaluation of the web interface, Table 5.4 shows the results of the following questions:

- **Q1:** I find this web interface easy to use (True - False)
- **Q2:** I think I would need the assistance of a technician to use this web interface (True - False)
- **Q3:** I was feeling confident using this web interface (True - False)
- **Q4:** I found this web interface cumbersome to use (True - False)

We can see that all participants gave the same answers for this part, finding the web interface easy to use. The experimenter was there to guide them for the first policy creation but left them on their own for the second policy creation which was naturally easier with prior knowledge. This could explain why every participants gave the same maximum score for the web interface.

A 1 to 5 scale would have probably given different, and more elaborated, results but the goal here was not to give the web interface a grade, rather to get an impression.

5.2.4 Dialogue-based interface

Before taking a look at the results of the two forms we presented earlier, we will show what a typical interaction with the voice assistant looked like. The following dialogue was recorded on a real participant and lead to the creation of two new rules.

Assistant: Bonjour. Êtes-vous disposé à répondre à quelques questions?

User: oui

A: Très bien. Voici la question. Si votre pression artérielle est anormalement haute, dois-je envoyer un sms à votre voisin ou votre fils?

U: non

A: Voulez-vous que j'effectue une autre action?

U: non

A: Voulez-vous que je prévienne une autre personne?

U: oui

A: Voulez-vous que je prévienne votre fille ou votre aide-soignant?

U: mon aide-soignant

A: Très bien, je vais envoyer un sms à votre aide-soignant si votre pression artérielle est anormalement haute. Pouvez-vous confirmer? Dites oui ou non.

U: oui

A: Votre nouvelle règle de sécurité a bien été enregistrée. Voulez-vous en créer une autre?

U: oui

A: Très bien. Voici la question.

A: Dois-je notifier votre fils ou votre fille si vous perdez beaucoup de poids?

U: oui

A: Voulez-vous que je prévienne votre fils ou votre fille ou les deux?

U: les deux

A: Très bien, je vais notifier votre fils et votre fille si vous perdez beaucoup de poids. Pouvez-vous confirmer? Dites oui ou non.

U: oui

A: Votre nouvelle règle de sécurité a bien été enregistrée. Voulez-vous en créer une autre?

U: nan

A: Merci pour votre aide. Je vais prévenir vos proches des changements que vous avez effectué.

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS score
1	4	1	5	2	3	2	4	1	4	1	82,5
2	4	1	5	1	4	1	4	1	5	1	92,5
3	4	1	4	1	5	1	5	1	5	3	90
4	5	1	5	2	5	1	4	1	5	2	92,5
5	5	2	4	2	4	2	5	1	5	2	85

Table 5.5: SUS form results

SUS results

The SUS form gave us some really nice results on the usability and learnability of our voice assistant. As we can see from Table 5.5, the score of each participant is greater than 80, which is following the grading scale of Aaron Bangor et. al. excellent for almost every participant. The questions for the SUS form were the following:

- **Q1:** I think that I would like to use this system frequently.
- **Q2:** I found the system unnecessarily complex.
- **Q3:** I thought the system was easy to use.
- **Q4:** I think that I would need the support of a technical person to be able to use this system.
- **Q5:** I found the various functions in this system were well integrated.
- **Q6:** I thought there was too much inconsistency in this system.
- **Q7:** I would imagine that most people would learn to use this system very quickly.
- **Q8:** I found the system very cumbersome to use.
- **Q9:** I felt very confident using the system.
- **Q10:** I needed to learn a lot of things before I could get going with this system.

The two participants that interest us the most are the two oldest, number 2 and 3, as they are the typical target for such a system. The SUS score we get from them is higher than 90 which is actually a very impressive and encouraging score.

We can however see, for participant 3, a score of 3 for **Q10** which indicates that the system was not so easy to learn to use. After a discussion with participant 3, It became clear that the issue was not regarding the voice assistant but rather the way the voice was recorded. Indeed, to ensure that the voice recognition tool would be efficient, we prevented any noises or silences in the audio by asking the participants to use a **RECORD** button that they had to press once to start the recording and press again to stop the recording and send the audio file to Rhasspy. This was an issue for older people and specifically for participant 3.

We can furthermore analyze the answers of some specific questions such as **Q2** which show that 4 out of the 5 participants strongly disagreed in finding the system unnecessarily complex which is valuable information in the validation of a simple-to-use voice assistant.

We can also see that **Q8** presents a score of 1 for every participant. This is a huge indication of the ease of use as well as in the learnability of our system. All the participants strongly disagreed in finding the voice assistant cumbersome which is an important factor specifically for elderly people. They might be less comfortable with technology by default and adding some more weight on their shoulder is something we really wanted to avoid.

Another encouraging aspect is the results for **Q9** showing that 4 out of the 5 participants felt confident using the system. This is also something we value as very important in the context of defining security rules.

Feedback

In addition to the SUS form, the participants were to give feedback, positive and negative, on the voice assistant they just used. They would mostly describe it with adjectives such as *simple*, *relevant*, *easy*, ...

The positive aspects that were most emphasized are the following:

- (1) **Simplicity**: the positive aspect that was common for most of the participants is how simple it is to interact with the assistant. While performing the experiment, it was also really encouraging to see that they had no need for help from the experimenter and that it was a nice and fluid conversation with the assistant. Some deadlocks occurred where the assistant didn't know how to interpret the participant's answer but the participants restarted the assistant without flinching.
- (2) **Relevance**: another positive aspect that really encouraged us was the fact that most users found this tool relevant, either for their parents but also for

them when they would get older.

- (3) **Confidence:** the participants felt more confident while performing policy management with the voice assistant than with the web interface. The fact that there is a voice instead of a screen is actually a huge factor for some participant that felt that this assistant was "*more human and the voice was better than a robotic interface*(web interface)".

There are of course some negative aspects that were given by the participants during this feedback session. Most of the comments from the participants were regarding the integration of the assistant rather than the dialogue system, however. The few comments that came back a lot were the following:

- (1) **Voice recognition:** it happened for some participants that the answer *Oui* was poorly recognized resulting in frustration from them. The assistant would simply exit after the first question because it would understand that the user does not want to create some new policies. For this specific issue, the experimenter, whose voice tone was very well recognized, took the role of the speaker and said the participant's answers out loud for them.
- (2) **Robotic voice:** for the evaluation, we used the Firefox text-to-speech tool that felt very robotic for some sentences. Some participants didn't feel confident with that and said that they would for example prefer a calmer voice, and preferably a female one.
- (3) **Missing actors:** another negative aspect about the assistant was that the choice of an actor was very restrictive and was missing some key actors such as the doctor. The absence of the doctor as an available was a choice we made based on some testimony of elderly people feeling that notifying their doctor would just result in inconveniences. The fact that some people would actually like to have the doctor as an actor shows that more freedom should be given to the user when defining the different actors

5.3 Further discussion

Regarding the recognition rate of the different speech-to-text tools, we should note that it is not the most effective way of testing the accuracy of each and every one of them. Indeed, a more correct approach would have been to sample each sentence several times and then check then transcribe them. By doing this, we can see how well a speech-to-text tool performs on the same sentence with some small voice disparities. Getting a 90% recognition rate on a set of 100 samples of the same sentence would give more insight than getting 90% of the voice commands being

recognized. To give you an example, we noticed later on that there were some background noises for participant 5 which might indicate why the recognition rate was lower.

Nonetheless, because the same samples were used on the different speech-to-text tools, we can identify **Kaldi** as being the most reliable and efficient of the ones available **for this specific use case**. We are not taking any assumptions about how well the other tools perform in a different context but only judge them in our own scope.

The results regarding the web interface need to be taken with a grain of salt as the form the participants filled was incomplete and does not give empirical results. We can however note that, as a whole, the web interface is something that might be used in the context of setting up the assistant with the definition of actors and the first policies. We believe that the voice assistant should be the main method for managing security rules but that a web interface should still be available for some specific scenarios such as a relative checking that the policies are in fact coherent. The two interfaces should thus be complementary.

Chapter 6

Conclusion

This thesis aimed to design, develop, and evaluate an alternative way of managing access control policies for the elderly in the form of a voice assistant. The tool we propose has been carefully designed and implemented to allow its usage on real hubs by following the policy format definition of SmartThings. By testing this prototype on real users that might be directly concerned with this topic, or soon will be, we have shown that such a system is a well-accepted, relevant, and more humane alternative to the more common web interfaces.

The evaluation we made suffered a lack of participants that do not give empirical results and thus does not allow a generalization of the acceptability of our system for all elderly people. But it does provide some positive and encouraging insights that lead to believe that voice assistant is a preferred solution in this context.

Further research needs to be done regarding the technologies used for speech-to-text, text-to-speech, intent recognition, and intent recognition to determine which ones are most suited for interactions with elderly people.

Based on the results from our evaluation, we can show that managing security policy rules with the help of a voice assistant is something that can be done in an easy and comprehensive manner by elderly people. This solution can therefore be further studied, specifically regarding the heuristics for creating new questions and monitoring existing policies. We have given some examples of heuristics that we found relevant but a more scientifically studied approach is needed.

Bibliography

- [1] Pardis Emami Naeini; Sruti Bhagavatula; Hana Habib; Martin Degeling; Lujó Bauer; Lorrie Cranor; and Norman Sadeh. *Privacy Expectations and Preferences in an IoT World*. 2017.
- [2] Alisa Frik ; Leysan Nurgalieva ; Julia Bernd ; Joyce Lee ; Florian Schaub; Serge Egelman. *Privacy and Security Threat Models and Mitigation Strategies of Older Adults*. 2019.
- [3] Paola Mello; Sabato Mellone; Giuseppe Mincoelli; Carla Raffaelli; Luca Roffia; Tullio Salmon Cinotti; Carlo Tacconi; Paola Tamburini; Marco Zoli Elena Borelli; Giacomo Paolini; Francesco Antoniazzi; Marina Barbiroli; Francesca Benassi; Federico Chesani; Lorenzo Chiari; Massimiliano Fantini; Franco Fuschini; Andrea Galassi; Gian Andrea Giacobone; Silvia Imbesi; Melissa Licciardello; Daniela Loreti; Michele Marchi; Diego Masotti and Alessandra Costanzo. *HABITAT: An IoT Solution for Independent Elderly*.
- [4] *Evolution de la part des 65 ans et plus dans la population (en pourcentage)*. URL: <https://www.iweeps.be/indicateur-statistique/population-des-65-ans-et/>.
- [5] Serena Zheng; Noah Apthorpe; Marshini Chetty; Nick Feamster. *User Perceptions of Smart Home IoT Privacy*. 2018.
- [6] Norbert Nthala; Ivan Flechais. *Informal Support Networks: an investigation into Home Data Security Practices*. 2018.
- [7] AlbertDarshan Chandrashekhar Khedekar; Amelia Carrera Truco; Diego A. Oteyza; Guillermo Florez Huertas. “Home Automation—A Fast - Expanding Market”. In: *Thunderbird, International Business Review* 59.10 (2016), pp. 79–91. DOI: 10.1002/tie.21829.
- [8] Andreas Jacobsson and Paul Davidsson. *Towards a model of privacy and security for smart homes*. 2015. DOI: 10.1109/WF-IoT.2015.7389144.

- [9] Hezam Akram Abdulghani; Niels Alexander Nijdam; Anastasija Collen; Dimitri Konstantas. *A Study on Security and Privacy Guidelines, Countermeasures, Threats: IoT Data at Rest Perspective*. 2019. DOI: [10.3390/sym11060774](https://doi.org/10.3390/sym11060774).
- [10] Juan Manuel Lucas-Cuesta et al. *Dialogue-based Management of user Feedback in an Autonomous Preference Learning System*. 2010. URL: https://www.researchgate.net/publication/221539775_Dialogue-based_Management_of_user_Feedback_in_an_Autonomous_Preference_Learning_System.
- [11] Haotian Chi; Qiang Zeng; Xiaojiang Du; Lannan Luo. *PFirewall: Semantics-Aware Customizable Data Flow Control for Home Automation Systems*. 2021. DOI: <https://dx.doi.org/10.14722/ndss.2021.24464>.
- [12] Eric Zeng; Shrirang Mare; and Franziska Roesner. *End User Security Privacy Concerns with Smart Homes*. 2017.
- [13] Z. Berkay Celik; Patrick McDaniel; and Gang Tan. *Soteria: Automated IoT Safety and Security Analysis*. 2018. DOI: <https://www.usenix.org/system/files/conference/atc18/atc18-celik.pdf>.
- [14] François Portet ; Michel Vacher ; Caroline Golanski ; Camille Roux • Brigitte Meillon. *Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects*.
- [15] Melinda Hamill; Vicky Young; Jennifer Boger; Alex Mihailidis. *Development of an automated speech recognition interface for personal emergency response systems*. 2009.
- [16] Aaron Bango; Philip Kortum; James Miller. “Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale”. In: *Journal of usability studies* Vol 4.3 (2009), pp. 114–123. DOI: [10.1.1.177.1240](https://doi.org/10.1.1.177.1240).
- [17] Kristel De Vlieghe; Bert Aertgeerts; Anja Declercq; Koen Milisen; Walter Sermeus; Philip Moons. *Shifting care from hospital to home: a qualitative study*. 2015. DOI: [10.7748/phc.25.9.26.s28](https://doi.org/10.7748/phc.25.9.26.s28).
- [18] Jeyanthi Nagamalai. *Internet of things (IoT) security: Current status, challenges and prospective measures*. 2016. DOI: [10.1109/ICITST.2015.7412116](https://doi.org/10.1109/ICITST.2015.7412116).
- [19] Z. Berkay Celik; Gang Tan; and Patrick McDaniel. *IOTGUARD: Dynamic Enforcement of Security and Safety Policy in Commodity IoT*. 2019. DOI: <https://dx.doi.org/10.14722/ndss.2019.23326>.

- [20] Luca Caviglione; Jean-François Lalande; Wojciech Mazurczyk; Wojciech Mazurczyk; Steffen Wendzel. *Analysis of Human Awareness of Security and Privacy Threats in Smart Environments*. 2015. DOI: 10.1007/978-3-319-20376-8_15.
- [21] Alma Whitten and J. D. Tygar. *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0*. 1999. URL: <https://www.usenix.org/conference/8th-usenix-security-symposium/why-johnny-cant-encrypt-usability-evaluation-pgp-50>.
- [22] *Worldwide Smart Home Devices market*. URL: <https://www.businesswire.com/news/home/20190923005071/en/Worldwide-Smart-Home-Devices-Forecast-to-Maintain-Steady-Growth-Through-2023-Says-IDC>.
- [23] Haotian Chi; Qiang Zeng; Xiaojiang Du; Jiaping Yu. *Cross-App Interference Threats in Smart Homes: Categorization, Detection and Handling*. 2018. DOI: <https://arxiv.org/pdf/1808.02125.pdf>.

Appendices

Appendix A

Classification

Listing A.1 and Table A.1 show the events that we defined in our system. The Listing represents them in our policy format while the Table shows the different properties we gave to the events for the sake of classification.

Listing A.1: Events listing presented in our policy format

```
1  TRIGGER: {
2      match (device).(monitoring).(heart-rate)
3      satisfy (<)->(60)
4  }
5
6  TRIGGER: {
7      match (device).(monitoring).(heart-rate)
8      satisfy (>)->(130)
9  }
10
11 TRIGGER: {
12     match (device).(monitoring).(bloodpressure)
13     satisfy (==)->(high)
14 }
15
16 TRIGGER: {
17     match (device).(monitoring).(bloodpressure)
18     satisfy (==)->(low)
19 }
20
21 TRIGGER: {
22     match (device).(monitoring).(weight)
23     satisfy (==)->(strong increase)
24 }
25
```

```

26     TRIGGER: {
27         match (device).(monitoring).(weight)
28         satisfy (==)->(strong decrease)
29     }
30
31     TRIGGER: {
32         match (device).(monitoring).(sleep)
33         satisfy (==)->(anomalous)
34     }
35
36     TRIGGER: {
37         match (device).(motion).(event)
38         satisfy (==)->(fall)
39     }
40
41     TRIGGER: {
42         match (device).(motion).(event)
43         satisfy (==)->(inactive)
44     }
45
46     TRIGGER: {
47         match (device).(agenda).(visits_last_month)
48         satisfy (<)->(2)
49     }
50
51     TRIGGER: {
52         match (device).(medication_sensor).(last_taken)
53         satisfy (>)->(2)
54     }
55

```

Name	Device	Attribute	Operator	Value	Mental health	Physical health	Reactivity	Privacy
Heartbeat increase	monitoring	heartbeat	>	130	0	3	2	3
Heartbeat decrease	monitoring	heartbeat	<	60	0	3	2	3
High bloodpressure	monitoring	bloodpressure	==	high	0	3	2	3
Weight increase	monitoring	weight	==	increase	2	3	1	3
Weight decrease	monitoring	weight	==	decrease	2	3	1	3
Anomalous sleep	monitoring	sleep	==	anomalous	1	1	1	3
Fall	motion	event	==	fall	0	3	3	0
Inactivity	motion	event	==	inactive	1	1	1	0
Not visited	agenda	visit_last_month	<	2	2	0	1	0
Medication not taken	medication sensor	last_taken	>	2	2	1	1	2

Table A.1: Events listing

Appendix B

Implementation

Listing B.1 shows the docker-compose file used to deploy our system.

Remark: The Docker image for mysql is only available for x86 and x86-64 processors and is therefore not compatible with Raspberry ARM processors. We thus suggest using the alternate `rpi-mysql` Docker image¹ instead.

Listing B.1: docker-compose.yml file

```
1 version: '3'
2 services:
3     laravel.test:
4         build:
5             context: ./docker/8.0
6             dockerfile: Dockerfile
7             args:
8                 WWWGROUP: '${WWWGROUP}'
9         image: sail-8.0/app
10        ports:
11            - '${APP_PORT:-80}:80'
12        environment:
13            WWWUSER: '${WWWUSER}'
14            LARAVEL_SAIL: 1
15        volumes:
16            - './var/www/html:'
17        networks:
18            - sail
19        depends_on:
20            - mysql
21    mysql:
22        image: 'mysql:8.0'
```

¹<https://hub.docker.com/r/hypriot/rpi-mysql/>

```

23     ports:
24         - '${FORWARD_DB_PORT:-3306}:3306'
25     environment:
26         MYSQL_ROOT_PASSWORD: '${DB_PASSWORD}'
27         MYSQL_DATABASE: '${DB_DATABASE}'
28         MYSQL_USER: '${DB_USERNAME}'
29         MYSQL_PASSWORD: '${DB_PASSWORD}'
30         MYSQL_ALLOW_EMPTY_PASSWORD: 'yes'
31     volumes:
32         - 'sailmysql:/var/lib/mysql'
33     networks:
34         - sail
35     healthcheck:
36         test: ["CMD", "mysqladmin", "ping"]
37 redis:
38     image: 'redis:alpine'
39     ports:
40         - '${FORWARD_REDIS_PORT:-6379}:6379'
41     volumes:
42         - 'sailredis:/data'
43     networks:
44         - sail
45     healthcheck:
46         test: [ "CMD", "redis-cli", "ping" ]
47 rhasspy:
48     image: "rhasspy/rhasspy"
49     container_name: rhasspy
50     ports:
51         - "12101:12101"
52         - "12183:12183"
53     volumes:
54         - "$HOME/.config/rhasspy/profiles:/profiles"
55         - "/etc/localtime:/etc/localtime:ro"
56
57     - networks:
58         - sail
59     command: —user—profiles /profiles —profile fr
60 networks:
61     sail:
62         driver: bridge
63 volumes:
64     sailmysql:
65         driver: local

```

```
66     sailredis:  
67         driver: local  
68
```

Appendix C

Evaluation

QUESTIONNAIRE POUR PARTICIPER A L'EXPERIENCE

Titre de l'étude: Voice assistant as interfaces for social home care security: Acceptability by the elderly
Professeur: Prof. Etienne Rivière, UCLouvain
Expérimentateurs: Donatien Schmitz
Email de contact: etienne.riviere@uclouvain.be, donatien.schmitz@student.uclouvain.be

Données personnelles (qui seront anonymisées)

Code anonyme: (à attribuer par l'expérimentateur)
Prénom: (anonymisés dans la base de données)
Nom de famille:
Genre: ☐ Femme ☐ Homme
Age: (années)
Profession: ☐ Etudiant ☐ Cadre ☐ Employé
☐ Indépendant ☐ Pensionné ☐ Chômeur
☐ Autre: (préciser s.v.p.)
Domaine d'activité: (p. ex., administration, enseignement)
Etudes: (domaine de votre dernier diplôme)

J'utilise un ordinateur fréquemment: Pas d'accord 1 2 3 4 5 6 7 D'accord
J'utilise un smartphone fréquemment: Pas d'accord 1 2 3 4 5 6 7 D'accord
J'utilise une tablette fréquemment: Pas d'accord 1 2 3 4 5 6 7 D'accord
J'utilise une tv connectée fréquemment: Pas d'accord 1 2 3 4 5 6 7 D'accord

Données expérimentales

Je suis conscient des données collectées par certaines applications: ☐ Oui ☐ Non
J'ai déjà modifié les permissions de collecte de données par certaines applications: ☐ Oui ☐ Non
J'accepte systématiquement les permissions demandées par les applications: ☐ Oui ☐ Non
Je refuse systématiquement les permissions demandées par les applications: ☐ Oui ☐ Non
Je pense être en contrôle de mes données personnelles: ☐ Oui ☐ Non
J'aimerais avoir plus de contrôle sur mes données personnelles: ☐ Oui ☐ Non
Je m'inquiète souvent des données collectées par certaines applications:
Rarement 1 2 3 4 5 6 7 Fréquemment

Figure C.1: Demographic form
75

Questionnaire

Partie 1

Après l'utilisation de l'interface web, veuillez compléter les questions ci-dessous.

1. Je pense que cette interface est facile à utiliser: ☐ Oui ☐ Non
2. Je pense que j'aurai besoin de l'aide d'un technicien pour utiliser cette interface: ☐ Oui ☐ Non
3. Je me sentais très en confiance en utilisant cette interface: ☐ Oui ☐ Non
4. J'ai trouvé cette interface très lourde à utiliser: ☐ Oui ☐ Non

Partie 2

Critères favorables à l'expérience utilisateur classé par importance (1 étant l'aspect le plus agréable) :

- 1)
- 2)
- 3)

Critères non-favorables à l'expérience utilisateur par importance (1 étant l'aspect le plus désagréable):

- 1)
- 2)
- 3)

Remarques :

.....

.....

.....

Figure C.2: questionnaire for part 6 and 8 of the experiment protocol

	Pas du tout d'accord				Tout à fait d'accord
1. Je pense que je vais utiliser ce service fréquemment					
	1	2	3	4	5
2. Je trouve ce service inutilement complexe					
	1	2	3	4	5
3. Je pense que ce service est facile à utiliser					
	1	2	3	4	5
4. Je pense que j'aurai besoin de l'aide d'un technicien pour être capable d'utiliser ce service					
	1	2	3	4	5
5. J'ai trouvé que les différentes fonctions de ce service ont été bien intégrées					
	1	2	3	4	5
6. Je pense qu'il y a trop d'incohérence dans ce service.					
	1	2	3	4	5
7. J'imagine que la plupart des gens serait capable d'apprendre à utiliser ce services très rapidement.					
	1	2	3	4	5
8. J'ai trouvé ce service très lourd à utiliser.					
	1	2	3	4	5
9. Je me sentais très en confiance en utilisant ce service.					
	1	2	3	4	5
10. J'ai besoin d'apprendre beaucoup de choses avant de pouvoir utiliser ce service.					
	1	2	3	4	5

Figure C.3: SUS questionnaire for part 8 of experiment protocol

Appendix D

Web interface

ID	EVENT	ACTION(S)	
3	Si votre rythme cardiaque augmente anormalement	appeler Raphaëlle envoyer un sms à Bernard	
4	Si vous chutez	appeler Bernard	
5	Si vous n'avez pas reçu de visite depuis longtemps	notifier Raphaëlle	
6	Si votre rythme cardiaque diminue anormalement	envoyer un sms à Donatien	
7	Si votre pression artérielle est anormalement haute	envoyer un sms à Donatien	
8	Si vous prenez beaucoup de poids	notifier Donatien notifier Raphaëlle	

Figure D.1: Interface for managing the user's policies

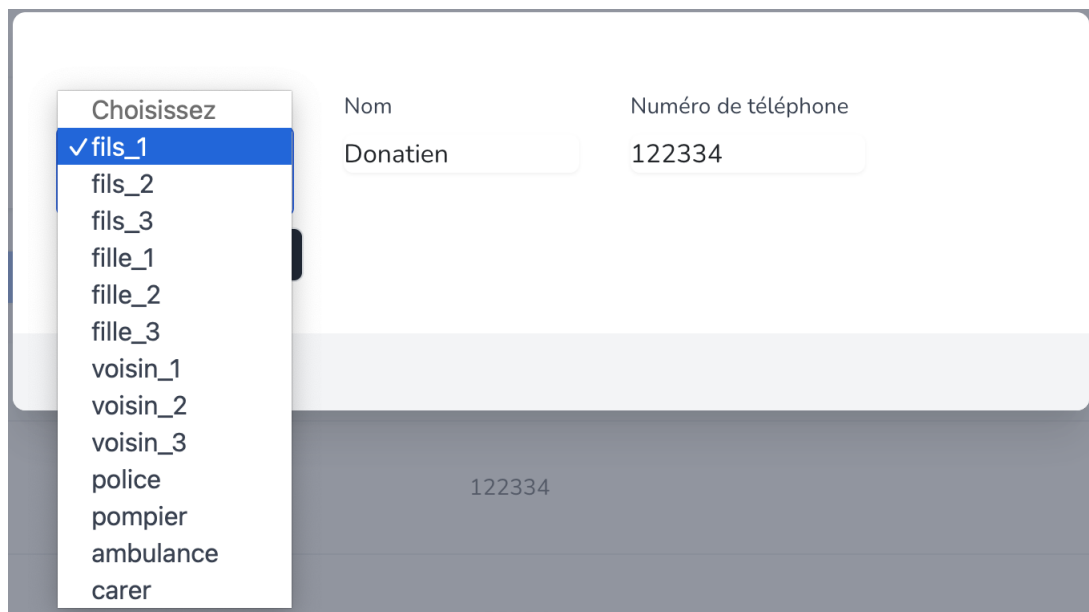
This appendix will present different screenshots of the web interface we created for managing policies. The dashboard we created can be seen in Figure D.1 where the defined policies along with their actions can be updated or deleted.

Figure D.2 shows the list of actors for the current user along with the ability to update or delete them. As explained before, if an actor is deleted, it is in fact *soft-deleted* meaning that the model is still present in the database but will automatically not be fetched by Eloquent (unless specified with the `withTrashed()` method).

TYPE	NOM	NUMÉRO DE TÉLÉPHONE	
votre fils	Donatien	122334	
votre fille	Raphaëlle	122334	
votre voisin	Bernard	567689	

Figure D.2: Interface for managing the user's actors

Figure D.3 shows the form for creating an actor. The user has to pick from one of the available relationships, then fill in some information fields including the phone number that will be used for a call or a message.

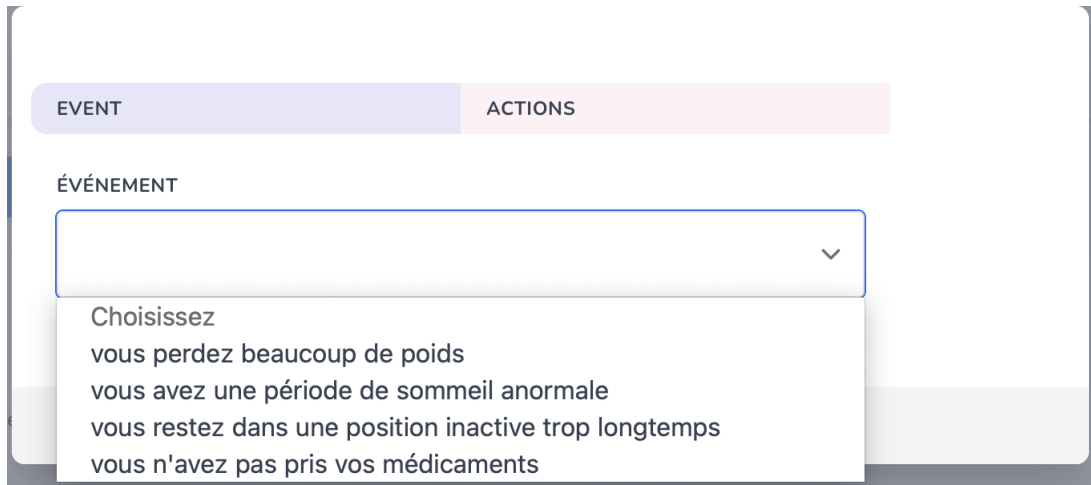


The form for creating a new actor consists of a dropdown menu on the left and two input fields on the right. The dropdown menu is titled 'Choisissez' and lists various relationships: 'fils_1' (selected with a checkmark), 'fils_2', 'fils_3', 'fille_1', 'fille_2', 'fille_3', 'voisin_1', 'voisin_2', 'voisin_3', 'police', 'pompier', 'ambulance', and 'carer'. The two input fields are labeled 'Nom' and 'Numéro de téléphone'. The 'Nom' field contains the text 'Donatien' and the 'Numéro de téléphone' field contains the text '122334'.

Figure D.3: Form to create a new actor

When the user has some actors defined, he will be able to create some policies with the two-tabs form seen on Figures D.4 and D.5. The first Figure shows how the events were presented in a formal language to reduce confusion. In the next Figure, we can see that the user can create some action-actor pairs and add them with the + button on the right. The already created action-actor pairs are shown

above the **Précédent** button. We can also see that if the user wishes, he can still create some more actors with the blue button below the event sentence.



The image shows a web form for creating a new policy. At the top, there are two tabs: 'EVENT' (highlighted in light purple) and 'ACTIONS' (highlighted in light pink). Below the 'EVENT' tab, the label 'ÉVÉNEMENT' is displayed. Underneath, there is a dropdown menu with a blue border and a downward arrow icon. The dropdown is open, showing a list of four options in French: 'Choisissez', 'vous perdez beaucoup de poids', 'vous avez une période de sommeil anormale', 'vous restez dans une position inactive trop longtemps', and 'vous n'avez pas pris vos médicaments'.

Figure D.4: Form to create a new policy (event tab)

EVENT

ACTIONS

Si vous n'avez pas pris vos médicaments

AJOUTER NOUVEL ACTEUR

ACTION

envoyer un sms à ▾

ACTIONS

appeler Donatien

envoyer un sms à Raphaëlle

ACTEUR(S)

▾

Choisissez

Donatien (fils_1)

Raphaëlle (fille_1)

Bernard (voisin_1)

+

PRÉCÉDENT

SAUVEGARDER

Figure D.5: Form to create a new policy (action-actor pairs tab)

UNIVERSITÉ CATHOLIQUE DE LOUVAIN

École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl