

Design a platform for providing home-based services concerning: care, safety, comfort and fun

MASTER THESIS

Diego Vettorel

diegovettorel@gmail.com | +31 619988253

Confidential version – confidential until the 17/07/2017

Master of Science in Business Information Technology
School of Management and Governance
University of Twente
Enschede, The Netherlands
17 July 2015

Graduation Committee:

Dr. Ir. E.J.A. Folmer, University of Twente
Dr. Ir. Marten J. van Sinderen, University of Twente
Drs Paul Francissen, Envolve owner, thesis and internship supervisor and HuisKluis concept's creator

HuisKluis concept's creators

Paul Francissen - paul.francissen@envolve.nl - +31610714772
Rob Van Dort - rob.van.dort@gmail.com - +31629525229
John Van Echtelt - john@van-echtelt.nl - +31610850096

UNIVERSITY OF TWENTE.



Preface

Everything begun in October 2014. Rob Van Dort explained to me, during the University of Twente Hackathon, the idea and concept of HuisKluis (the foundation that provided the business case for this thesis). I was fascinated by the idea and we decided to meet with Paul Francissen to discuss it. When we met, Paul and Rob told me their idea to make a platform to import, see, use and share data of end-users about their houses. I was enthusiastic about the idea and we decided that it would be a great project for my master thesis. This project took the form of an internship of six months and was financed by the Envolve company together with Foundation HuisKluis. Envolve was one of the initiator of the HuisKluis foundation. I worked with them for six months to make this thesis that summarizes all their ideas and concepts and includes my analysis as well.

Paul, Rob and John Van Echtelt had this beautiful idea and I personally think that it is a great idea that will become a successful platform. During my internship they were always willing to give me help, information, support and feedback.

This thesis is the summary of all the ideas, concepts, use cases, interface design and end-user's analysis that they had, and adds my analysis of and designs for those concepts. Every time that "the platform" is mentioned, I am referring to the HuisKluis platform. This thesis contains all the ideas that refer to the HuisKluis concept. Therefore the thesis will remain confidential and not public for 24 months from the date of my graduation.

This thesis was possible just because of the time and effort that all the different participants put in it. I would like to say thank you to my project supervisors Paul, Rob and John. We did a tremendous amount of work together in developing this project. We did a great job guys, this will be a successful project when it will be realized. A special thanks to Envolve for financing this thesis, without which this thesis would not have been possible.

If you are reading this thesis and would like more information about the concept you can contact the following end-users:

Diego Vettorel - diegovettorel@gmail.com - +31619988253 (author of this thesis)

Paul Francissen - paul.francissen@envolve.nl - +31610714772 (HuisKluis concept's creator)

Rob Van Dort - rob.van.dort@gmail.com - +31629525229 (HuisKluis concept's creator)

John Van Echtelt - john@van-echtelt.nl - +31610850096 (HuisKluis concept's creator)

I would like to thank my university supervisors Marten Van Sinderen and Erwin Folmer. They always helped me with a lot of useful feedback to improve the thesis. They were always honest and reasonable in the development of the thesis and their contribution to the project was indispensable.

I would like to thank the University of Twente that offers this master program. I think that the Business Information Technology Master program really helped me in expanding my knowledge of

both the business and the IT side. I think that it is a unique combination that I hope will give me a brilliant future in the IT and business environment.

I want to say thank you to all the people that supported me in my Dutch university career. My parents Roberto and Djamilia always supported me during all my studies. My flatmates Sigrid, Marga, Vincent, Rene and Danielle were always willing to help me and to support me in any situation. Thanks guys for all the support. Thanks to Michele for the support that you have given me in the last years, that was very important to me. Finally thanks to all my friends that in the past years supported me in everything, sorry if I do not mention everybody here, the list will occupy a lot of pages. Guys, I would not have succeed without you. Thanks a lot.

Enschede, 17 July 2015
Diego Vettorel

Abstract

Currently the information about a house is scattered over different websites and stored on different servers on the internet. House owners, inhabitants and renters perceive this to be a problem. Additionally, they often have problems sharing data with third parties in a simple and secure way, without any external applications using their private data without their permission. Besides end-users, several stakeholders are involved in this problem such as service providers and data providers. These different stakeholders need to have a platform where their needs can be met by cooperating with each other to offer the best service possible.

The problem that is addressed in this thesis is how to create this platform: to decide which services can be implemented in the platform that will use the end-users' data, to choose the software architecture and how to solve the trust issue between end-users and the different parties involved.

This thesis investigates these problems and provides a solution for each of them. Research pointed out which requirements the platform will need to have to fulfill the needs of the stakeholders. These requirements include different levels of authentication for end-users, the possibility to import external data in the platform, the possibility to develop external services that can be integrated in the platform, the possibility to share end-users' data with external websites and the alert function.

Several services that will use the end-users' data are described, such as the alarm sharing service, the photo retrieval and upload service, the company information service and a service that provides information about the appliances in the house. Additionally, there are many types of data sources that can be imported in the platform such as sources that give information about security, energy consumption, flood risks, garbage collection and municipality contact information.

Mockups and a software prototype of the platform are provided to show the different parts of the user interface and how the platform will work. The software architecture of the platform is divided in different parts. The first part is the user interface that includes different ways end-users will be able to access the platform, for instance by using a tablet and mobile browsing. The second part is the platform where the different services and servers are located. These services and servers include the linked data server, the file server, the data storage and retrieval service, the authentication service and the data conversion service. The third part consists of external actors that are connected to the platform. These actors include external websites with which users are able to share their data, the external alert function, the static data sources and the dynamic data sources.

The trust issue that plays a part in making this platform a success is also addressed. End-users do not want to share personal data with companies and other parties without their consent. However, data sharing by end-users is an essential assumption of our project, and therefore do users need to feel safe about sharing their data with the external services of the platform.

This thesis proposes a fourfold solution to this problem. The first solution is to create a foundation that will own the platform. The foundation will have the role of controlling and supervising the development and the governance of the platform. Since a foundation is not able to share its profits with shareholders, end-users will know that their data is not used for commercial purposes. Secondly, a manifesto of the foundation was written that states that the end-users are the owners of their data and that the foundation cannot share it without their consent. The third solution is to provide different levels of authentication for end-users, this will increase the security of the platform. The fourth solution is that the end-user will be able to control every single data source that they will share with external parties. End-users will have total control over the decision with whom they share their data.

Table of contents

Preface	ii
Abstract	iv
List of abbreviations.....	xi
List of tables	xii
List of figures	xiv
1. Introduction.....	1
1.1. Context	1
1.2. Problem statement.....	1
1.3. Motivation	2
1.4. Research objectives	3
1.5. Research questions	4
1.6. Research methodology	4
1.7. Report structure.....	7
2. Problem analysis and current state of research and technology.....	8
2.1. Problem analysis.....	8
2.1.1. Key concept definitions	8
2.1.2. Stakeholders identification, their needs, concerns and requirements.....	8
2.1.3. Different levels of identity certification for end-users.....	16
2.1.4. Software requirements.....	17
2.2. Current state of the research and of the technology.....	20
2.2.1. Current state of research	20
2.2.2. Existing technology solutions.....	26
2.2.3. Products and services on the market	34
2.2.4. Current problems and limitations of existing products and services.....	38
3. The business case	40
3.1. External view.....	40
3.2. End-user perspective services.....	41
3.2.1. Different levels of end-user authentication and certification.....	41

3.2.2.	Import of external data.....	41
3.2.3.	Storage of data inserted by the end-user	41
3.2.4.	Sharing data.....	42
3.2.5.	External services integration.....	42
3.2.6.	Dashboard	42
3.2.7.	Newsfeed.....	43
3.2.8.	Alert function	43
3.2.9.	Social function.....	43
3.2.10.	Wallet function	43
3.3.	Use cases summary	45
3.4.	Most important use cases	56
3.4.1.	Security alarm sharing	56
3.4.2.	Photo retrieval and upload	59
3.4.3.	Chamber of commerce company information.....	60
3.4.4.	Maintenance and various information about appliances.....	63
3.4.5.	Security information.....	65
3.4.6.	Energy information	65
3.4.7.	Flood risk information	66
3.4.8.	Garbage collection information	66
3.4.9.	Municipality contact information	67
3.5.	Data sources	68
3.6.	Revenue models	70
3.6.1.	Web advertisement.....	70
3.6.2.	Fees from end-users	71
3.6.3.	Fees from service providers and data source providers	73
3.6.4.	Consulting for external companies	74
3.6.5.	Selling the complete platform to external companies	76
3.7.	Foundation model and open source model.....	78
3.7.1.	Foundation model.....	78
3.7.2.	Open source model.....	81

4.	Software architecture	84
4.1.	Software architecture design method	84
4.2.	Requirement analysis	86
4.3.	Technical problems analysis.....	87
4.3.1.	R1. Different levels of authentication for end-users.....	87
4.3.2.	R2. The possibility to import external data	87
4.3.3.	R3. The possibility to store data inserted by end-users.....	87
4.3.4.	R4. External services development.....	88
4.3.5.	R5. Dashboard	88
4.3.6.	R6. News feed	88
4.3.7.	R7. End-users will be able to receive alerts.....	88
4.3.8.	R8. Social function	89
4.3.9.	R9. Wallet function	89
4.3.10.	R10. Tablet and mobile	89
4.4.	Solution domains knowledge analysis.....	90
4.4.1.	T1. How is the platform going to provide the different levels of authentication for end-users? 90	
4.4.2.	T2: how will the system store the authentication data in a reliable way for the authentication component?	90
4.4.3.	T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?.....	90
4.4.4.	T4: How is the platform going to connect to external data sources?	91
4.4.5.	T5: How will the system store the end-users' data?	92
4.4.6.	T6: How are we going to integrate the external services in the platform?.....	92
4.4.7.	T7: How are we going to develop the news feed for every end-user?	93
4.4.8.	T8: How are we going to implement a reliable alert function?	94
4.4.9.	T9: How are we going to implement the social component in the software architecture that will permit end-users to connect with other end-users in the platform?.....	94
4.4.10.	T10: How are we going to implement the wallet component that will share the end-users' data with external websites in the software architecture?.....	95

4.4.11.	T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?.....	96
4.5.	Solution analysis	97
4.5.1.	T1: How is the platform going to provide the different levels of authentication for end-users?	97
4.5.2.	T2: How will the system store the authentication data in a reliable way for the authentication component?.....	97
4.5.3.	T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?.....	97
4.5.4.	T4: How is the platform going to connect to external data sources?	97
4.5.5.	T5: How will the system store the end-users' data?	98
4.5.6.	T6: How are we going to integrate the external services in the platform?.....	99
4.5.7.	T7: How are we going to develop the news feed for every end-user?	99
4.5.8.	T8: How are we going to implement a reliable alert function?	100
4.5.9.	T9: How are we going to implement the social component in the software architecture that will permit end-users to connect with other end-users in the platform?.....	100
4.5.10.	T10: How are we going to implement the wallet component that will share the end-users' data with external websites in the software architecture?.....	101
4.5.11.	T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?.....	101
4.6.	Synthesis of the overall architecture.....	107
4.7.	Scenario analysis	115
5.	Mockups of the UI.....	118
5.1.	Interface description	118
5.1.1.	Homepage	118
5.1.2.	Subcategories webpage	119
5.2.	Mockups of the interface	120
5.3.	Analysis and testing of the UI	124
6.	Software prototype.....	126
6.1.	Test of the software prototype.....	126
7.	The trust issue	128
7.1.	Problem description	128

7.2. The solution to the trust issue in this platform	129
8. Discussion	131
9. Conclusion	132
9.1. Reviewing the research questions	132
9.2. Scientific contributions	135
9.2.1. Theoretical contributions.....	135
9.2.2. Practical contributions.....	137
9.3. Research limitation.....	138
9.4. Recommendations for future work and research	139
10. References	141
Appendix A: end-user interviews.....	143
First round.....	143
Second round	146
Appendix B: data sources that will not be used for the software architecture in this platform but that can be used for future use cases.	150
Appendix C: mockups of the interface.....	152

List of abbreviations

OSS:	Open Source Software
MM:	MissionMode
ADS:	Advertisements
SA:	Software Architecture
IdM:	Identity Management
JS:	JavaScript
SOA:	Service Oriented Architecture
CSS:	Cascading Style Sheets
API:	Application Programming Interface
GPL:	General Public License
#:	Number
UI	User Interface
RQ	Research question
R:	Requirement
T:	Technical problem

List of tables

Table 1: Stakeholders analysis methods.....	9
Table 2: Pains and gains of end-users.....	10
Table 3: Pain and gains of data source providers.....	12
Table 4: Pain and gains of service providers.....	13
Table 5: Pain and gains of public administration services	15
Table 6: Pain and gains of end-users	16
Table 7: Software requirements.....	18
Table 8: Software requirements linked with the different stakeholders involved.....	20
Table 9: Requirements and technologies comparison	39
Table 10: Use cases list explanation.....	45
Table 11: Use cases summary	55
Table 12: Description of main aspects of the most important use cases	56
Table 13: Alarm use case benefits analysis.....	59
Table 14: Archief Eemland photo retrieval and upload benefit analysis.....	60
Table 15: Chamber of commerce company information benefit analysis	63
Table 16: Maintenance and various information about appliances benefit analysis	65
Table 17: Security information benefit analysis.....	65
Table 18: Energy information benefit analysis	66
Table 19: Flood risk information benefit analysis.....	66
Table 20: Garbage collection information benefit analysis	67
Table 21: Municipality contact information benefit analysis.....	67
Table 22: Data source list explanation.....	68
Table 23: Data sources used for the first release of the platform	69
Table 24: Description of the revenue models.....	70
Table 25: Foundation board activities.....	79
Table 26: Foundation employee activities.....	80
Table 27: Consultancy services.....	80
Table 28: Foundation additional revenue models	80
Table 29: OSS advantages	82
Table 30: OSS disadvantages	83
Table 31: Software requirements.....	86
Table 32: Service layer details	110
Table 33: Questions for interface graphic designers.....	124
Table 34: Interface graphic designers feedback.....	125
Table 35: Details of end-users interviewed concerning the software prototype	126
Table 36: Answers of the prototype's testers.....	127

Table 37: Details of end-users interviewed in the first round	143
Table 38: First round questions	143
Table 39: First round contexts	144
Table 40: First round interview results	145
Table 41: Details of interviewed end-users in the second round.....	146
Table 42: Second round interview questions.....	147
Table 43: Second round results	148
Table 44: Data sources that will not be used for the software architecture of this platform but that can be used for future use cases	151

List of figures

Figure 1: Research framework.....	6
Figure 2: Conceptual view	21
Figure 3: Architecture of the virtual sensor system	22
Figure 4: MyPHRMachines architecture, Van Gorp et al (2014)	24
Figure 5: Existing technology solution schema.....	27
Figure 6: Tim Berners-Lee's rules for Linked data	28
Figure 7: Berners-Lee's ratings of linked data	29
Figure 8: RDF triple structure	29
Figure 9: RDF graph for a sentence with to objects.....	30
Figure 10: RDF data.....	30
Figure 11: SPARQL query	31
Figure 12: SPARQL query result.....	31
Figure 13: SPARQL normal querying	32
Figure 14: Local endpoint data dumps	32
Figure 15: LDF technology	32
Figure 16: Software architecture of Apache Syncope.....	34
Figure 17: External view	40
Figure 18: The end-user registers a new alarm with the correct alarm number and password	58
Figure 19: The end-user registers a new alarm with an incorrect alarm number and password.....	58
Figure 20: End-user uploads and shares a photo with Archief Eemland	60
Figure 21: An end-user wants to have paid information about a company.....	62
Figure 22: Adding a new appliance in the system	64
Figure 23: E3Value proposition of the Web advertisement revenue model.....	71
Figure 24: E3Value proposition of the fee from end-user revenue model.....	72
Figure 25: E3Value proposition of the fee from service providers and data source providers revenue model	74
Figure 26: E3Value proposition of consulting for external companies revenue model.....	75
Figure 27: E3Value proposition of the complete platform revenue model.....	77
Figure 28: Software architecture design method.....	84
Figure 29: LinkedIn - A Professional Network built with Java Technologies and Agile Practices	93
Figure 30: LinkedIn SA.....	95
Figure 31: Cross-Platform Authentication with Google+ Sign-In.....	96
Figure 32: SA to connect the platform with external data sources.....	98
Figure 33: SA of the storing data component of the platform	99
Figure 34: Newsfeed SA.....	99
Figure 35: Social component SA	100

Figure 36: Wallet function SA	101
Figure 37: Board SA.....	102
Figure 38: PNMsoft SA	103
Figure 39: Different components of a SOA SA	104
Figure 40: Overall SOA architecture	105
Figure 41: Complete SA of the platform	107
Figure 42: Client layer and frontend component	108
Figure 43: Service layer and the frontend component	109
Figure 44: Data storage and retrieval service, file server and SPARQL server.....	111
Figure 45: Data conversion service and static and dynamic external data sources.....	112
Figure 46: Authentication service and MySQL	113
Figure 47: Wallet and alert services connection with external actors	114
Figure 48: Scenario process scheme	116
Figure 49: Homepage without the end-user logged.....	121
Figure 50: Homepage with the end-user logged	122
Figure 51: Alarm information and use case webpage	123
Figure 52: The solution to the trust issue of this platform.....	129
Figure 53: Combining open and closed data with a location.....	136
Figure 54: Importing and selling data from other companies' datasets	137
Figure 54: Mockup of the homepage without an end-user logged in	152
Figure 55: Mockup of the homepage with the end-user logged in	153
Figure 56: Mockup of the homepage with the end-user settings menu.....	154
Figure 57: Mockup of the homepage with the gas submenu	155
Figure 58: Mockup of the gas webpage.....	156
Figure 59: Mockup of the homepage with the alarm submenu	157
Figure 60: Mockup of the alarm webpage.....	158
Figure 61: Mockup number one of the alarm sharing use case.....	159
Figure 62: Mockup number two of the alarm sharing use case.....	160
Figure 63: Mockup number three of the alarm sharing use case	161
Figure 64: Mockup number four of the alarm sharing use case	162

1. Introduction

In this thesis we will present a complete business case for the design of a platform that provides the end-user with useful information and functions that will permit him to interact with different data connected with the system and his house. Keywords for these services are care, safety, comfort and fun. In this thesis we will use the word "use case" to refer to a service that, with sources of information, will provide useful functions to end-users. A "use case" is also a service that provides just information to end-users. In this thesis I will use any plural personal and possessive pronouns (we, our) to describe the work of the author of this thesis (Diego Vettorel), Envolve's owner and HuisKluis concept's creator (Paul Francissen) and the HuisKluis concept's creators (Rob Van Dort and John Van Echtelt). In this thesis we will use the word "inhabitant" to refer to a person that live in a house.

In this introductory chapter we present an introduction on the topic, the motivation that lead us to write this thesis and the research objectives and methodology of this thesis.

1.1. Context

These days, open data is becoming more and more important. We are going to enter a new era: the Web 3.0 era. In this era, data will be better connected to the internet (for instance with Linked Data). People will be able to search and use data stored in different servers in an easy and efficient way. Data will be structured and easily accessible.

The problem is that at this moment an easy and widely used platform to collect and access information does not exist. A new way to manage this standard information will be used in the future by all internet end-users. The situation is similar to the early age of social media. Many different companies are trying to make their interfaces and platforms the dominant factor but none prevails over the others.

The technology and the ideas are already on the market. What the different parts involved really need is an organization that will lead the market and that will provide a real standard and a platform for house-related data and services. This to collect all the information in a simple way and to give access to as many end-users as possible. We have put a lot of consideration into the linked-data technology.

A few different companies are trying to solve this problem, but in this thesis we will focus on the collection of data regarding houses and on providing useful services that use this data.

1.2. Problem statement

The main problem that we want to solve is how to make the life of end-users easier by connecting the different sources of data about houses in one interface. At the moment information about houses and their qualities is scattered over different sources. Accessing these sources separately and collecting the data this way is time consuming and bothersome. With our platform this problem will be solved.

In addition, we will solve the problem of sharing of this data with other websites and with other applications. Right now, it is difficult for a company to access data of a customer in a simple way and respecting the privacy of the customer. With our system this problem will be solved by permitting every end-user to choose with who he will share his data. Different companies will be able to access the end-user data in a simple way.

In this thesis we will provide an analysis of a future platform that will solve these problems. These kind of problems, extended to all the sources of information and not limited to the house field, will be a growing problem in the next few years.

With our research we will contribute to solve this problem also in others fields than the house-related information. Our contribution will help future research in this field by providing a complete business case, a complete SA and a complete software prototype of the platform.

1.3. Motivation

This thesis shows a future way to connect different sources of information linked to a house with the owner of the house or with its inhabitants. This idea is the essence of the project. The provided information can be used by external services and applications that will benefit the inhabitants and the owner of the house.

At the moment a lot of information about houses is available for end-users. The problem is that this information is stored in different places, and scattered over different companies and different servers. Also it is not simple to access this information by end-users since not all companies provide an end-user friendly interface.

This thesis contains the analysis for a platform where end-users can access their house's data with a simple and end-user friendly interface. They will also be able to use different apps, developed by third party companies, that will use their houses data. Doing so enables end-users to share their data with external websites and give them the opportunity to decide which data they want to see in their personalized interface. Additionally, end-users can connect with external services that will use the information from the portal to provide useful functions to them. These functions are not available with the current products and technologies. We aim to make them possible with this platform.

This project may be the future of house data sharing. Using our platform a person will be able to access the data of his house from a single and integrated interface, and can use external applications that access this data. For instance, a person can choose to share his alarm security information with his neighbors and with the police. In this way, if a malicious person enters his house, the neighbors and the police will be alerted immediately.

Another example can be sharing information about different rooms in a house with the firefighter department of the city. In addition, there are many different other uses of the system.

End-users will be able to access to this platform with different devices such as tablets and smart phones.

1.4. Research objectives

The objective of this thesis is to design a platform to provide home-based services concerning care, safety, comfort and fun. This includes a business case of the system, a detailed SA of the future system and a detailed description of some use cases.

This is a very innovative field. For instance, we have researched how to aggregate data about houses and how to use this data to help end-users' lives. We researched how to link this data with external services and websites. We have created use cases that, when they will be implemented, will help end-users and will provide them useful functions. We have also done case study research on the existing platforms that are similar to this one.

Currently end-users cannot access a simple interface where all their house data is shown a simple way. They also cannot use this data with external applications by deciding which application can access which data resource. Data providers now do not have an integrated and simple system to provide their data to end-users. Additionally, app developers do not have an integrated source of information from end-users. With this platform these problems will be solved and the different parties can start to provide these services.

Another objective of this thesis is to provide a software architecture design to implement the platform. The software architecture needs to satisfy the requirements that we have gained from analysis of the wishes of different parties involved. We analyzed different parts of the architecture, for instance the central platform, the user interface, the data import and the physical layer. The architecture supports the import of linked data and of non-linked data. This data will be processed and made accessible for third party companies and services.

More specifically the objectives of the thesis can be divided into four main categories:

1. The business case:

- Provide a business case of the platform.
- Provide a functional description of the different functions of the system.
- An alert function that will permit end-users to receive an alert on the mobile app (or by e-mail or by any other communication system) when an alert is sent from the system.
- A wallet function where end-users will be able to share their data information in a simple way. End-users will be able to share the information of their wallet with every application or website that they want.
- Provide different identification methods for different levels of trustiness of the end-users.
- Provide the description and the analysis of several use cases for the system.
- Provide different examples of data sources from where the system will take the data and how they will connect to the system.
- Provide different examples of ways to access the data (web interface, mobile app).

2. The software architecture:

- Provide a software architecture for the system.

3. The trust framework:

- Provide the solution of the “trust” issue between this platform and the end-users (end-users may be scared to share their data).

4. The proof of concept:

- Provide the design of the interface that the final dashboard (where all the different information of the house will be shown to the end-user) will have.
- Provide the design of the interface of several use cases.

1.5. Research questions

The research questions are:

- **RQ1:** What are the characteristics of a platform where end-users will be able to see, share and analyze information regarding their houses?
- **RQ2:** Which use cases will be implemented that use information about houses and that will provide useful functions to end-users?
- **RQ3:** What will the software architecture for such a system look like?
- **RQ4:** What can be a solution for the trust issue between end-users and the information sharing platform?

1.6. Research methodology

For this thesis a design-science approach was used. First interviews were held with the different stakeholders involved. Interviews with end-users, public administration services and social organizations, service providers and data source providers shared their needs and requirements. At the same time a discussion between the owners of the platform was held to find out our needs and requirements.

From the interviews and discussion a requirement list was compiled. Then a market analysis of the existing solutions that have already been developed was done to find out if any organizations had already developed a platform with these requirement combinations.

Then the business case of the platform was made using all of these requirements and the SA of the platform was designed. At the same time the an analysis of the “current state of technology and research” was made and the information that was gained here was used for the business case creation and for the SA design. Furthermore, at the same time several analyses of external companies to research the possibilities of partnerships were made and the information gained from these analyses were also integrated in the business case and in the SA design.

The next major step was to design the UI of the platform by making some mockups. After the designing the mockups were tested with two different user interface designer experts that given us feedbacks that we integrated in the mockups.

Then a software prototype was made and it was tested with potential end-users of the platform to find out if the software requirements were satisfied. Then the final software prototype was made.

Then the problem of the trust issue was analyzed and solved using all the information that we gained from the previous steps.

Finally the discussion of the results and the conclusion of the thesis was made.

Figure 1 shows this research method process.

Additionally the E3value model is used to research which stakeholders will benefit from the platform for each revenue model. Using the E3 value model gives insight into the question when the different stakeholders will be satisfied and which revenue streams they will take and give.

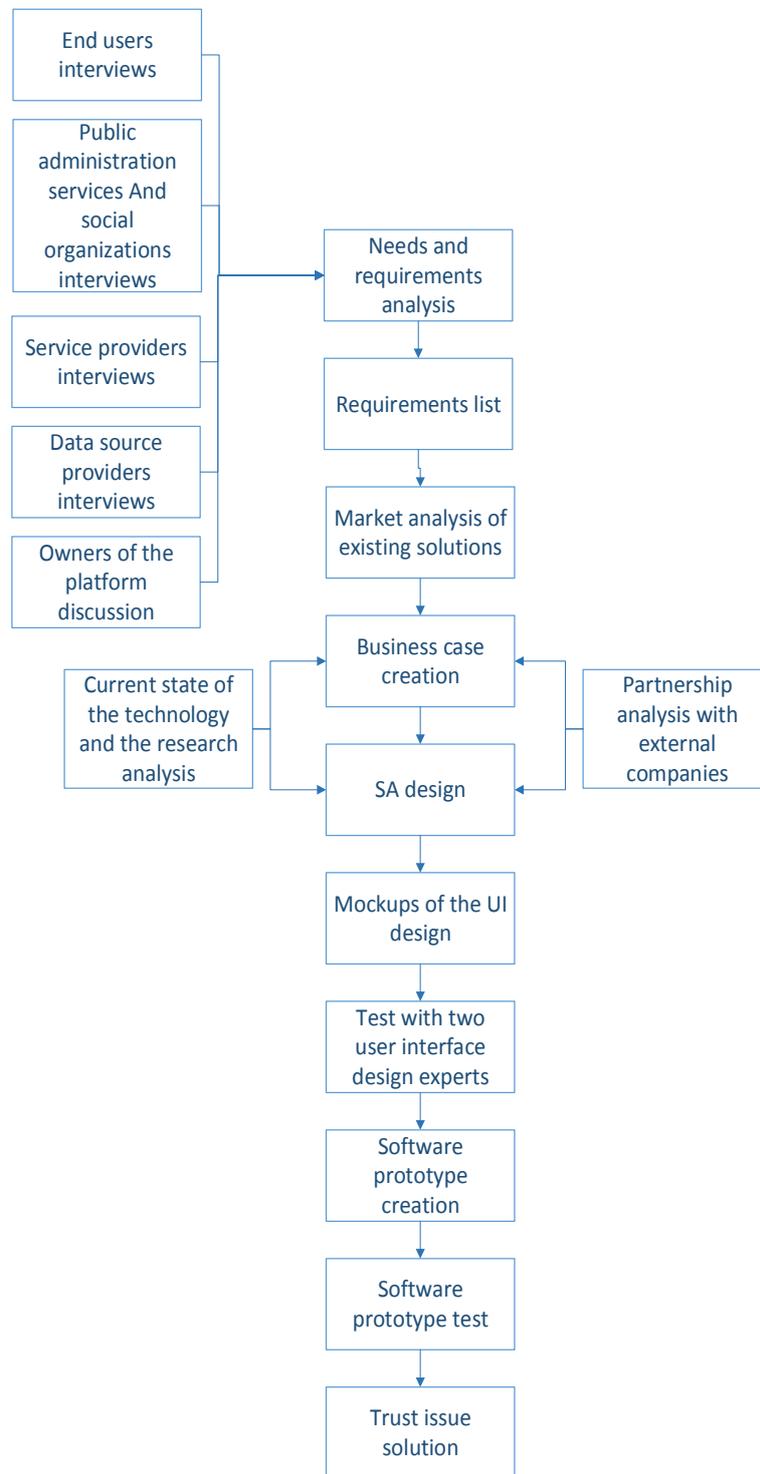


Figure 1: Research framework

1.7. Report structure

In the introduction chapter it is presented a description of the context, the motivation and the methods used for this research.

In chapter two we present an analysis of the problem, the stakeholders and the end-users analysis, the requirements of the future platform and a complete analysis of the current state of the technology.

In chapter three we present the complete description of the business case that we made during our research. A user perspective services analysis was made and there were analyzed some use cases, some data sources and some revenues models. Finally an analysis of the foundation governance model and of the OSS was made.

In chapter four we present the software architecture that the system will have and the process that we followed to make it.

In chapter five we present the description of the mockups of the platform and the analysis of the test that we made of them. The reader can find mockups of the interface that the system will have.

In chapter six the description of the software prototype of the platform is made and some test of it are described.

In chapter seven the trust issue between the platform, the end-users and the external companies is analyzed and there is described why our platform will solve this trust issue.

In chapter eight the discussion of the results is made.

In chapter nine the conclusion is made that includes the review of the research questions, the theoretical and the practical scientific contributions, the research limitation and the recommendations for future work and research.

In the appendixes there are explained some details about the interviews with the end-users that we made, some data sources and some mockups of the interface.

2. Problem analysis and current state of research and technology

In this chapter a problem analysis is provided, along with an analysis of the current state of technology and of research. In the problem analysis part the stakeholders, their needs, their concerns and their requirements are discussed. Then an analysis of the different levels of identity certification for end-users and the end-user requirements analysis is given. Finally, in the current state of research and technology section, existing platforms that provide similar services and functions to this platform are scrutinized. In that section we will also analyze the current state of research and existing technology solutions connected to this platform.

2.1. Problem analysis

2.1.1. Key concept definitions

In this section definitions of the three main concepts are stated. These definitions are the general agreed definitions for the most important concepts contained in this thesis.

2.1.1.1. Integrated platform

An integrated platform is software that integrates different applications and services. Within the boundaries of this thesis “integrated platform” refers to a platform where different data sources, applications and services are integrated.

2.1.1.2. Linked-Data

Linked Data provides a method to publish structured data. The term Linked Data, following Tim Berners Lee (2009), refers to practices for publishing and connecting structured data on the web. These pieces of structured data can be linked together and can be retrieved using the SPARQL language. Currently many sources of Linked Data exist on the internet, which can provide useful data for this platform.

2.1.1.3. Use case

A use case, following Cockburn (2002), is a description of a system’s behavior when interacting with the outside world. In this thesis a use case is a service that uses external data sources, new data provided by the end-user, connections with other services and end-user input to provide useful functions to the end-user.

2.1.2. Stakeholders identification, their needs, concerns and requirements

In this section the different stakeholders involved in the platform are analyzed. A stakeholders analysis is important because it clarifies which requirements this platform will need to have to satisfy the stakeholders’ needs.

Determination of the shareholders followed the results of analyzing the different functions and services that will be provided by the platform. The different stakeholders that were found are connected in different ways

to the system. Five groups of stakeholders have been identified: end-users, public administration services and social organizations, service providers, data source providers and the owners of the platform. End-users are the customers that will use the system, that will see the information provided and that will use its functions. Public administration services and social organizations will collaborate in the creation of the platform and the services connected with them. Service providers will make useful services for end-users. Data source providers will provide the data sources of the platform. The owners of the platform will be the people that will develop and manage this platform.

For each stakeholder involved a complete analysis of their needs, pain, gains and the value that they add to the platform was made. This analysis was made by analyzing services and functions that each of them will provide or use. Interviews with them were held to better understand the different needs and qualities of each of them. For each pain and gain a vote of importance is given based on the analysis of each stakeholder.

The methods of analysis of the stakeholders involved are presented in the next table.

Name	Description
Research methods	How we researched the needs and the information regarding the stakeholder.
Description	Description of the stakeholder.
Vision	The vision that we have of the stakeholder. This vision is created on the basis of the analysis that we made of each stakeholder involved.
Analysis	A brief analysis of the stakeholder.
Needs	The different needs of the stakeholder.
Desired product	The product that the stakeholder wants to have.
Pains and gains	An overview of the advantages and disadvantages of the platform for each stakeholder. An importance evaluation of each pain and gain is made. The pains and the gains that are already mentioned in the analysis may not be present in this section.
Value that they will bring to the platform	The additional value that the stakeholder will bring to the platform.

Table 1: Stakeholders analysis methods

2.1.2.1. End-users

Research methods

As explained in

Appendix A: end-user interviews, we held interviews with potential end-users of the platform. These interviews gave insight into different details of end-users and to make the software requirements that are explained in section 2.1.3.

Description

End-users are the users that will use the platform. They will register to the platform, retrieve data and use the services. They will not provide any data sources (except the data that they will insert in the platform) and they will not develop any services. They will be able to provide feedback to us.

Vision

We will provide information and services that allow end-users to make better choices regarding their home/life.

Analysis

End-users will provide data subscription, the end-user base and feedback to improve the system. We will reach these end-users by promoting the service with the partnership of national consumer organizations and governments. End-users will also be reached via data sources and service providers companies.

Needs

End-users have the following needs:

- A safer living environment: they want to have more control of their living environment in terms of safety. They also want to be able to control their living environment remotely and to share their alarms with other end-users.
- Better care at their home: they want to have better care in their home, this is important especially for old end-users.
- Save costs: they want to spend less money for the services and for everything connected to their house.

Desired product

End-users want a secure platform that provides useful information by aggregating different services and different data sources.

Pains and gains

	Description	Importance
Pains	Difficult to manage information about their house/neighborhoods	Medium
	Difficult to find/access services related to their house/neighborhoods	High
Gains	Integrated platform for services and data	High

Table 2: Pains and gains of end-users

Value that they will bring to the platform

- User base: they will be the end-user base for the platform. Having a user base is important because it permits to have the a lot of end-users that use the services and that provide feedback. A lot of end-users also attract more end-users and free advertisements.
- User profiling: they will provide a lot of useful information about the different types of end-users and their needs, this information will be used to improve the platform.
- Feedback: they will provide feedback to improve the platform.

2.1.2.2. *Data source providers*

Research methods

We interviewed and made some research on several possible data source providers that want to offer their data on our platform.

Description:

Data source providers will provide the data that will be connected to the platform. This data will be shown to the end-user on the dashboard and it will be used by the services.

Vision

We will provide a platform to data source providers to share the data that they provide in a simple way. They will be able to get revenues from the sharing of this data with us.

Analysis

They will provide data for the platform. Data suppliers will receive a fee for the data sold via the platform. We will reach data suppliers by looking at the open data that is now available and by creating partnerships with companies that offer data on a commercial basis.

Needs

Data source providers have the following needs:

- To have an already existing and simple reachable end-user base: this guarantees an end-user base that will use the data source.
- To have the validated identity of end-users when data is retrieved: this will permit to use also private data sources that share their information after the end-user authenticated to the system.
- To have companies that, by using their data sources, provide useful functions to end-users: this will permit data source providers to have more end-users that access their data.

Desired product

Data source providers want a platform that:

- Retrieves information from their data sources and, by using them, provides useful functions to end-users: this will make possible to use data sources and to provide useful function to end-users by using them.

Pains and gains

	Description	Importance
Pains	Difficult to advertise their shared data to potential end-users.	Medium
	Difficult to provide a simple interface to show their data to end-users.	High
	Difficult to make revenues from selling their data sources.	High
Gains	Statistics containing information about the usage of their data sources.	Medium
	A platform where end-users can use their data.	High

Table 3: Pain and gains of data source providers

Value that they will bring to the platform

- Data sources: this is one of the main functions of the platform. Without data sources this platform will not be able to satisfy the needs of its stakeholders.
- Advertising on their website and channels: this will probably attract more end-users to the platform.

2.1.2.3. Service providers

Research methods

We interviewed possible service providers that want to offer their services on our platform.

Description

Service providers will develop services for this platform. The services that they will develop are shown in the interface of the platform. These services are connected to the data sources of this platform and to the data provided by the end-users.

Vision

We will empower service providers by creating new services for inhabitants that will use our platform and data sources.

Analysis

Service providers will provide services that will connect to our data sources and that will be integrated in our interface. They will have access to the platform's end-user base and they will get access to the data sources of the system, provided that the end-users have given permission. We will reach them by creating partnerships with external companies.

Needs

Service providers have the following needs:

- To have an already existing and simply reachable end-user base: this gives them a first initial base of end-users that will use their services.

- To access end-users' data to deliver a personalized service: they will need to access end-users' data to provide a personalized service to end-users based on their open data.
- To have the validated identity of end-users and their belongings: they need different levels of certification of end-users to provide privacy-sensitive services to end-users.

Desired product

Service providers want a platform that:

- Provides a service for retrieving data: this permits them to use data for their services
- Provides a standard user interface for services: this provides to end-users an interface that they already know, permitting end-users to learn how to use services more quickly.
- Provides a secure way to transfer and store information: this will convince end-users to have more trust in the services and in the platform and so to use more of them.

Pains and gains

	Description	Importance
Pains	Difficulty in making data sources available.	High
	Difficult to reach a lot of end-users.	Medium
	Difficult to have a trusted identity from the end-user.	High
Gains	A platform that provides data and the end-user base.	High

Table 4: Pain and gains of service providers

Value that they will bring to the platform

- Fees: they will pay a fee based on the revenue models of chapter 3.
- Added value to end-users and to the platform: the services that they will develop will provide an added value to the platform because they will provide useful functions to end-users.

2.1.2.4. Public administration services and social organizations

Research methods

For this group of stakeholders we held interviews with Dutch municipalities that clarified their needs and requirements.

Description

Public administration will be involved in the developing of the platform because they will provide information, advice, funds and services for the platform. They are similar to the data-providers and to the services providers because they will also provide data and services to the platform. However, they have special needs that are explained in this section and that requires them to be considered as a standalone stakeholder.

Vision

This platform will be able to empower individuals and communities to solve societal challenges by themselves. These challenges include the retrieve and the share of municipalities information and the use

of municipalities services in a simple way. They will solve these challenges by providing data and services to our platform that end-users will use.

Analysis

They will use this platform to improve public services to residents and homeowners. They will be able to share government open data with end-users and with services providers. We will reach them by making a partnership with them and with different government departments.

Needs

Public administration services have the following needs:

- Innovative image: they want to have an innovative image to end-users that live in their municipalities.
- Empower inhabitants: municipalities want to empower their inhabitants by permitting them to take care of processes that they currently cannot manage themselves. These processes include services that are managed by municipalities at the moment and that will be managed autonomously by end-users.
- Provide inhabitants with better, cheaper and more extensive services in the fields of safety, care and information supply.
- Learn and experience service innovation: municipalities want to experience new techniques and methods in the field of service innovation.
- Save money: municipalities want to save money by simplifying their internal processes or by permitting end-users to manage a number of services by themselves.

Desired product

Public administration services want a platform that:

- Is based on standard protocols: they want a platform based on standard protocols that gives end-users better access to their information.
- Has the trust of inhabitants: for them it is important to have the trust of their inhabitants because this will increase their public image.
- Provides a validated identity function of end-users: this is important because services that they will offer through this platform will require levels of validated identification from end-users.

Pains and gains

	Description	Importance
Pains	Challenge in how to deal with social changes.	Medium
	Provide proper digital tools to empower individuals.	High
Gains	Innovative image.	High
	A platform to provide services and data to inhabitants.	High
	Learn and experience service innovation.	Medium

Table 5: Pain and gains of public administration services

Value that they will bring to the platform

- Direct financing: they will finance this platform.
- Assistance in receiving indirect finance: they will help us to receive indirect financing from third parties.
- Exposure : they will provide free advertising for this platform.
- Access to inhabitants: they will provide large user bases consisting of their inhabitants.
- Access to service providers: they will help us to contact external service providers.
- Data providers: they will provide data to the platform.
- Service providers: they will provide services to the platform.

2.1.2.5. Owners of the platform

Research methods

We will be the owner of the platform. Our needs come from the discussions that we had about the developing and the managing of the platform.

Description

The owners of the platform will be the owners of the shares of the company that will provide the platform. We assume that we will not use a foundation model for this section but a normal limited responsibility company.

Vision

The platform will provide us with a good job as managers of the company and with the dividends of the company. Moreover, this platform will help communities to access information and services in a safe, reliable and simple way.

Analysis

We will provide the vision of the platform and the management part. Initially we will also manage the other parts of the company such the marketing and development parts.

Needs

We have the following needs:

- To develop a platform that will help end-users all over the world regarding data and services connected to their houses.
- An interesting job in the company or in the foundation.
- Profits or salary from the company or from the foundation.

Desired product

We want a product that will be successful in satisfying the needs of the other stakeholders involved.

Pains and gains

	Description	Importance
Pains	The problem of sharing information about houses in a lot of different websites.	Medium
Gains	Great experience in the creation of the platform.	High

Table 6: Pain and gains of end-users

Value that we will bring to the platform

- Designing, creating, managing and maintaining the platform.

2.1.3. Different levels of identity certification for end-users

The platform needs different levels of identity certification for end-users. Each service and data source will be linked to a different level of end-user certification permitting end-users that have the maximum level of certification to have access to all the available data sources and services. In this thesis the words "authentication" and "certification" are used to express the same concept of having different levels of identity certification for end-users.

To establish these different levels of certification we analyzed the needs of the stakeholders that will provide services and data sources to the platform. This was done by interviewing stakeholders and by analyzing the different services, data sources and functions that they will provide. For each service, data source and function we analyzed the possible level of authentication that the end-users will need to have for each of them. The results are presented in this section.

The different levels of certification for the end-users need to be:

1. **Email confirmation:** this is a low level security measure as anybody can use fake credentials to register to the platform and can use a fake email address.
2. **Address confirmation:** a letter with a security code will be sent to the address of the end-user. The user needs to insert this code into the system to prove that he lives on that address. This level of certification is medium-low because the end-user can use a fake name and surname. Interception of the letter is another threat to this method.
3. **Name, surname and address confirmation with an identity document verification:** end-users will send a copy of their identity document to us. We will verify that the data inserted by the end-user matches the details of the provided identity document. This is a low security certification measure because an identity document image can easily be faked.
4. **Name, surname and address certification from the government:** our system will request certification from the government that proves that the end-user really lives at that address. This is a medium security certification measure because government data is carefully checked. On the

other hand, with this security method we can just prove that a particular person lives at a particular address but we cannot prove that the person that registers to the platform uses his real name and surname.

5. **Bank name, surname and address certification:** the credentials of the end-user are checked against his bank account details. This method provides a high certificated identity because every bank end-user is checked personally with his identity document.
6. **DigiD certification:** this certification proves the identity of one person and his address. This provides a high certificated identity because every DigiD identity has the name, the surname and the address of users checked. To have this level of certification we will need to make an arrangement with the Dutch government because private organizations cannot access the DigiD technology.

2.1.4. Software requirements

In this section the software requirements of the platform are described. These findings are based on the information we gleaned from the interviews with stakeholders. These requirements will be mandatory for the final platform to satisfy the needs of all the stakeholders involved.

R#	Title	Description
R1	Different levels of authentication for end-users	Every service, function and data source will require a different level of authentication from the end-user. The section 2.1.2.1 describes in detail the different levels of identity certification.
R2	The possibility to import external data	The platform allows importing data from external sources in the system. This data can be static or dynamic (for instance from external sensors).
R3	The possibility to store the data inserted by end-users	End-users will be able to insert data in the system that is stored on the platform. End-users will also be able to manage and delete this data.
R4	External services development	External companies will be able to access end-users' data and to develop services providing useful functions to end-users using that data. A list of the available services will be available on the dashboard. These services will access the external data sources connected to the end-users. End-users will be able to control which data sources will be shared with the external services.
R5	Dashboard	End-users will be able to see in their dashboard all the data connected to their house, the list of services, the newsfeed and links to the other functions of the platform like a messaging system. End-users will be able to see everything about their home in one place.

R6	News feed	End-users will be able to see a newsfeed on the dashboard. The newsfeed will contain news about the system like new services, new data sources and important data retrieved from external data sources.
R7	End-users will be able to receive alerts	Alerts are messages sent from services and from the platform to end-users by email, mobile app, SMS and automatic calls.
R8	Social function	End-users will be able to add other end-users as friend, to share data with them and to exchange messages with them.
R9	Wallet function	End-users will be able to share their data with external websites with a wallet function.
R10	Tablet and mobile	The user interface website will have a modality that will support tablet and mobile browsing.

Table 7: Software requirements

Table 8 below summarizes the information from the stakeholders that lead us to formulate these requirements. Each requirement is linked to the stakeholder involved. A blank cell means that the requirement is not relevant for the stakeholder.

R#	End-users	Public administration and social organizations	Service providers	Data source providers	Owners of the platform
R1	They want to retrieve and share personal information, in order to do this different levels of certification of end-users are needed	They want to provide their inhabitants with services that will need specific authentication requirements	They want to offer services to end-users that will need to have different levels of certification of end-users' identity	They want to provide personal data to end-users. End-user identity certification is therefore extremely important.	We want to provide a safe service to end-users with our platform. Because of this different levels of certification of end-users are required
R2	End-users want to retrieve data from different sources	They want to provide inhabitants with data that will be imported in the platform	They want to use the data imported by the end-user from different sources for the services	They want to provide their data to the platform	We want to make the life of end-users easier by permitting them to import their external data

R3	End-users want to insert data in the platform, for instance an overview of what they are going to share with neighbors		They want to use the data that end-users stored on the platform		We want to permit end-users to insert their data to provide them with useful functions and relevant services
R4	They want to have services available through the platform, for instance a service indicates that the energy saving option is available	They want to provide services to their inhabitants	They want to provide services to end-users	They want their data to be used by services, because then it becomes more useful and valuable	By offering a lot of external services, end-users will be more satisfied of the platform and thus more willing to use it.
R5	End-users want to see all their data accessible through one dashboard because they perceive the scattering of their information as a problem		They want a dashboard where end-users will be able to see a list of their services	They want a dashboard where end-users will be able to access their data through one single platform	By using the dashboard the end-users will be able to find their data in a single place. This will make them more willing to use the platform.
R6		They want to show news about their data and services in the newsfeed	They want to show news about their services in the newsfeed	They want to show news about their data in the newsfeed	In the newsfeed the end-user will be able to see their alerts and news. This will lead to more satisfied end-users
R7			They want to be able to send alerts to end-users		The alert function will lead to more satisfied end-users

R8	They want to share their information with other end-users				The social function will lead to more satisfied end-users
R9			They want their end-users to be able to easily share data with/within the service offered	They want the data that is imported to the platform to be shared with external websites by end-users	The wallet function will lead to more satisfied end-users
R10					End-users will be more satisfied if they are able to see the platform from different types of devices.

Table 8: Software requirements linked with the different stakeholders involved

2.2. Current state of the research and of the technology

2.2.1. Current state of research

In this section current scientific papers relating to our platform are analyzed and discussed. The main catalogue where these papers were found was Sciencedirect (<http://www.sciencedirect.com/>). Some of the researches presented in this section are not used in the thesis because we realized, during the later analysis, that they were not useful. They are included in this section to show a complete overview of the research the we made.

2.2.1.1. *A flexible platform for synchronized measurements, data aggregation and information retrieval*

Alexandru Nechifora, Mihaela Albul, Richard Hairc and Vladimir Terzijaa (2015) explained a way to realize a platform that receives measurements from Phasor measurements units (PMUs) and then aggregates and shows their data.

This paper can be helpful for the development of our platform because it explains how to retrieve real time data from measurement units. In our platform we will need to import data from real time sources like thermostats and weather stations. These data then need to be shown to end-users and need to be analyzed in order to provide services and statistics to end-users.

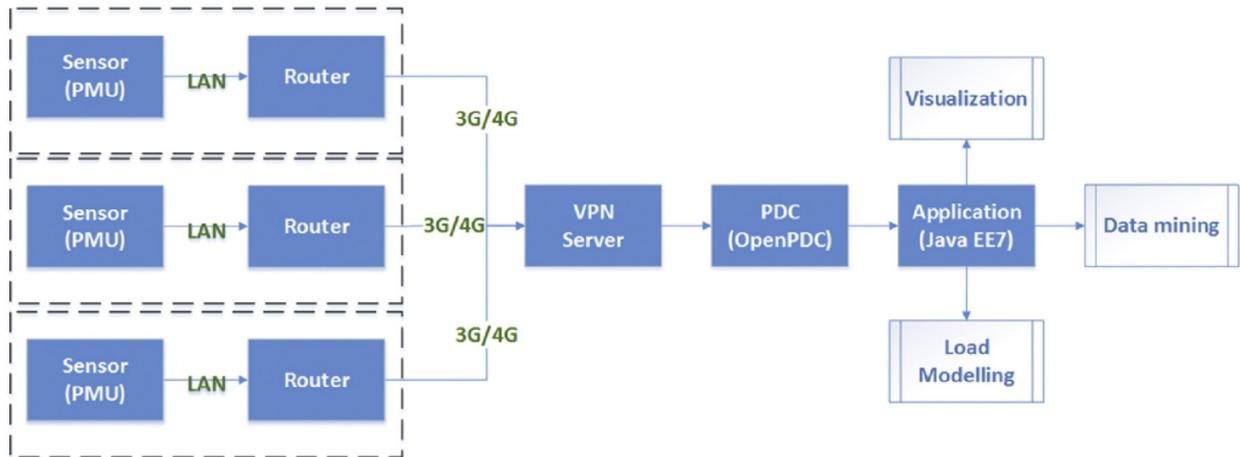


Figure 2: Conceptual view

As shown in Figure 2 above the communication infrastructure proposed in their paper consists of a layer where different sensors are connected to routers through a local LAN network. Then routers are connected to the PDC server using a VPN to increase the security of the connection. The PDC server is then connected to the application layer that is responsible for visualizing and mining the data. Sensors can be connected to the internet with a 3G and 4G wireless connection if there is no possibility to use cables in the position of the sensor.

It is possible to use this infrastructure in the SA of our platform.

2.2.1.2. A virtual sensor system for end-user-generated, real-time environmental data products

Hill et al. (2011) describes a method to retrieve data from sensors. The prototype that they developed receives real-time data from sensors, enhances it and then shows it through a Web interface. These data streams are provided with metadata including the position of the sensor. This can be helpful for our platform because it shows a way to structure real time data retrieval of our platform. We will need to connect real time data retrieval sensors to the platform like thermostats and weather stations and this paper will be helpful for this purpose.

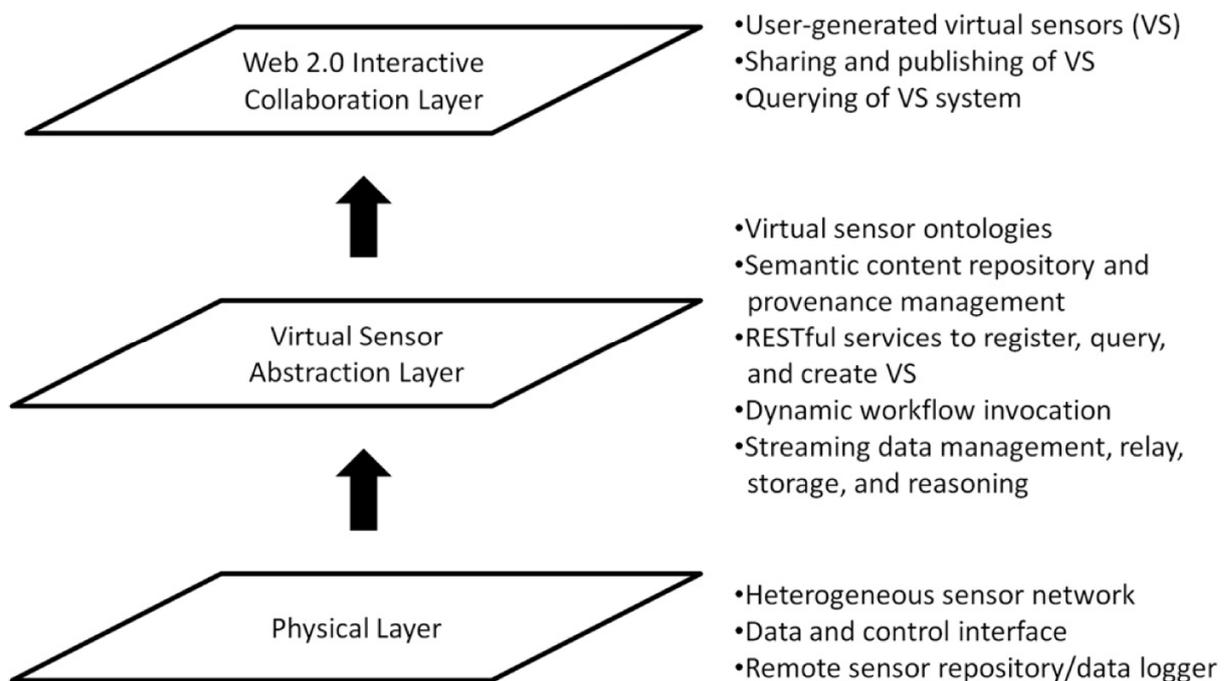


Figure 3: Architecture of the virtual sensor system

The architecture of the platform as proposed in this paper is divided in three layers. The first layer is where the remote sensor is located. It includes the sensor and the network required to connect the sensor to the internet. The authors assume that the sensors are accessible through FTP and HTTP protocols. Data is stored in the sensors and needs to be retrieved with FTP or HTTP protocols. The repositories of the public sensors are distributed in the world and they provide a list of sensors that are publicly accessible with their specification and IP address.

The middle layer is made to facilitate the retrieval, enhancement and publication of the sensor's data. This is done by implementing a "workflow-based processing (Cyberintegrator) over a semantic content repository abstraction (Tupelo), augmented by a temporal stream management layer", Hill et al. (2011). The result of this layer is a set of file-like datasets with additional information such as descriptive information and temporal relations with other datasets. It also provides a RDF statement (see section 2.2.2.2) in already defined vocabularies. These data are then accessible, for instance, through SPARQL queries.

The third layer is the Web interface where end-users are able to see the data that comes from the workflows of layer two. Visualization capabilities are provided with the Web interface such as the visualization of time-series of data that is sent through streaming by a graph.

2.2.1.3. How to Ensure the Economic Viability of an Open Data Platform

Duval & Brasse (2014) made an analysis of different business models for open data-providers. This research can be helpful for our platform because we will provide open data to external parties and customers that we will retrieve from external data sources and from the data inserted by end-users in the system. Three different business models are discussed in the paper: the free model, the freemium model and the premium model.

The first is the free model where all the income comes from advertisement (see section 3.6.1), end-users do not pay any fees to access the data and the only revenues are from advertisements that are shown to end-users.

The second business model is the freemium model where end-users are able to subscribe (and have to pay a fee) to get more functionalities of the platform. We will use this model for our platform by providing access to specific services and use cases after the end-users subscribed to a pay account (see section 3.6.2).

The third business model is the premium model where all end-users have to pay a fee to have access to the platform. This business model will be implemented for a few data sources that are requested by specific external companies. These external companies will pay a fee for every query made to our data sources using a premium business model.

The authors found that different features can be accessed by different end-users using different business models. They divided end-users in three categories: free (user is not registered, anonymous end-users), signed in (end-users that created a free account) and subscriber (end-users that created a pay account). In the free business model the free end-users have access to simple research only and they have limited access to metadata. For signed in end-users all the functionalities of the platform are provided (such as advanced search, access to API, visualization tools, social features and multilingual interface).

In the second model (the freemium model) the free users have the same rights as the free end-users in the free model but with a more limited access to data. The signed in end-users have limitations on metadata and a loss of API access. The subscriber can access the full functionalities of the system (such as API, automatic translation and full metadata).

In the premium model just the subscriber type of end-users exist, because the other types of end-users are not paying end-users and so they cannot fit in a premium business model that requires that all the end-users pay a fee to use the platform. For the subscribers all the functionalities of the system are accessible.

2.2.1.4. An open platform for personal health record apps with platform-level privacy protection

Van Gorp et al (2014) discuss a system to share health data with external applications and still maintain the privacy of that data. In the health care industry, data is difficult to share with external applications and parties because it is privacy sensitive. People do not want their privacy sensitive data to be shared with parties that in turn can share this data with other commercial companies.

The authors of this papers provided a solution to this problem. This is relevant for our platform because we

have the same problem. People do not want their data to be shared and exchanged with commercial companies and they want to remain in control of their data. Using the technologies contained in this paper we will be able to provide external parties with the data of end-users and, at the same time, maintain control over this data.

Figure 4 shows the architecture of the platform as presented in the paper, called MyPHRMachines.

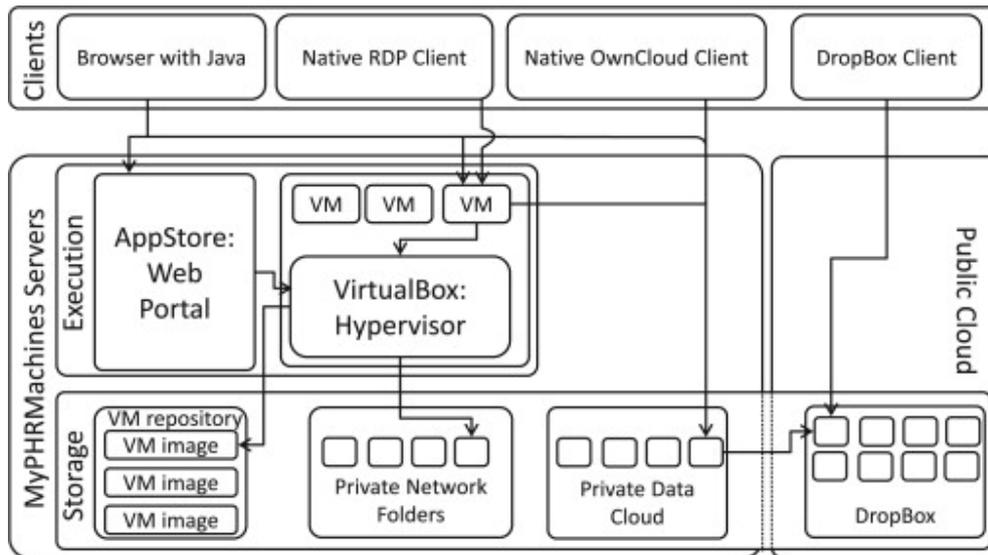


Figure 4: MyPHRMachines architecture

The architecture shows two main layers, the client layer and the server layer. The client layer includes all the methods that an end-user can use to access his data and the apps contained in the system. End-users can access the apps through a browser with Java or by a native RDP client. End-users can also access their private data through the native OwnCloud client. The OwnCloud component provides secure storage and access of the data of end-users.

In the server layer all the processing, execution and storing parts are found. The execution part, as can be seen in figure 4, is where the AppStore Web portal where the list of apps that are available is stored. The Hypervisor is a layer that includes the virtual machines that run the external applications. Every time a virtual machine is executed the end-user can access it through the Java Web interface or through a RDP client.

In the storage layer the Virtual Machines repository can be found, wherein all the images of the virtual machines are stored. Developers can upload a new version of their services and apps in this layer. Another part of the storage layer are the private network folders and the private cloud data where all the data of end-users are stored. Private network folders have better security of data because they are not connected directly to the internet. This way the virtual machines cannot exchange any information with external parties or servers because they do not have any internet connection with the outside world.

In the public cloud layer there are files of end-users that are not inside this system. End-users can connect their public cloud (defined as a cloud service with a personal account of the end-user) with the system data cloud by connecting it with the Private Data layer.

Using this software architecture the data is protected and a malicious person that wants to make a malicious app that steals information of end-users cannot do it. This is because every virtual machine is not directly connected to the internet and does not have a direct network connection. However, it is still possible to share data with other parties. If a person wants to share his data with external end-users he can provide access directly to his data through the OwnCloud client or provide access to his apps to an external person.

This platform is focused on privacy. It satisfies two main requirements: maintaining the privacy of the data and, at the same time, providing a way to use and share the data with external applications and parties.

Developers can develop their apps locally and then upload the new version of the apps in the VM repository. Every VM session is stateless. This means that every VM execution terminates when the end-user finishes to use it and then all data written on the local disk on a VM will be deleted. So developers do not need to care about the migration of running VM session. All the data that a VM wants to store must be stored in the private network folder layer, preventing any leaks of information to external parties and companies.

It is possible to use parts of this SA in our platform.

2.2.1.5. Integrating Public Datasets Using Linked Data: Challenges and Design Principles

Omitola et al. (2010) describe a method to convert datasets of different locations and formats into linked data.

The knowledge described in this article can be interesting for our platform because we will need to import data from different sources and then convert them into linked data to be able to offer them to external service providers and customers.

The authors could not take advantage of the automatic resource discovery process because the data that they wanted to import was not in the linked data format. So they sourced their datasets by looking source by source and site by site. The data that was found was in many different formats such as PDF, HTML and XLS.

It was decided to use RDF (see section 2.2.2.2) as the normal dataset form and the turtle representation of RFD triples. They chose the 4store system to manage and store RDF triples. Furthermore, the Exhibit publishing framework (see section 2.2.2.5) was used to develop the interface for the consumption of data.

The paper includes the complete step-by step process that was followed to convert the different types of data to linked-data. First of all they modelled the datasets and found links with existing well-known vocabularies to re-use terms. Then the datasets were converted to the RDF format. The problem of this step was the little semantic description of the data involved. In some of the public data there was no explanation of the relationship with other data. Another problem was that it was difficult to integrate and link data from disparate data sources.

They developed scripts that automatically converted the spreadsheet data into RDF triples. They also used the Jena Semantic Web Framework (see section 2.2.2.6) to convert types of data. Then they aligned the different datasets by aligning the different identification of relation between the different relevant entities of the datasets. Finally they developed the interface for the linked data consumption.

2.2.2. Existing technology solutions

In this section we will analyze and describe solutions using existing technology that are related to our platform. We found these technologies by searching on the internet for the main technologies related to our platform. The main search engine that we used was Google.

In Figure 5 below all the different existing solutions that are relevant to our platform are shown.

The Linked data section shows different technologies such as RDF, SPARQL and Linked data fragment technologies.

In the Linked data framework section there are the two frameworks discussed in this section: the Exhibit framework and the Apache Jena framework.

In the Identity access framework section two different identity frameworks are described that we may implement in our platform. Both the Linked data framework and the Identity access framework technology may be implemented in the software architecture of our platform.

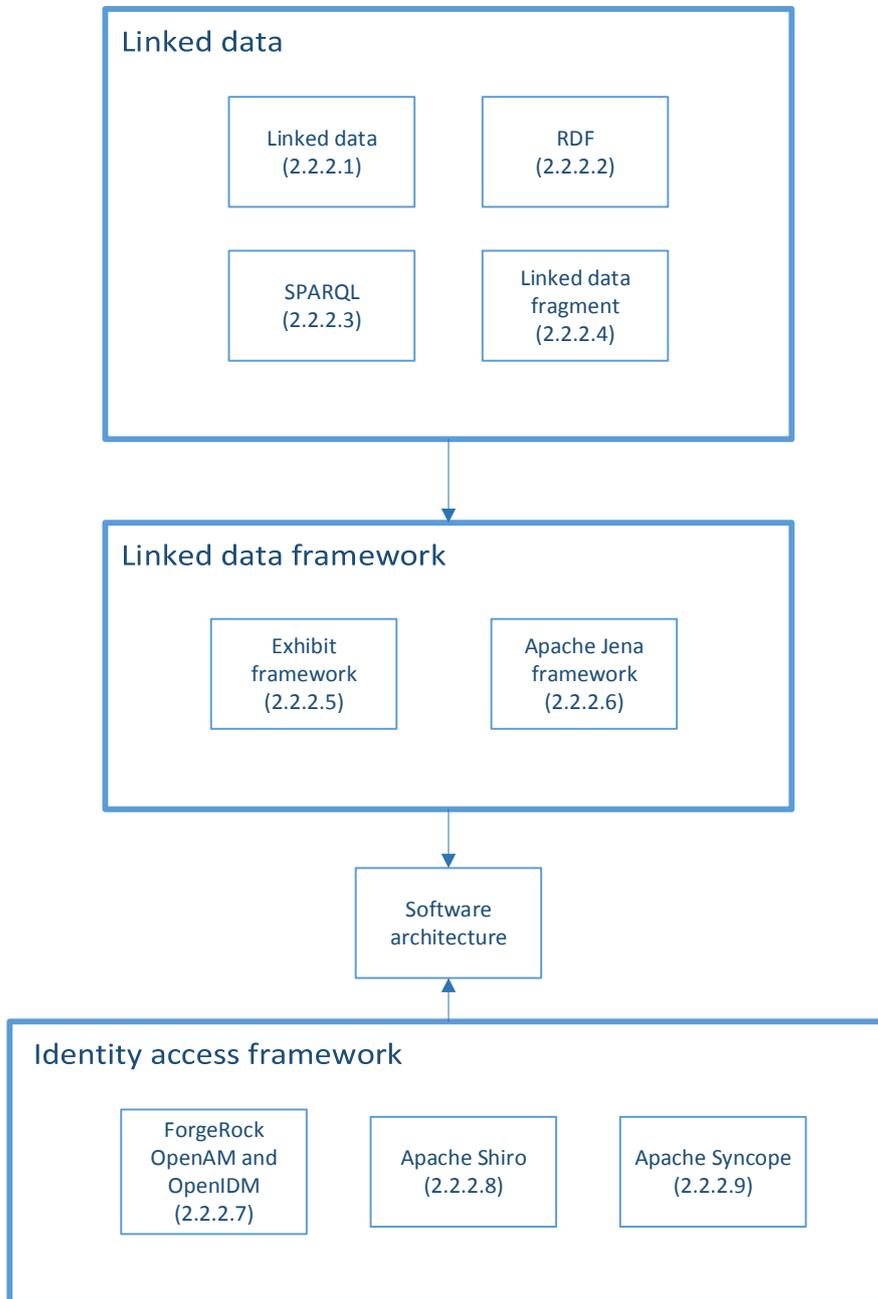


Figure 5: Existing technology solution schema

2.2.2.1. *Linked data*

Linked data refers to using the Web to make typed links between data from different sources. Technically speaking, Linked data refers to data that is readable from a machine and that is published on the Web (Bizer et al., 2009)

On the Web, hypertexts are programmed using HTML documents. HTML allows the programmer to connect different documents together by using hyperlinks. In Linked Data there is RDF (Resource Description Framework). The RDF defines how different objects link together. In the next section a more accurate definition of RDF is provided. Also in Linked data every object or concept is identified by an URI , this is similar to the Web (Berners-Lee, 2006). An URI (uniform reference identifier) is a string of characters that refers to a resource.

Linked data, following Tim Berners-Lee, should follow rules (Berners-Lee, 2006). The following rules are the original rules of Tim Berners-Lee:

1. Use URIs as names for things.
2. Use HTTP URIs so that end-users can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).
4. Include links to other URIs. so that they can discover more things.

Figure 6: Tim Berners-Lee's rules for Linked data

Open data can be viewed also as a browsable graph. With a browsable graph a person can see in a graphical way the connections of every node. When one looks for a node in the RDF graph, the server returns all the information about the arcs in and out of that node. Thus it is possible to know all the statements where the node is a subject or an object and the description of all the blank nodes attached to the node. (Berners-Lee, 2006)

Tim Berners-Lee also made an analysis of the different ratings of Linked data. These levels can be used for Linked Open Data and for Linked Data. The only difference is that Linked Open Data provides data that is accessible under an open license.

Figure 7 shows the ratings, starting from the lowest (1) to the highest (5) (Berners-Lee, 2006).

1. Data is available on the web (in whatever format).
2. Data is available as machine-readable structured data.
3. Data is available as machine-readable structured data plus non-proprietary format.
4. Data is available as machine-readable structured data, non-proprietary format plus uses open standards from W3C (RDF and SPARQL) to identify things.
5. Data is available as machine-readable structured data, non-proprietary format, uses open standards from W3C (RDF and SPARQL) to identify things plus data is linked to other users' data to provide context.

Figure 7: Berners-Lee's ratings of linked data

2.2.2.2. RDF

The RDF (Resource description framework) is a data model for publishing structured data on the Web. (Heath & Bizer, 2011). The RDF data model can represent information with a node and an arc labeled directed graphs. The objective of RDF is to moderate between other data models used on the Web (Heath & Bizer, 2011).

With RDF, the description of a node is represented with triples. The different parts of each triple are called subject, predicate and object. A triple can be seen as a basic sentence structured, for instance:

"Diego is a student"

Becomes:

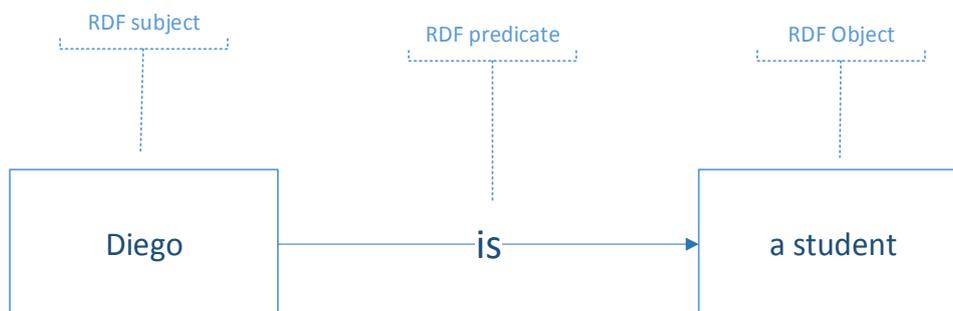


Figure 8: RDF triple structure

Each part of a triple can be a URI that connects to an external object. Each part of a triple can also be a literal value, like a number, date or string. The predicate indicates which relationship exists between the subject and the object. A predicate can be also a URI and can be taken from vocabularies, those are collections of URIs that contain predicates about a specific domain.

The following example converts a sentence to a RDF graph.

Suppose that we have the sentence:

"Diego is a student and works as a business consultant"

The **subject** is "Diego".

The **objects** are "student" and "business consultant".

The **predicates** are "is" and "works as".

Then the according graph will be:

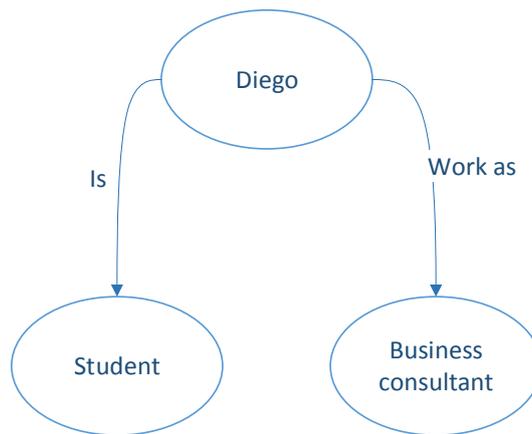


Figure 9: RDF graph for a sentence with two objects

2.2.2.3. SPARQL

The SPARQL language is used for querying RDF data. It enables the user to retrieve and manipulate data stored in the RDF format. The current last version of the language is the 1.1 version that was released in March 2013. ("SPARQL 1.1 Overview,")

In the following example taken from the W3C SPARQL Working Group ("SPARQL 1.1 Query Language,") we will make a SPARQL query to retrieve the title of a book.

Suppose to have this data:

```
<http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial" .
```

Figure 10: RDF data

And to have this SPARQL query:

```
SELECT ?title
WHERE
{
  <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title .
}
```

Figure 11: SPARQL query

The select clause identifies the data that will appear in the results, and the where clause provides the RDF data that needs to match.

The result of this query will be:

Title
"SPARQL Tutorial"

Figure 12: SPARQL query result

2.2.2.4. *Linked data fragments*

Linked data fragments (LDF) (<http://linkeddatafragments.org/>) is a method that needs minimal server effort during the solving of SPARQL queries and that allows to switch the effort of the queries to the client. This method was introduced because public SPARQL endpoints that reply to SPARQL queries are often unavailable due to the high numbers of queries that they need to reply to.

One solution to this problem was to create local endpoints using data dumps, but the data of the local endpoint was not up to date and the client could not query every dataset but just the local data of the local endpoint. To solve this problem the LDF was invented, making it possible to switch the effort from the server to the client.

The Linked Data Fragment of a Linked Data dataset consists of the triples of the dataset that match a specific selector. A LDF server consists of a HTTP server that offers the service of LDF with one or more of its datasets. The LDF technology can be installed in any HTTP server that wants to offer this service.

The next diagrams compares the data fragments solution with the normal SPARQL data retrieving and with the local endpoint data dumps.

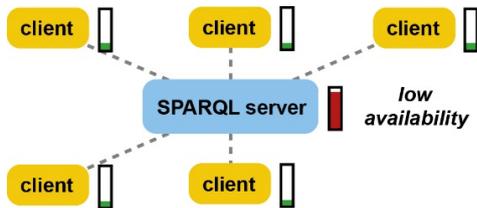


Figure 13: SPARQL normal querying



Figure 14: Local endpoint data dumps

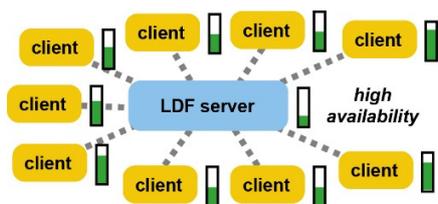


Figure 15: LDF technology

As is shown in the above figures, the normal SPARQL querying relies on the server for the querying effort. The local endpoint data dumps relies on the local server but the data may not be updated. With the LDF technology the server needs to put less effort in the query and the clients do a big part of the effort.

2.2.2.5. The Exhibit publishing framework

Exhibit (<http://www.simile-widgets.org/exhibit/>) is a publishing framework to integrate data in web pages. It enables incorporating tabular data into webpages using simple tags to incorporate in the HTML code. Then the framework automatically imports the linked data from the tabular data sources. To attach the data a developer just needs to specify the data columns that contain coordinates, dates or attributes and then they will be visualized in the web interface.

The framework does not need installation because it is an open source JavaScript library that relies on the client browser. So it can be integrated in already existing CMS in a simple way.

This framework may be used for our platform.

2.2.2.6. Apache Jena

Apache Jena (<https://jena.apache.org/>) is an open source framework to build linked data and semantic web applications. The framework is composed by different APIs that process RDF data.

This framework permits users to develop linked data applications in a faster a simpler way by offering APIs that can be used in the development of the software architecture of a platform.

It offers TDB, which is a component for RDF storage and query. It can be accessed and managed with scripts and by using the Jena API. It can be used between multiple applications by the Fuseky component that a SPARQL server provides.

The ARQ is the query engine of Jena. It supports the SPARQL RDF query language and many other functions such as free text search, access of the SPARQL algebra, custom filter functions, property functions for custom processing of semantic relationships, aggregation, GROUP BY and assignment as SPARQL extensions. ("ARQ - A SPARQL Processor for Jena,")

It implements an advanced text function that allows making free text searches within SPARQL queries. ("Text searches with SPARQL,")

We will be able to use this software in the SA of our platform.

2.2.2.7. ForgeRock OpenAM and OpenIDM

OpenAm (<https://www.forgerock.com/>) is an open source identity provider that manages the identity of end-users. It offers a modular platform where end-users can store their login details, their policies and their details. It provides easy integration with external services and web platforms.

With this platform end-users are able to securely manage the access to their services. It provides a personalized experience on any digital channel, such as mobile, cars, home appliances or anything connected with the internet. The authentication is based on context access that includes location, IP address and time of the day. Its policy engine tool permits managing access to the different services that an end-user can have access to. It eliminates the need to use multiple passwords to use different services by providing a unique identity manager. It allows a rapid customer adoption by permitting quick integration options such as the "sign up and log in with Facebook" function.

OpenIDM provides a complete identity management platform that supports following the customer's login in every device and service. It permits an end-user provisioning based on existing rules and permissions. End-users can manage registration and password reset themselves without the help of an external system administrator. It also allows to connect the identity of end-users with cloud services like Google and Office365.

2.2.2.8. Apache Shiro

Apache Shiro (<http://shiro.apache.org/index.html>) is a Java security framework that has the following functions: authentication of end-users, authorization, cryptography and session management. Apache Shiro provides easy to use APIs to provide a secure authentication to a web application.

Using Shiro, managing security rules for every end-users registered in the system becomes simple . It permits to react to events during the different phases of end-user's authentication. It also allows to

aggregate more data sources of end-user security data and to present this data with a single composite view. It also offers the “remember me” function for end-user access without login.

Data stored with Shiro can be cryptographed to increase its security. It supports the “run as” function that allows an administrator to login with the credentials of an end-user This can be useful in administrative scenarios.

We may be able to use this software in the SA of our platform.

2.2.2.9. Apache Syncope

Apache Syncope (<http://syncope.apache.org/>) is an open source system that can be used to manage digital identities in enterprise environments. It provides different functions such as account and role administration, domain integration with external resources and a web-based interface for the administration console.

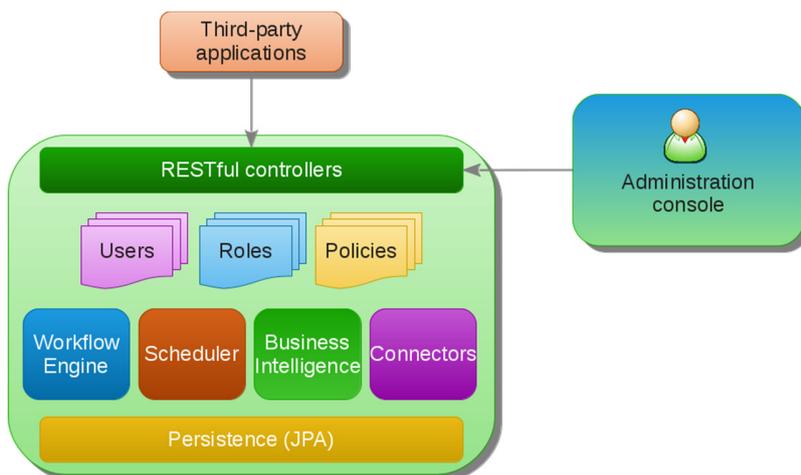


Figure 16: Software architecture of Apache Syncope

The figure 16 shows the SA of Apache Syncope. The core part is formed by the web application that implements the IdM features. The central component manages the data flow through the system. It handles RESTful call, processes data and propagates it to and from external data sources.

The administration console allows managing the identity of end-users on the web interface.

The connection layer connects the IdM part with the other parts of the SA of our platform.

We may be able to use this software in the SA of our platform.

2.2.3. Products and services on the market

In this section organizations that are already developing or researching a data-collecting platform are analyzed and discussed. In addition to our own search results several experts gave us useful information about organizations that are of interest to our project.

For each company we state its name, web address, current status of development, its description and its current provided functions. This analysis was made in January and February 2015 so the information provided in this section is fairly up to date.

2.2.3.1. My data store

Company

Telecom Italia Group

Website

<http://jol.telecomitalia.com/jolskil/personal-data-store/>

Status:

Prototype testing with MTL community members

Description:

Telecom Italia is creating an aggregator of Linked Data that will permit end-users to decide their privacy setting for each data source.

Functions:

- Collecting of personal information: end-users can collect data regarding themselves. End-users can decide to collect particular data just during specific time-slots.
- Sharing of information: end-users are able to share their personal information with third parties. End-users are able to set different disclosure levels for each piece of information.
- Deleting of information: end-users are able to delete information that they do not want to have anymore.
- Interface for aggregating information: information is aggregated into a single interface that permits end-users to have a complete view of their information. This is done to increase the end-users' consciousness of daily behaviors.
- Compare data with other end-users: end-users are able to compare the data that they share with data shared by other end-users.

2.2.3.2. The HAT project

Company

HAT Research Team

Website

<http://hubofallthings.com/>

Status:

First beta launch in June 2015 in UK and Singapore. They are collecting marketing data through the mandatory survey to register to the service on the website.

Description:

The HAT project team is developing a platform to collect Linked Data. Data can be shared with external applications. People will be able to set privacy settings for their data and to analyze their data. External companies will be able to set-up their HAT servers and to store personal data of end-users inside them. The HAT project will validate external apps, maintain a catalog of the external HAT servers, be an intermediate gateway for inter-platform payments, manage the APIs, manage the universal HAT unique ID database, install and update the external HAT servers and give advice about business and revenue models to external companies.

Functions:

- Collecting of personal data: end-users can collect and store data that comes from different sensors, devices, internet connected objects and services. End-users can store this data in a HAT server.
- External applications and services: data can be shared with external applications and services that will use and analyze it.
- End-users will be able to see in one integrated interface the apps that they can use with a specific HAT server.

2.2.3.3. *CommonSense*

Company

Sense Observation Systems

Website

<http://www.sense-os.nl/home>

Status:

The beta version of the platform and of the mobile app have already been developed. There is the possibility to have free and open access to a beta account to try the platform.

Description:

CommonSense provides a platform where data from different input sources is stored. Data can be retrieved from the mobile app developed by the company and from a Fitbit device. The system can retrieve and store the following types of data:

- Sleep information
- Time activity information
- Location information

End-users are able to share their data with other end-users.

Functions:

- Collecting information: end-users can collect information using the mobile app, Twitter and a Fitbit device.
- Presentation of information: end-users can see their information on the dashboard.

- Sharing of information: end-users can share their information with other end-users and see information shared by other end-users.
- Open API: external companies and developers can build apps for the platform using the CommonSense's APIs.

2.2.3.4. *Tippiq*

Company

Tippiq (Alliander)

Website

<https://www.tippiq.nl>

Status:

The first release is public and everyone can access and register to it.

Description:

Tippiq provides a platform to find information about events, projects and initiatives of an area.

End-users can search with their postal code to information about their area and register to the system.

Functions:

- Presentation of information: end-users can see information about their surrounding on the dashboard.
- Inclusion of information: end-users can provide information about their area to the system that can be shared with other end-users of that area.
- Privacy of information: end-users are able to decide with whom to share their information.
- End-users are able to set up an initiative in their neighborhood and share it with other Tippiq end-users.

2.2.3.5. *Huislink*

Company

Huislink

Website

<https://www.huislink.nl/>

Status:

The first beta release is public and everyone can request an account.

Description:

Huislink provides a platform to store all information regarding a house. For instance photos, characteristics and general information about a house. This data can be shared with other end-users that can search the houses database to find a house to buy or rent.

Functions:

- Storing of information: end-users can store information regarding their house on the platform.
- Sharing of information: end-users can share their information with other end-users.
- Searching of information: end-users can search for information regarding houses in the system and they are able to contact the owner of the house to rent or buy the house.
- Messaging system: end-users are able to add other end-users to their profile and to exchange messages with them.

2.2.4. Current problems and limitations of existing products and services

Currently a solution that provides a platform that satisfies all our stakeholders' needs and requirements (as discussed in section 2.1.2) does not exist. There is a gap between what we need and what has already been developed.

In the next table a comparison is made between different solutions available with the requirements that are described in section 2.1.3. The requirements list formed the bases of this comparison and we compared it with the functions offered by each existing solution. For a better description of each requirement please refer to section 2.1.3. A blank spot in the table means that a requirement is not implemented for the specific solution.

Requirements/existing solutions	My data store	The HAT project	CommonSense	Tippiq	Huislink
Different levels of authentication for end-users					
The possibility to import external data	Present	Present	Present	Present	Present
The possibility to store data inserted by end-users					Present
External services development		Present	Present		
Dashboard	Present just for the aggregation of information	Present without the newsfeed			
News feed					
End-users receive alerts					
Social function					Present
End-user function to share their data with external websites with a wallet function					
Tablet and mobile browsing			Present		

Table 9: Requirements and technologies comparison

Some of the requirements are met but in none of the platforms analyzed all the requirements are met by one single platform. Moreover, several of the mentioned platforms do not focus on data specifically regarding houses but they focus on different types of data.

3. The business case

3.1. External view

In the external view (see Figure 17: External view) there are three different main entities that will link to our platform. The first group consists of the service providers that will provide the services in the platform (the complete description is in section 2.1.2.3). The second group is made up by the data-providers, they will provide the data sources that will connect to the platform (the complete description is in section 2.1.2.2). The third group are the end-users. They will use the system, its services, functions and data sources (the complete description is in section 0).

A service provider can also be a data provider (and vice versa) because every company or organization can both share data with our platform and offer services that link to these data. An end-user can also be a data provider by entering his personal data in the platform.

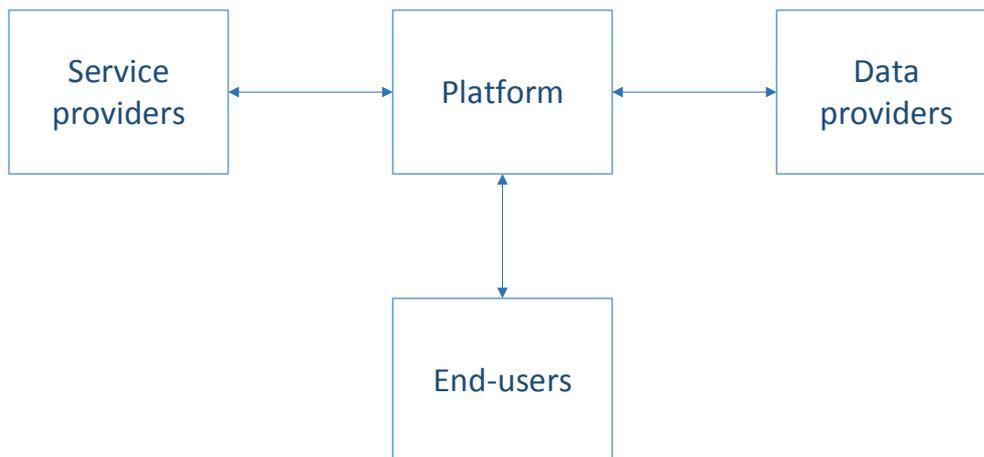


Figure 17: External view

3.2. End-user perspective services

In this section the services and functions that will be offered to the end-users of the platform are presented. These functions are developed following the requirements described in section 2.1.3.

3.2.1. Different levels of end-user authentication and certification

Every service will need a different level of certification of end-users to be used. Privacy sensitive services will need a stronger authentication security from the end-user. The different levels of authentication that we will implement in the platform are described in section 2.1.2.1.

To implement the authentication in the software architecture of our platform we may use the two frameworks described in sections 2.2.2.7, 2.2.2.8 and 2.2.2.9.

The end-user will be able to upgrade his level of identity certification when he wants to access a service that needs a new level of identity certification.

Some data sources, to be displayed from an end-user, also need to have a determined identity certification from the end-user.

3.2.2. Import of external data

End-users will be able to import external data from external sources and to visualize and manage it through use cases. The data that they will be able to import and use will depend on the identity certification that they have and on the amount of data that can be connected to them.

After the first login the end-user will be able to see just the basic data linked to his profile. Then he will be able to add more data to the interface by selecting it from the data list. In the data list there will be a list of all the available data sources that can be linked to this specific end-user.

The data that can be imported in the platform will be decided by us. We will carefully analyze the data that will be available for end-users by selecting data just from trusted sources.

Data will consist of two types: static and dynamic. The static data will include all the data that is not updated in real time but that is updated sometimes. The dynamic will data include all the data that is updated in real time. This type of data will include all the real time sensor data sources.

Data sources will also be used by the external services described in section 3.3 to provide useful functions to end-users.

3.2.3. Storage of data inserted by the end-user

End-users will be able to insert data regarding their personal details in the platform. This data will be stored in various storage devices that are discussed in section 0.

This data can be used by the services discussed in section 3.3 and shared with other end-users.

3.2.4. Sharing data

Every end-user will be able to share their data with external companies and end-users. Before sharing a particular type of data the end-user will need to accept the sharing permission of it and he can revoke the sharing permission when he wants.

The data that can be shared can be of two types: the data imported from external sources and the data inserted locally by the end-user. End-users will be able to share both types of data. Regarding the data imported from external sources, end-users will be able to share just the data linked to them.

Data can also be shared with services. Every time a service needs a new type of data or during the first use of it, the end-user will be asked to give permission to use his data.

3.2.5. External services integration

External companies and parties will be able to develop external services to integrate in the platform.

These services (called "use cases" in section 3.3) will retrieve data of end-users from external sources and from data that the end-user inserted in the system. These services then will elaborate that data and they will provide to the end-users functions that will come from the output of this elaboration.

The complete list of the services that may be implemented in the platform are in section 3.3.

Services will be developed by us and by external companies that want to offer useful functions to end-users. External companies will be able to collect and analyze data of end-users (after having got the permission from them) in a simple way. With the current state of technology companies need to retrieve data of end-users from a lot of different sources. With our platform they will be able to retrieve this data just from one source and they will be able to ask for permission just once to use the data of the end-user.

The list of the services will be shown to the end-user in the interface of the platform and will be divided by categories. The access to the different services will be regulated by the authentication level of the end-user. If a service needs a high authentication level of an end-user the platform will ask for it.

A service can show information in the interface of our platform. There will be the possibility for a service to just show information in the user interface without taking any input from the end-user.

3.2.6. Dashboard

The dashboard will be the user interface that end-users will use. On the dashboard the following things will be shown: data of the end-user (both from external sources and from the end-user), the list of available services, the newsfeed and the other functions of the platform. All these parts of the dashboard will be divided into categories.

3.2.7. Newsfeed

On the dashboard end-users will be able to see the newsfeed. The newsfeed consists in short messages (that can include photos, videos and links) that will be shown on the dashboard. These messages will be generated by external services, by data sources or by other parts of the platform.

If there are new types of important data available for the end-user, or important new services available, a link to them will be shown in the newsfeed. If a person that is in contact with the end-user shared important information with the neighbors it will be possible to show this information also in the newsfeed.

3.2.8. Alert function

The alert function will consist of alert messages sent to the end-user with different communication interfaces. They can be sent through a mobile app, SMS, e-mail or automatic calls.

Users will be able to reply to the messages that they will receive and we will be able to collect their replies in our system. The reply types that they give can be text information or a multi-choice reply.

These messages will be generated by the platform and they will come from the services. When an external service wants to send a message to a specific end-user the message will be sent to our platform and then be directed to the end-user. The platform will also collect the replies of end-users and then send them to the different services.

End-users will be able to decide which services will be enabled to send alerts to them. This function is different from a normal messaging function because it will just be used for important information messages of the services.

3.2.9. Social function

End-users will be able to make a profile in their account and to connect with other end-users of the platform. They will be able to share data with these end-users and to see their profiles. End-users can exchange messages with other end-users in the system and form groups with other end-users to share information inside the group.

Services will be able to connect with these groups and use the information of the group to offer useful functions to end-users.

People will be able to share photos, videos, links and other types of data with other group members.

3.2.10. Wallet function

People will be able to share their data with external websites or services through the wallet function.

The wallet function consists of an open API that external websites will be able to use to retrieve data about the end-user. When the end-user will access an external website that requires information, he will be able

to share the information stored in our platform directly within the website by just giving permission to share the data.

Giving permission to share data with external websites will be always managed by the end-user and they will be able to interrupt the sharing of their data at any time. End-users will also be able to decide which data they want to share with each specific website.

3.3. Use cases summary

In this section there is a list and a brief description of the possible use cases that will be implemented in the platform. This list will be used to decide which are the use cases that we will focus on in the next section. These use cases were determined by analyzing the needs of the different parties involved in the project and by analyzing how to satisfy these needs with the functions that we will be able to provide through the platform.

The next two tables show a summary of the use cases that were examined. The next table is a preliminary of table 11 and explains what is meant by the labels in table 11. In the "possible values" column there are explained just the possible values that are not self-explanatory in Table 11.

Name of the column	Description	Possible values	Overlapping between values
ID	Sequential number that identifies the use case	N/A	N/A
Name	Name that identifies the use case	N/A	N/A
Description	A brief description of the use case	N/A	N/A
Target group	The target group of end-users that will use the use case	Inhabitants of the house: end-users that live in the house Senior end-users: end-users that are older and have problems connected with their age	The inhabitants of the house can include other target groups such as the senior end-users.
Category	The category of the use case	Simplify life: it simplifies the life of end-users Save money: end-users will be able to save money with it Social: end-users will be able to socialize with other end-users Safety: it increases the safety of end-users Care: it increases the health of end-users Government: it offers a government service	There is some overlap of the different categories but the category indicated refers to the main category of the use case

Table 10: Use cases list explanation

ID	Name	Description	Target group	Category
1	Appliance information	End-users will be able to see information about the appliances that they have in their house in the portal. This information can be static (for instance manuals) or dynamic (for instance real time data from the appliance). End-users will also be able to share their manuals and details of their appliances with other end-users in the community	Inhabitants of the house	Simplify life
2	Manual detection of energy counters	End-users will be able to put information about energy counters in their house directly in the platform. This will be possible for the water, electricity and gas counter. End-users will be able to take a photo of the counter and an OCR program will detect the correct value of it.	People that do not have an automatic counter information sharing	Simplify life
3	Wi-Fi sharing	End-users will be able to share their wireless connection with their neighbors.	Inhabitants of the house	Save money
4	Meet your neighbors	End-users will be able to see their neighbors' list and to send messages to them. They will be able to share information from the portal with them. If an end-user does not want to be contacted this can be indicated in the settings interface.	Inhabitants of the house	Social
5	Fire alarm sharing	The fire alarm will be connected to the neighbors' system and to friends of the end-user. They will be alerted if the alarm goes. If nobody replies to the signal it is sent to the firefighters. The end-user will be able to insert the details of the end-users that will be contacted if the alarm turns on in the platform interface.	Inhabitants of the house	Safety

6	Security alarm sharing	The security alarm will be connected to the neighbors' system and to the friends of the end-user. They will be alerted if the alarm goes. If nobody replies to the signal the end-user can choose to hire an external alarm company that will receive the signal and manage it. The end-user will be able to insert the details of the end-users that will be contacted if the alarm goes in the platform interface.	Inhabitants of the house	Safety
7	Health alarm sharing	The health alarm will be connected to the neighbors' system and to the friends of the end-user. They will be alerted if the alarm goes. If nobody replies to the signal the end-user can choose to hire an external health company that will receive and manage it. The health alarm consists of a machine with a button that is put around the neck of the end-user. If the end-user presses the button the alarm is activated. There is the possibility to install sensors that can detect if a person is in a specific room. The end-user will be able to insert the details of the end-users that will be contacted if the alarm goes in the platform interface.	Old end-users, end-users taking care of their old parents	Care
8	Doorbell managing	People will be able to see in the portal if anybody is ringing the doorbell. They will be able to see the image of the person using the camera of the doorbell and they will be able to speak with him. End-users will be able to open the door using the portal.	Inhabitants of the house	Safety
9	Thermostat remote controlling	End-users will be able to remote control the thermostats of the house. If there are different thermostats for different parts of the house they will be able to program the temperature of the different parts of the house from the portal.	Inhabitants of the house	Save money
10	Photo retrieval and upload	End-users will be able to share their photos with a photo archive. The photos can be old photos of the house, the town or any other type.	Inhabitants of the house	Government

11	Chamber of commerce company information	The Dutch chamber of commerce will provide information about registered companies near the address of the end-user. They will provide basic information about the companies for free (name, address, company number, year of foundation). If the end-user wants to see the complete information about the company, he will be able to buy them from the platform interface. The information about new companies will also be shown in the newsfeed. End-users will be able to send to the chamber of commerce a warning if there is any incorrect information about a company on the platform.	Inhabitants of the house	Simplify life
12	Equipment sharing	End-users will be able to share a list of equipment that they have with their neighbors. Neighbors will be able to see this list and then contact the owner to ask if it is possible to borrow the equipment that they need. This use case can be combined with the appliance information service.	Inhabitants of the house	Save money
13	House value estimator and mortgage calculator	End-users will be able to see their house value and calculate their mortgage. End-users will be able to share the house's value with banks to have a possible mortgage calculated.	Owner of houses	Save money
14	Solar panel estimator	End-users will be able to see how much money they will save by installing solar panels. They will be able to compare different offers of solar panels from the portal.	Inhabitants of the house	Save money
15	Collective purchase assistant	End-users will be able to make collective purchases with their neighbors. For instance supplies to paint the house.	Inhabitants of the house	Save money

16	Delivery assistant	<p>End-users will be able to indicate if they are at home to collect a delivery. They will indicate it with a mobile app that will check if the end-user is at home using the internal mobile GPS.</p> <p>End-users will be able to make a list of the neighbors that will receive the delivery if they are not home.</p> <p>If the end-user is not home the delivery company will be able to see the list of neighbors that will receive the delivery and if they are at home or not.</p>	Inhabitants of the house	Save money
17	Collective energy management	<p>End-users will be able to manage electricity usage and costs with their neighbors. They will be able to share information between them.</p>	Inhabitants of the house	Government
18	Interior design sharing	<p>End-users will be able to share photos of the interior design of their house with other end-users. End-users will be able to see the interior design ideas of other end-users with the same room dimensions and structure.</p>	Inhabitants of the house	Simplify life
19	Renovation assistant	<p>End-users will be able to see where to buy products that they need to renovate their house.</p>	People that want to renovate their house	Simplify life
20	Energy consumption assistant	<p>End-users will be able to control the energy consumption of the house from the platform and to share it with other end-users.</p>	Inhabitants of the house	Save money

21	Information for firefighters, the police and ambulance	<p>People will be able to insert information about their houses in the system that will be useful for:</p> <ul style="list-style-type: none"> - firefighters: in case of a fire - the police in case of a theft or in case that of a hostage situation - the ambulance in case of a health emergency <p>This information will include: insurance of the house and personal insurance, the purpose of each room of the house, where end-users sleep in the house, where the gas stove and the boiler are, if there is an oxygen tank, if there are any disabled persons, contact information of the end-user's family, the doctor to contact in case of a medical emergency, how to contact the neighbors.</p>	Inhabitants of the house	Safety
22	Insurance information	<p>End-users will be able to store information about their insurance in the system. For instance, users can store:</p> <ul style="list-style-type: none"> - pictures of the house and of insured items - invoices - list of proprieties - value of the house including renovation costs - measures used to prevent robberies/fires - insurance documents/contracts - contact information about insurance <p>This information will be accessible also by the emergency people (see use case 21)</p>	Inhabitants of the house	Simplify life

23	Permit managing	<p>End-users will be able to ask for a building permit through the platform. End-users will be able to put, find and store the information needed to request a building permit on the platform, for example:</p> <ul style="list-style-type: none"> - house drawings - house ownership information <p>End-users will be able to share this information with other end-users (for instance with potential buyers of the house).</p> <p>End-users will be able to see in which cases they need a permit and they will be able to have more information about those.</p> <p>End-users will be able to see the permits that neighbors asked for and copy data from them.</p>	People that want to build a new construction	Government services
24	Past maintenance details	<p>End-users will be able to put information about the maintenance history of the house in the system. For instance:</p> <ul style="list-style-type: none"> - which color of paint was used to paint the house - which window cleaner is usually contacted - what are the cleaning and maintenance contracts of the house - which renovations were carried out and how much was spent on them <p>End-users will also be alerted when it is time for maintenance of the house and the estimated correlated costs.</p>	Inhabitants of the house	Simplify life
25	Garden maintenance details	<p>End-users will be able to put, store and see information about their garden on the platform. For instance:</p> <ul style="list-style-type: none"> - which trees are in the garden - how many daily sun hours there are in the current period - advise on how to maintain the garden - help to design the garden <p>There will be the possibility to compare the garden to the other end-users' gardens.</p>	Inhabitants of houses with a garden	Simplify life

26	Information about the different parts of the house	End-users will be able to store and retrieve information about their house: - the manufacturer and the details of windows and doors - the manufacturer and the details of the wiring system - the manufacturer and the details of the fireplace	Inhabitants of the house	Simplify life
27	Creating a dossier of the house	End-users will be able to create a complete dossier of the house and to share it with other end-users (for instance a future buyer). This dossier will contain all relevant information about the house and can be personalized by the end-user.	House vendors	Simplify life
28	House sale	End-users will be able to put their house for sale or for rent in the system. They will also be able to share the dossier of the house. Other end-users will be able to see advertisements of the houses that are free to buy or to rent. There will be the possibility to make a hidden/silent sale by sharing just little information about the house and by letting a buyer contact the seller to obtain more information.	House vendors	Save money
29	Ask help to neighbors	End-users will be able to ask for help to their neighbors. They will be able to see which neighbors are at home and they will be able to contact them.	Inhabitants of the house	Simplify life
30	Public space cleaning alert	End-users will be able to report to the municipality if there are public spaces with problems or that need to be cleaned.	Inhabitants of the house	Government services
31	Robbery alert	If there is a robbery, all the end-users near it will be alerted. A description of the robber will be shared (if possible).	Inhabitants of the house	Safety
32	Children's friend finder	Children will be able to find and contact other children in the street to play with.	Inhabitants of the house with children	Social
33	Meet for a coffee	People will be able to see when their neighbors are at home and to contact them to ask to drink a coffee together.	Inhabitants of the house	Social

34	Illegal building report	End-users will be able to report anonymously if a person is making or has made an illegal construction.	Inhabitants of the house	Government services
35	Food delivery from restaurants	End-users will be able to order food online that will be delivered at their home.	Inhabitants of the house	Simplify life
36	Pet registration	End-users will be able to register the details of their pets in the system and to share them. If somebody finds a pet he will be able to look for the owner within the system. If an end-user loses a pet he will be able to contact the neighbors to ask if someone found it.	People that own pets	Simplify life
37	Death of a person	End-users will be able to register their last wishes. If the person dies his last wishes will be automatically shared with the end-users that the person selected.	Inhabitants of the house	Simplify life
38	Services at home for senior end-users	End-users will be able to see a list of services available to senior end-users like meals, hairdresser, pedicure etc. All these services will be delivered at home.	Senior end-users, end-users taking care of their senior parents	Care
39	Consumer offers	End-users will be able to see offers for goods. They will include energy offers and up to date info on products. It will be possible to make collective purchases of goods.	Inhabitants of the house	Save money
40	Flooding emergency	In case of a flooding or flood threat end-users will be able to help each other by sharing information about the risk. End-users will also be able to receive alerts if an emergency is occurring.	People that live in flooding risk areas	Government services
41	Health information sharing	End-users will be able to see and share information from sensors that can measure variables of the end-users life. These sensors can include: check for daily pill intake, check the quantity of water that the person drinks every day. End-users will be able to see and share this information in the platform.	Senior end-users, end-users taking care of their senior parents	Care

42	Ask end-users' opinions about government	The government and municipalities will be able to ask for the opinion of end-users about different topics such as public space maintenance and building plans. The end-users that will be contacted will be selected depending on the area where they are residents.	Inhabitants of the house	Government services
43	Babysitter sharing	People will be able to share the babysitting service with other neighbors in the street. The children will go to one house with a single babysitter.	Inhabitants of the house with children	Save money
44	House tax value calculator	End-users will be able to share data about the variables that are used to calculate municipality taxes levied on their houses. End-users will also be able to import this data from external data sources (for instance the square meters of the house from the land register database).	Inhabitants of the house	Save money
45	Offence alert	Police will be able to advise end-users if there was any offence in the surrounding of the house. Police will be able to send messages to end-users through the platform to find potential testimony of an offense.	Inhabitants of the house, police	Safety
46	Information on improving safety	End-users will be able to receive information about how to make their house more safe. They will be able to certificate their house for the level of safety and to put this certificate outside the house to scare potential robbers. This service can be combined with the services of companies that sell safety equipment for houses. Certificates can be stored in the platform and end-users will be able to share it with their insurance.	Inhabitants of the house	Safety
47	Start an initiative to improve the area	End-users will be able to start initiatives to improve the area near them. They will be able to use a template in the system and they will be able to reach end-users in the surrounding that are registered in the platform.	Inhabitants of the house	Government services

48	Information about roads, public spaces and facility maintenance	End-users will receive information about roads, public spaces and facility maintenance that will be done near them. This information will come from the municipality and from the government.	Inhabitants of the house	Government services
49	Security information	End-users will be able to see information about current criminal offences that have been committed in their surroundings. This use case will just show this information in the user interface.	Inhabitants of the house	Government services
50	Energy information	End-users will be able to see information about their energy consumption regarding gas, water and electricity. This use case will just show this information in the user interface.	Inhabitants of the house	Save money
51	Flood risk information	End-users will be able to see information of flood risk of their area. This use case will just show this information in the user interface.	Inhabitants of the house	Safety
52	Garbage collection information	End-users will be able to see information about the current garbage collecting timeline of their area. This use case will just show this information in the user interface.	Inhabitants of the house	Government services
53	Municipality contact information	End-users will be able to see the contact details of their municipality. This use case will just show this information in the user interface.	Inhabitants of the house	Government services
54	Information on government regulations	End-users will be able to see government regulations concerning their house and the environment.	Inhabitants of the house	Government services
55	Tool to make houses more energy friendly	End-users will be able to get help from the platform to make their house more energy friendly. This tool will permit them to make decisions about their houses regarding thermal isolation, solar panels etc.	Inhabitants of the house	Save money

Table 11: Use cases summary

3.4. Most important use cases

In this section the most important use cases are examined in more detail. The cases are taken from the list in the previous section. The use cases discussed here will be included in the software architecture of the platform as is described in chapter 0. These use cases were chosen because they have the best combination of utility for the different parties involved and simplicity to implement in the first release of the platform.

Each use case is described by analyzing several aspects, as is explained in the next table.

Title	Description
Description	A brief description of the use case and the possible types of partnerships with external companies.
Sequence diagrams	Sequence diagrams of the use case. In this section not all possibilities are analyzed but just the most important examples. For the use cases that provide only information in the user interface the sequence diagrams are not provided because the information is automatically retrieved by our platform without the end-user's input.
Analysis	In this part there is an estimation of the simplicity to implement the use case and a benefits analysis of the different parts involved.

Table 12: Description of main aspects of the most important use cases

3.4.1. Security alarm sharing

3.4.1.1. Description

This section describes in detail use case number 6 (briefly described section 3.3).

Alarms are devices that can be installed in a house and that send a signal when a thief or an unidentified person enters in the house. The end-user can then receive the signal on the mobile app or by other communication tools and decide what to do: to switch it off or to let the central emergency office take care of it. End-users will be able to share the alarm signal with other end-users inside the platform community. A summary of the information of the alarm will also be visualized in the interface of the platform. This information will include the current state of the alarm, the last alarm activation and the list of the end-users that will be contacted if the alarm activates.

3.4.1.2. Partnership with MissionMode

To create this use case we collaborated with MissionMode (<http://www.missionmode.com/>). This service can be integrated with any company that offers a similar service and a similar API of MM.

MissionMode is a company that provides a mobile app and a platform that collect alarm signals and the data of end-users that need to be contacted in case the signal is activated. When the signal activates it is sent to the mobile app of the end-users involved, who can reply if they will handle the signal or not.

MissionMode is an interesting partner they provide a reliable platform to handle alarm signals that, due to their critical importance, must be totally reliable. The role of our platform will be to register the alarm in the system and to add the list of end-users that will be contacted in case the alarm is activated.

When the alarm is activated, the signal is sent to MissionMode. Then the MissionMode system checks which end-users must be contacted and which methods are available (email, mobile app etc.). Then it sends the signal to all

these end-users. The end-users involved will see the message that the alarm is activated and they can respond to that message in two ways:

7. The person addressed will take care of it, so the central emergency system does not need to do anything and the system sends the “yes” reply to our platform and the alarm manufacturer that will store it for statistical purposes.
8. Nobody involved will take care of it, so the signal is switched to the central emergency system. The system sends the “no” reply also to our platform and the alarm manufacturer that will store it for statistical purposes.

The role of our platform in this use case will be to provide the interface where the end-user will be able to insert the data of the end-users that need to be contacted if the alarm goes. This data is then sent to the MissionMode servers.

The end-user will also need to insert the serial number of the alarm and its password (that is written on the alarm itself) in our platform’s interface. Then our platform checks in the MissionMode system if this data is correct. Finally, if it is necessary to use the MissionMode mobile app, our platform will send the link to the end-users involved to download.

3.4.1.3. Sequence diagrams

The following two sequence diagrams explain two different situations that can happen regarding this use case.

In Figure 18 the sequence diagram shows what happens in case an end-user registers the alarm with the correct serial number and password in our platform. This sequence diagram also assumes that the details of the end-users that will be contacted are correct and accepted by MissionMode. The end-user first inserts the serial number and password of the alarm. Then our system asks to MissionMode if the alarm number and password are correct. Then MissionMode replies that they are correct and we provide the interface to insert the details of the end-users that need to be contacted if the alarm is activated. Then this data is sent to MissionMode. Finally we will send all the end-users that will receive the alarm signal a notification and the link to download the MissionMode mobile app (in the diagram they are called “PeopleAlarmActivates”).

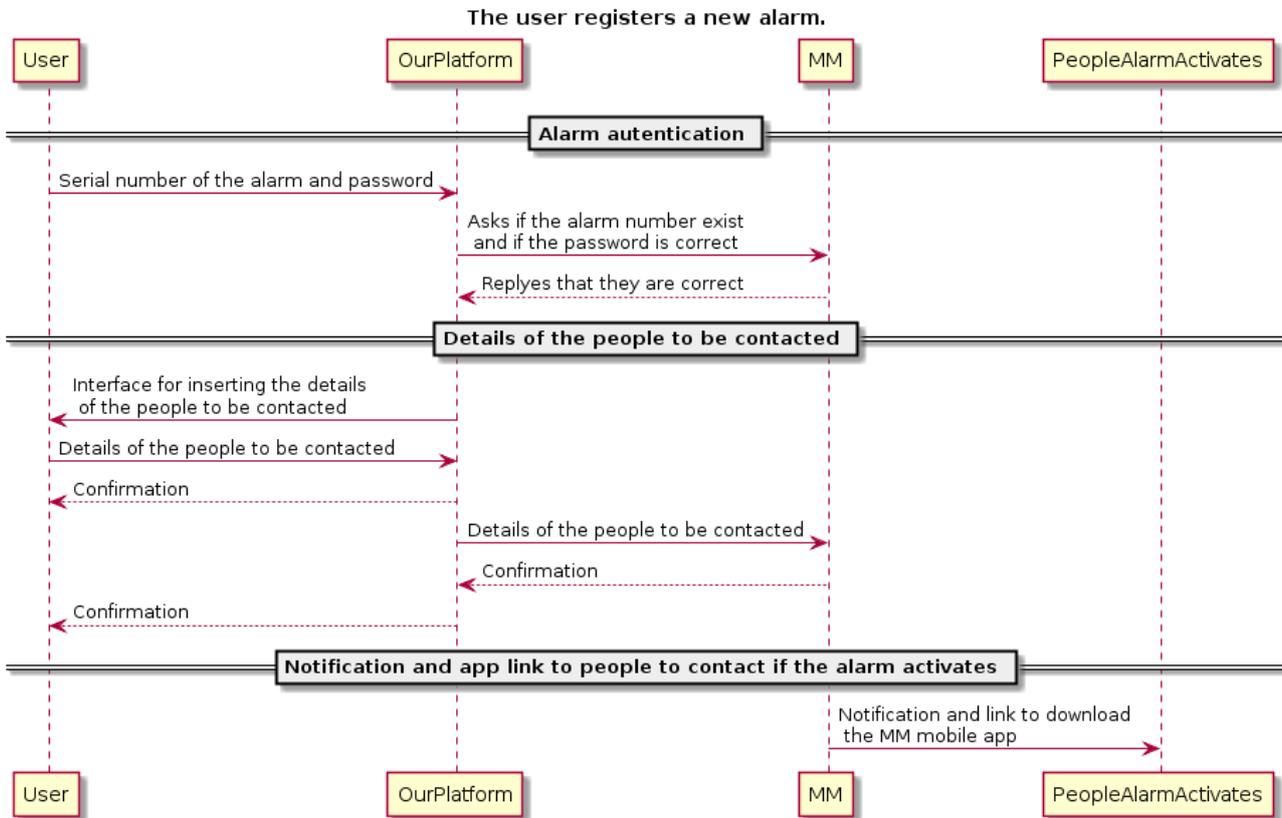


Figure 18: The end-user registers a new alarm with the correct alarm number and password

In Figure 19 the end-user inserts an incorrect serial number and password of the alarm in the system. Then we ask to MissionMode if they are correct and MissionMode replies that they are incorrect. Finally, we give this information to the end-user.

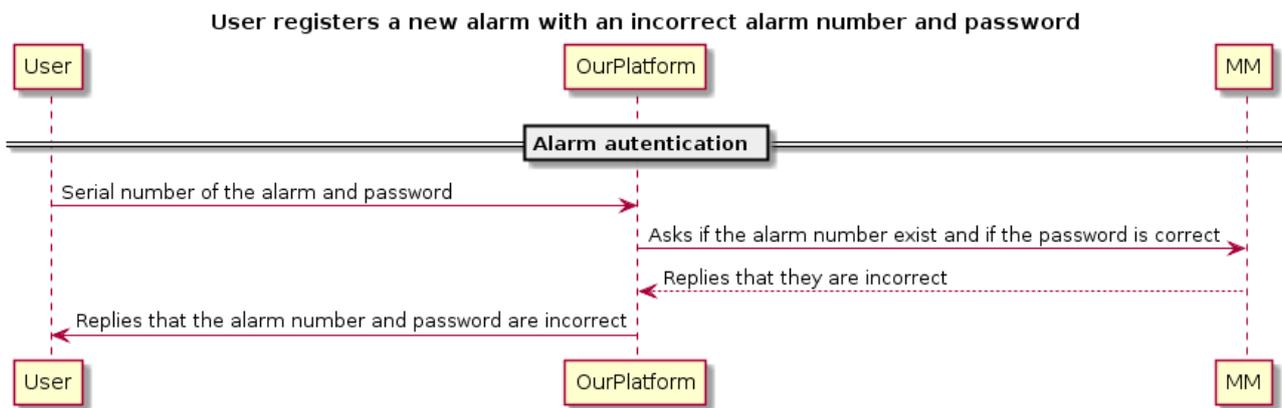


Figure 19: The end-user registers a new alarm with an incorrect alarm number and password

3.4.1.4. Analysis

Simplicity to implement:

The simplicity to implement this service, regarding our part and assuming that our platform is implemented, will be high because our platform will just send MissionMode the contact details of the end-users that will be contacted in case the alarm is activated.

Benefits:

Beneficiary	Description of the benefit
End-users	Simple system to store details of end-users that will receive the signal.
	Since our platform is an integrated portal, end-users will also be able to access other services from the same portal.
	Simple mobile app from MissionMode to receive the alert and reply to it.
MissionMode	Simple way to receive information about to whom the end-user wants to send the signal.
	Fee for every alert.
Alarm manufacturers	Alert service that can be offered to customers.
	Customer base and advertising possibilities when MissionMode and our platform have many end-users.
Our platform	Fee for every alarm added.
	End-user base and advertising from alarm producers and MissionMode.

Table 13: Alarm use case benefits analysis

3.4.2. Photo retrieval and upload

3.4.2.1. Description

This section describes in detail use case number 10 in section 3.3.

We analyzed this use case in collaboration with the Archief Eemland photo database (<http://www.archief.eemland.nl/>). This service can be integrated with any company that offers a similar service and a similar API as Archief Eemland.

With this use case end-users will be able to see old photos of their house in the data information dashboard. End-users will also be able to upload old photos that they have of their home and street and share these photos with the photo archive. We will use this database also for data retrieval purposes (as explained in section 3.5). End-users will be able to upload their photos through the service interface.

In the future there will be the possibility to import photos from other archives but in the beginning the only photo archive that will be available will be the Archief Eemland archive. End-users will also be able to share their photos with other end-users and groups inside our platform.

When a new photo of the end-user's house or street is available the end-user will see that photo in the newsfeed interface.

Archief Eemland makes profit by selling pictures so, if an end-user shares his pictures with Archief Eemland he will get a part of the fee that the company makes by selling them. Our platform will get a part of that fee too.

3.4.2.2. Sequence diagrams

In Figure 20 the process of uploading a new photo to the Archief Eemland database by the end-user is shown. The end-user uploads the photo in our platform and then we send the photo to the Archief Eemland database. A confirmation is sent by Archief Eemland to us and then from us to the end-user.

User uploads and share a photo with ArchiefEemland

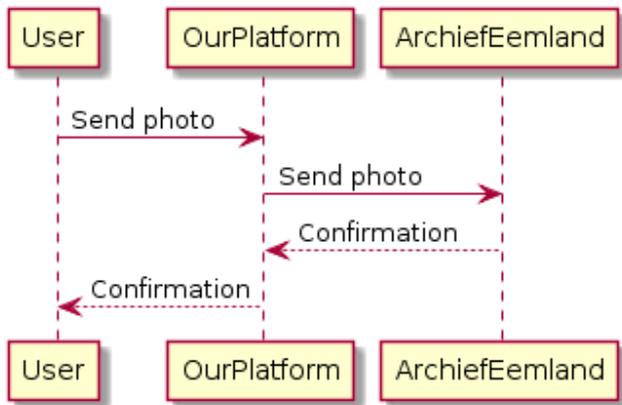


Figure 20: End-user uploads and shares a photo with Archief Eemland

3.4.2.3. Analysis

Simplicity to implement:

The simplicity to implement this service is high, we will only need to program the archive service, the service interface and the storing system for the new photos. But, assuming that the central system will already have been developed, development of this use case will be simple.

Benefits:

Beneficiary	Description of the benefit
End-users	They will be able to see old photos of their house and their street.
	They will be able to share their photos with the photo archive and they will be able to get money if those photos are sold.
Archief Eemland	New photos from end-users that can be sold.
	Free end-user base and advertisement.
Our platform	Fee for every photo sent by end-users that is sold to the photo archive.
	Interesting service and data source to offer to end-users.

Table 14: Archief Eemland photo retrieval and upload benefit analysis

3.4.3. Chamber of commerce company information

3.4.3.1. Description

This section describes in detail use case number 11 in section 3.3.

In this use case the Dutch chamber of commerce will provide the information about registered companies near the address of the end-user. They will provide basic information about the companies for free (name, address, company number, year of foundation).

If the end-user will need the complete information sheet from the chamber of commerce a fee needs to be paid. We will get a part of that fee every time an end-user buys complete information sheets of a company.

The information about the companies that are nearby will be visible to the end-user in the data interface. If the end-user will click on the name of the company in that interface, the chamber of commerce service will open and he will be able to buy the complete information sheet of the company.

There will be also a chamber of commerce service in the services list, where the end-user will be able to search for a specific company or for a list of companies in a specific area. Every time that a new company opens in the end-user's area the end-user will receive a message in the newsfeed.

End-users will also be able to send information to the chamber of commerce if wrong information about companies is listed. We will get a fee for every warning of incorrect information that every end-user sends to the chamber of commerce.

3.4.3.2. Sequence diagrams

In Figure 21 the process of retrieving paid information about a company is shown. First the end-user requests, through the service, the information. Then he is redirected to the payment webpage and makes the payment to us. After this a fee for the information and a query for the information are sent to the chamber of commerce that replies to us with the information needed. Finally the information is sent to the end-user.

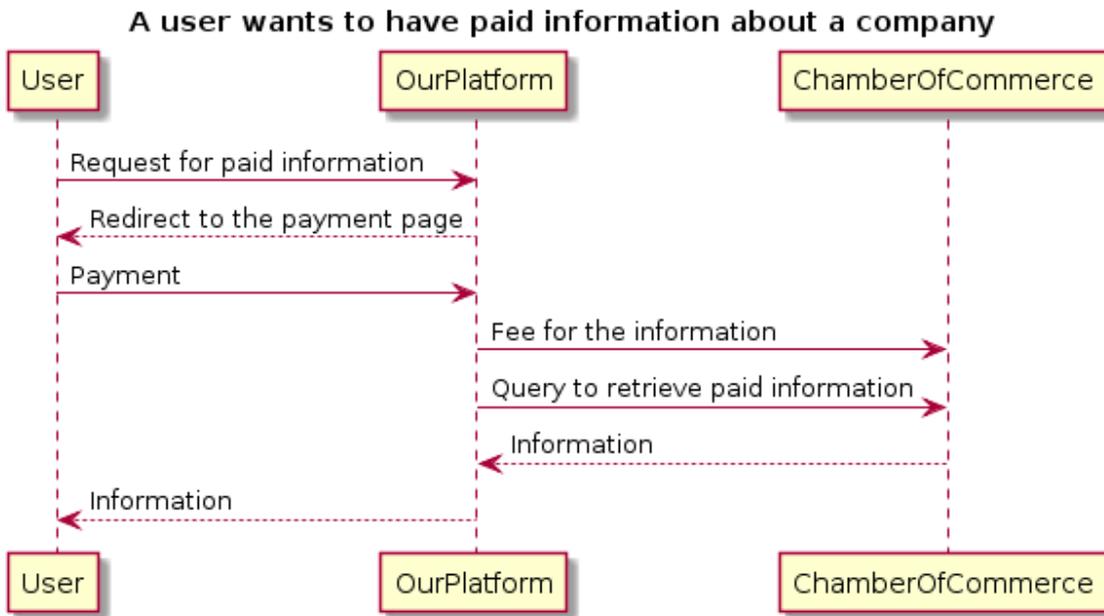


Figure 21: An end-user wants to have paid information about a company

3.4.3.3. Analysis

Simplicity to implement:

The simplicity to implement this service is high, once we have developed the central system, we will just need to develop the connection with the chamber of commerce, the storage to store the company information and the service that will manage the corrections made by end-users.

It may be difficult to implement a function that, when a new company is added to the chamber of commerce database and it matches the address of one of the platform’s end-users, the information is automatically sent to our platform. This is because we will need a signal from the chamber of commerce every time that a new company is added (so they will need to develop this function in their systems) or we will need to send a query to look for new companies every time that the end-user logs in to the system.

Benefits:

Beneficiary	Description of the benefit
End-users	They will be able to see the list of companies near their house and to buy information about them.
	They will be able to send corrections to the chamber of commerce.
Chamber of commerce	Possibility to receive corrections from end-users.

	Possibility to sell information about companies to our end-users.
	Our platform's end-user base and advertisement.
Our platform	Fee for every correction sent by end-users to the chamber of commerce.
	Fee for every purchase of information about a company made by an end-user.
	An interesting service and source of information to offer to end-users.

Table 15: Chamber of commerce company information benefit analysis

3.4.4. Maintenance and various information about appliances

3.4.4.1. Description

This section describes in detail use case number 1 in section 3.3.

With this use case end-users will be able to see different kinds of information about their appliances on the interface of our platform.

They will be able to see the following information:

9. Maintenance schedule
10. User manual
11. Warranty contract
12. Receipt
13. Information about past maintenance and installation
14. Emergency number in case of appliance failure

The supported appliances will be (for instance):

1. Washing machine
2. Dishwasher
3. Boiler
4. Television
5. Router
6. Freezer
7. Refrigerator
8. Dryer
9. Computer

End-users will be able to recognize if an appliance is compatible with our system by a label or an advice that will be put on the box of the appliance.

When the end-user buys one of these compatible appliances, he will insert the serial number and the name of the producer in the appliance service in our platform. Then he will see all the information in the appliance service and the data interface.

We will get a fee from manufacturers for every appliance registered in our platform.

The platform will also be able to show dynamic data that comes from an appliance, for instance if it is turned on, the last time that it was turned on, statistics about usage and the current working mode.

3.4.4.2. *Sequence diagrams*

In Figure 22 the process of adding a new appliance in the system by an end-user and retrieval of the correlating information is shown.

First the end-user inserts the serial number and the producer details of the appliance in the platform. Then our system sends the query to the producer database to retrieve the appliance’s information. Then the producer sends us this information and finally we (after storing this information in our platform) send this information to the end-user.

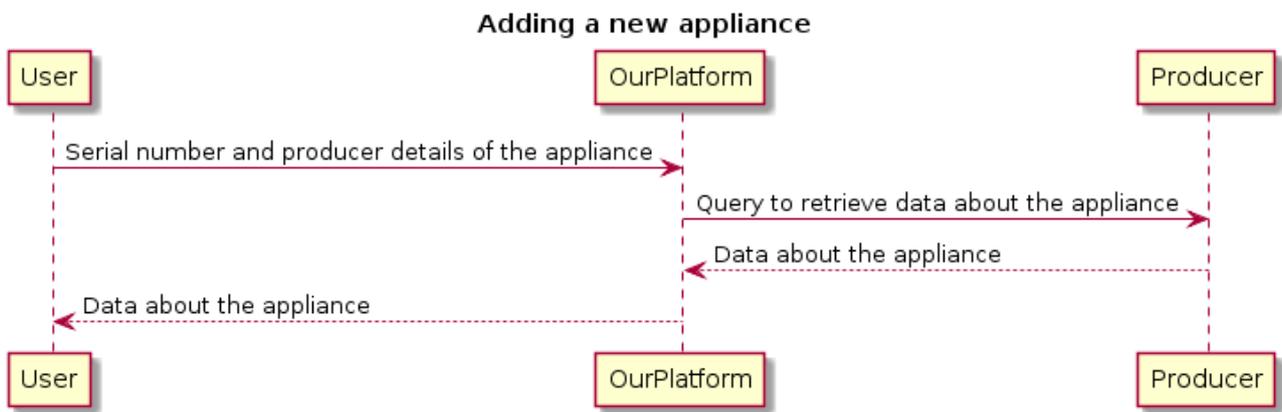


Figure 22: Adding a new appliance in the system

3.4.4.3. *Analysis*

Simplicity to implement:

The simplicity to implement this service depends on the current database and the quality of information of the producers. If the producers have complete and public information about the different appliances that they produce and this information is well structured and accessible it will be simple to implement this service.

Otherwise, if the producers do not have this information public and it is not well-structured it will be difficult to implement this service. Then we will need to convince the producers to reorganize this information in their internal servers.

We would advise to make an agreement with one producer of appliances for one or two appliances for the first public version of the platform. In the future new producers and appliances can then be added to the system.

Benefits:

Beneficiary	Description of the benefit
End-users	They will have all the information about their appliances accessible in an easy and integrated way.

Producers of appliances	Less resources needed for their customer helpdesk department because end-users will retrieve a lot of information from our platform.
	Good service to offer to their customers.
Our platform	Fee for every appliance added in the system.
	Good service to offer to end-users.
	Free advertisement from producers.

Table 16: Maintenance and various information about appliances benefit analysis

3.4.5. Security information

3.4.5.1. Description

This section describes in detail use case number 49 in section 3.3.

End-users will be able to see in their user interface current and past criminal offences that have happened in the surroundings of their house. This information will be retrieved from the Burgernet datasets (<https://www.burgernet.nl/>)

3.4.5.2. Analysis

Simplicity to implement:

The simplicity to implement this service is high because we will just need to retrieve the information from the Burgernet datasets, link it with the address of the end-user and then show this information in the user interface.

Benefits:

Beneficiary	Description of the benefit
End-users	They will be able to see information about criminal offences in their area in the platform.
Burgernet	They will be able to inform end-users about criminal offences in a simple way.
Out platform	Good service to offer to end-users.

Table 17: Security information benefit analysis

3.4.6. Energy information

3.4.6.1. Description

This section describes in detail use case number 50 in section 3.3.

End-users will be able to see details of their energy consumption regarding gas, water and electricity in their user interface. This information will include the daily consumption, the weekly consumption, the current consumption, the current energy rate, the cost of energy divided in the current day, week, month and year. End-users will also be able to see statistics about their consumption in a graph in the user interface. These statistics will be elaborated by the service using data imported from the energy providers.

3.4.6.2. Analysis

Simplicity to implement:

The simplicity to implement this service is medium because we will need to connect to the datasets of the different energy companies involved. There is also dynamic information that needs to be imported such as the current energy consumption that can make the implementation of this use case even more difficult.

Benefits:

Beneficiary	Description of the benefit
End-users	They will be able to see information about their energy consumption in the platform.
Energy providers	They will be able to inform end-users about their consumption in a simple way.
Out platform	Good service to offer to end-users.

Table 18: Energy information benefit analysis

3.4.7. Flood risk information

3.4.7.1. Description

This section describes in detail use case number 51 in section 3.3.

End-users will be able to see in their user interface the risk of flood of their area.

3.4.7.2. Analysis

Simplicity to implement:

The simplicity to implement this service is high because we will just need to retrieve the information of the flood risk connected to the address of the end-user.

Benefits:

Beneficiary	Description of the benefit
End-users	They will be able to see the information about their area flood risk in the platform. This can prevent damages cause by floods.
Government	They will be able to inform end-users about their current flood risk in a simple way.
Out platform	Good service to offer to end-users.

Table 19: Flood risk information benefit analysis

3.4.8. Garbage collection information

3.4.8.1. Description

This section describes in detail use case number 52 in section 3.3.

End-users will be able to see the current garbage collection timeline of their area in their user interface. This information will be shown for each type of garbage such as plastic and organic garbage.

3.4.8.2. Analysis

Simplicity to implement:

The simplicity to implement this service is high because we will just need to retrieve the information of the garbage collection timeline linked to the address of the end-user.

Benefits:

Beneficiary	Description of the benefit
End-users	They will be able to see information about their garbage collection timeline in the platform.
Government	They will be able to inform end-users about their garbage collection timeline in a simple way.
Out platform	Good service to offer to end-users.

Table 20: Garbage collection information benefit analysis

3.4.9. Municipality contact information

3.4.9.1. Description

This section describes in detail use case number 53 in section 3.3.

End-users will be able to see contact details of their municipality in their user interface. This information will include the address, telephone number, fax number, services offered and opening hours of the municipality.

3.4.9.2. Analysis

Simplicity to implement:

The simplicity to implement this service is medium because we will need to retrieve this information from the government datasets. This information may be incomplete or not open to external access.

Benefits:

Beneficiary	Description of the benefit
End-users	They will be able to see information about their municipality on the platform.
Municipalities	They will be able to inform end-users about their details in a simple way.
Out platform	Good service to offer to end-users.

Table 21: Municipality contact information benefit analysis

3.5. Data sources

In this section the data sources that will be used in the use cases that are implemented in the first release of the platform are presented (as described in section 3.1).

These data sources were found by analyzing different use cases, as described in section 3.1, and by finding out which data sources were needed for each use case.

In Appendix A a list of the data sources that will not be implemented in the first release of this platform is shown, and that are not connected with any use cases described in section 3.1. A person can use these data sources for future work and for future development of new use cases.

In the next table the meaning of different sections of the Table 23: Data sources is explained.

Name	Description
ID	Sequential ID number of the data source.
Name	Name of the data source.
Description	Description of the data source.
Category	Category of the data source. This category will be used to group the different data sources in the user interface.
How to retrieve it	How to retrieve the data source.
Use case that uses it	The corresponding use case that uses the data source.

Table 22: Data source list explanation

In the next table the different data sources are listed.

ID	Name	Description	Category	How to retrieve it	Use case that uses it
1	Alarm data	Information about the alarm (the current state of the alarm)	Security	From the alarm	Security alarm sharing
2	Old photos of the surroundings	Old photos of the surroundings of the house where the end-user lives	History	Archief Eemland photo database	Photo retrieval and upload
3	Companies in the surrounding area information	Information about companies that are in the surrounding area of the end-user's home	Various	Dutch chamber of commerce	Chamber of commerce company information

4	Information about appliances	Static and dynamic information about appliances	Various	Appliance producers' datasets	Maintenance and various information about appliances
5	Security information	List of criminal offences in the area	Security	Burgernet datasets	Security information
6	Energy information	Details about the end-user's energy consumption regarding gas, water and electricity	Energy	Energy providers' datasets	Energy information
7	Flood risk information	Risk of flood in the end-user's area	Security	Government datasets	Flood risk information
8	Garbage collection information	Garbage collection timeline of the end-user's area	Various	Government datasets	Garbage collection information
9	Municipality contact information	Contact information about the end-user's municipality	Various	Municipality datasets	Municipality contact information

Table 23: Data sources used for the first release of the platform

3.6. Revenue models

In this section different revenue models of the platform are presented. We found these revenue models by analyzing the different revenues from the different parts involved. For each revenue model we made a description of the model, the customer target of the revenue model, the key activities of the revenue model, the key resources, the value proposition explained with the E3Value methodology and the channels to find new customers. For more information about the E3Value methodology please refer to the website: <http://e3value.few.vu.nl/>.

The next table shows the different sections used to present the different revenue models.

Title	Description
Description	The description of the revenue model.
Customer targets	The customer target of the revenue model.
Key activities	The key activities to implement the revenue model.
Key resources	The key resources needed to implement the revenue model.
Value proposition	The value proposition of the revenue model, determined using the E3Value methodology.
Channels	The different channels that we will use to get customers for the revenue model.

Table 24: Description of the revenue models

3.6.1. Web advertisement

3.6.1.1. Description

In the interface of the platform advertisements will be shown. The advertisements can be linked to the geographical location of the end-user's home to show just relevant advertising for each end-user.

In the SA and in the prototype of the platform described in this thesis, the relevant parts of the SA and the user interface linked with this revenue model are not included. This is because for the first release of the platform this revenue model will not be included.

3.6.1.2. Customer targets

Our customer target are companies that want to offer online advertisements based on the geographical location of the end-users. The customer target of these companies will be the end-users of the platform that will be segmented using their residential address.

3.6.1.3. Key activities

The key activities that we will need to do are:

- Find advertising companies that are able to put their advertisements with a geographical location link in our platform.
- Develop, in the user interface, the parts where the advertisements will be shown.

3.6.1.4. Key resources

The key resources that we will need are:

- The advertising companies that will give us the advertisements.

- The end-user base of our platform.

3.6.1.5. Value proposition

In Figure 23 the value proposition is explained, using the E3Value methodology.

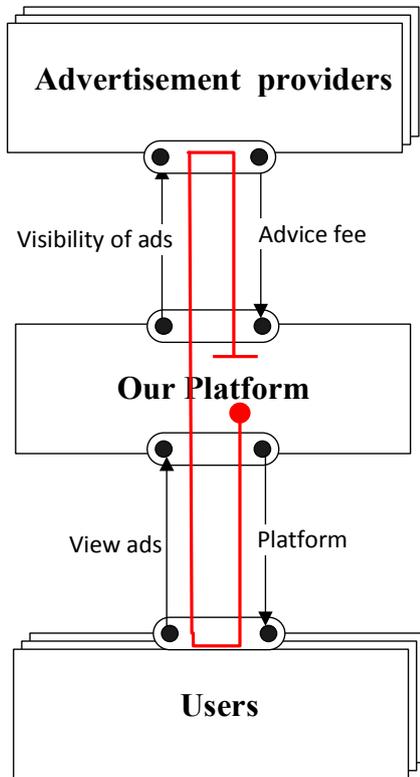


Figure 23: E3Value proposition of the Web advertisement revenue model

3.6.1.6. Channels

We will find new customers through these channels:

- Direct contact with advertisement companies.
- Companies that are currently using our advertising services can introduce us to new advertisement companies.
- Service providers and data source providers that will put advertisements of their company on our platform
- Direct online advertisement.
- Agencies that provide online advertisement (for instance Google AdSense).

3.6.2. Fees from end-users

3.6.2.1. Description

When the platform will reach a large user base it will be possible to ask end-users to pay a small fee to use the platform. The risk of this strategy is that our main resource (the end-users) will be disappointed by the introduction of a fee and that they will leave and go to similar competing companies.

This risk is quite real if we implement an open-source and foundation model as described in section 3.7. This is because it will be simpler for other companies to make a platform similar to ours. However, there will be the possibility to ask a fee to end-users just for using special services, functions and data sources of the platform.

3.6.2.2. Customer targets

Our customer target will be the end-users that have a special advantage in using our platform. They will be more willing to pay a fee without going to our future competitors.

Another customer target will be the customers that will use particular services, data sources or functions of our platform and that are going to pay for using them.

3.6.2.3. Key activities

The key activities that we will need to do are:

- Reach a large initial user base.
- Prevent the switching of end-users to competitors that offer the same service as us at a lower price.
- Raise the entry barriers for new competitors that offer a platform similar to ours.

3.6.2.4. Key resources

The key resources that we will need are:

- Special functionalities or methods that will prevent customers from switching to competitors.

3.6.2.5. Value proposition

In Figure 24 the value proposition is explained, using the E3Value methodology.

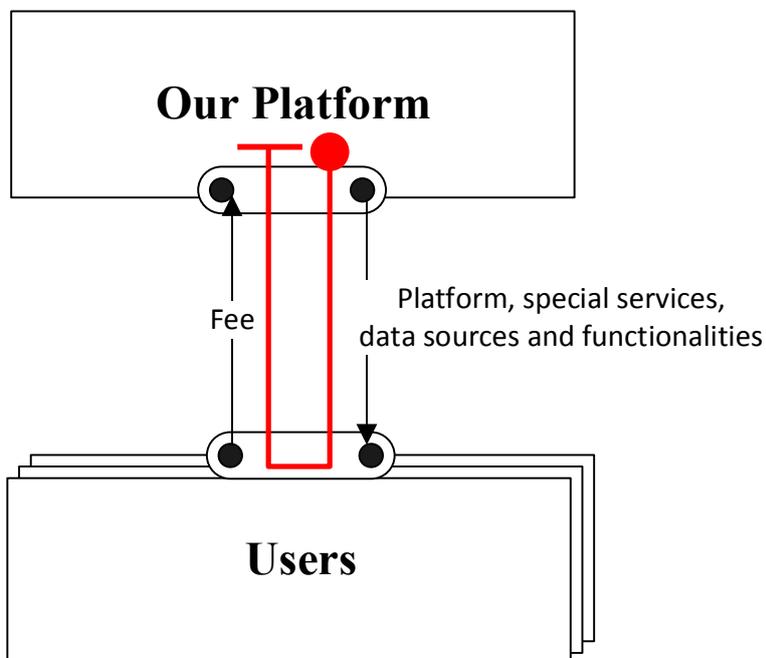


Figure 24: E3Value proposition of the fee from end-user revenue model

3.6.2.6. Channels

We will find new customers through these channels:

- Already existing user base that will pay a fee if they want to keep using functionalities, services and data sources of our platform.
- New end-users from online advertisements.
- Free online marketing (Facebook and LinkedIn profiles).

3.6.3. Fees from service providers and data source providers

3.6.3.1. Description

We will ask a fee from the service providers and the data source providers that will offer particular services and/or data to end-users.

3.6.3.2. Customer targets

Our customer target will be the service providers and data source providers that offer particular services and data sources for which we will be able to ask a fee from them.

If a service or a data source will be useful just for our user base and it will not give any revenue or advantage to the service provider or to the data source provider, we will not ask any fees for it.

3.6.3.3. Key activities

The key activities that we will need to do are:

- Reach a large initial user base.
- Find service providers and data source providers that will develop services and sources for which we will be able to ask a fee for them.

3.6.3.4. Key resources

The key resources that we will need are:

- A large user base.
- Service providers and data source providers that are willing to pay a fee for being able to put their service in our platform.

3.6.3.5. Value proposition

In Figure 25 the value proposition is explained, using the E3Value methodology.

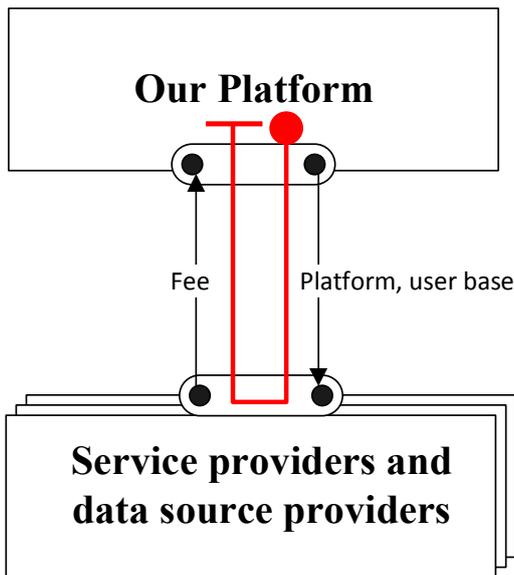


Figure 25: E3Value proposition of the fee from service providers and data source providers revenue model

3.6.3.6. Channels

We will find new customers through these channels:

- Already existing service and data-providers that will be able to offer services for which we will be able to ask a fee.
- Direct contact with new service and data providers.

3.6.4. Consulting for external companies

3.6.4.1. Description

It will be possible to provide consulting services for external companies, municipalities and associations regarding open data and implementation of services.

The developers of this platform have an extremely detailed knowledge of the field of data and services that use data. This valuable knowledge can be used to do consulting services to external parties.

3.6.4.2. Customer targets

The different customer targets will be:

- Companies, municipalities and associations that want to develop services for this platform, data repositories and external services that use data.
- Companies, municipalities and associations that want to offer this platform as a personalized service for their customers.
- Companies, municipalities and associations that want to create a platform that will offer similar services to this one.

3.6.4.3. Key activities

The key activities that the consulting service will provide are:

- Provide help to external companies, associations and municipalities to create services linked to this platform, data repositories and external services that use data.
- Provide help to external companies, associations and municipalities to offer this platform as a personalized service for their customers.
- Provide help to external companies, associations and municipalities to create a similar platform.
- Provide help to external companies, associations and municipalities to create new revenue models connected to this platform or connected to open data.
- Certification of services and data sources that are included in this platform.
- Provide help for the different trust levels that an end-user will need to have to use a specific data source or service.

3.6.4.4. Key resources

The key resources will be:

- People that offer consulting services, these people will be skilled in data services consulting because they will create and develop this platform.

3.6.4.5. Value proposition

In Figure 26 the value proposition is explained, using the E3Value methodology.

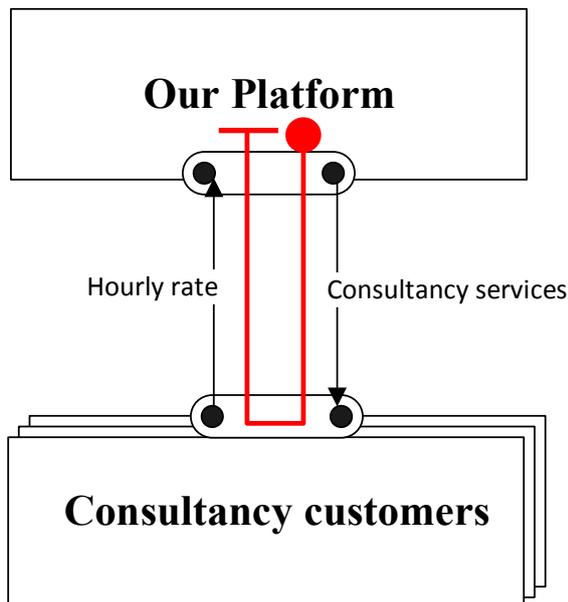


Figure 26: E3Value proposition of consulting for external companies revenue model

3.6.4.6. Channels

We will find new customers through these channels:

- Direct contact with companies, municipalities and associations.

- Current consultancy customers that will introduce us to new customers.
- Advertising from the buzz that this platform will generate when it will be released. We will be the first company that will provide a platform like this, and this will generate a lot of positive reviews from newspapers and specialized journals.
- Third parties that will work with us as partners and introduce us to new customers.
- Direct online advertising.
- Free online marketing (Facebook and LinkedIn profiles).
- Direct contact with municipalities that offer a RfP for innovative services to provide to end-users.

3.6.5. Selling the complete platform to external companies

3.6.5.1. *Description*

It will be possible to provide the complete platform to external companies that will personalize the portal with their logo and color scheme. We will manage the personalization of the portal and the services. Additionally, we will maintain the portal and provide all necessary training to the company to manage simple tasks themselves.

These companies will then be able to offer the portal to their customers, thus providing them with innovative services without the effort of implementing the platform. Once the platform is developed, it will be simple for us to replicate it for external customers. We can store and run the platform also on our servers, requiring little effort in management by external companies.

3.6.5.2. *Customer targets*

The customer targets are:

- Housing associations that want to provide the platform with their brand to their associates.
- Municipalities that want to provide access to the platform with their brand for their inhabitants. This could be a useful tool in a city marketing campaign.
- Student housing associations: they can offer their associates a personalized portal with data sources and services targeted at students.

3.6.5.3. *Key activities*

The key activities are:

- Provide customers consultation about their needs and requirements.
- Provide the platform and personalize it with the customer's brand and color scheme.
- Provide technical and business consultation about personalized services and data sources that the customer needs.
- Provide help to define services and data sources that the customer wants to include in his personalized platform.

3.6.5.4. *Key resources*

The key resources are:

- People that offer consultation to implement the personalized platform.
- Servers to store and run the personalized platform.

- The platform to personalize.

3.6.5.5. Value proposition

In Figure 27 the value proposition is explained, using the E3Value methodology.

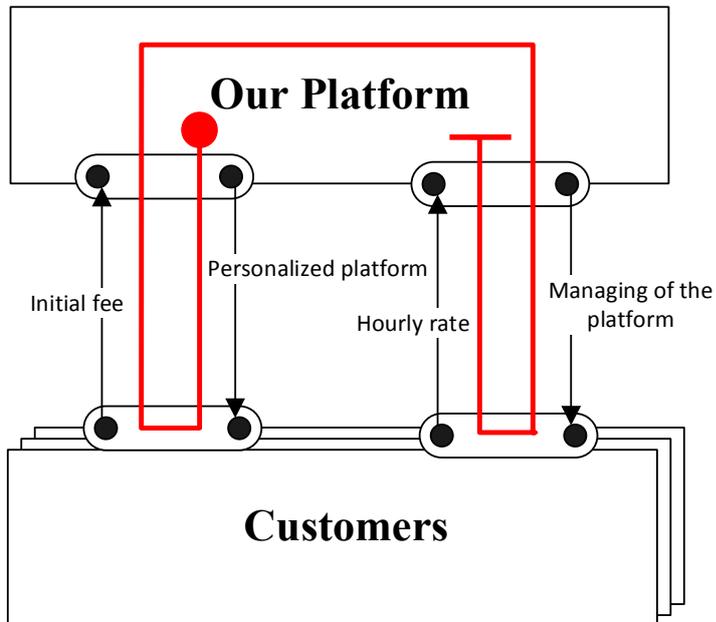


Figure 27: E3Value proposition of the complete platform revenue model

3.6.5.6. Channels

We will find new customers through these channels:

- Direct contact with companies.
- Customers that already have a personalized version of the platform that introduce us to new customers.
- Free online marketing (Facebook and LinkedIn profiles).
- Direct contact with municipalities that offer a RfP for innovative services to provide to end-users.

3.7. Foundation model and open source model

In this chapter the advantages and the disadvantages of creating a foundation and of developing the platform as OSS are described. The HuisKluis foundation already uses this model (for further information about HuisKluis please refer to the preface of this thesis).

3.7.1. Foundation model

3.7.1.1. Description

Using a foundation instead of a private company we will be able to gain more funds to develop the platform. Since a foundation is not allowed to make any profit and share it with the shareholders, the profit for the creators of the platform will come from other sources, as is explained in section 3.7.1.4.

A foundation normally consists of a foundation board and employees that work for the foundation. The foundation board normally consists of people that do not work as employees for the foundation. The members of the foundation board will be people from external parties that fund the foundation, other external parties or us.

3.7.1.2. Dutch foundation principles and regulations

In The Netherlands foundations have a great history. In 2001 the non-profit sector in Netherlands had about 669.000 full-time equivalent jobs that added up to 12.9% of non-agricultural paid employment. The amount of volunteers in the non-profit sector in Netherlands in the same year was about 400.000 full time equivalents jobs (Burger & Dekker, 2001).

Foundations are composed of two types of end-users, the board of directors and the employees. They have different key activities in the foundation as is explained in section 3.7.1.3. A foundation cannot share profit with its members but must use them for its purpose and mission. When a foundation is registered, several things need to be established: the objective of the foundation, the method of acquiring new board members and the resignation procedures of existing ones (Burger & Dekker, 2001). The foundation has full legal competence and the management board's is personally responsible for legal acts of the foundation (as long as the foundation is not entered at a commercial register) (Burger & Dekker, 2001).

The Dutch chamber of commerce requires an annual fee for the registration of the foundation (Burger & Dekker, 2001).

The management board is responsible for realization of the objectives and for representing the foundation. It is able to take independent decisions. There is the possibility to have an supervisory board (especially in large foundations) that supervise the actions of the management board (Burger & Dekker, 2001) but this is not mandatory. As a rule, the management board's directors cannot have any conflict of interests with the foundation.

In Netherlands just a few types of foundation can have privileged tax treatment and this is correlated with the type of services and activities that they offer. Foundations that pursue "religious, ideological, charitable, cultural, scientific or public interest objects" are eligible for this tax reduction and they are called "public benefit organizations". (Burger & Dekker, 2001)

A common way to use the foundation model in Netherlands nowadays is to create a foundation that owns the shares of a company. The foundation board has the right to vote for the decisions of the company whose shares have (Francken, 2013).

3.7.1.3. Foundation board and employee key activities

The foundation board will have the following responsibilities, as is explained in the next table.

Title	Description
Coordinate consulting services	The foundation board will coordinate consulting services that the employees do for external companies. There will be the possibility that external people offer private consulting services to external companies regarding this platform. In this case, the foundation board will coordinate just the relationship with them and will not coordinate their work.
Coordinate funds retrieval	The foundation board will coordinate funds retrieval from external parties, both public and private.
Coordinate partnerships	The foundation board will coordinate partnerships with external companies that provide services and data sources to the platform.
Coordinate the ethical compliance of all parties involved	The foundation board will constantly check that all different parts involved in the foundation, including the employees and external partners, will comply with the ethical principles of the foundation’s manifesto.
Coordinate marketing activities	The foundation board will coordinate the marketing and public relationship activities.
Coordinate employees	The foundation board will coordinate the main activities of employees of the foundation.

Table 25: Foundation board activities

The employees of the company will have the following tasks and responsibilities, as listed in the next table.

Title	Description
Managing the open source community	The project managers of the foundation will manage the open source community that develops the software (if the platform will be released as OSS).
Defining new business models for the platform	The economists of the foundation will search for new businesses and for new revenue models for the platform that will benefit all parts involved.
Providing business and strategic advice	Manager will provide business and strategic advice to the foundation board regarding the platform
Managing public relationships with external parties	The public relationship department of the foundation will manage the public relationship with external parties.
Managing the marketing of the foundation	The marketing department of the foundation will manage marketing of the platform.
Analyzing future functions, services and data sources of the platform	The innovation department will manage new functions, services and data sources that the platform can implement in the future.

Table 26: Foundation employee activities

3.7.1.4. Revenue models for us

In the foundation model we will work for the foundation as employees, or as foundation board members and for external companies as private consultants.

As employees of the foundation we will get a normal hourly rate. As private consultants we will be able to provide the services explained in the following table.

Title	Description
Consulting services to companies that want to connect their data sources to the system	We will provide consultation for external companies that need to connect external sources to the platform.
Consulting services to companies that want to provide services linked to the system	We will provide consultation for external companies that need to make services linked to the platform.
Consulting services for companies that want to offer a personalized version of the software to their customers.	We will provide consultation for external companies that need to make a personalized version of the platform to offer to their customers.
Consulting services for the foundation	We will provide consultation to the foundation as external consultants for various activities such as managing and innovation.

Table 27: Consultancy services

3.7.1.5. Revenue models of the foundation

In addition to the revenue models described in section 3.6, the foundation will also receive funds from the revenue models explained in the following table.

Title	Description
Private and public donations	People will be able to make donations to the project to support it. The donations will be managed by the foundation.
Certification of platforms implemented by other companies	If an external company wants to implement a copy of the platform and to have certification that its platform is compliant with the original project, it will need to pay a fee to the foundation.

Table 28: Foundation additional revenue models

3.7.1.6. Foundation manifesto

The foundation will follow ethical rules that are explained in the following manifesto.

The foundation board will always check that all parts of the foundation comply with these rules.

1. In the last 30 years the world changed into a digital world. We will help end-users to take full advantage of this digital world by providing them with useful and simple services.
2. Strangers cannot enter your house uninvited, it is prohibited by law. But big technology companies can record your data through your telephone, television and computer and use it for economic purposes. We will solve this problem by permitting you to share your data with just the external parties that you want.

3. Every Dutch person has the right to privacy. They have the right to choose with whom to share their data. Currently this is not possible, big technology companies use data of end-users, who do not have any control over them. We will give end-users the right of privacy in the digital world back.
4. National governments collect private data of citizens for anti-terrorism purposes. We will share data of end-users with national governments just if the national law forces us to share data with them. We will make a complete and transparent list of situations where we are forced to share data of end-users within the national government and we will share end-users' data just in these specific cases.
5. People must have the right to change their private data when they want. We will help end-users to change their data across the internet with any measure that we will be able to do.
6. We will only be successful when we reach millions of end-users. To do this we will provide easy, safe and useful services to end-users.
7. Open source software changed the world in many ways. Open source software is transparent, safe, reliable and innovative. The platform that we will make will be open source so end-users are able to benefit from the advantages of the open source system.
8. Some services on the internet are privacy sensitive, such as online banking. We will provide high security measures that control the identity of end-users that access highly private services.
9. To be really innovative, ideas and opinions of all parts involved in the project will be taken in consideration. We will carefully listen to every idea, concern and opinion that is voiced by our customers and partners to deliver the best innovative solution that will satisfy the needs of all the end-users involved.
10. Real innovation is created by honest and trustworthy actions. We will put all our statements in action by providing a platform that is safe, private, secure and simple, to satisfy the needs of end-users from all over the world.

3.7.2. Open source model

It will be possible to develop this platform as open source.

The objective of an open source strategy is to create an environment where open-source developers will develop the project without receiving a salary for developing the code. Following this strategy, the complete project will be released under a GPL license permitting anybody to obtain and modify the source code. Any modification of the source code, following the GPL license ("GNU GENERAL PUBLIC LICENSE," 2007), must be made public. There are also other types of licenses for OSS that have more or less restrictions than the GPL license. In this section we assume that we will use the GPL license.

Using this strategy, we will create a foundation that will own the platform brand, logo and domain. The foundation will coordinate the activities of the open source developers and will provide business, marketing and management activities.

3.7.2.1. Advantages and disadvantages

The advantages of implementing an open source model are explained in the following table (Morgan & Finnegan, 2014).

Title	Description
-------	-------------

Free developers	Most of the developers will work for free. This will lead to a big cost saving for us. We will still need to pay the management and leaders of the project that will coordinate the open source developers. There is the possibility that we will still need to hire developers that will work on the project if the reaction of the open source community to our platform will not be enthusiastic.
More trust from customers	Open source software is perceived to be safer, more reliable and trustworthy by customers, so they will have more trust in our platform. Customers will not have the vendor lock-in because in a few years there will be copies of our platform in the market and customers will be able to switch to our competitors without any problems.
Better quality of code	Due to a better peer review and because everybody will be able to access the code and report bugs, the quality of the code will increase. This will lead to having a more reliable and trustable product.
Simple for third parties to develop services and plugins for the platform	Since third parties will be able to access the source code, it will be simpler for them to develop services that connect to the platform.
Accelerate innovation and exchanging ideas	A lot of developers will be able to see the source code. Because of this, they will be able to implement and propose innovations and ideas to the project.
Access to superior knowledge	Since a lot of developers will be involved in the project, they may have better skills and knowledge of developing compared to internal paid developers.

Table 29: OSS advantages

The disadvantages of implementing an open source model are explained in the following table (Morgan & Finnegan, 2014).

Title	Description
Difficult to find OSS developers	According to Santos et al. (2013), for a new project like this it can be difficult to attract new open source developers in the initial stages of development due to the low attractiveness of this project. The low attractiveness to the OSS environment stems from the fact that this is a new project.
Difficult to manage the creation of an open source software	Because developers will work for free, we will not be able to control their work and schedule. We will also need to take their opinions into consideration during the decision making process. This will lead to a lack of centralized decision making. The overall cost of managing this may overshadow the initial cost saving of using free developers.
No ownership of the code	The code can be used by anybody. The source code is one of our main assets and, with open source developing, the entry barriers to future competitors will be low because they will be able to use our source code. This may lead to less revenue and profit.

Difficult to access top quality developers	Because developers will work for free, the top quality developers may not join our community because they are already developing for closed source companies and getting paid for it.
Difficult to plan a roadmap	Due to the difficulty of managing the community it will be difficult to schedule a roadmap and to force developers to follow it.

Table 30: OSS disadvantages

4. Software architecture

In this chapter the software architecture of the platform is described. The reader can find the requirement analysis, the technical problem analysis, the solution domain knowledge analysis, the solution analysis and then the synthesis of the overall software architecture in this chapter.

4.1. Software architecture design method

The software architecture design method that we will follow is described in Figure 28. The design is based on the process explained by Tekinerdoğan and Akşit in their book “Synthesis-based software architecture design” (Tekinerdoğan & Akşit, 2002). This design process involves the construction of sub-solutions for every different problem of the architecture and then the integration of these sub-solutions into one single solution.

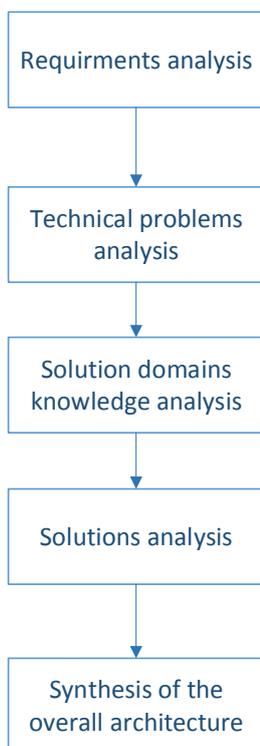


Figure 28: Software architecture design method

As explained in Figure 28 the first phase will be the “requirement analysis” phase. The requirement analysis consists of a list of the requirements that the platform needs to have from the client’s perspective.

Then, in the “technical problem analysis” phase, the client’s requirements are translated into technical requirements. Normally the client’s requirements lack technical specifications and the right details of the problem. In consequence, the gap between the client’s requirements and the software architecture is generally large.

In the "solution domains knowledge analysis" phase, the solution domains for each problem are mapped. Each solution domain is connected to a technical problem in the previous step. This phase is different from the "solution analysis" phase because "the solution domain itself may be large and include abstractions that are not relevant for solving the corresponding problem" (Tekinerdoğan & Akşit, 2002). This phase enables creators to have a better distinction between and better knowledge of the solution domains.

Then, in the "solution analysis" phase, the solution analysis for each technical problem is provided. The solution analysis includes the part of the software architecture that will solve that specific technical problem.

In the "solution analysis" phase the commonality and the variability analysis of the alternatives that can be developed for a solution domain are also included. During this phase alternatives of the same solution domain are analyzed and then the commonalities and the variability between them are found. Finally several of these variables and commonalities are taken into consideration to decide on the final solution for that particular technical problem. In this thesis the commonality and the variability analysis is done just for technical problem number 11: "what will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?". This is because it is the main technical problem of the software architecture.

In the "synthesis of the overall architecture" phase the different solutions of the previous phase are synthesized in the overall architecture of the platform.

4.2. Requirement analysis

These requirements are taken from section 2.1.3 and are discussed in more detail in the next table.

R#	Title	Description
R1	Different levels of authentication for end-users	Every service, function and data source will require a different level of authentication from the end-user. The section 2.1.2.1 describes in detail the different levels of identity certification.
R2	The possibility to import external data	The platform will allow data import in the system from external sources. This data can be static or dynamic (for instance from external sensors).
R3	The possibility to store data inserted by end-users	End-users will be able to insert data in the system, which will be stored in the platform. End-users will be able to manage and delete this data.
R4	External services development	External companies will be able to access end-users' data and to develop services that provide useful functions to end-users using that data. A list of the available services will be available on the dashboard. These services will access the external data sources connected to the end-users. End-users will be able to determined which data sources are shared with the external services.
R5	Dashboard	End-users will be able to see in their dashboard all the data connected to their house, the list of services, the newsfeed and the links to other functions of the platform, like the messaging system. End-users will be able to see everything about their home in one place.
R6	News feed	End-users will be able to see a newsfeed on the dashboard. The newsfeed will contain news about the system like new services, new data sources and important data retrieved from external data sources.
R7	End-users will be able to receive alerts	Alerts are messages sent by services and the platform to end-users by email, mobile app, SMS and automatic calls.
R8	Social function	End-users will be able to add other end-users as friend, to share data with them and to exchange messages with them.
R9	Wallet function	End-users will be able to share their data with external websites with a wallet function.
R10	Tablet and mobile	The user interface website will have a modality that will support tablet and mobile browsing.

Table 31: Software requirements

4.3. Technical problems analysis

In this section the technical problems related to each requirement of the previous section are discussed. Each requirement corresponds with one or more technical concern. Each technical concern can relate to one or more requirements.

4.3.1. R1. Different levels of authentication for end-users

End-users will need different levels of authentication to access each service (the different levels are described in section 2.1.2.1). For this SA we will not include level number 5 and 6 because they are not essential for the first release of the platform.

Concerns:

- T1: how is the platform going to provide the different levels of authentication for end-users?
- T2: how will the system store the authentication data in a reliable way for the authentication component?
- T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.3.2. R2. The possibility to import external data

The platform will allow data import in the system from external sources. This data can be static or dynamic (for instance from external sensors).

Concerns:

- T4: How is the platform going to connect to external data sources?
- T5: how will the system store the end-users' data?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.3.3. R3. The possibility to store data inserted by end-users

End-users will be able to insert data in the system, which will be stored in the platform. End-users will be able to manage and delete their data.

Concerns:

- T5: how will the system store the end-users' data?
- T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.3.4. R4. External services development

External companies will be able to access end-users' data and to develop services that provide useful functions to end-users using that data. A list of the available services will be available on the dashboard.

Concerns:

- T5: how will the system store the end-users' data?
- T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?
- T6: How are we going to integrate the external services in the platform?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.3.5. R5. Dashboard

End-users will be able to see all the data connected with their house, the list of services, the newsfeed in their dashboard. Links to other functions of the platform, like the messaging system, can also be found there.

Concerns:

- T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.3.6. R6. News feed

End-users will be able to see a newsfeed on the dashboard. The newsfeed will contain news about the system like new services, new data sources and important data retrieved from external data sources (for instance flood risks).

Concerns:

- T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?
- T7: How are we going to develop the news feed for every end-user?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.3.7. R7. End-users will be able to receive alerts

Alerts are messages sent by services and the platform to end-users by email, mobile app, SMS and automatic calls.

Concerns:

- T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?
- T8: How are we going to implement a reliable alert function?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.3.8. R8. Social function

End-users will be able to add other end-users as friend, to share data with them and to exchange messages.

Concerns:

- T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?
- T9: How are we going to implement the social component in the software architecture that allows end-users to connect with other end-users in the platform?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.3.9. R9. Wallet function

End-users will be able to share their data with external websites through a wallet function. This function permits users to share data with external websites.

Concerns:

- T10: How are we going to implement the wallet component that shares the end-users' data with external websites in the software architecture?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.3.10. R10. Tablet and mobile

The user interface website will have a modality that supports tablet and mobile browsing.

Concerns:

- T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?
- T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

4.4. Solution domains knowledge analysis

4.4.1. T1. How is the platform going to provide the different levels of authentication for end-users?

The platform will need to provide different levels of authentication to end-users. Each service will need a different level of authentication from the end-user and it will be connected to the authentication component of the platform.

4.4.1.1. S1. ForgeRock OpenAM and OpenIDM

As explained in section 2.2.2.7 OpenAm is an open source identity provider that can provide tools to manage the identity of end-users. It satisfies T1 because it can be integrated in our platform to provide the different authentication methods.

For more information please refer to section 2.2.2.7.

4.4.1.2. S2. Apache syncope

As explained in section 2.2.2.9 Apache Syncope permits managing the identity of end-users. It provides useful functions and the ability to be connected with other parts of our platform.

For more information please refer to section 2.2.2.9.

4.4.2. T2: how will the system store the authentication data in a reliable way for the authentication component?

The authentication component will need to store the credentials of end-users.

4.4.2.1. S3: MySQL

MySQL (<https://www.mysql.it/>) is a free and open-source database for storing data. It is compatible with Apache Syncope. It provides useful functions such as backup and encryption.

4.4.2.2. S4: PostgreSQL

PostgreSQL (<http://www.postgresql.org/>) is an open source database system that runs on all major operating systems, including Microsoft Windows, Linux and Unix. It is fully ACID compliant and gives full support for foreign keys, joins, views, triggers, and stored procedures. It is possible to store full binary objects such as photos and videos.

It natively supports C/C++, Java, .Net, Perl, Python and other programming interfaces.

4.4.3. T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?

The user interface will need to connect to the different components of the SA, it will need to show the different elements inside it such as data connected to the house, the list of services, the newsfeed and the links to the other functions of the platform like the messaging system.

The user interface will need also to be accessible from mobile and tablet devices.

4.4.3.1. S5: Bootstrap

Bootstrap (<http://getbootstrap.com/>) is a HTML, CSS and JS framework for developing websites and web applications. Its objective is to simplify web development. It provides HTML and CSS design templates with many different interface components such as buttons and forms.

It provides a simplified way to make websites accessible from tablets and by mobile browsing, by making customized CSS styles for every different device. It allows connecting the different components of our platform through JS components.

Bootstrap is free and open-source and it can be used in our platform for free.

4.4.3.2. S6: Webix

Webix (<http://webix.com/>) is a HTML and JS framework that simplifies the creation of web applications, for both desktop and mobile. It is released under the GNU as well as under a commercial license for commercial purposes.

It provides the possibility to add widgets to the web interface such as data tables, pivot tables, data list widgets, charts and organograms, different layout types, different data editing types such as forms and uploader widgets and different navigation methods.

4.4.3.3. S7: Exhibit

Exhibit is described in section 2.2.2.5 and it permits integration of data in web pages. The framework is able to automatically import linked data from tabular data sources. It can be integrated in existing CMS in a simple way because it is a JS library.

For more information please refer to section 2.2.2.5.

4.4.4. T4: How is the platform going to connect to external data sources?

The platform will need to import data from external data sources. These data sources will be static and dynamic.

4.4.4.1. S8: Sensors connected by routers through LAN network

According to Nechifor (Nechifor et al., 2015) and as described in section 2.2.1.1, it is possible to retrieve dynamic data through connecting the sensors to local routers, by a LAN network. Then the different routers will be connected with an external VPN server to increase the security of the connection.

For more information please refer to section 2.2.1.1.

4.4.4.2. S9: virtual sensor system

As described in section 2.2.1.2, Hill et al. (2011) describe a method to retrieve data from sensors. They divided their architecture in three layers: the physical layer, the virtual sensor abstraction layer and the web 2.0 interactive collaboration layer.

For more information please refer to section 2.2.1.2.

4.4.4.3. S10: converting data to Linked data

As described in section 2.2.1.5 it is possible to import data from external data sources and then convert this data to linked data. First, the external datasets need to be modelled. Then we will need to convert this dataset to the

RDF format (for more information see section 2.2.2.2). Finally, we will need to store this linked data in our platform.

For more information please refer to section 2.2.1.5.

4.4.5. T5: How will the system store the end-users' data?

The platform will need to store the end-users' data. This data will include the data inserted by the end-user in the platform and the data imported from external data sources.

4.4.5.1. S11: Apache Jena

Apache Jena (described in section 2.2.2.6) permits storage of RDF triples through its component TDB. It supports all the Jena APIs and SPARQL query language. It can be combined with the Apache Jena Fuseki SPARQL server to provide a "robust, transactional persistent storage layer, and incorporates Jena text query and Jena spatial query" ("Apache Jena Fuseki,").

For more information please refer to section 2.2.2.6.

4.4.5.2. S12: Virtuoso Universal Server

Virtuoso Universal Server (<http://virtuoso.openlinksw.com/>) is an enterprise multi-model data server that permits users to manage different types of server functionalities into a single product.

Its functions include (<http://virtuoso.openlinksw.com/>):

- SQL Relational Tables Data Management
- RDF Relational Property Graphs Data Management
- Content Management
- Web and other Document File Services
- Five-Star Linked Open Data Deployment
- Web Application Server

It is distributed with two separated licenses, one free and open-source and one commercial.

4.4.5.3. S13: private network folders

Following Van et al (2014), a good way to store the end-users' files is to store them into private network folders and in a private data cloud. Private network folders have a better security because they are not connected directly to the internet.

For more information please refer to section 2.2.1.4.

4.4.6. T6: How are we going to integrate the external services in the platform?

External services need to be integrated in the platform and they need to connect with the other components of the platform. In this section we discuss just one solution domain because S14 provides a perfect solution for this technical problem.

4.4.6.1. S14: MyPHRMachines

Following section 2.2.1.4 the architecture of MyPHRMachines can be a good example of how to implement external services in our platform.

They implemented a layer where the different virtual machines on which the services run are implemented. When an external company wants to insert a new service or update an existing one, it will first need to send it to us. Then we will check that the service is safe and reliable by testing it, and finally we will upload it in the VirtualBox layer where all the services run.

In a separate layer the AppStore was put. In the AppStore there is the list of all the services available to end-users. We will implement a similar layer where all the different services are listed.

4.4.7. T7: How are we going to develop the news feed for every end-user?

The news feed of every end-user need to be processed and then sent to the user interface. In this section we present just one solution domain because S15 provides a perfect solution for this technical problem.

4.4.7.1. S15: LinkedIn SA

The LinkedIn company made a SA for their platform that includes a news service that provides news to the web app. It has similar specifications as our newsfeed because it provides news about companies and other end-users registered in the platform.

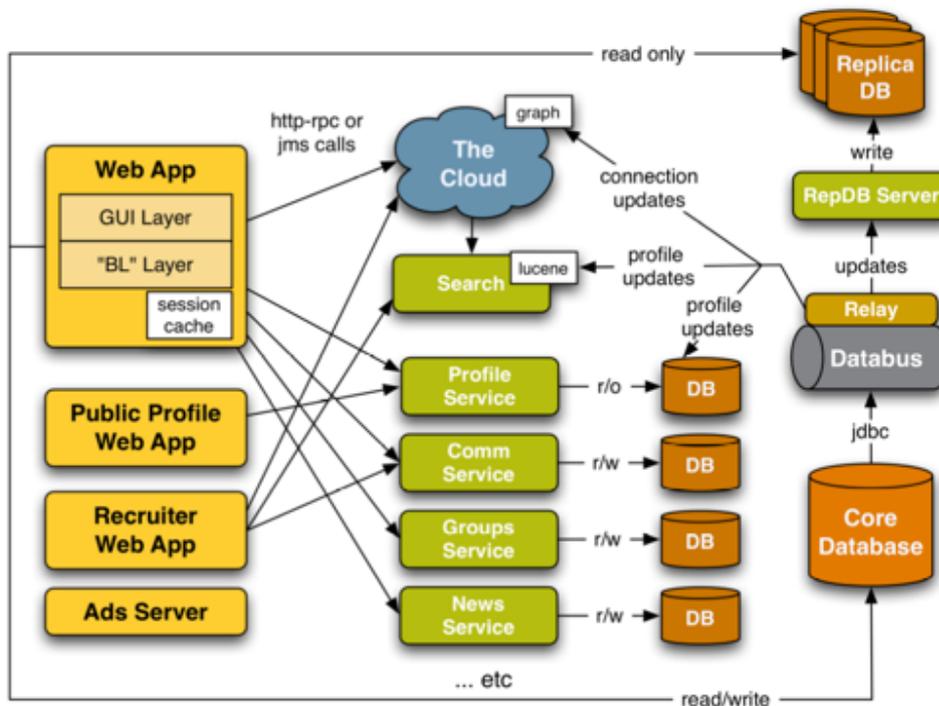


Figure 29: LinkedIn - A Professional Network built with Java Technologies and Agile Practices

Following Figure 29 ("LinkedIn - A Professional Network built with Java Technologies and Agile Practices,"), LinkedIn connected the different services to the Web App and to the databases. As explained in Figure 29 the

news service is connected to the LinkedIn database that contains updated profiles of every end-user. The Web App that provides the GUI layer (that provides the user interface) is then connected with the news service for retrieval of the news feed.

4.4.8. T8: How are we going to implement a reliable alert function?

The SA will need a component that provides the alert function to end-users. This alert function will need to be reliable because it will be used for the alarm sharing use case (see section 3.4.1). In this section we analyzed just one solution domain because the S16 MissionMode solution provides a perfect solution for this technical problem.

4.4.8.1. S16: MissionMode

As explained in section 3.4.1.2 the MissionMode company provides reliable software for the alert function. They provide the alert function in different ways such as for a mobile app, by email, telephone and fax.

We will be able to connect to their system and receive the feedback signal from the end-user directly from their SA.

They support the function of storing details of the end-users that need to be contacted in case the alarm is activated. When the alarm is activated they will receive a signal from the alarm in their system and then they will send the signal to the corresponding end-user.

For more information please refer to section 3.4.1.2.

4.4.9. T9: How are we going to implement the social component in the software architecture that will permit end-users to connect with other end-users in the platform?

End-users will need to connect with other end-users of the platform. In this section we analyzed just one solution domain because S17 provides a perfect solution for this technical problem.

4.4.9.1. S17: LinkedIn SA

We can use the software architecture of LinkedIn, as mentioned in section 4.4.7.1, to also implement the social component.

The social component is divided in two sub-components: the profile service and the communication service.

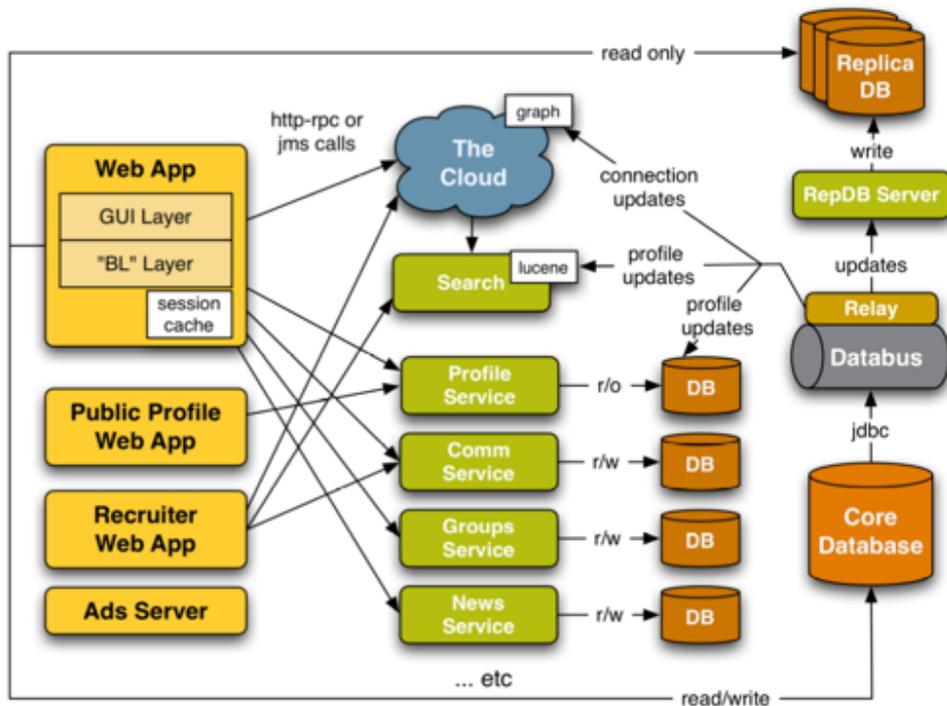


Figure 30: LinkedIn SA

As explained in Figure 30: "The LinkedIn SA" the social function is divided in two services: the profile and the communication service. Both services are linked with the database and with the Web App to enable exchanging information.

4.4.10. T10: How are we going to implement the wallet component that will share the end-users' data with external websites in the software architecture?

We will need to implement the wallet function in the platform to permit end-users to share their information with external websites in a simple way. In this section we analyzed just one solution domain because S18 provides a perfect solution for this technical problem.

4.4.10.1. S18: Google authorization SA

Google offers a similar service that has similar functions to our desired wallet function. Its platform permits sharing of information with third parties and websites. This information includes the authentication data and information about the Google account.

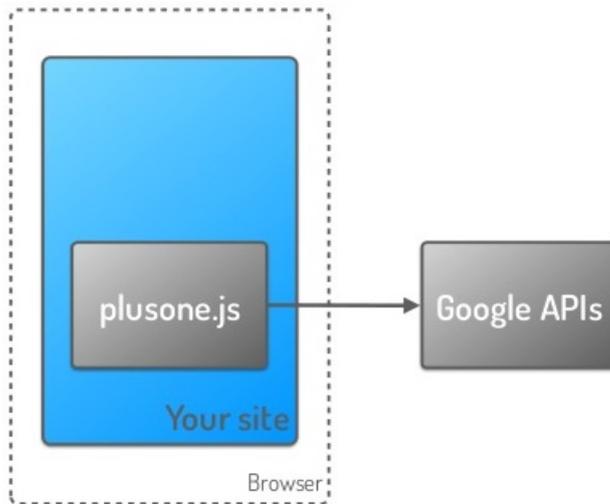


Figure 31: Cross-Platform Authentication with Google+ Sign-In

In Figure 31 ("Cross-Platform Authentication with Google+ Sign-In,") it is explained how the Google SA handles this function. The external website connects with the Google APIs to ask for authentication of the end-user. Then a window is shown to the end-user to ask permission to share data with the external website. Finally the data is shared with the external website. The plusone.js JavaScript library is a library that needs to be imported in the external website to be able to use the wallet function.

4.4.11. T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

The platform will need a suitable architecture that will arrange the different components in the system. This architecture will need to be reliable and suitable for all the components involved. In this section we analyzed just one solution domain because S19 "SOA architecture" provides a perfect solution for this technical problem

4.4.11.1. S19: SOA architecture

Following Microsoft (Wilkes & Forum, 2004) a solution to the question of the possible architecture is to use a Service Oriented Architecture. With a service oriented architecture the system will exchange information with the different layers in an independent way. They will connect to the different services inside the system using a SOA architecture.

There are multiple ways to implement a SOA architecture, for instance the Jboss platform (<http://www.jboss.org/>) is free and open source service oriented architecture software. It enables enterprises to automate business processes, link IT resources with data, services and applications and to integrate services.

Microsoft also released an architectural pattern for implementing the SOA architecture ("What Is an Enterprise Service Bus?"). This system provides an endpoint run-time discovery and virtualization, a loosely coupled service composition, a dynamic message transformation and translation and centralized exception management. The license of the system is the standard, free, Microsoft binary-only license.

4.5. Solution analysis

4.5.1. T1: How is the platform going to provide the different levels of authentication for end-users?

For this technical problem we will choose the Apache Syncope technology as described in section 2.2.2.9.

This technology permits to connect to an external database to store the authentication information. It permits to connect the authentication layer with the other parts of our architecture by providing a domain integration. It also permits managing the end-users' details from a web interface. Finally, it is free and open source so it will be cost effective to implement in our platform.

It will need to connect with a database to store the information, with the other parts of the platform to provide the authentication information and with the user interface.

4.5.2. T2: How will the system store the authentication data in a reliable way for the authentication component?

We decided to use the MySQL database due to better compliance on the official SQL language and because it has high support from a large community of end-users (Mikoluk, 2014).

4.5.3. T3: How are we going to implement the user interface, to connect the different components to it and also provide a mobile and a tablet version of it?

We decided to use the Bootstrap technology because it is free, open-source and widely used. The community behind this software is huge and it has a lot of documentation that supports it.

It satisfies the technical concern by providing a simplified way to implement our interface and the possibility to design the interface for mobile and tablet browsing by modifying different CSS styles.

4.5.4. T4: How is the platform going to connect to external data sources?

We decided to use a mix of the solutions provided in the previous section for this technical problem.

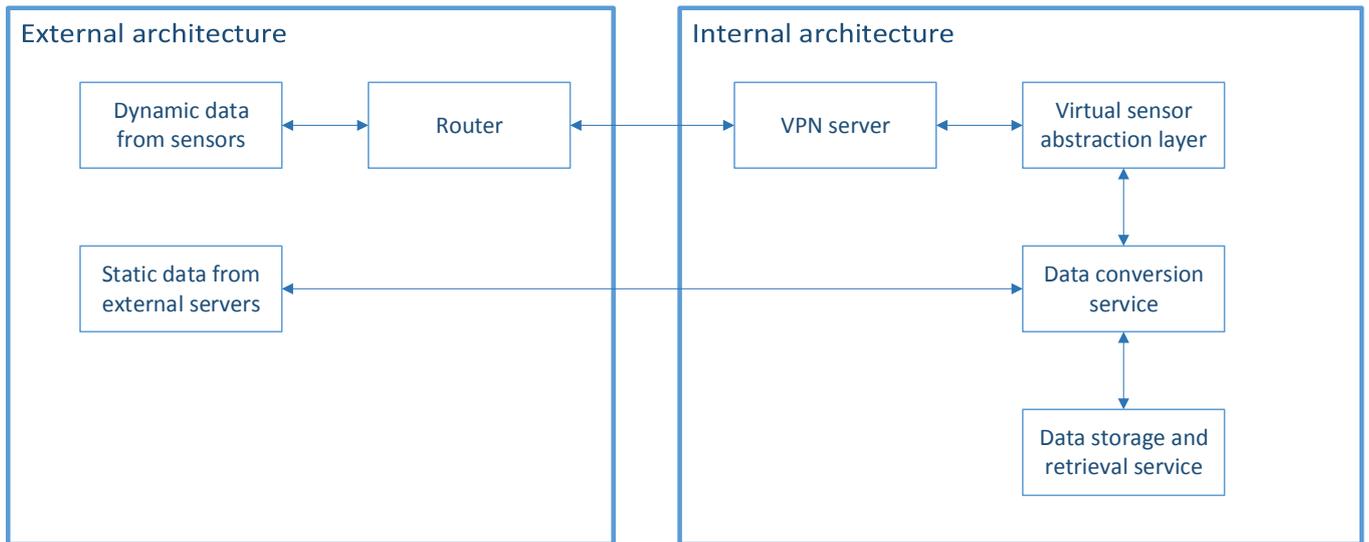


Figure 32: SA to connect the platform with external data sources

To import the data from dynamic sources we will connect the sensors to local routers that will in turn connect to our servers via a VPN. This will increase the security of the connections. Then the data is sent to the virtual sensor abstraction layer that manages the stream of data. Then the data is converted into linked data by the Data conversion service and finally the data is stored in our platform.

To import static data from external data sources we will connect the architecture to the external data sources. Then the data is converted into linked data by the Data conversion service and finally the data is stored in our platform.

4.5.5. T5: How will the system store the end-users' data?

To solve this technical problem we decided to use a mix of the solutions mentioned in the previous section.

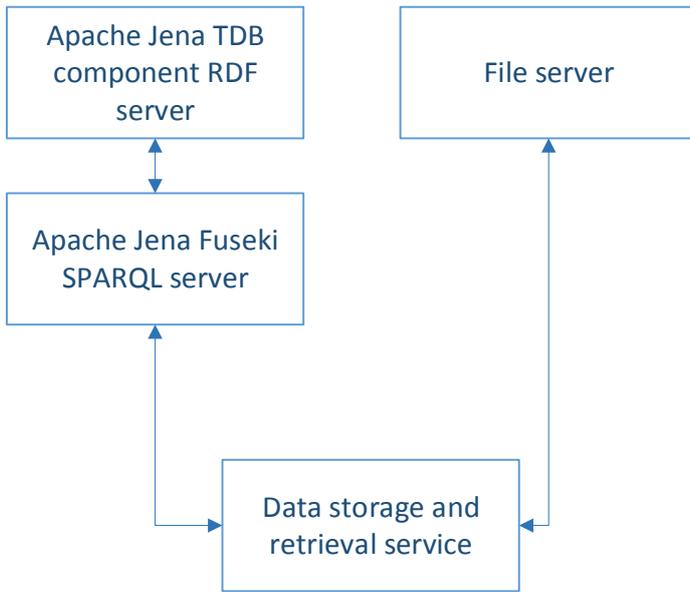


Figure 33: SA of the storing data component of the platform

As described in Figure 33 there will be two different servers that store the end-users' data. The linked data will be stored in the Apache Jena TDB RDF server and the files of the end-users will be stored in a file server.

When data needs to be inserted or retrieved, the corresponding service of our platform will contact the two different types of data servers. In case of linked data, the service will contact the Apache Jena Fuseki SPARQL server that will then contact the Apache Jena TDB RDF server. In case of binary files the service will contact the file server.

4.5.6. T6: How are we going to integrate the external services in the platform?

We decided to use the structure of MyPHRMachines to manage the external services. There will be a component that manages the services in the platform. This component will connect with the data storing and retrieving service, with the user interface service and with the other services of the platform.

4.5.7. T7: How are we going to develop the news feed for every end-user?

We will follow the software architecture made by LinkedIn due to its speed and reliability.



Figure 34: Newsfeed SA

As explained in Figure 34 the Web interface layer will connect directly to the newsfeed service. The newsfeed service will then connect with the data storage and retrieval service to retrieve news that will be shown to the end-user.

Every time that a news item is spread it is also stored in the data storage. Each piece of news will be associated with end-users. This way it is possible to respond in a quick way to the news retrieval queries.

4.5.8. T8: How are we going to implement a reliable alert function?

For this technical problem we will use the MissionMode solution. We have chosen this solution because we already analyzed the problem with help from that company and we already analyzed the alarm sharing use case with MissionMode support.

4.5.9. T9: How are we going to implement the social component in the software architecture that will permit end-users to connect with other end-users in the platform?

We will follow the software architecture made by LinkedIn due to its speed and reliability.

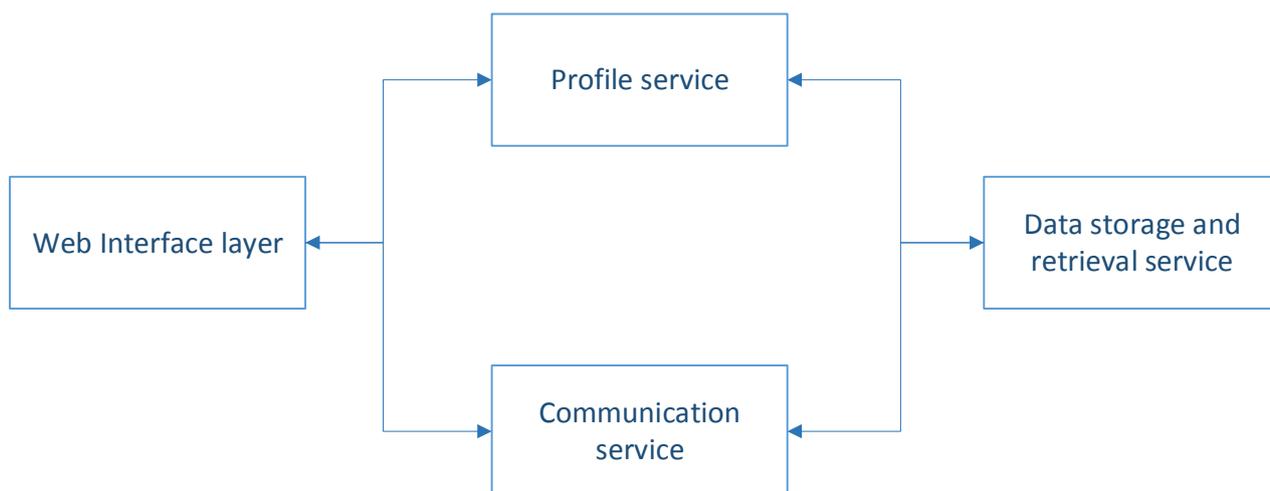


Figure 35: Social component SA

In Figure 35 the social component of our SA is shown. The first component is the profile service that will manage the profiles of end-users registered in the platform. Each profile will contain details of the person such as address, name and photo. The second component is the communication service that will manage the communication between end-users. End-users will be able to send to each other messages and to chat in the platform.

Both services are connected with the Data storage and retrieval service, to enable storing and retrieving information about the end-users' profiles and their messages.

Both services are also connected with the web interface layer that shows information about the services to end-users.

4.5.10. T10: How are we going to implement the wallet component that will share the end-users' data with external websites in the software architecture?

We choose to use the SA of Google due to its reliability, performance and success.

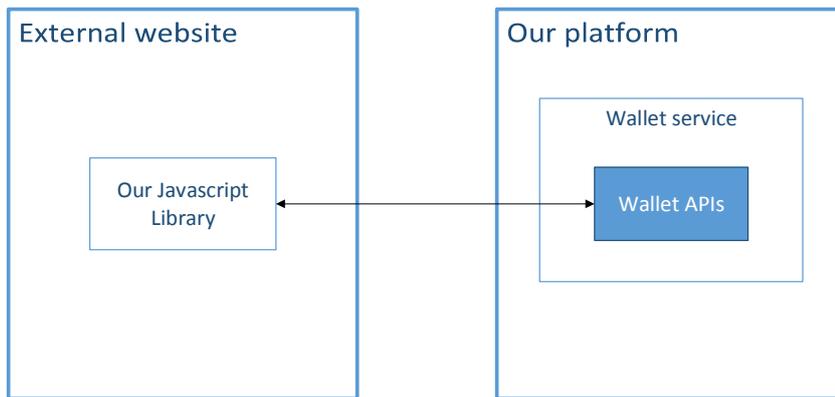


Figure 36: Wallet function SA

As is explained in Figure 36, the external websites will need to import our JS library in their code to access our wallet function. Then, when the external website needs to authenticate an end-user with our platform and to take information about the end-user from our platform it needs to connect to our wallet APIs that are managed from the wallet service. Then the wallet service asks the end-user if he wants to share the information with the external website and finally it will share the requested information to it.

4.5.11. T11: What will be a suitable architecture for our system that provides a secure and reliable system and that will arrange all the components within the system?

The best architecture for our system will be the SOA architecture because it will permit exchanging information with the different layers of the architecture in an independent way.

It will have a better return on investment because the different services work independently (Stevens, 2002). The code mobility will be improved because the different services can be moved to different machines or even to external providers (Stevens, 2002).

The roles of the developers will be more focused because each layer of the SOA architecture will need to be programmed by different programmers with specific skills. Testing of the application will be simpler because services have published interfaces that can be tested in an easy way by the developers.

The support for multiple client types is another advantage because there are multiple ways to access the services in the architecture. All the services implemented will be listed in a catalog.

To design a suitable SOA architecture we analyzed four examples of SA of companies that use this type of architecture in their system. Then we made a commonality and a variability analysis of the different architectures

to find out the parts of the architecture that need to be implemented in our architecture. Finally, we will show the SOA architecture made from this analysis.

The solution domain of the SOA architecture is quite wide. As already explained before, the SOA architecture has a lot of advantages compared to the other types of architecture. The SOA system consists of different services that can connect with external applications. Every service is independent from the other layers. This allows a better and simpler implementation of the system.

In this section we will take four already existing SOA architectures, analyze and compare them by finding their commonalities and the variables. Finally we will provide a final architecture, based on the findings of the analysis.

The four SOA architectures that are used as example are:

- the Board software architecture ("Board Service Oriented Architecture,")
- the COMM-IT software architecture ("Education - Campus Management System,")
- the PNMsoft software architecture ("What is Service Oriented Architecture?,")
- the SOA component scheme from the book "Enterprise SOA" (Krafzig, Banke, & Slama, 2005)

The Board software architecture

The first one is the "Board" software ("Board Service Oriented Architecture,") and its system's architecture is shown in the next figure:

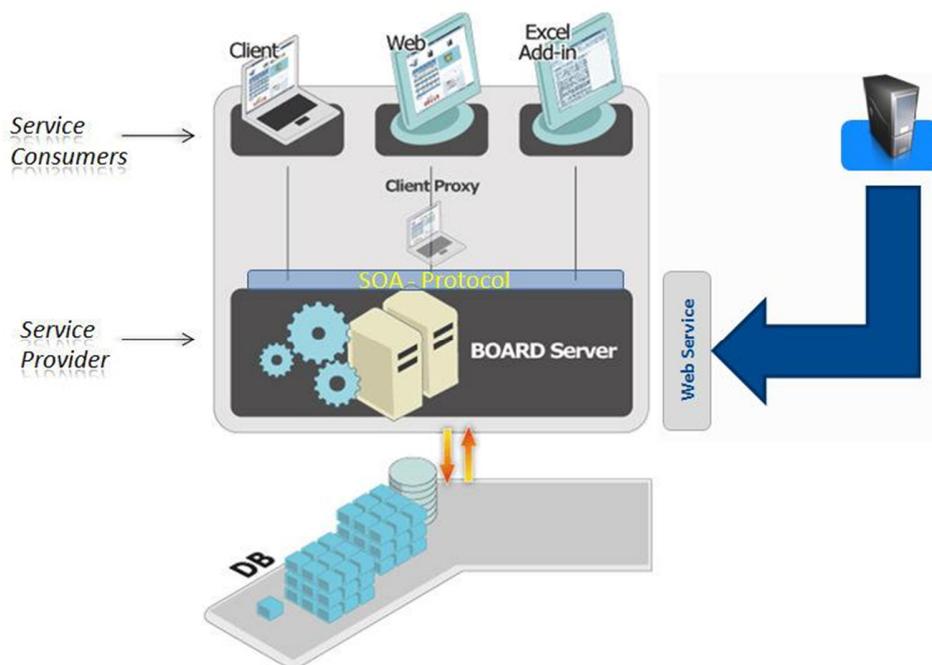


Figure 37: Board SA

As explained in Figure 37 the different clients access the board with a SOA protocol that connects directly with the different services of the system. The system then connects to an external database. All services of the system are implemented in the BOARD Server that uses SOA architecture.

COMM-IT Education-Campus management software architecture

The second system that is taken as a reference is the COMM-IT Education-Campus management system ("Education - Campus Management System") that is also based on SOA architecture.

In this architecture the devices are connect to the system with interfaces that are different for each type of end-user that needs to access the system. Then these interfaces are connected through a "XML web service layer" that connects them to the services of the system.

After that, the service layer connects with the external services, such as the reporting component and the file server.

PNMsoft software architecture

The third example that we will take as a model is from PNMsoft ("What is Service Oriented Architecture?") and it has the following architecture.

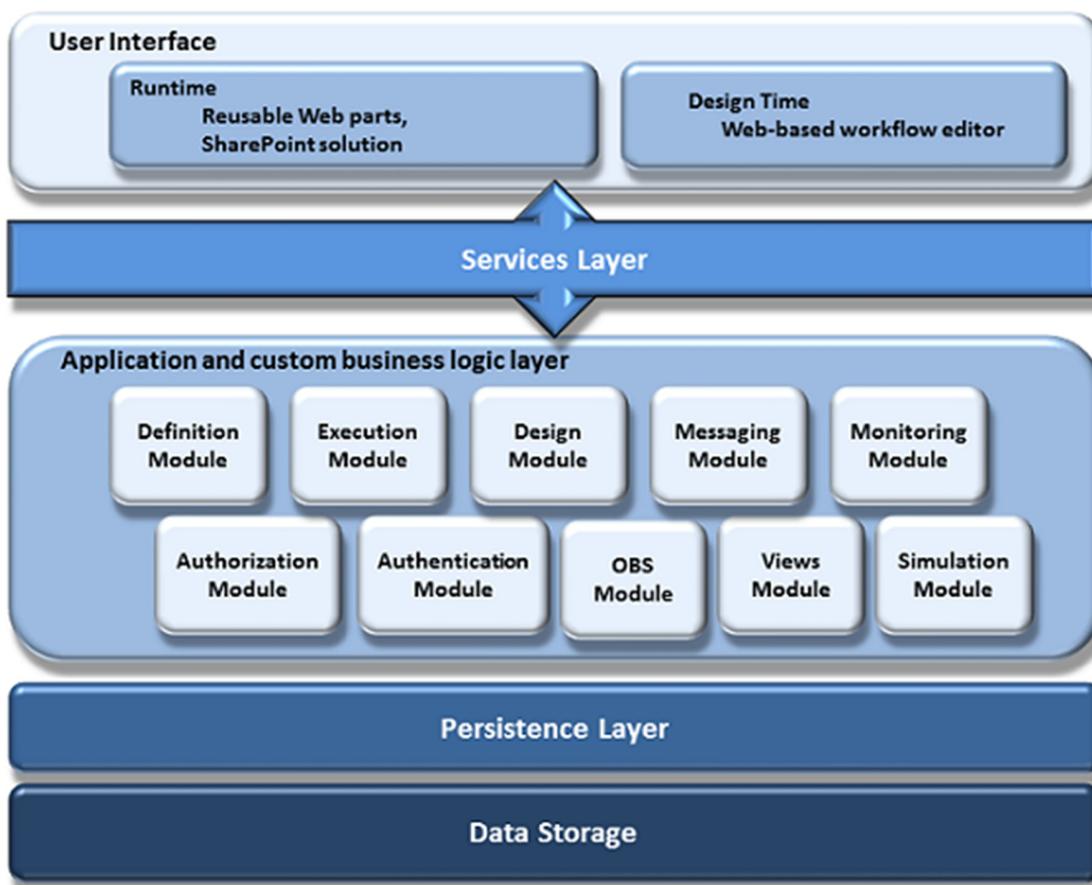


Figure 38: PNMsoft SA

As the reader can see the user interface is connected to the services through a service layer. Then the services are connected to a persistence layer and a data storage layer.

Enterprise SOA

The fourth example that serves as an inspiration for our SA is from the book Enterprise SOA (Krafzig et al., 2005). In this book the author presents the components that a SOA architecture should have.

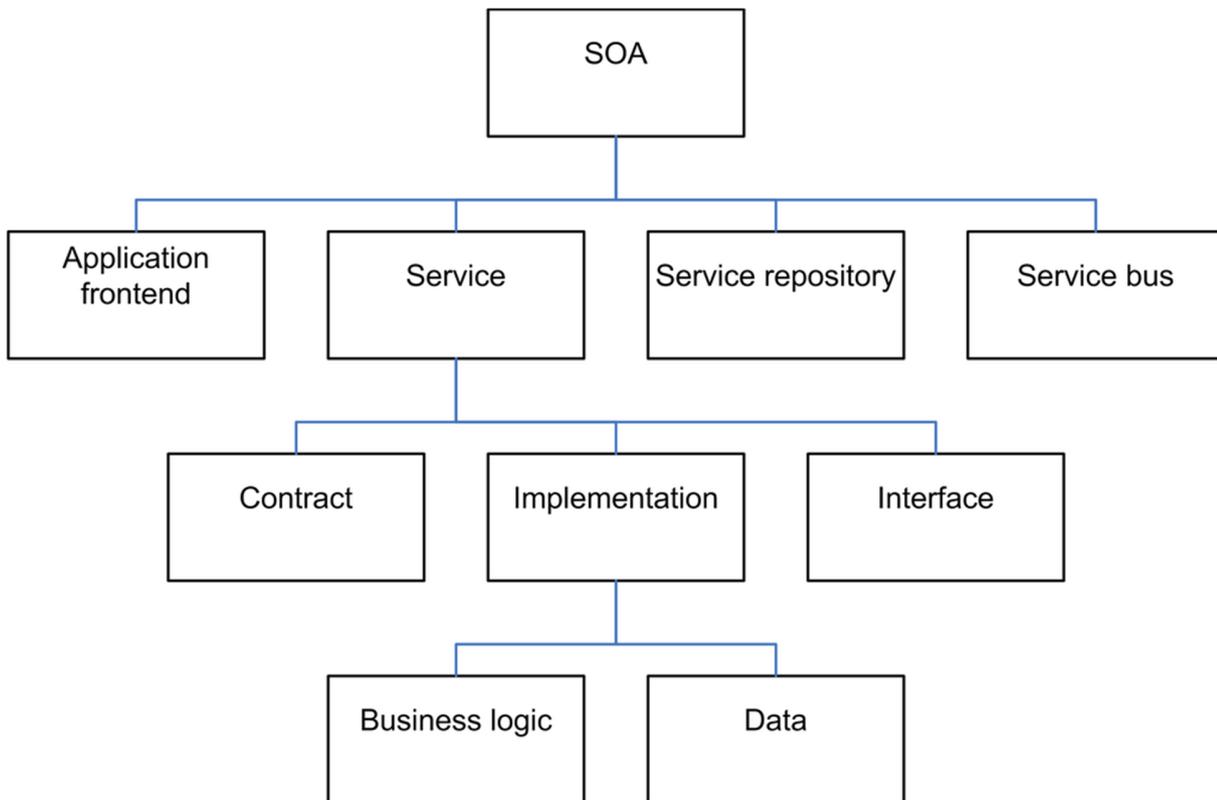


Figure 39: Different components of a SOA SA

In Figure 39 the different components of a SOA architecture are shown, including how they link with each other.

4.5.11.1. Commonality and variability analysis

In this section we will analyze the commonality and the variability of the previous SOA architecture examples.

Commonality analysis:

All the previous architectures have the following parts in common:

- They have a central system (that we will call "service layer" in Figure 41) that manages the different services. This central system that manages all the services is essential because it runs all the services. As the reader can see all the architectures mentioned in the previous section have this central service system so it is essential in a SOA architecture.
- They have an user interface (that is called "client layer" in Figure 41) that connects with the different services. This user interface is essential because it enables connecting the different devices to the central system. All the architectures mentioned have an user interface that connects to the central system.

However, it has several names, depending on the system. For instance, in the BOARD software this layer is called "service customers".

- They have servers (for example the database and the file server) that connect with the different services (that is called the "physical layer" in Figure 41). This external physical layer is essential because it permits to connect the external servers with the central system. Also, using an external physical layer, the external servers can be managed independently.

Variability analysis

The different architectures mentioned have the following variability:

- The COMM-IT has different types of devices that connect with the application layer. In our system we will manage the different devices connected to the system in a separate layer (the clients layer).
- The scheme from the Enterprise SOA book has a detailed description of the parts of a service. In our system we assume that the contract, the implementation and the interface of a service are embedded in the different services.
- In the COMM-IT example there are more external servers connected with the system, such as the OLTP and the reporting server, than in the other examples. In our system we decided to implement the different external servers in the physical layer.
- The COMM-IT example has the XML web service layer between the presentation layer and the service layer. The XML web service layer parses the input that comes from the presentation layer and sends it to the service layer. The XML web service layer is also embedded in the architecture of the other examples but it is not directly mentioned.

4.5.11.2. The architecture extracted

The next figure summarizes the architecture extracted from the previous section.

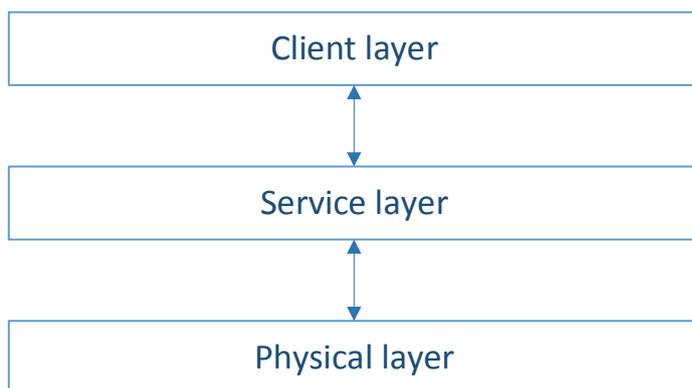


Figure 40: Overall SOA architecture

The client layer will contain the different user interfaces of the platform. We decided to implement it because all the SOA architectures mentioned in section 4.5.11 contain a client layer.

The service layer will contain and manage the different services of the platform. We decided to implement it because all the SOA architectures mentioned in section 4.5.11 contain a service layer and so it is essential in SOA architecture.

The physical layer will contain all the external servers of the platform, such as the database server and file server. We decided to implement it because all the SOA architectures mentioned in section 4.5.11 contain a physical layer and so it is essential in SOA architecture.

4.6. Synthesis of the overall architecture

This section presents the results of the previous sections: the complete SA of our platform can be found here.

The following figure explains the SA of our platform.

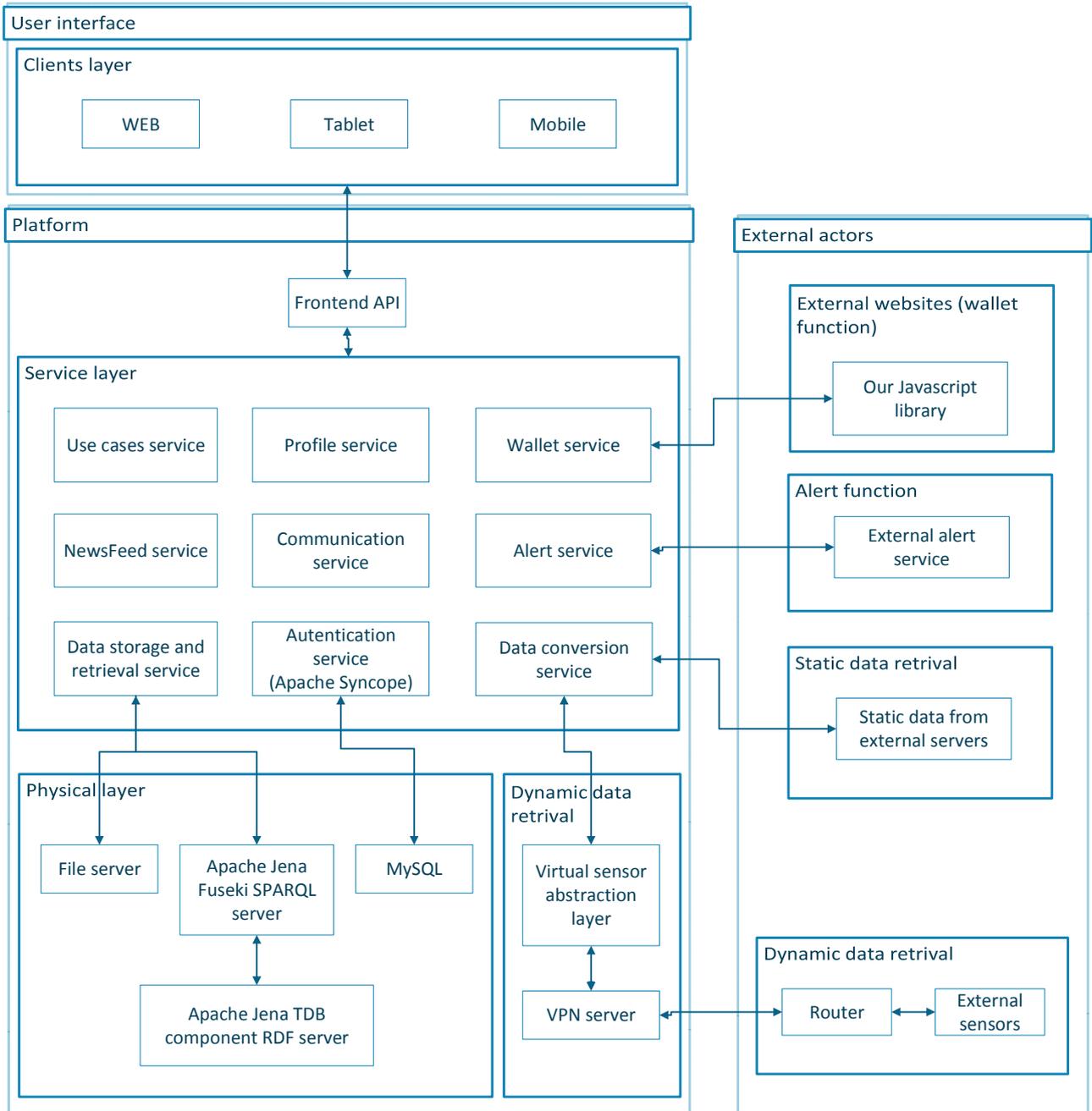


Figure 41: Complete SA of the platform

As shown in the previous figure, the SA is divided into three main parts, the "user interface", the "platform" and the "external actors".

In the "user interface" the interface for end-users is found. It contains the client layer where interfaces for the different types of devices are provided.

The "platform" contains all the components of our platform. It contains the frontend component, the service layer, the physical layer and the dynamic data retrieval layer.

In the "external actors" part the different external servers with which our platform will connect are placed. It contains external websites for the wallet function, the external alert function, the external static data retrieval and the external dynamic data retrieval functions.

On the next pages we will explain in detail every part of the SA. Figure 41 is divided into several sub-figures that show the different parts of the SA.

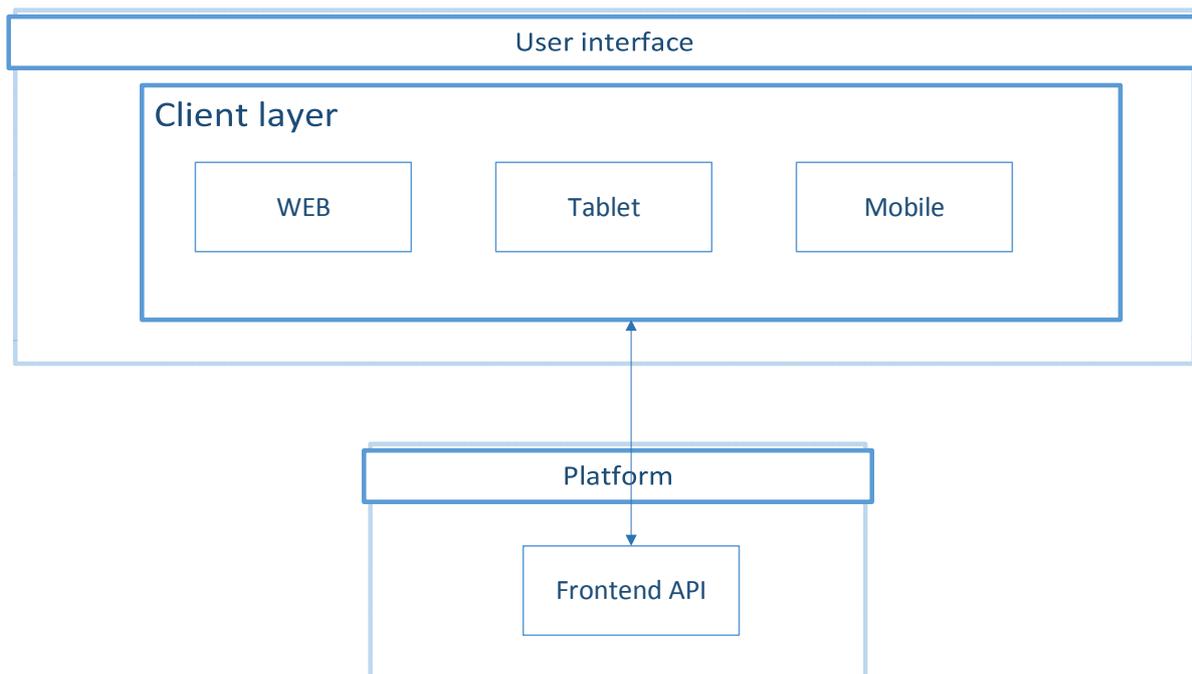


Figure 42: Client layer and frontend component

In the previous figure the client layer is shown that connects with the frontend API component of our platform. The client layer is divided into the WEB, tablet and mobile user interfaces. Each type of user interface has a different CSS style that will be able to adapt the user interface to different devices.

The user interface is connected with the frontend API component of the platform. They exchange HTML and REST/SOAP data over the HTTP protocol.

The frontend API component will include a web server that provides the HTML and CSS data to end-users.

The frontend API component connects to the services of the platform to provide the user interface. As explained in section 4.4.3.1, we decided to use Bootstrap as a design HTML tool to provide the different CSS for the different types of clients that will connect to the platform.

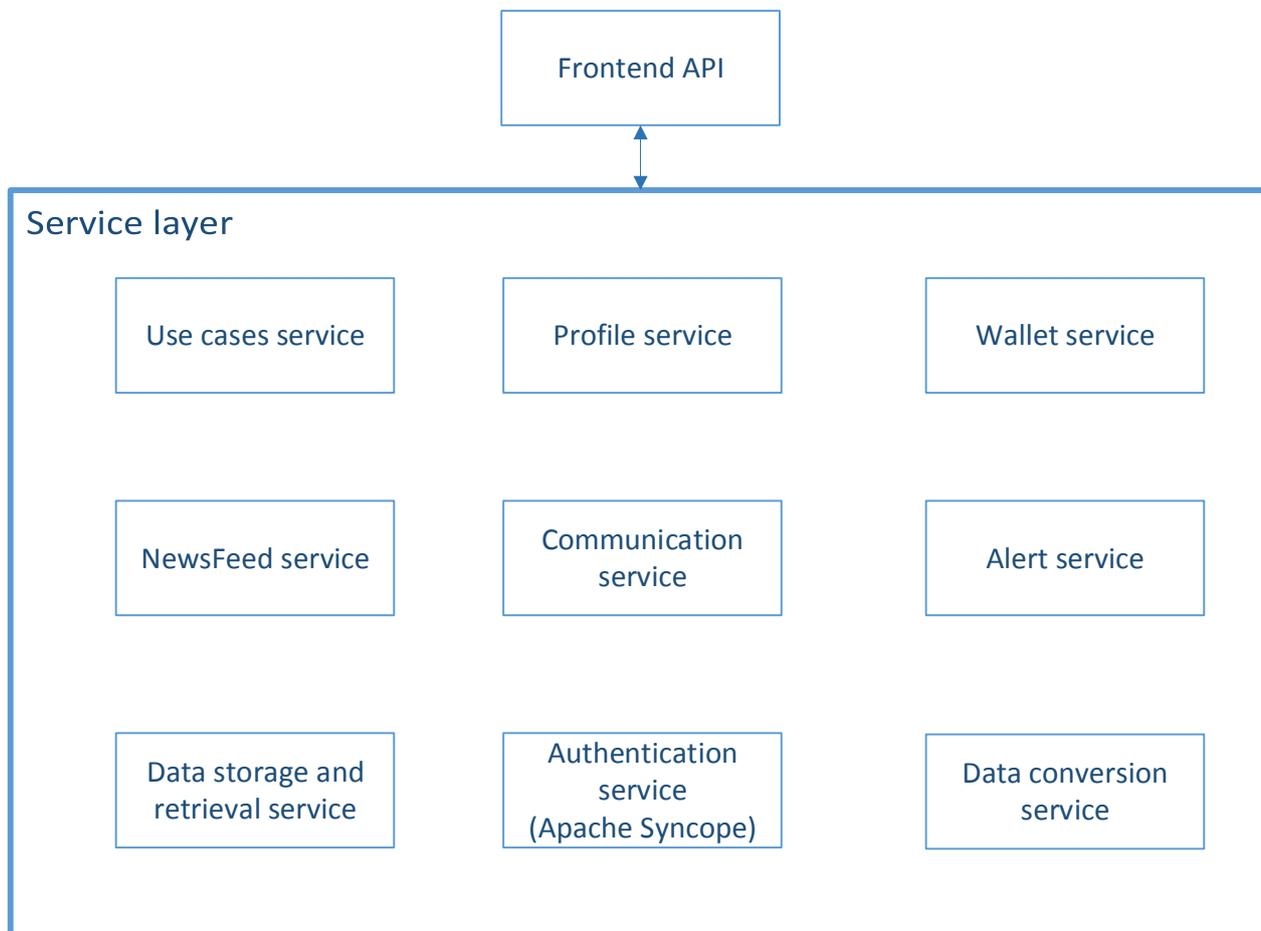


Figure 43: Service layer and the frontend component

As explained in the previous figure, the service layer is connected to the frontend API component. They exchange REST and SOAP requests to provide the user interface to end-users.

The service layer contains all the services of the platform. These components are stateless and they can handle multiple requests at the same time.

In the next table a summary of the list of services, their functionalities and their input/output capabilities is shown.

Name	Functionalities	Input	Output
Use case service	Manages the use cases of the platform	User's data, use cases activation event	User's data to the frontend, end-user's data to store, data for the alert service.

Profile service	Manages the profile of the end-user	User's data, end-user's profile data requests	User's data to the frontend, end-user's data to store.
Wallet service	Allows sharing of information by the end-user with external websites	Website requests	User's data to external websites
Newsfeed service	Provides the news to show in the newsfeed of end-users	News list request	News list for every end-user
Communication service	Provides the messaging between the end-users	Text and recipients of the message	Messages to every end-user
Alert service	Provides the alerts for each end-user	Alerts text (that will be a string) and recipients	Alerts text and recipients to the external alert service
Data storage and retrieval service	Provides the data storage and retrieval for every end-user	Data input, request input	Storing and retrieving queries, data retrieved
Authentication service	Provides different levels of authentication for each end-user, provides account creation and managing of the platform	Authentication requests, account creation and modification requests	Authentication responses, data about the account creation and update
Data conversion service	Provides the conversion between different types of data and linked data	Static and dynamic data	Linked data

Table 32: Service layer details

The next figure explains how the data storage and retrieval service is connected with the file service and the Apache Jena Fuseki SPARQL server.

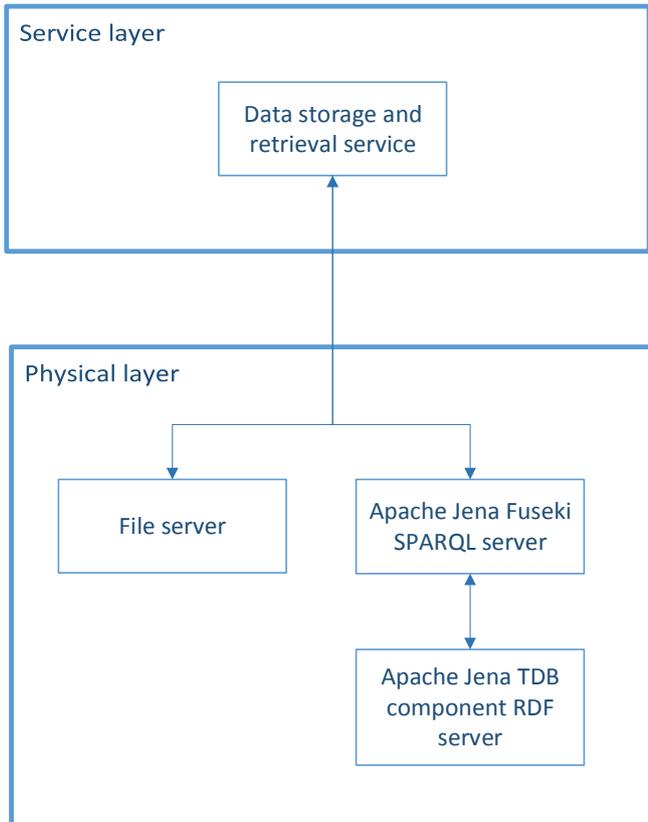


Figure 44: Data storage and retrieval service, file server and SPARQL server

The data storage and retrieval service provides the insertion, update and retrieval of data with the File and the SPARQL servers. It exchanges the files that need to be stored with the file server. It exchanges linked data that need to be stored with the SPARQL server. Then the SPARQL server stores the linked data in the Apache Jena TDB RDF server.

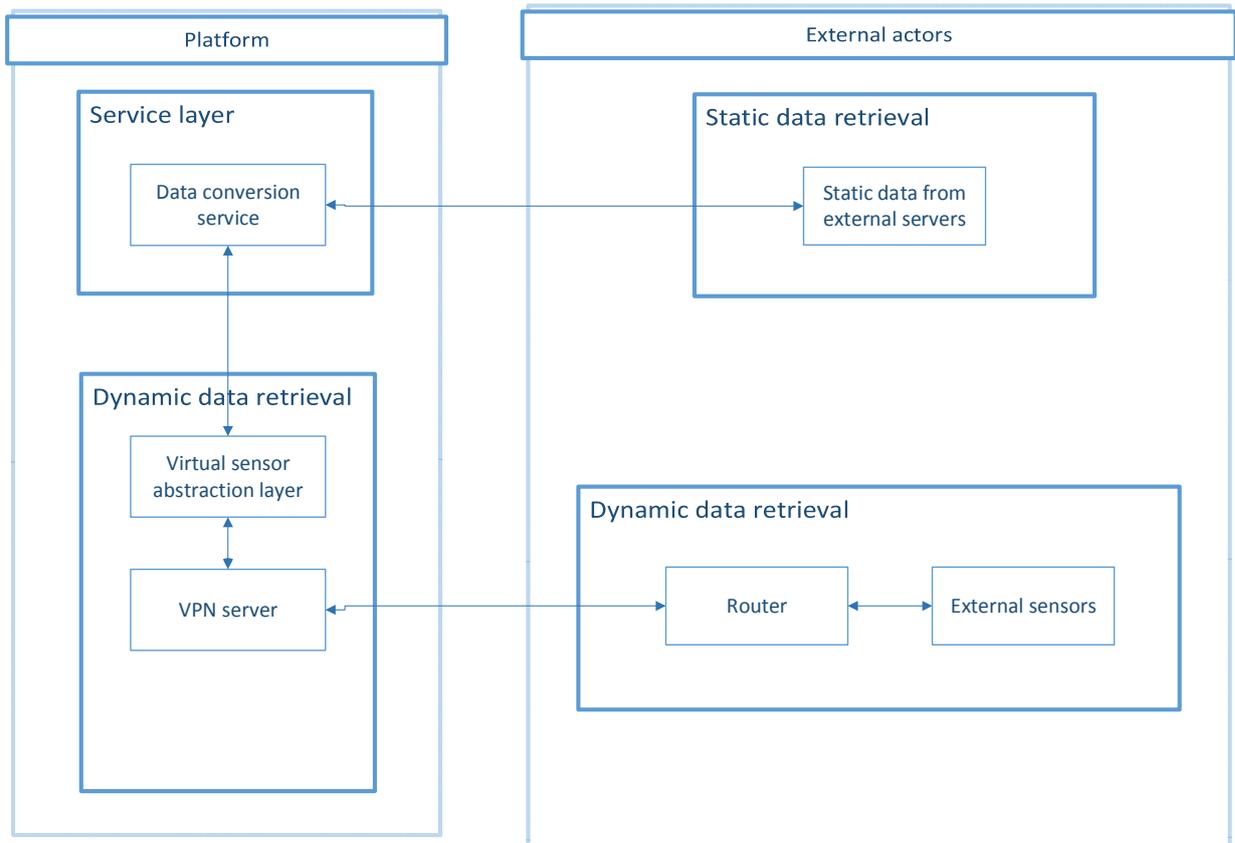


Figure 45: Data conversion service and static and dynamic external data sources

Figure 45 shows the connection between the data conversion service and the static and the dynamic external data sources.

For the static data sources, the data conversion service imports the data directly from the external servers. The information that is exchanged are queries and data.

For the dynamic data sources, as is explained in section 4.5.4, the data is sent from the sensors to the routers, which are connected to the internet. The data that is exchanged in this process are queries and data from the sensors. Then the data is sent from the router to our VPN server to increase security. Then the data is sent to the virtual sensor abstraction layer and finally to the data conversion service that will convert it into linked data. All the data that passes through these connections are queries and data from the sensors.

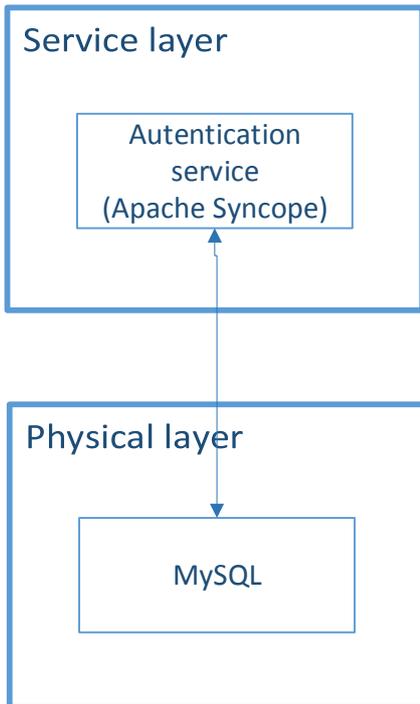


Figure 46: Authentication service and MySQL

In the previous figure the connection between the authentication service (implemented with Apache Syncope) and the database MySQL is shown. The information exchanged over this connection consists of queries and authentication data.

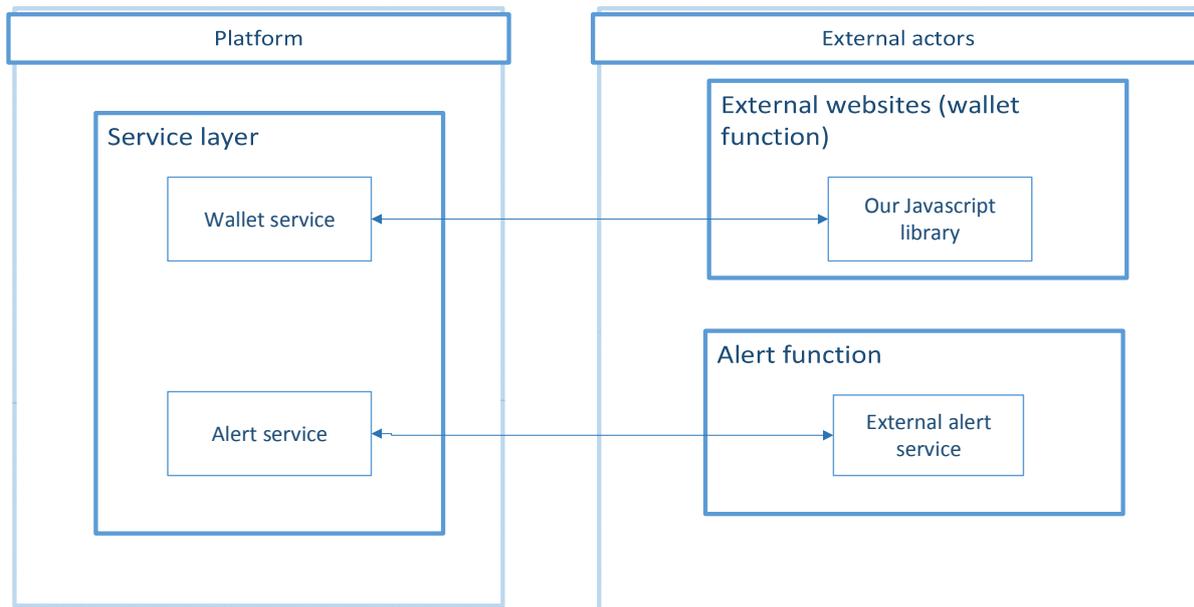


Figure 47: Wallet and alert services connection with external actors

Figure 47 shows two different connections: the wallet service with the external websites connection and the alert service to the external alert function.

The wallet service connects with external websites through implementation of a JavaScript library in the code of the external websites. The data that is exchanged over this connection consists of queries, end-user's data and authentication data.

The alert service connects with the external alert service. The data that is exchanged is the authentication data, the text and the recipients of the alerts. In the first release of the platform the external alert service will be MM.

4.7. Scenario analysis

In this section a scenario of the usage of the platform is presented. Then, following this scenario, a detailed description of the flow of information in the SA (which is explained in the previous section) is given. This section is helpful for the reader to have an example of how the SA of our platform will really work.

The scenario that we analyzed is of an end-user that performs a correct login in the platform and then the platform shows him the security information linked to his address in the interface.

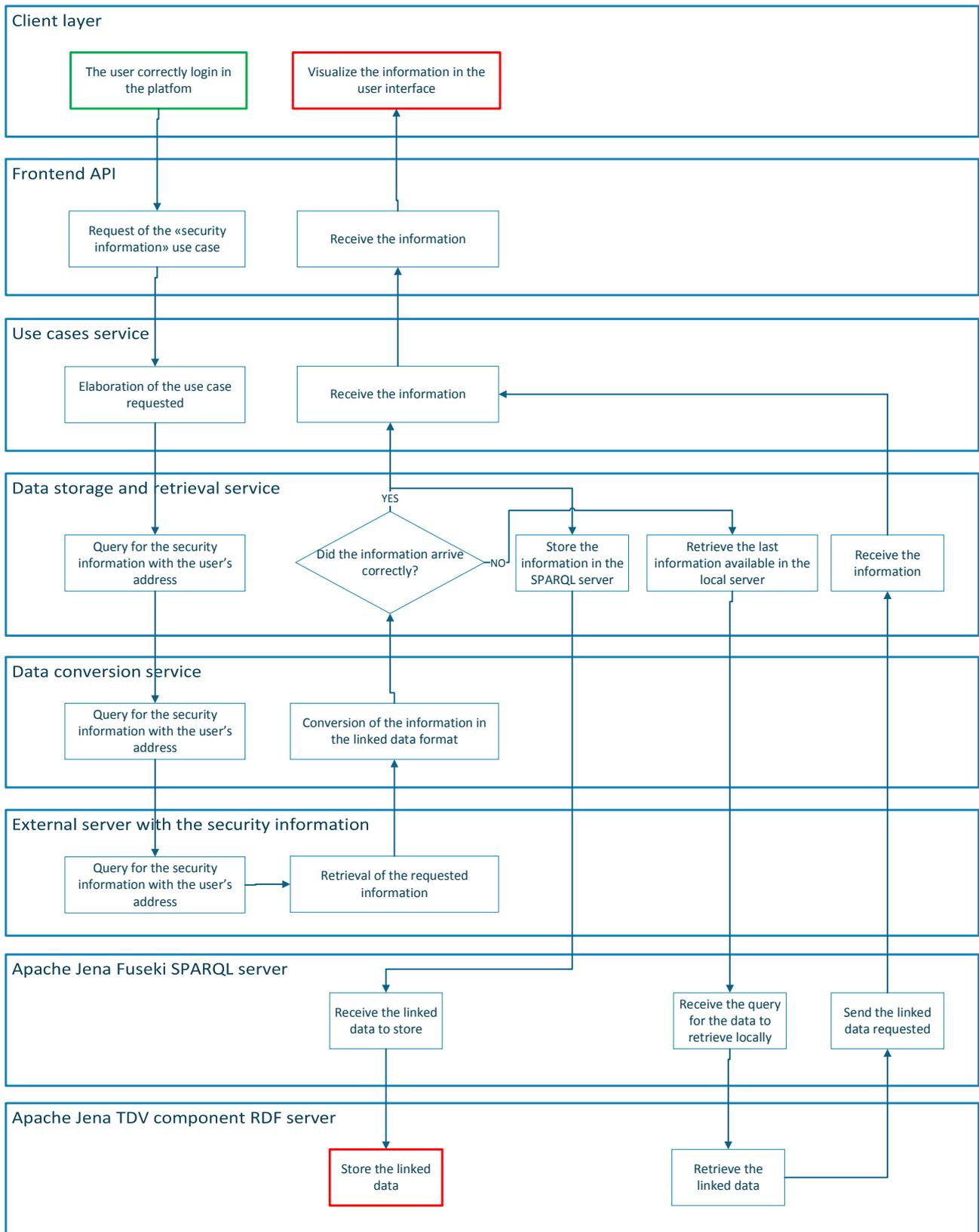


Figure 48: Scenario process scheme

In the previous figure the process analysis of the scenario mentioned is visualized. The blue rectangles represent intermediate processes, the green rectangle represents the initial process and the red rectangles represent the end process.

The scenario begins with the end-user that correctly logs in on the platform, so it is assumed that the login process is already done correctly. Then the "frontend API" component receives the request for the user interface and, by analyzing the different data needed to make the user interface, it finds out that the "security information" use case is needed. Then the "use cases" service receives the request of the mentioned case and elaborates it.

Then the "use case" service sends the query to request security information to the "data storage and retrieval" service, adding the information that is needed and the address of the end-user. Then the request is sent to the "data conversion" service that queries the external server to retrieve the information needed. Then the data is sent back to the "data conversion" service that will convert the data into linked data. Then the "data conversion" service sends the linked and elaborated data to the "data storage and retrieval service". If the retrieved data contains an error it is sent to the "data storage and retrieval service".

Then the "data storage and retrieval" service analyzes the received data. If the data is correct, it is sent to both the "use case service" and to the "Apache Jena Fuseki SPARQL" server. From the "use case" service the data is then sent to the "frontend" component and then to the user interface of the end-user. From the "Apache Jena Fuseki SPARQL" server the data is sent to the "Apache Jena TDV component RDF" server to store it.

In case the received data is not correct, the "data storage and retrieval" service retrieves the last data stored in the system by sending a query to the "Apache Jena Fuseki SPARQL" server that retrieves the data from the "Apache Jena TDV component RDF". Then the data is sent back to the "data storage and retrieval" service and then to the "use case" service. From the "use case" service the data is sent then to the "frontend" component and then to the user interface of the end-user.

This scenario uses a "fall back cache" method to store and retrieve data. This method will be used for several data sources, depending on their reliability. We assume that for the data source used in this scenario we need this method.

5. Mockups of the UI

In this chapter parts of the mockups of the platform are shown and explained. Mockups of the user interface of the platform were made, and the different parts of the user interface are described. In this chapter just the mockups of the interface can be found. These mockups are just images that have no functionalities: their function is to explain how the UI will work on a basic level. In the next chapter a software prototype of the platform is presented that offers the basic functionalities of the UI. To make the mockups presented in this chapter Adobe Illustrator CC 2014 was used.

5.1. Interface description

5.1.1. Homepage

In this section the different parts of the homepage of the platform are shown and explained.

5.1.1.1. Navigation menu

In this part the menu with the categories of the different use cases will be shown. Each category will contain subcategories and then every subcategory will connect with a webpage, as is explained in the next section.

5.1.1.2. Submenu

For each subcategory, there will be a submenu where the end-user can see the most important information about that subcategory and the most important services connected with that subcategory. If the end-user wants to go to the subcategory webpage he will need to click on the "All the information" button.

5.1.1.3. User information and menu

In the upper part of the interface the name, city and country of the current end-user will be shown, and a submenu where the end-user will be able to access the setting page, the profile page and the log-out function.

5.1.1.4. Slider

Under the navigation menu there will be a slider that contains the main news of the platform. The slider will slide news that can be changed from the administration panel of the platform.

5.1.1.5. Alerts

Under the slider there will be an alerts box that contains the main alerts of the end-user. The alerts of the end-user will be important information that comes from the different use cases.

5.1.1.6. The newsfeed

In this section every alert that comes from the newsfeed service is shown. News is updated regularly. Availability of new services and advertisements of the functionalities are shown also in this feed.

This section will be similar to Facebook's main page where the end-user is able to see information that comes from different friends and pages.

5.1.1.7. Use cases list

In this section there will be a list of the services that the end-user is able to use. The main service list will be shown on the homepage. This service list contains just the main services that are available. Then in each subcategory webpage there will be the list of services connected with that subcategory.

5.1.1.8. Data from use cases

In this section the end-user will be able to see all the data connected with his profile that comes from a use case. On the homepage the most important data will be shown to the end-user. Then, on each subcategory webpage, more data is shown related to that subcategory.

5.1.2. Subcategories webpage

This webpage will contain the information and services related to each subcategory. Some of the menus are equal to the homepage so they will not be mentioned again in this section.

5.1.2.1. Information

In this part information related to the subcategory is shown. There will also be the possibility to show graphs and charts.

5.1.2.2. Use cases list

This section contains a list of the services that the end-user is able to use, related to the subcategory.

5.2. Mockups of the interface

In this section the reader will find mockups of the interface that the platform will have.

Each part of the interface is connected to a part of the software architecture of the platform and this information is shown in every mockup.

In this section just the main mockups are shown. For the complete and clean mockups please refer to Appendix C: mockups of the interface.

All the different mockups contain the "HuisKluis" logo: that will be the name of the future platform and that is also the name of the foundation where I did research for this thesis.

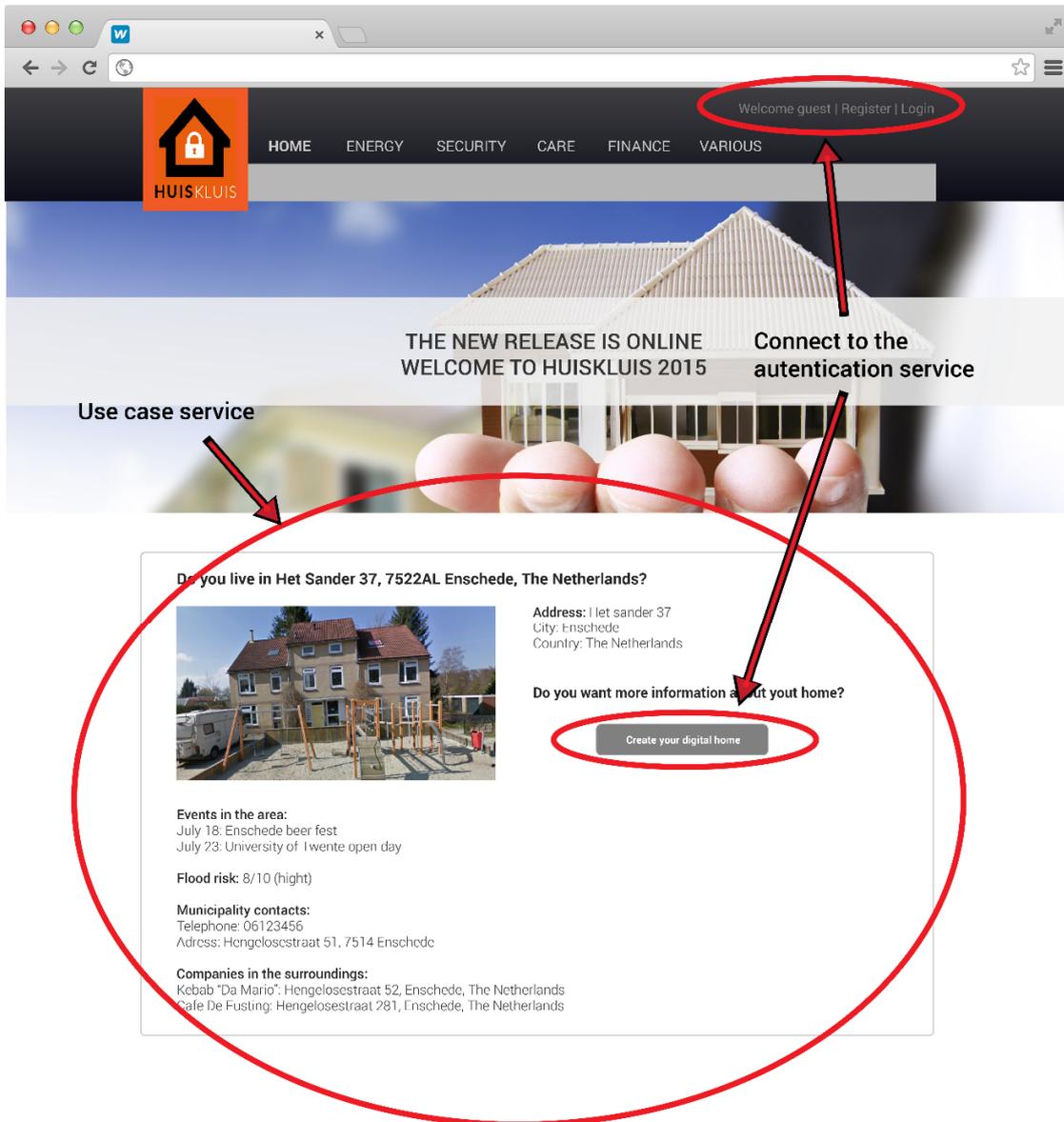


Figure 49: Homepage without the end-user logged

In Figure 49 the homepage of the platform is shown. In that mockup we assume that the end-user still did not register in the platform and that we have the referral data of his address. To get hold of this data we assume that the end-user searched in a search engine for his address and that he found our website. This way we will have the referral information of his search.

The two circles called "connect to the authentication service" mean that the two components will connect the end-user to a webpage that connects with the authentication service.

The circle connected with the “use case service” means that the information that is displayed comes from the “use case” service. The particular use case that shows the basic information connected with the address of the end-user is not explained in section 3.1 because we decided to make the mockups also with this additional use case. This use case just shows the basic information linked with an address.

The other parts of the interface are just HTML code that does not connect with any service.

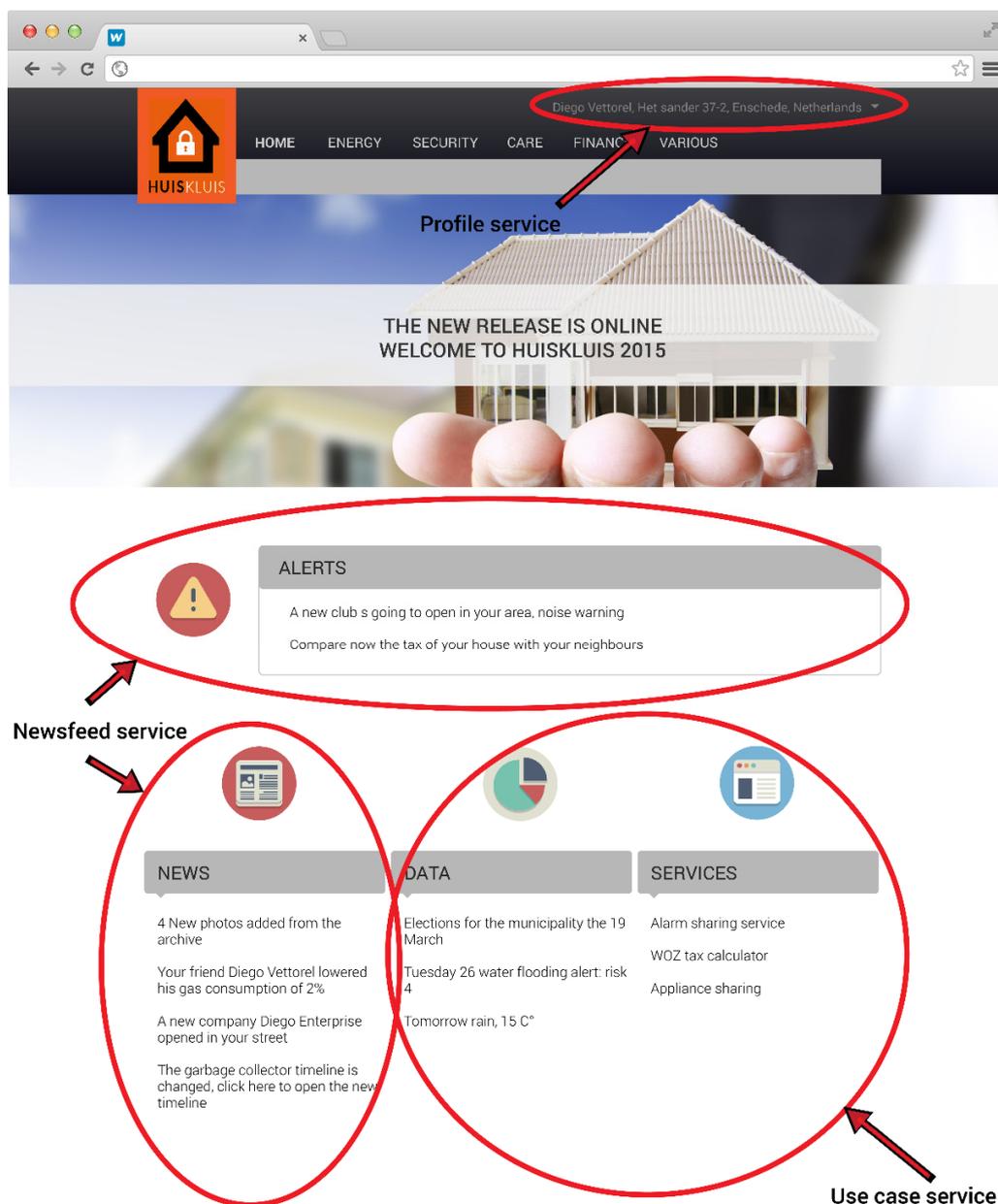


Figure 50: Homepage with the end-user logged

Figure 50 shows the homepage of the platform assuming that the end-user correctly logged in on the platform.

The circle connected with the "profile service" indicates that this end-user's information comes from the profile service of the platform.

The circles connected with the "newsfeed service" show that this information comes from the newsfeed service of the platform. This information is divided into "alerts" and "news". The alerts will contain the most important information and the news will contain more general information.

The two circles connected with the "use case service" show that this data comes from the use case service.

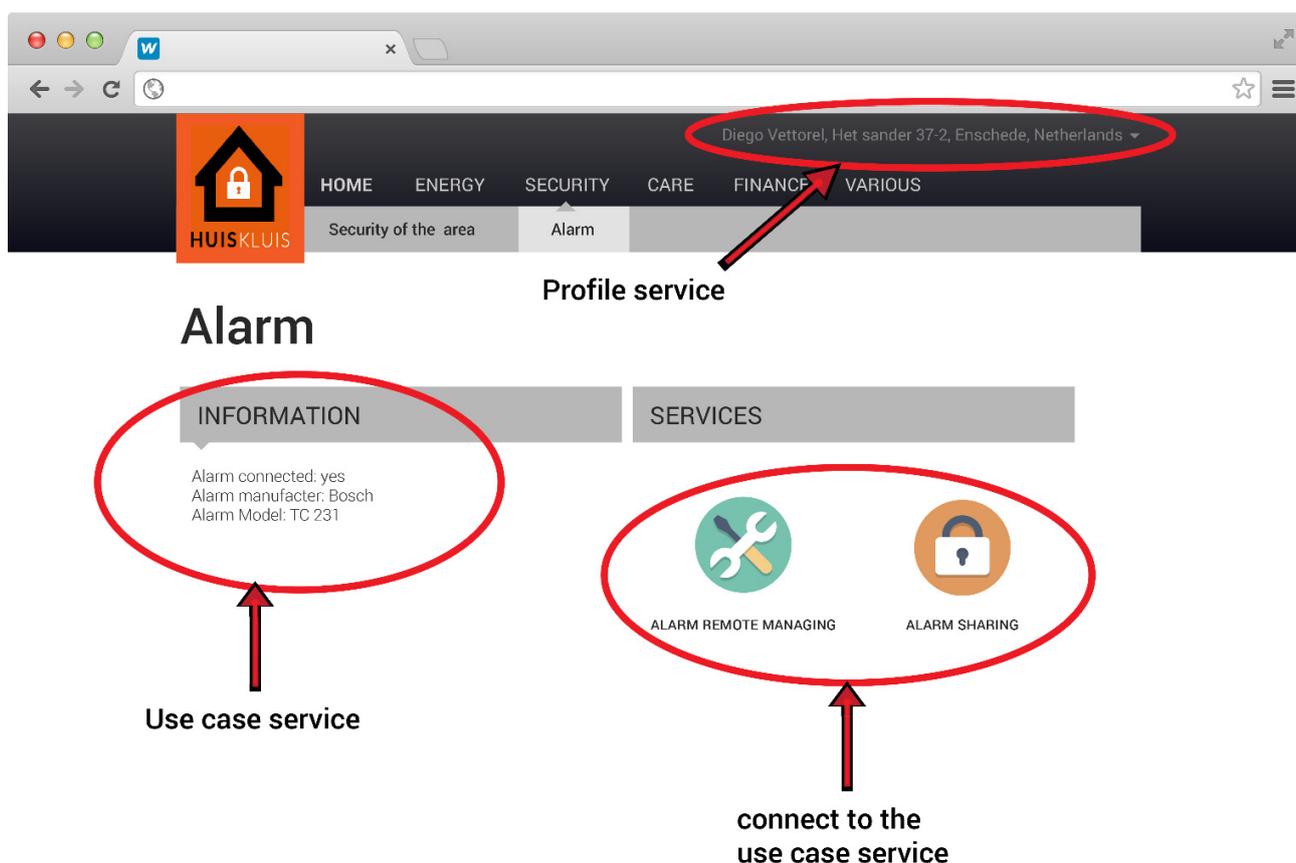


Figure 51: Alarm information and use case webpage

In Figure 51 the "alarm" webpage is shown, and also the information and the services about the alarm's category.

The circle surrounding the heading "profile service" shows that this end-user's information comes from the profile service of the platform.

The circle around the "use case service" means that this data comes from the use case service of the platform.

The circle called “connect with the use case service” means that the two services’ links will connect the end-user to a webpage that contains information that is connected with the “use case” service.

The “remote alarm managing” and the “alarm information” use cases (that you can see in Figure 51) are not mentioned in section 3.1 because we decided to include the output and the link of these additional use cases only in the mockups.

5.3. Analysis and testing of the UI

This section includes the results of testing the user interface and their analysis. This analysis is based on the mockups of the previous section and the mockups contained in Appendix C: mockups of the interface.

Following the method presented by Wieringa (Wieringa, 2014), we used “expert opinion[s]” to validate the artifact. Two people, both interface graphic designer, were interviewed and have given me feedback on the user interface. The main question was if they thought the user interface is user-friendly and well designed.

The questions that we asked them are listed in the next table.

Question #	Question text
1	Do you think that the interface is user-friendly and that it has a good usability level? How can we improve it?
2	Do you think that the color scheme is nice? How can we improve it?
3	Do you think that the menus are implemented and positioned correctly? How can we improve them?
4	Do you think that the different components of the user interface are well positioned? How can we improve them?

Table 33: Questions for interface graphic designers

They gave us useful feedback that we implemented in the mockups that you can see in this thesis.

In the next table the feedback that we got from them is summarized.

Question #	Feedback	Implementation
1	You can improve it by using the material design icons to describe the different components and services of the platform.	We implemented the material design icons in the mockups. Previously, we were using normal images that did not comply with the material design.
2	You can use a grey color scheme, that will present the user interface as a professional platform. The orange logo of “HuisKluis” will fit in this color scheme.	We used the grey color scheme for different parts of the interface.
3	The different menus are correctly implemented and positioned	We did not modify the mockups since the feedback was positive

4	You should put a slider on the homepage to show the last news of the platform	We implemented a slider on the homepage.
4	You should divide the different components of the user interface by putting the title in a rectangle and at the bottom of it the information about the components of the user interface	We used this system to design the different parts of the interface in the homepage and in the subcategories webpages.

Table 34: Interface graphic designers feedback

6. Software prototype

After the realization of the mockups (shown in the previous chapter), we made a complete software prototype of the platform. This software prototype can be found on the web address <http://huiskluis.altervista.org/> (this web address is subject to change so, if you need more information about the software prototype, please contact the reference people mentioned in the preface of this thesis).

This software prototype was made by using Axure RP 7.0 prototyping software. It shows several functions of the platform and the links between the different web pages. This type of software prototype can be helpful to explain to the different stakeholders how the platform will work, by showing them a complete prototype of it. Using the prototype it is possible to see the links between the different webpages, the functions of use cases, the settings webpage, the homepage, the registration webpage, the login webpage and other webpages of the platform. This software prototype can have minor differences in colors and menu artifacts compared to the UI of the mockups presented in the previous chapter, due to the different software used to make this prototype and the mockups.

6.1. Test of the software prototype

After making the software prototype, we decided to test it with 5 different end-users. These end-users are potential future end-users of the platform. We already analyzed the UI design in the previous chapter with two UI graphic designers to test if the interface has a good design, usability and if it is end-user friendly.

In this chapter we report on the tests that were done to check if the software prototype satisfies the requirements mentioned in section 2.1.3. In order to do this we presented the software prototype to 5 different end-users that answered a few questions.

The details of the end-users interviewed are summarized in the next table.

Person #	Sex	Age
1	Male	20-30
2	Female	40-50
3	Female	40-50
4	Male	30-40
5	Female	40-50

Table 35: Details of end-users interviewed concerning the software prototype

For each requirement mentioned in section 2.1.3, we asked the end-users if the requirement, in their opinion, was correctly implemented in the prototype. The possible answers were "yes" and "no". In both cases a motivation was possible to be given for each answer. During the interviews each end-user was guided through the prototype and we always explained all the different functions and parts to them.

In the prototype that was made we did not implement requirements number 8 and 10 due to their complexity and their low importance for this initial stage of development.

Regarding requirement number one, just the first 3 levels of authentication described in section 2.1.2.1 are implemented in the prototype.

Regarding requirement 4 (external services development): it was not included in the interviews because it concerns the "service provider" stakeholders and end-users cannot see if this requirement is correctly implemented in the prototype.

Regarding requirement 7 (users will be able to receive alerts): we did not implement it in the prototype because it concerns an external application that shows alerts to end-users.

Regarding requirement 9 (wallet function) we did not implement it in the prototype because it concerns external websites where the end-user will be able to share his information.

In the next table the answers that were given are summarized plus the possible motivations given for each answer. The columns "yes" and "no" summarize the count of answers that we got for each requirement. The columns "answer#" shows the sequential number of the motivation explained in the next column. The column "motivation" summarizes the motivation that we got for each answer, in brackets there is the answer connected to the motivation. It was not compulsory to write a motivation so the motivations are just few compared with the total number of the answers. A blank cell means that we did not get any motivations for that requirement.

R#	Yes	No	Answer#	(Answer) motivation
R1	3	2	1	(no) it is difficult to understand which services and data sources are connected with each authentication level.
R2	4	1	2	(no) it is not clear if it is possible to add new data sources to the system.
R3	4	1	3	(no) it is not clear where the data inserted by the end-user can be managed .
R5	5	0		
R6	5	0		

Table 36: Answers of the prototype's testers

From the results shown in the previous table we can glean different observations. Concerning R1: end-users found it difficult to understand the connection between the services and the different authentication levels. This was because we assumed that all the services implemented in the prototype need authentication level number one (that is the one that the end-user that we assume is logged in on the prototype has). This can be seen on the "authentication level" webpage of the prototype.

Concerning R2: end-users did not understand that all the data sources implemented in the prototype were already shown. In the prototype we assumed that data sources were already chosen by us to show in the user interface, and that other future data sources can be added by the end-user using future services. However, we did not implement these services in the current prototype.

Concerning R3: an end-user said that he would like to have a page where all the different data that he inserted in the platform is put together. We will implement this webpage in a next version of the prototype.

Concerning R5 and R6: the end-users said that these requirements were correctly implemented in the prototype.

7. The trust issue

In this chapter trust issues of the end-users are discussed. End-users will need to trust our platform to insert and import their private data into it. Privacy of end-users' data is important, and guaranteeing privacy and security are a key factor for the success of our platform. The potential problems, solutions and implications of this topic are analyzed in this chapter.

7.1. Problem description

The topic of trust between end-users and companies regarding private data of end-users is current nowadays because more and more private data of end-users is becoming available on the internet. As stated in the paper "Privacy op zijn plaats: Tussen willen weten en wetten" (Van Oortmarssen, De Vries, & Van Loenen, 2015) we are entering in an "era of datafication, featuring previously unknown capabilities to generate, store, analyze and distribute enormous amounts of data with the blink of an eye at minimal costs". This will lead to more and more potential privacy problems for end-users. End-users have two different and seemingly contradicting purposes: to store and use their data on an IT platform to maximize usefulness of them and, at the same time, protect that data from malicious entities and unauthorized access to it.

Companies that provide services using end-users' data have a similar problem: they want to maximize their and end-users' benefits by developing useful services but they need the trust of the end-users to convince them to share their private data.

As stated in the paper mentioned above a quick fix still does not exist. The information communities tend to rely on bottom up consensus models (for instance the process of setting standards) and the privacy lawyers appear to put more trust in top down regulatory mechanisms (Van Oortmarssen et al., 2015).

The result of this problem is a digital world where the potential of data is underutilized. Currently there are a lot of possibilities to use the large amount of data available, but these possibilities are underdeveloped because of the trust issue between companies that should provide services from data and the end-users.

7.2. The solution to the trust issue in this platform

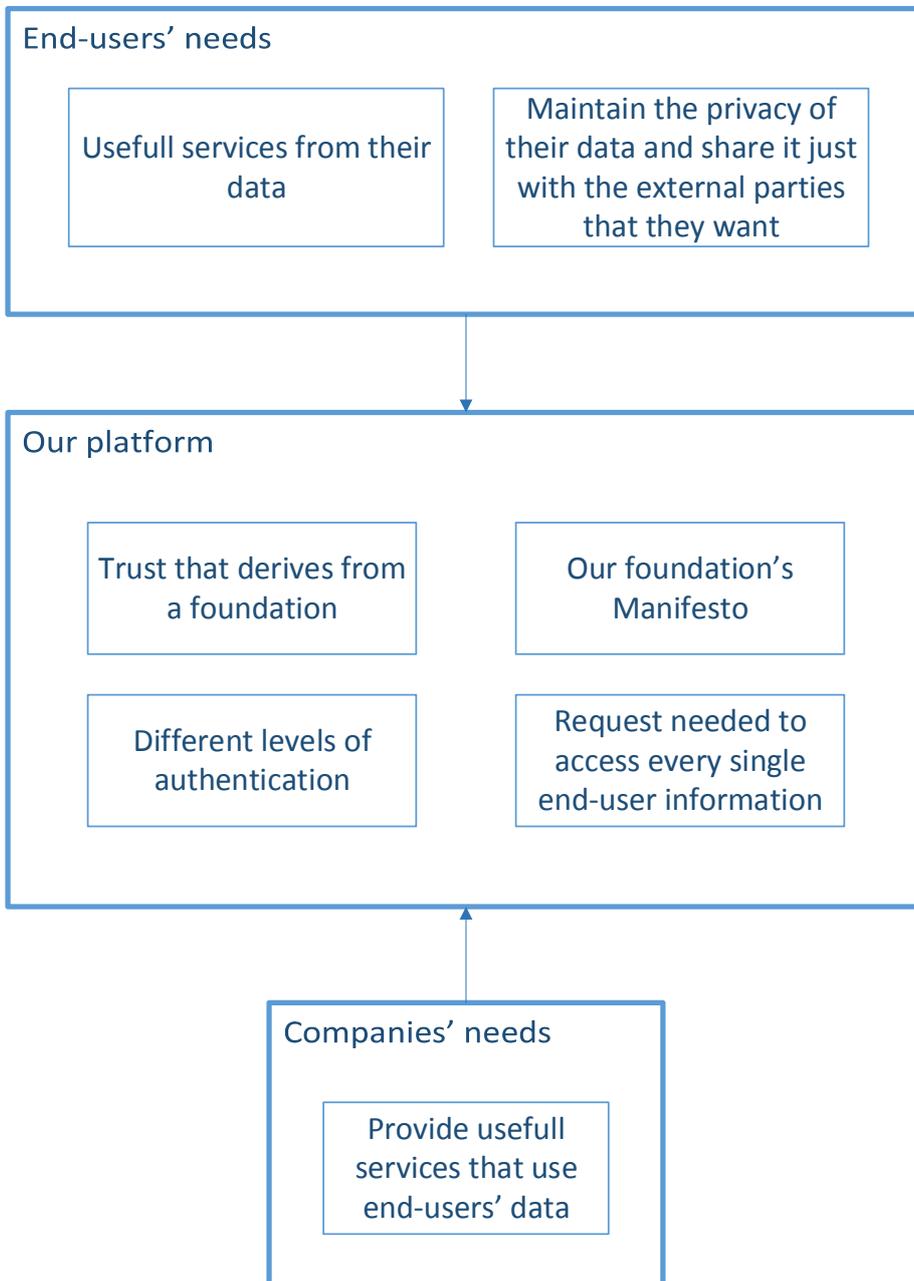


Figure 52: The solution to the trust issue of this platform

As shown in the previous figure, our platform will solve the trust issue by implementing a platform and foundation governance model that will earn trust from end-users. As explained in the scheme, the end-users' needs are useful services that come from their private data and the necessity to share this private data just with trustworthy external parties. The external companies need to provide useful services to end-users and, to do this, they need the end-users' trust and private data.

Our platform will satisfy the needs of all of them by permitting end-users to trust our platform to share their private data. This is firstly because our governance model will be a foundation model. As described in section 3.7.1, the foundation model implies that the foundation does not make nor share profits with the shareholders involved. This will be a convincing reason for end-users to trust us more, because we will not be able to make direct profit from their data.

Secondly, our foundation will follow the Manifesto (described in section 3.7.1.6) that states that every end-user will be able to decide with whom to share their data.

Thirdly, we will implement different levels of authentication in the platform (as described in section 2.1.2.1). These different levels of authentication will increase the security of our platform by permitting just the high certificated end-users to use services that use private information. If an end-user wants to use a service that uses his private data, he will need to use a secure level of authentication that will certify that he is the real person that he claims to be. This will increase the difficulty of unauthorized access by malicious people and thus it increases trust in our platform.

Fourth, every end-user will be able to decide which services and external websites share his private information. He will be able to withdraw this decision at any time from the user interface of our platform. This grants end-users more power over their private data and it is estimated that this feature will increase the trust in our platform.

8. Discussion

This thesis contains a design for a platform to provide home based services concerning care, safety, comfort and fun. This section will analyze the results that were achieved during this project.

The first result that we obtained is a list of requirements of the different stakeholders that are involved. These requirements were taken from interviews with the stakeholders. The interviews that we conducted demonstrated that several needs of the stakeholders are currently not met and that our platform can fulfill them. The research that we did to find these requirements was extensive, but could be done more thoroughly. The number of end-users that have been interviewed is low and further research on this should be done to confirm these requirements. Further research can also be done using the prototype described in chapter 6 to test if the requirements are really needed by the different stakeholders involved.

Regarding the existing available technological solutions and the current products and services on the market, most of them were analyzed in chapter 2 and were integrated in the technological solutions of the SA of the platform. However, more of them could be analyzed to have a wider spectrum of solutions to implement in the platform and to know more about our current competitors.

In section 3.3 a summary of the possible use cases to implement in the platform was made. This list contains 55 use cases and they are briefly described in that section. However, more use cases can be found by deeply analyzing the needs of the stakeholders and by showing them the prototype described in chapter 6. This will lead to an even more extensive list of possible use cases.

In section 3.4 nine use cases are deeply analyzed. We made a deep analysis with sequence diagrams for most of them. However, they are a low number. More use cases should be thoroughly analyzed in the future.

The data sources described in section 3.5 are strictly connected to the use cases described in section 3.4. In Appendix B other data sources are mentioned that can lead to more future use cases. In the future, new data sources should be analyzed to be implemented in the platform.

The software architecture described in chapter 4 contains and satisfies all the requirements mentioned before. However, we did not test this SA by implementing it. When it will be implemented, maybe future research will need to be done in order to improve it.

In chapter 5 the mockups of the UI are shown. This UI was tested by two professional user interface graphic designers. However, the real test will be when the platform will be developed and when it will be tested by the end-users. Because of this, more research on the UI of the platform will need to be done. The same advice should be given about the software prototype described in chapter 6. It was tested with 5 possible end-users to check if the requirements that it is supposed to satisfy have really been satisfied. The results were positive overall but the real test will be when the platform will be created and more end-users will be able to use and test it.

The solution to the trust issue described in chapter 7 is innovative but it still needs to be tested in the real world. When the platform will be created we will be able to test if our trust solution will lead to end-users sharing their private data with us and the services of the platform.

9. Conclusion

In this chapter conclusions about the project are drawn. The different research questions stated in chapter 1 are answered. The scientific contributions, both theoretical and practical, are stated and described. Finally, the research limitations and the recommendations are provided.

9.1. Reviewing the research questions

In this section we will review the research questions that we posed in chapter 1.

- **RQ1: What are the characteristics a platform where end-users will be able to see, share and analyze information regarding their houses?**

This research question is quite broad and has few limitations on scope. For this thesis a platform was developed where end-users will be able to see, share and analyze information regarding their houses. We made a problem analysis to find out which requirements the platform will need to have by analyzing the needs of the stakeholders involved. This analysis was made by interviewing different parties involved and by collecting their needs, opinions and requirements.

In chapter 3 the reader can find the complete business case of the platform. The end-user perspective services are provided with a complete description of each service. The platform will need different levels of authentication for end-users to comply with the different services' security needs. The platform will be able to import data from external data sources. This data will be both in the linked and in the non-linked data format. The platform will be able to store data inserted by the end-user in the platform's servers. End-users will be able to share their data with external websites by using the wallet function. External services can be integrated in the platform to provide end-users useful functions combined with external imported data.

In the dashboard the end-users will be able to find the different functions of the platform and they will be able to see their services and their private data. The alert function enables end-users to receive alerts through the mobile app, by email, by phone calls and by SMS from the use cases of the platform. Finally, the social function will permit end-users to manage their social profile and to connect with other end-users of the platform.

In section 3 a complete list of the different use case that can be implemented in the platform is provided, and in that same chapter, some use cases are deeply analyzed. This deep analysis is helpful because it shows in detail how some use cases really work and how they are integrated in the platform.

In chapter 4 the SA of the platform is provided. It explains from an IT perspective how this future platform will work and what will be the different components of it.

In chapter 5 and 6 the mockups and the software prototype of the platform are provided to permit the reader to better understand how the final platform will be, as seen from an user interface perspective.

For further information please refer to chapters 3, 4, 5 and 6 of this thesis.

- **RQ2: Which use cases will be implemented that use information about houses and that will provide useful functions to end-users?**

In section 3.3 a complete overview of the different use cases that can be implemented is provided. This list contains 55 use cases and provides a brief description and categorization of each of them.

In section 3.4 several use cases are deeply analyzed. These use cases were the most useful for the different parts involved and they will be implemented in the first release of the platform.

The "security alarm sharing" use case will permit end-users to share their alarm signal with other end-users. These end-users will be reached by the mobile app, by phone, by SMS and by email. This use case will permit end-users to be safer in their house because, in case of a robbery, their neighbors and their friends will be alerted immediately.

The "photo retrieval and upload" use case will be helpful to end-users because it enables them to see old photos of their house and of their surroundings. It will also permit end-users to upload their old photos and to gain money from doing so.

The "chamber of commerce company information" gives end-users an overview of the companies that are in their surroundings. It will make the life of end-users easier and more comfortable by providing them information about companies in their surroundings.

The "maintenance and various information about appliances" use case will permit end-users to manage and to store information about their appliances in the platform. This will make the life of end-users more comfortable because end-users will be able to have all the information about their appliances in one single portal.

The "security information", "energy information", "flood risk information", "garbage collection information" and the "municipality contact information" use cases allow end-users to see in the platform information linked to their house. These use cases will not get any input from the end-users but they will just show and retrieve the information linked to the end-users.

For more information please refer to chapter 3 of this thesis.

- **RQ3: What will the software architecture for such a system look like?**

The SA of the platform is completely explained in chapter 4. The SA that we developed is described in detail in section 4.6 and it is divided into several parts.

The first part is the user interface that will contain the client layer. This client layer will contain the different user interfaces linked with the devices that will connect to the platform. The devices' interfaces will be managed using different CSS.

The central part of the SA is the platform. It contains the frontend API, the service layer where the different services of the SOA architecture are implemented, the physical layer where the file server and the databases are implemented and the dynamic data retrieval where the data retrieved from dynamic sources is parsed.

The last part of the SA consists of the external actors part that contains all external servers and actors that will connect to our platform. This part includes the external websites where end-users' information will be shared using the wallet function. It will contain the external alert function and it will contain the static and dynamic external data sources where the different types of data will be retrieved and then imported in the platform.

For further information please refer to chapter 4 of this thesis.

- **RQ4: What will be the solution to the trust issue between end-users and the information sharing platform?**

As explained in chapter 7 end-users will need to trust our platform to store their private data. This trust will need to be gained and we will gain it with four main solutions.

The first solution is by making our governance model as a foundation. A foundation cannot share its profits with its shareholders and so end-users will be more willing to share their private information with us. This is because being a foundation will increase the end-users' trust that we will not share their private data to make profits from it.

The second solution is to follow the manifesto that we made for our foundation. This manifesto states that every end-user will own their data and that they will be able to decide with which external services and websites they want to share their private data. This will increase the trust of end-users in our foundation.

The third solution is to implement different level of authentication for end-users. This will increase the overall security of the platform and so the trust of end-users.

The fourth solution is to permit end-users to decide with which services and external websites they will share their private data. This will be possible by using our UI and our SA explained earlier in this thesis.

For more information please refer to chapter 7 and to section 9.2.1.1.

9.2. Scientific contributions

9.2.1. Theoretical contributions

9.2.1.1. Create trust with users that share their private data

As explained in chapter 7, big data will open up new possibilities to create innovative, interesting and useful services. If end-users share their private data with us and the services developed by external companies, we will be able to offer them many useful services that will improve their life. This applies also if they share their private data with external websites by using our wallet function.

End-users want to have access to data regarding their home through an integrated platform, where they will be able to find all of their houses' information in one single place. This will lead to an increase in the quality of their life because, this way, one of their needs will be satisfied.

A major obstacle to sharing of private data by the end-users is the trust issue between them and the companies with which they will share data. The outcome of this obstacle is, as stated in the paper "Privacy op zijn plaats: Tussen willen weten en wetten" (Van Oortmarsen et al., 2015), that "artificial and suboptimal solutions are sought, leading to the underuse of the location data's full potential". This is because users are skeptical to share their private data with external websites because they are afraid that their data will be used against them.

Our platform offers an innovative and unique solution to this problem. As described in chapter 7, our platform will gain the trust of end-users to share private data with four solutions. The first one is that we will be a foundation. Foundations in The Netherlands cannot share profits with their shareholders. Presumably the end-users will therefore be more willing to share their private data with us, since we will not be able to make any profit using their data.

The second solution is the Manifesto, described in section 3.7.1.6, which states that every end-user will own their data and that they will be able to decide with which external services and websites they want to share their private data. Our foundation will follow this Manifesto.

The third solution is that we will implement different levels of authentication for end-users. Each level of authentication will permit end-users to access different types of services and data sources. This will increase the overall security of the platform and thus it will increase the trust that the end-users will have in our platform.

The last solution is that the end-users will be able to decide, through the user interface of the platform, with which services and external websites they will share their private data. This function will give end-users the power to control their private data and the power to decide with whom to share their private data. This will lead to more trust from end-users in our platform.

We will combine all these solutions in our platform. These solutions to the trust issue are innovative and unique because currently there is no company that implements all these solutions into one single platform. These solutions combined will solve the trust issue and this will lead to better development and usage of services that use private data of end-users.

9.2.1.2. Combine open (public) and closed (private) data with a location

In our platform the data sources that will be imported will be of various types. They can be closed (private) data and open (public) data sources and they can be both linked-data and non-linked-data. The data that can be imported will be linked to the name of the end-user and to his address.

The combination of data with the location and the name of the end-user is new and unique because nobody has implemented it yet.

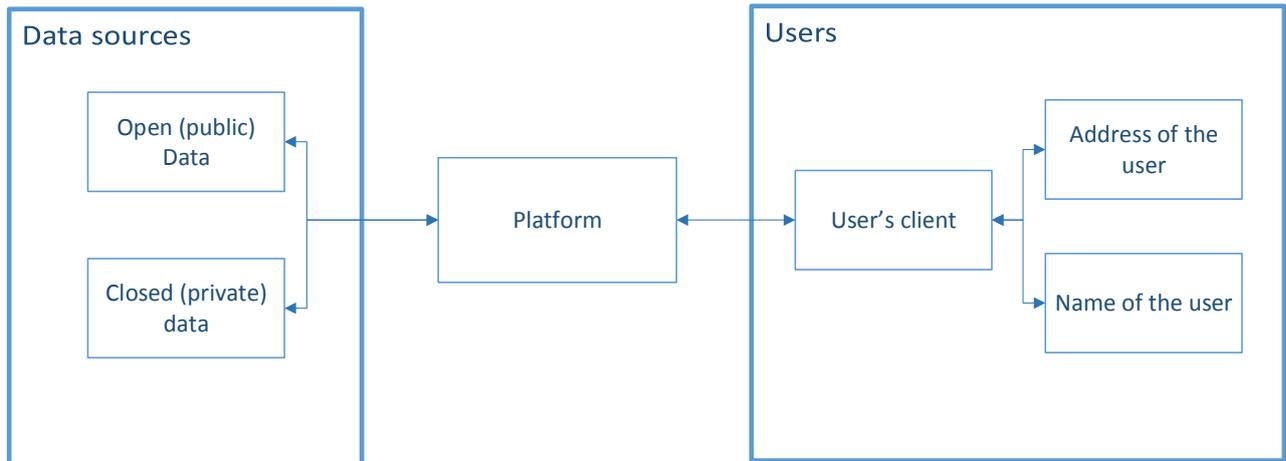


Figure 53: Combining open and closed data with a location

As the reader can see in Figure 53 the data sources are combined with the details of the end-user such as address and name.

This model is made into a unique software architecture. Data will be imported from the “data conversion” service that connects with the “data storage and retrieval” service, which will in turn connect to our database where the details (address and name) of each end-user are stored. These details are used to query the external datasets by the “data conversion” service.

So this new and innovative requirement will provide an innovative software architecture that will support it.

For instance, if an end-user wants to retrieve security information about his home’s area, the platform will retrieve his address from the internal database and then the “data conversion” service will request the external data source for security information of the area by making a query with the address of the end-user.

For further information about the SA please refer to chapter 4 of this thesis.

9.2.1.3. Importing and selling linked data from other companies’ datasets

The data that will be imported in our platform will come from third party companies that will provide their datasets to us. A portion of this data will be imported and copied in our database and some of it will be just imported.

The data that will be imported will be both linked and non-linked data. The non-linked data that will be imported will be converted into linked data by our platform.

According to our revenue model, we will be able to sell data that has been imported from external data sources and then converted into the linked format by our platform.

This is new and innovative because nobody has had the idea to import data from different sources, convert it to linked data, and finally sell it to third party companies.

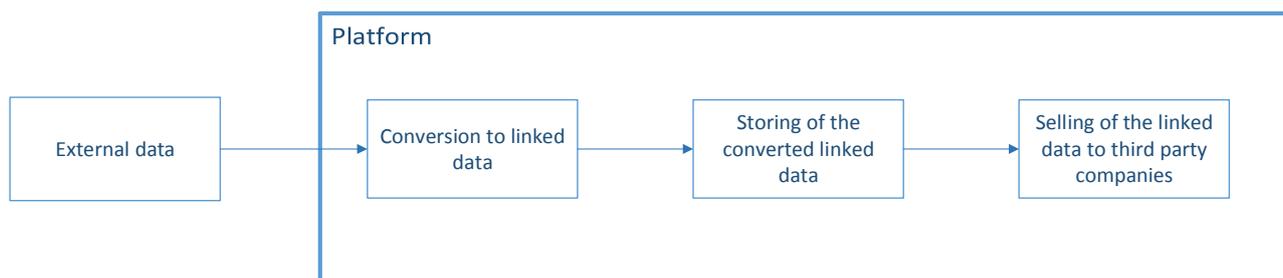


Figure 54: Importing and selling data from other companies' datasets

As the reader can see in Figure 54, the data is imported from external datasets into the platform. Then the data is converted into linked-data and then it is stored on our servers. Finally, the data is sold to external companies.

This requirement is included in our unique and innovative software architecture. Data will be imported from a service in the platform. Then data will be converted and stored in our databases by the Apache Jena Fuseky server and the Apache Jena TDB RDF component server. Finally, the data is retrieved by third party companies using a service of the platform.

9.2.2. Practical contributions

9.2.2.1. The software architecture

The software architecture that we made and that is presented in chapter 4 is unique and innovative because it satisfies a new and innovative combination of software requirements. The software requirements that we found are unique because, currently, there is no company that has already implemented them into a SA. We found this by analyzing the current competitors that are on the market and we found that our current competitors did not implement our unique requirement combination in their SA (for further information about the current market analysis please refer to section 2.2.2.1).

The SA that we made has a unique combination of components that satisfies our requirements. The most important components of our SA are summarized below. For a more detailed account please refer to chapter 4. The "data conversion" service that converts the data imported from both static and dynamic external data sources into linked data is unique when combined with the other parts of our SA. The data is stored in our database by the "data storage and retrieval" service that will store the linked data of the end-users into our database. The data of end-users can be shared with external websites using the wallet service of our platform. External websites need just to implement our JavaScript code in their web pages to be able to import our end-users' data (obviously after the end-user's confirmation to share that data). The external data sources that provide dynamic data can use our VPN

and our “virtual sensor abstraction layer” to import data in the platform. An external sensor just needs to be connected to the internet and then needs to connect to our VPN. This makes the retrieval of dynamic data simpler for us and for the sensor’s manufacturers. The external alert service permits end-users to receive alerts from use cases in many ways such as by mobile app, by voice calls, by email or by SMS.

These components are already partially present in competitors’ SA but our SA combine all of them thus creating a unique SA.

For further information about the SA please refer to chapter 4 of this thesis.

9.2.2.2. The mockups and software prototype

The mockups and software prototype that we made are described in chapters 5 and 6. They include all the different parts of the interface that the end-users will see of our platform. They are important for the scientific community and for every reader of this thesis because they show how the interface of the platform will really look like and how it will work.

Companies can modify the user interface and use it for their own platform, by customizing the platform or taking components and creating their own unique user interface. The different parts of the user interface are connected with one part of the SA described in chapter 4 and so someone can use this link to understand how the user interface and the SA of the platform will work together. The different connections of the user interface and of the SA are explained in chapter 5.

The user interface that we designed includes the homepage of the platform, the menus, use cases and information provided to end-users. Potential clients will be able to reuse the different components of this user interface in their own platform and so save time and money instead of making a complete new user interface. This user interface has already been tested by two professional user interface designers that gave us feedback. So, potential clients will not need to test it themselves.

Using the user interface designed and explained in chapter 5 and 6 can lead to money and time saving for anybody that needs to design an user interface for a platform similar to this one.

For further information about the user interface please refer to chapters 5 and 6 of this thesis.

9.3. Research limitation

A number of limitations of this research are present.

The most important limitation of this project is the small sample size of the interviews of the end-users described in Appendix A. Due to time constraints and practical limitations we could only work with a small sample size. However, a bigger sample size would have provided a more thorough analysis of the end-users’ needs and requirements, and would have permitted to generalize the results contained in this thesis in a better way. So we advise to work on this particular part of the project.

Another limitation is the small amount of use cases that have been scrutinized in section 3.1. We made a complete list of the different use cases available for this platform in section 3.3 but we analyzed just 9 of them (we analyzed the most important ones following the needs of the stakeholders). So future research on more use cases will be

important when the platform will grow. Also, more research needs to be done on the possible use cases to implement in the platform to increase the list on section 3.3.

More existing technological solutions and current products and services on the market should be analyzed. A deep analysis of them will give us a better understanding of our competitors' products and the technologies that can be used in the SA of our platform.

Another limitation of this thesis is the superficial analysis made of single data sources that will be used to retrieve data needed for the use cases. We do not know completely the state and the accessibility of the data that is needed for these use cases so more research in this field would be needed. However, we implemented in our SA the "data conversion" service that will be able to convert data imported from many types of data sources into linked data. This SA permits us to manage different types of data sources and this smooth the poor analysis that we made for the available data sources. In addition, more data sources should be analyzed in order to be able to develop more use cases.

The SA and the UI of the platform should be tested in the real world when the platform is made. This is because we did not test them in the real world but we just tested the mockups and the software prototype with two professional user interface graphic designers and with five possible end-users.

Our solution to the trust issue should also be tested when the platform will be realized to know if the end-users are willing to trust us and to share their private information with us.

9.4. Recommendations for future work and research

Taking into consideration the mentioned research limitations and the overall project, recommendations for future work and research can be made.

The first recommendation is to better analyze the needs and requirements of the end-users. For instance by using a bigger sample of end-users to interview. This makes it possible to generalize the findings that we derived from a small number of interviews into more broadly tested findings.

The second recommendation is to look deeper into existing technological solutions and current products and services on the market. Thus it is possible to have a better SA with the latest technology available and it will give us better understanding of our competitors.

The third recommendation is to create more use cases when the platform will be implemented and will start growing. When the use base will start reaching a good base number, more use cases will need to be implemented. This is important because end-users will constantly need new use cases in the platform to take the best advantage from it. These new use cases can be found by finding data sources to connect to the platform and then by creating use cases that will use them.

The fourth recommendation is to analyze the different data sources that will be connected to the platform. This step will be important when the platform will be developed and when the first release of it will be made.

The next step of this project will be to develop the platform. We will use the SA and the UI explained in this thesis. When the code of the platform will be written, an deep analysis of the different parts of the SA and of the UI will need to be made. The different REST calls of the different SOA's services will need to be made and the different classes of every part of the SA need to be developed. The user interface will need to be converted into an HTML and CSS interface to permit the different end-user's devices to use it. The development of the platform will be the next main challenge because it will be both time and money consuming. However, it is a mandatory stage in publishing the first release of the platform.

Finally, when the platform is developed, our solution of the trust issue can be tested and so we will be able to find if the end-users will share their private information with our platform.

10. References

- Apache Jena Fuseki. Retrieved 09/05/2015, from <http://jena.apache.org/documentation/fuseki2/>
- ARQ - A SPARQL Processor for Jena. Retrieved 12/04/2015, from <https://jena.apache.org/documentation/query/>
- Berners-Lee, T. (2006). Linked Data. Retrieved 10/04/2015, from <http://www.w3.org/DesignIssues/LinkedData.html>
- Bizer, C., Heath, T., & Tim Berners-Lee, T. (2009). *Linked Data-the story so far*.
- Board Service Oriented Architecture. Retrieved 11/05/2015, from http://help.board.com/Chapter_2_-_Board_Architecture_files/The_Client_Master_Server_Architecture.htm
- Burger, A., & Dekker, P. (2001). The nonprofit sector in the Netherlands.
- Cockburn, A. (2002). Use cases, ten years later. *Software Testing and Quality Engineering Magazine* (pp. 37-40).
- Cross-Platform Authentication with Google+ Sign-In. Retrieved 11/05/2015, from <http://www.slideshare.net/peterfrieese/crossplatform-authentication-with-google-signin>
- Duval, A., & Brasse, V. (2014). How to ensure the economic viability of an open data platform.
- Education - Campus Management System. Retrieved 11/05/2015, from <http://www.commit.in/form/education.aspx>
- Francken, N. (2013). The Dutch foundation. Retrieved 03/05/2015, from <http://www.step.org/dutch-foundation>
- GNU GENERAL PUBLIC LICENSE. (2007). Retrieved 26/04/2014, from <https://www.gnu.org/copyleft/gpl.html>
- Heath, T., & Bizer, C. (2011). Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1), 1-136.
- Hill, D. J., Liu, Y., Marini, L., Kopper, R., Rodriguez, A., Futrelle, J., & McLaren, T. (2011). A virtual sensor system for user-generated, real-time environmental data products *Environmental Modelling & Software* (pp. 1710-1724).
- Krafzig, D., Banke, K., & Slama, D. (2005). *Enterprise SOA: service-oriented architecture best practices*: Prentice Hall Professional.
- LinkedIn - A Professional Network built with Java Technologies and Agile Practices. Retrieved 10/05/2015, from <http://www.slideshare.net/linkedin/linkedins-communication-architecture>
- Mikoluk, K. (2014). MySQL vs PostgreSQL: Why MySQL Is Superior To PostgreSQL. Retrieved 09/05/2015, from <https://blog.udemy.com/mysql-vs-postgresql/>
- Morgan, L., & Finnegan, P. (2014). Beyond free software: An exploration of the business value of strategic open source. *The Journal of Strategic Information Systems*, 23(3), 226-238.
- Nechifor, A., Albu, M., Hair, R., & Terzija, V. (2015). A flexible platform for synchronized measurements, data aggregation and information retrieval *Electric Power Systems Research* (pp. 20-31).
- Omitola, T., Koumenides, C. L., Popov, I. O., Yang, Y., Salvadores, M., Correndo, G., . . . Shadbolt, N. (2010). Integrating public datasets using linked data: challenges and design principles.
- Santos, C., Kuk, G., Kon, F., & Pearson, J. (2013). The attraction of contributors in free and open source software projects. *The Journal of Strategic Information Systems*, 22(1), 26-45. doi: <http://dx.doi.org/10.1016/j.jsis.2012.07.004>
- SPARQL 1.1 Overview. Retrieved 22/04/2015, from <http://www.w3.org/TR/sparql11-overview/>
- SPARQL 1.1 Query Language. Retrieved 22/04/2015, from <http://www.w3.org/TR/sparql11-query/>
- Stevens, M. (2002). The Benefits of a Service-Oriented Architecture. Retrieved 11/05/2015
- Tekinerdoğan, B., & Akşit, M. (2002). *Synthesis-based software architecture design*: Springer.
- Text searches with SPARQL. Retrieved 12/04/2015, from <https://jena.apache.org/documentation/query/text-query.html>

- Van Gorp, P., Comuzzi, M., Jahnen, A., Kaymak, U., & Middleton, B. (2014). An open platform for personal health record apps with platform-level privacy protection. *Computers in biology and medicine*, 51, 14-23.
- Van Oortmarssen, A., De Vries, M., & Van Loenen, B. (2015). Privacy op zijn plaats: Tussen willen weten en wetten: Geonovum.
- What Is an Enterprise Service Bus? Retrieved 11/05/2015, from <https://msdn.microsoft.com/en-us/biztalk/dd876606.aspx>
- What is Service Oriented Architecture? Retrieved 11/05/2015, from <http://www.pnmsoft.com/resources/bpm-tutorial/soa-tutorial/>
- Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*: Springer.
- Wilkes, D. S. a. L., & Forum, C. (2004). Understanding Service-Oriented Architecture. Retrieved 11/05/2015, from <https://msdn.microsoft.com/en-us/library/aa480021.aspx>

Appendix A: end-user interviews

We interviewed potential end-users in two interview rounds. We used the survey method to know the needs and the requirements of the end-users because “surveys are useful for implementation evaluation and problem investigation because they can provide information about real-world phenomena” (Wieringa, 2014). The disadvantages of surveys are that they can disturb the phenomena investigated and that they can investigate only a few aspects of the phenomena (Wieringa, 2014).

First round

In the first interview round we asked general questions that are summarized in Table 38: First round questions. We asked four potential end-users to answer these questions.

The details of each person that we interviewed are shown in Table 37. These end-users are representative of our end-users because we assume that every person that we interviewed will be able to use our platform. The sex of the person is not relevant because both females and males will be able to use the platform. Age is relevant because old end-users will have more trouble to use a web-platform, so the end-users that we interviewed are less than 50 years old. Whether they have children is not important because both end-users with and without children will be able to use our platform. Sex and the fact of having children are mentioned in Table 37 because we collected this information during the interviews but they are not relevant for this research.

Person #	Sex	Age	Children
1	Female	20-30	No
2	Female	30-40	Yes
3	Male	40-50	Yes
4	Female	30-40	yes

Table 37: Details of end-users interviewed in the first round

In Table 38 are the questions that we asked in the first round. For each question the person was asked to refer to the different contexts included in Table 39. We asked these questions because we wanted to know what end-users do when something is happening, what they are going to do when something is happening, what is important for them, how they know that something is happening, how they will solve the problem and if they are going to share relevant information about the situation with external parties.

Question #	Question description
1	How do you know that it is happening?
2	Which actions are you going to take?
3	What is the most important thing for you?
4	Who has to know that it is happening?
5	How to solve this problem?
6	Are you willing to share information with trusted end-users?

Table 38: First round questions

In Table 39 the different contexts that the person should have in mind when replying to the different questions are shown.

Context #	Context description
1	Fire in my home.
2	Theft in my home.
3	I have broken a body part.

Table 39: First round contexts

In the next table the replies that were given by the interviewees are shown. If replies were similar they were merged into one single sentence in the table. A blank cell means that no reply for that particular combination of question and context was received.

Questions/context	Fire in my home	Theft in my home	I have a part of the body broken
How do you know that it is happening?	<ul style="list-style-type: none"> • From neighbors • Alarm signal and police calling 	<ul style="list-style-type: none"> • Alarm signal 	N/A
Which actions are you going to take?	<ul style="list-style-type: none"> • Call parents • Go home • Alert neighbors 	<ul style="list-style-type: none"> • Call police, parents and neighbors 	<ul style="list-style-type: none"> • Call for help of friends, parents and neighbors
What is the most important thing for you?	<ul style="list-style-type: none"> • Children, humans, photos, things that cannot be replaced • Mobile telephone for data • Documents and things with emotional value 		
Who has to know that it is happening?	<ul style="list-style-type: none"> • Parents and friends • Firefighters 	<ul style="list-style-type: none"> • Police, parents and neighbors 	<ul style="list-style-type: none"> • Friends, parents and neighbors
How to solve this problem?	<ul style="list-style-type: none"> • Call the insurance company for compensation • Wait • If the fire is small try to stop it, if the fire is big do nothing 	<ul style="list-style-type: none"> • Call the police, parents and neighbors 	
Are you willing to share information with trusted end-users?	<ul style="list-style-type: none"> • Yes for a map of the house and where the gas and explosive materials are stored in the house 	<ul style="list-style-type: none"> • Yes to share information with neighbors about a robbery that is going on in the house 	<ul style="list-style-type: none"> • Yes to share information about my health problems with the neighbors to get help

	<ul style="list-style-type: none"> • Yes for maps and all the information that can be helpful to prevent and extinguish the fire. • Not for sharing general information but in case of a fire it is ok to share relevant information but just by selecting the exact end-users that will receive this information 	<ul style="list-style-type: none"> • Not for sharing general information. In case of a robbery it is ok to share relevant information but just by selecting the exact end-users that will receive this information 	<ul style="list-style-type: none"> • We have a good relationship with neighbors so we do not need a platform to contact them
--	---	---	---

Table 40: First round interview results

These results were helpful to get a general view of what end-users do in a particular context. We found that end-users normally know if there is a robbery or a fire in their house by the alarm signal, from the neighbors or from the police.

We found that end-users, when they know that there is a fire or robbery in their home, call their parents, police and neighbors and then they go home. This information can be important for the alarm sharing use case explained in section 3.4.1. If a person has a part of his body broken and cannot move outside the house, he will call his friends, family and neighbors for help.

People think that, when there is a fire in their home, the most important things for them are their children, any humans in the house, photos and things that cannot be replaced, data from mobile telephones, documents and things with emotional value.

They are willing to share their information, in case of a fire or a robbery, with their parents, friends, neighbors, firefighters and the police. In case they have a broken body part they are willing to share this information with friends, parents and neighbors.

When there is a fire at home, a person will call the insurance for get reimbursed, he will wait or, if the fire is small he will try to stop it. If a robbery is happening in the person's home, he will call the police, his parents or his neighbors.

To prevent fires in the house and to increase the probability of extinguishing the fire in a short time, end-users are willing to share information about the house with specific end-users (this information includes a map of the house and the position of the gas stove). In case of a robbery, end-users are willing to share this information by selecting exactly the end-users that will receive this information. In case of a broken body part, a person is willing to share the information about his situation with his neighbors. However, they replied that they are already receiving help from their neighbors if they are in this situation so they do not need a web-platform to contact their neighbors.

Second round

In the second round we interviewed more end-users and we asked more specific questions to them. These questions were about the information that they would want to get from a web-portal, if they perceive the division of information over many different websites as annoying and if they would like to have any other functions in the platform.

The details of the interviewed end-users are summarized in Table 41. A blank cell means that the information was not provided. The end-users were both female and male and they had a wide age range. We decided to interview older end-users as well, because we wanted to include them in the end-users of the platform. In this second round we did not ask if people have children because that is irrelevant for this platform.

Person #	Sex	Age
6		70-80
7	Male	50
8	Female	40
9	Male	90
10	Female	40
11	Male	35
12	Female	45
13	Female	50
14	Male	65
15	Female	40
16	Female	35-40
17	Female	50
18	Female	40-50
19	Female	70
20	Female	30-40

Table 41: Details of interviewed end-users in the second round

The questions that we asked are listed in the next table. We asked questions that were divided into three categories. The first category is the "surrounding" category that contains all the information and services related to the surroundings of the end-user. The second category is the "house" category that includes all the information and services about the house of end-users. The last category is the "energy" category that includes all the services and information related to energy and collective purchasing of things with neighbors.

The aim of these questions was to know more about end-users' needs and how to satisfy them.

Question #	Category	Question
1	Surrounding	Do you want to know the level of traffic congestion in your surroundings?
2	Surrounding	Right now, do you not know about road construction or maintenance in the surrounding in time?
3	Surrounding	Do you want to know if your neighbors ask for a building permit?
4	Surrounding	Do you want to know about a power failure in your surroundings?

5	Surrounding	Do you think that the alternatives in case of power failure are difficult to find?
6	Surrounding	Are you willing to share your belongings with your neighbors?
7	Surrounding	Are you familiar with an appliance sharing website?
8	Surrounding	Are you willing to lend belongings if your neighbors ask for it?
9	Surrounding	Do you know what is happening in your area?
10	Surrounding	Would you like to know what is happening in your area?
11	Surrounding	Do you know when there are local events in your area?
12	Surrounding	If you knew when there were local events in your area, would you go to them more often?
13	House	Do you perceive that the information about your house is scattered over a lot of different websites?
14	House	Do you perceive this scattering as annoying?
15	House	Would you share information about your house with other end-users?
16	House	Do you know how much your house is worth?
17	House	Do you want to know how much your house is worth several times each year?
18	House	Would you like to share pictures of the interior of your house with other end-users?
19	House	Would you like to see old pictures of the surroundings of your house in the platform?
20	Energy	Do you know how your energy usage is compared to your neighbors?
21	Energy	Do you know which devices in your house are consuming more energy?
22	Energy	If you knew which devices are consuming more energy, would you be able to save energy?
23	Energy	Would you like to know which energy saving options are available?
24	Energy	Would you like to know about collective purchases?

Table 42: Second round interview questions

The results are summarized in the next table, and are summarized in a binary way. We analyzed the information by labeling each answer as a yes/no binary answer. Then we summed them and derived the conclusion. The conclusion is yes or no if the answers were more than 70% in favor of one of these answers. In the other cases, the conclusion is neutral.

Question #	Yes	No	Conclusion
1	7	6	Neutral
2	5	7	Neutral
3	7	6	Neutral
4	10	2	Yes
5	7	3	Yes
6	8	2	Yes

7	0	9	No
8	8	4	Yes
9	5	8	No
10	8	2	Yes
11	4	6	No
12	7	3	Yes
13	5	1	Yes
14	4	2	Yes
15	0	3	No
16	2	0	Yes
17	1	2	Neutral
18	1	2	Neutral
19	1	3	No
20	0	1	Neutral
21	1	2	Neutral
22	1	3	No
23	1	0	Yes
24	2	2	Neutral

Table 43: Second round results

People seem neutral about the traffic congestion information, road construction information and about their neighbors building permits. Based on the answers that were received it can be concluded that end-users already know if there is road maintenance in their street because they receive a letter from the government with details of the construction work.

People think that alternatives, in case of a power failure, are difficult to find. End-users would like to know when the electricity will come back after a power failure.

End-users are willing to share belongings with neighbors, but they do not know the existing platforms that offer this service. If they knew these websites they would use them.

People do not know about local events of their area and, if they knew about them, they would join them more often.

People perceive that their information is scattered over different websites and this scattering is perceived as annoying.

People are not willing to share their information with other end-users. However, during the interviews, most of them said that they would share their information if it is not important information or if sharing it is really important, for example during a robbery. They are also willing to share their information just with specific end-users and with specific third parties.

People seem to know how much their house is worth and they seem neutral about getting information about the value of their house several times each year.

People are neutral about sharing of pictures of the interior of their house with external end-users.

People do not care about seeing old pictures of their house and of their surrounding in the platform.

People are neutral about information about the energy consumption of their neighbors and a comparison with their energy usage. People are also neutral about knowing which devices in their house are consuming more energy and they do not think that, with this information, they will be able to save energy. People would like to know which energy saving options are available for them.

Finally, end-users are neutral about collective purchasing of goods with neighbors.

Using these results we can determine the necessary requirements for the platform. These requirements are discussed in section 2.1.3 where the complete list of requirements is provided.

Appendix B: data sources that will not be used for the software architecture in this platform but that can be used for future use cases.

The following data sources were found using several methods. First, we searched for data sources in data source collection websites such as "data.overheid.nl" and "nationaalgeoregister.nl". These websites collect a huge amount of public datasets that can be used in our platform.

Secondly, we found data sources through partnerships and external parties that will develop use cases for the platform.

Finally, other data sources were found using web search engines such as Google.

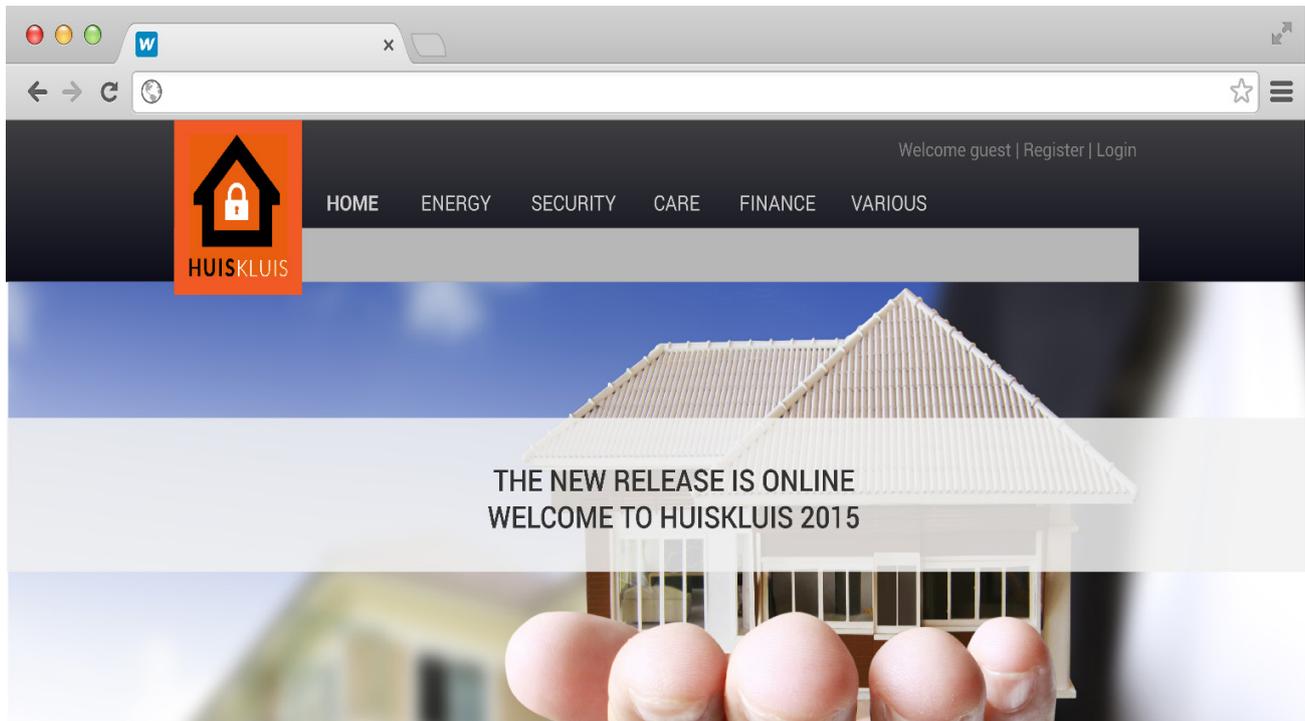
ID	Name	Description	How to retrieve it
1	Traffic density	The traffic density of the street where the end-user lives.	
2	Perceived safety	Perceived safety of end-users divided per postcode.	https://data.overheid.nl/data/dataset/onveiligheid-buurtkenmerken-2008-2009
3	Quality of living	Quality of life in the suburb.	https://data.overheid.nl/data/dataset/leefbaarometer-1998-2010
4	Weather forecast	Information about the weather forecast with added information connected to it. For instance: - garden info system: informing you about actions needed, watering based on weather, soil, ground water level, season, etc. - storm warning: close your windows. - heat levels: expected warm days in which rooms on the top floor will be too warm to sleep conveniently. This added information can be processed by our platform.	
5	Temperature inside the house	Temperature data from the temperature sensors placed in different parts of the house	
6	House tax details	Amount of taxes to pay for the house.	

7	Information for an activity in the city	End-users will receive an alert if there is an event near their postcode. The alert can be for traffic or for high sound levels. This can also be combined by showing the details of the activities that are planned in the city.	
8	Monuments	If the house is a monument there will be information about its history or for instance renovation restrictions.	
9	Information about the street	History of the street, important events, important people that lived there.	
10	Cables and pipes in the underground	Information about the pipes (water and gas) and cables (electricity and telephone) that are underground near the house.	
11	Local emergency telephone numbers	Local police telephone number and the nearest hospital telephone number.	
12	Nearest hydrant	Nearest hydrant of the end-user	
13	Map of the surroundings	A map of the surroundings of the end-user.	
14	Local news	Local news of the city and the surroundings.	
15	Parks information	The nearest park and its details.	
16	Pollution statistics and forecast	End-users will be able to see pollution statistics and forecast of their area and advice about how to protect themselves from the pollution.	
17	Parking places	Nearest public parking places and their hourly fees.	
18	Average temperature, rainfall and climate	Data about the average climate of the area.	
19	Energy class of the house	Energy class of the house.	

Table 44: Data sources that will not be used for the software architecture of this platform but that can be used for future use cases

Appendix C: mockups of the interface

In this section several mockups of the user interface are presented, without a connection with the SA. For each different mockup a brief description of it is provided. These mockups can differ from the software prototype presented in chapter 6 because the software prototype is more update than these mockups.



Do you live in Het Sander 37, 7522AL Enschede, The Netherlands?



Address: Het sander 37
City: Enschede
Country: The Netherlands

Do you want more information about your home?

[Create your digital home](#)

Events in the area:
July 18: Enschede beer fest
July 23: University of Twente open day

Flood risk: 8/10 (hight)

Municipality contacts:

Figure 55: Mockup of the homepage without an end-user logged in

In the previous figure the homepage of the platform is shown, assuming that the end-user did not login on it. Information about the address of the end-user (taken from the referral information) and the slider are also shown.

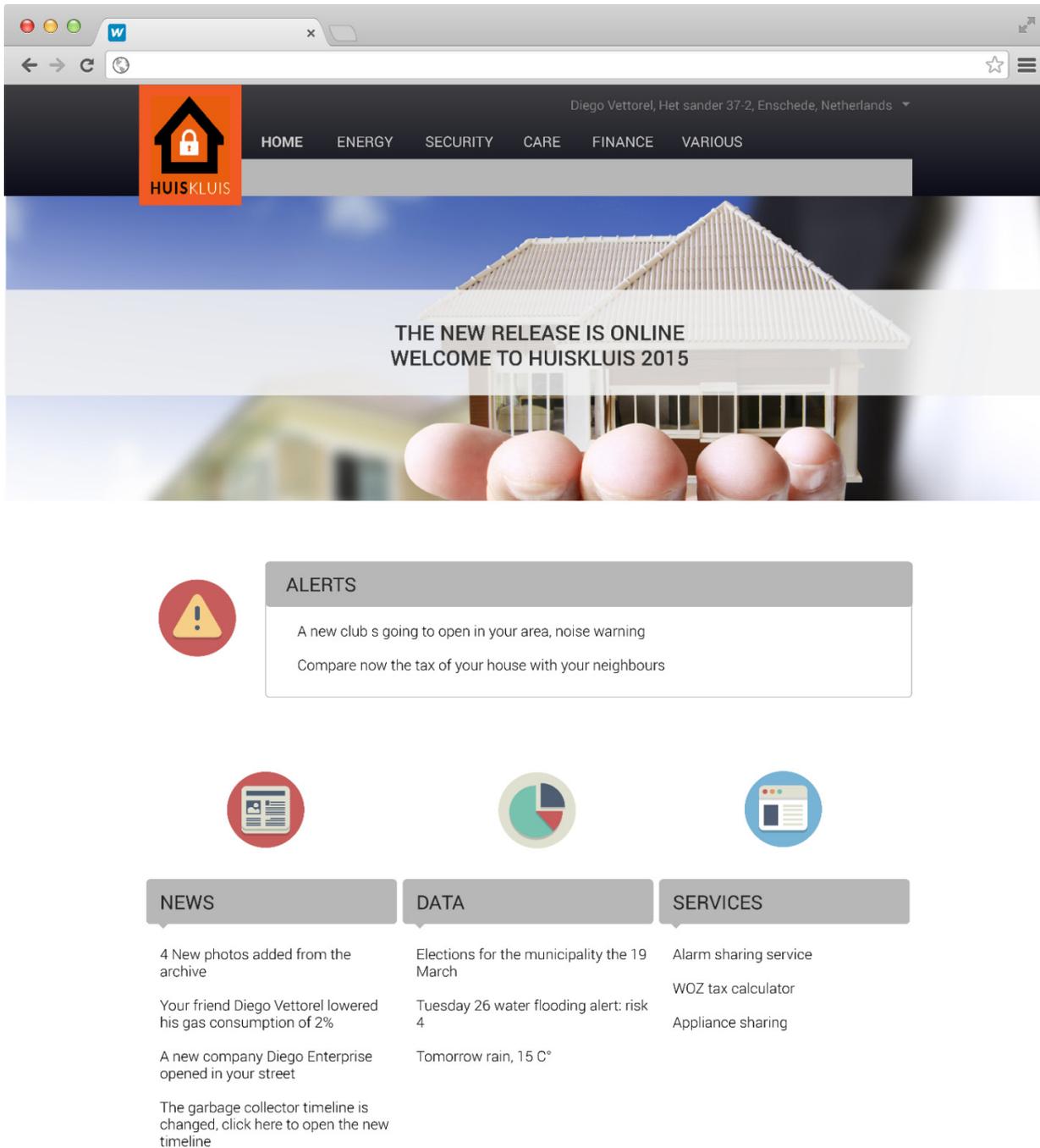


Figure 56: Mockup of the homepage with the end-user logged in

In Figure 55 the homepage of the platform is shown, assuming that the end-user already logged in on the platform. Included are the slider, alerts, news, main data and the main services.

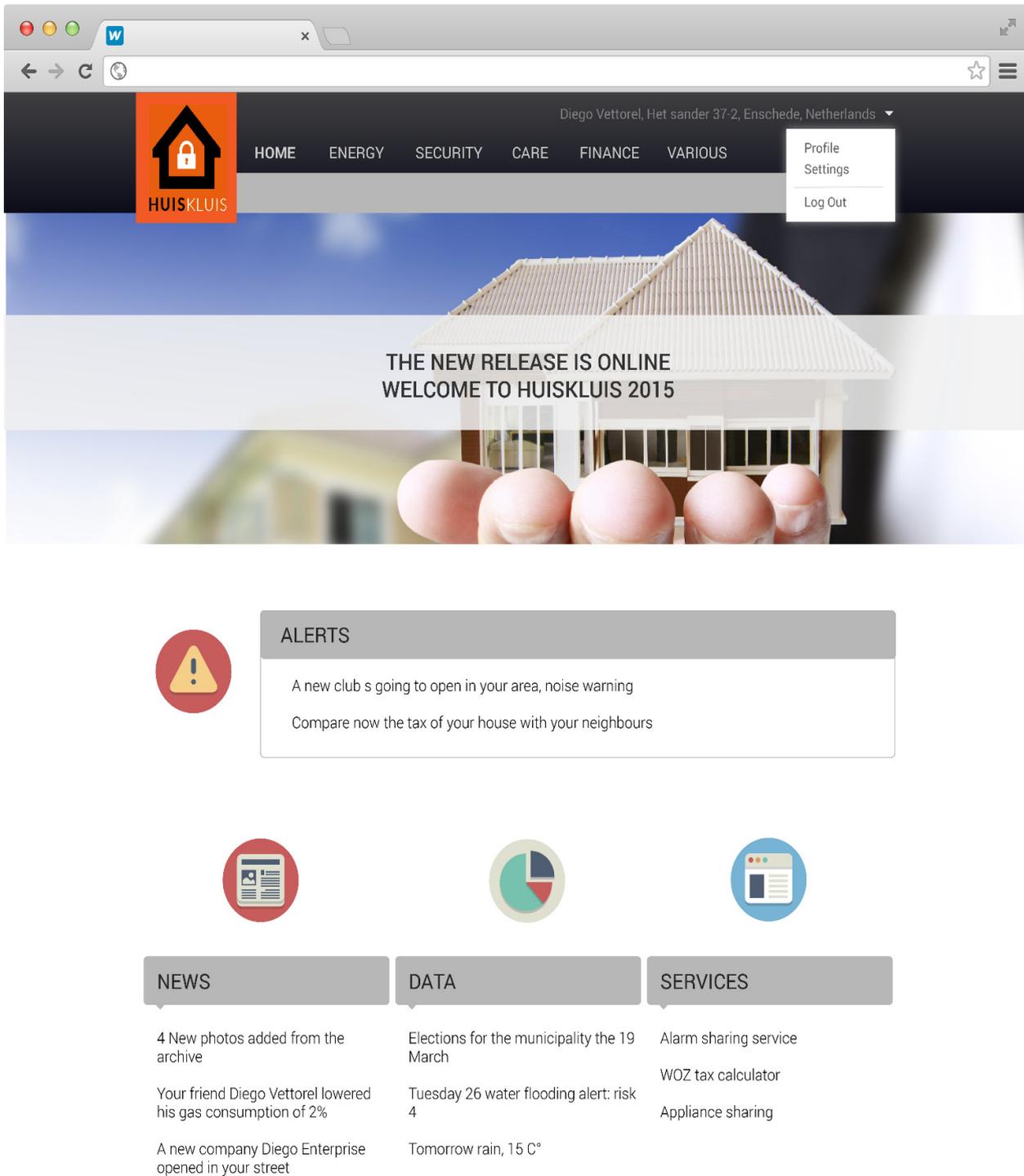


Figure 57: Mockup of the homepage with the end-user settings menu

In Figure 56 the homepage of the platform can be seen, it is assumed that the end-user did not login on it. In the image the reader can see the settings menu of the end-user in the top-right corner of the image. From it, the end-user can access his settings, profile and the log-out function.

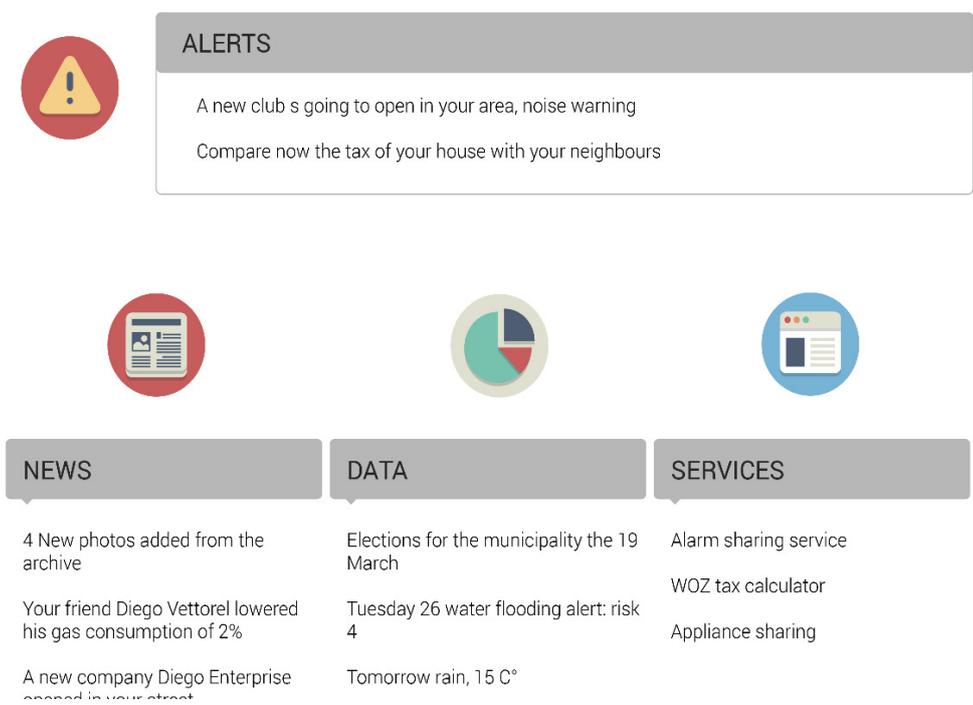
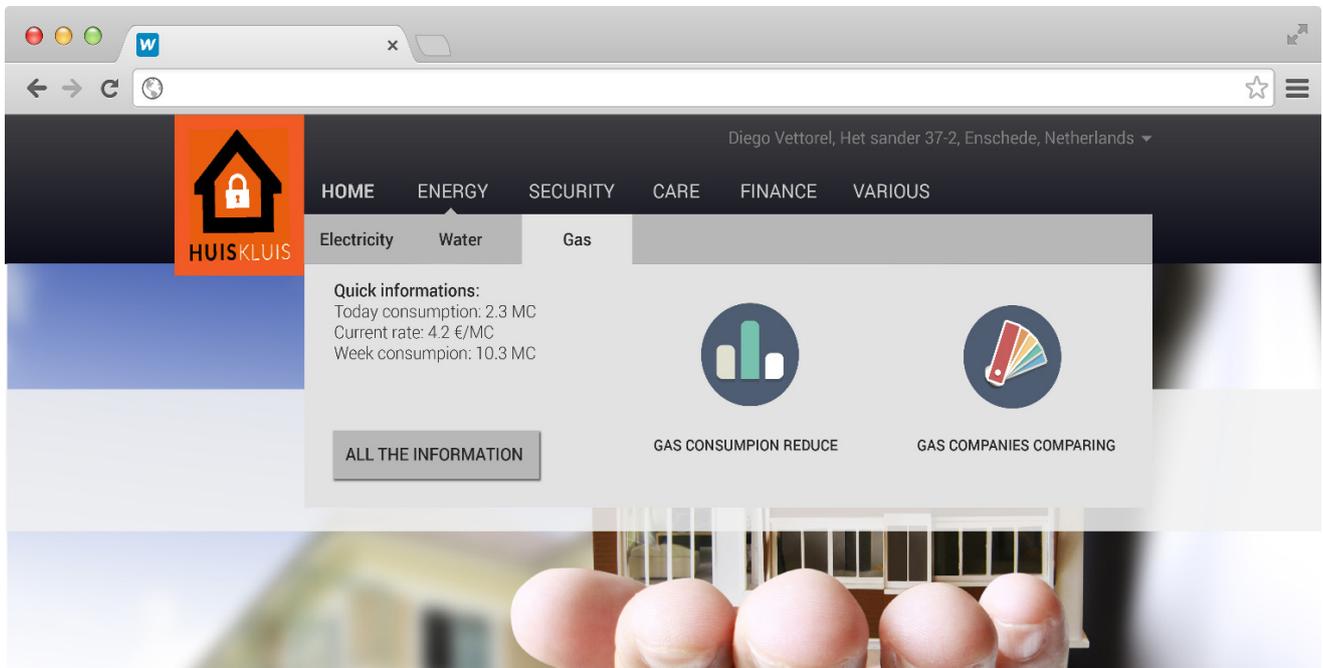


Figure 58: Mockup of the homepage with the gas submenu

In the previous figure the homepage of the platform is shown with the gas submenu open. As the reader can see, in this submenu the most important information and services about the gas category are grouped. If the end-user clicks on the "all the information" button he will be redirected to the gas webpage (shown in Figure 59).

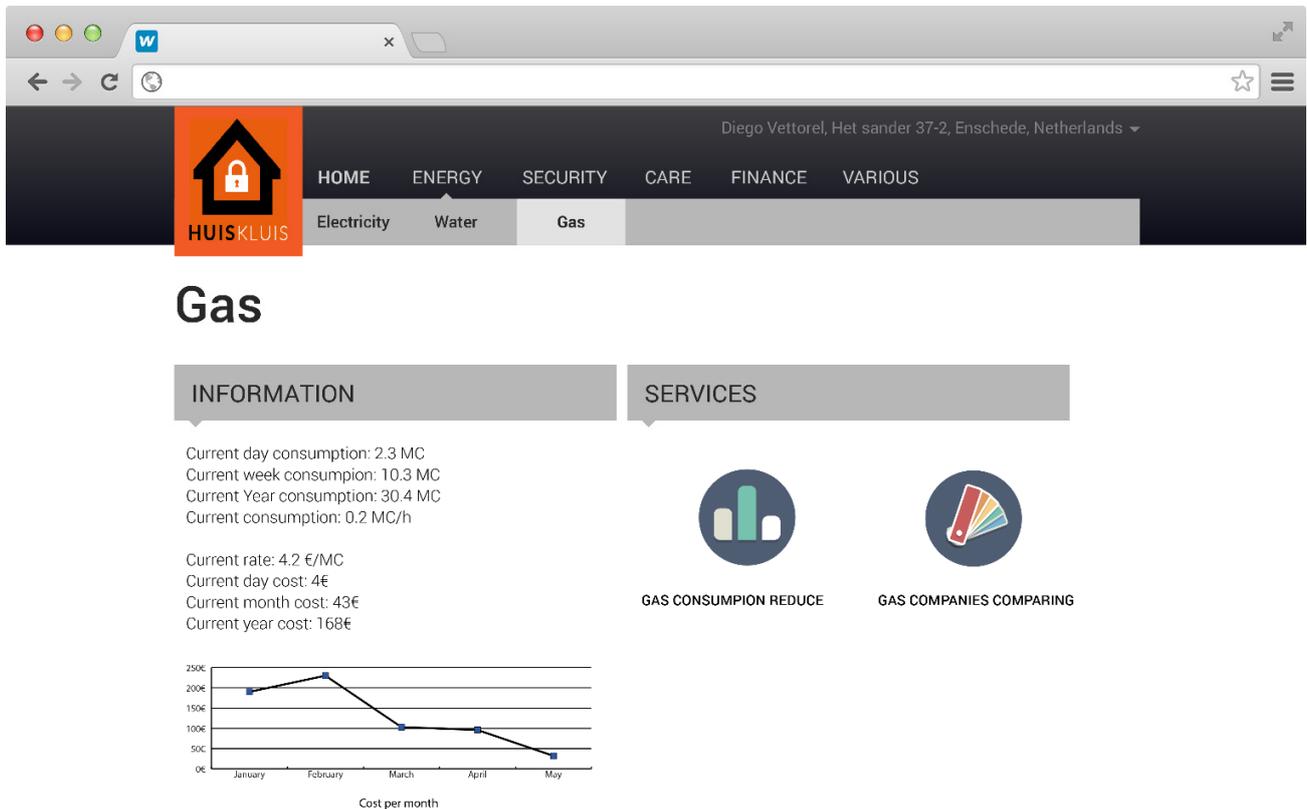


Figure 59: Mockup of the gas webpage

Figure 58 shows the gas category webpage. As the reader can see information and services connected with this category are included.

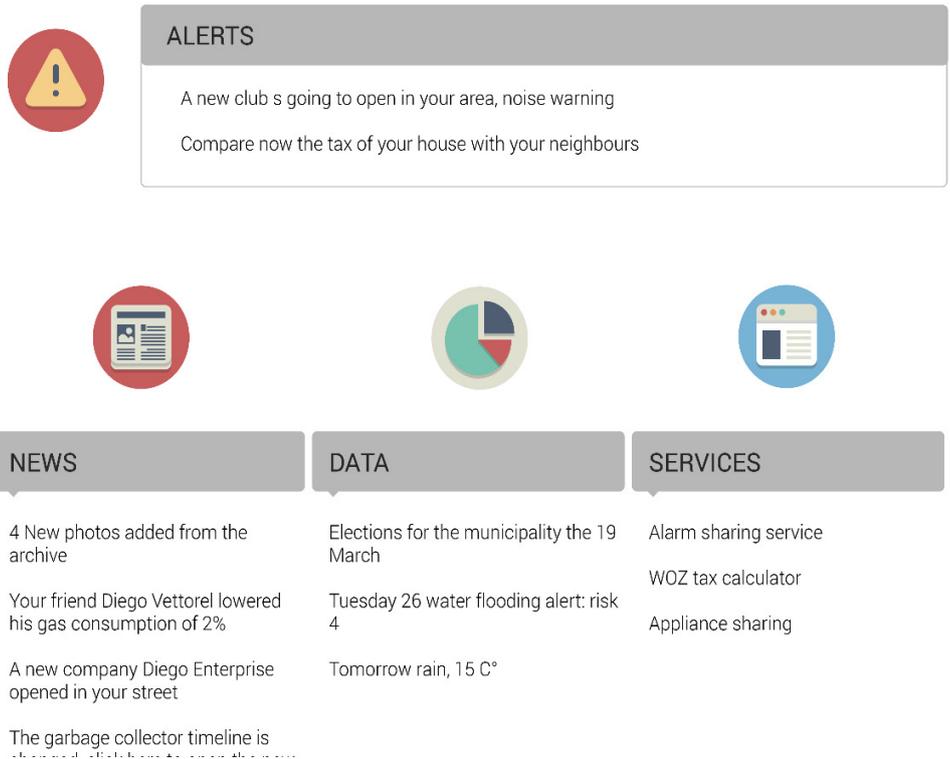
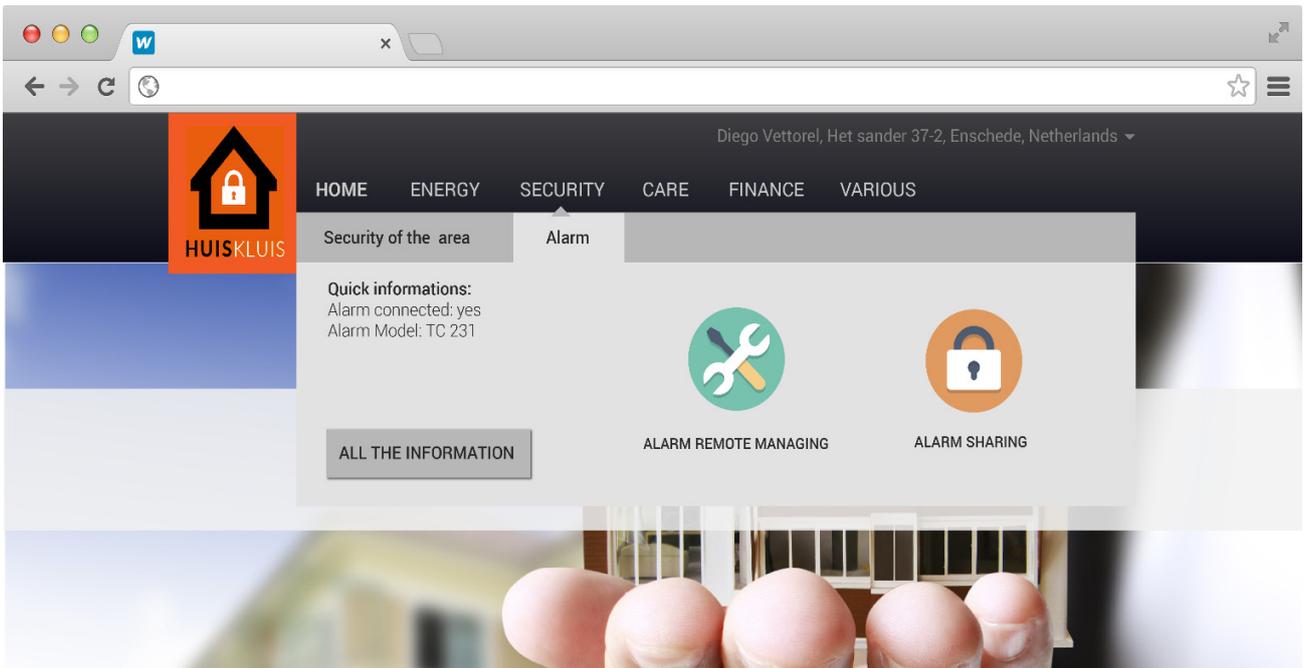


Figure 60: Mockup of the homepage with the alarm submenu

In the previous figure the homepage of the platform can be seen with the alarm submenu open. In this submenu the most important information and of the alarm category are grouped. If the end-user clicks on the “all the information” button he will be redirected to the alarm webpage (shown in Figure 61).

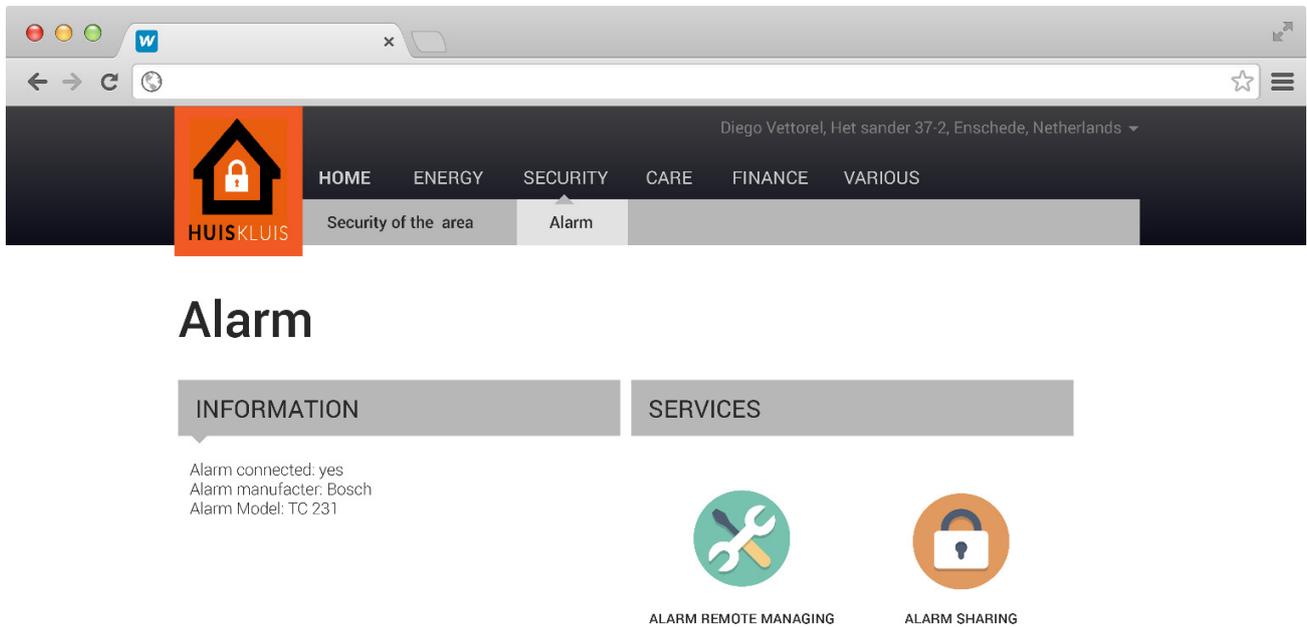


Figure 61: Mockup of the alarm webpage

Figure 60 shows the alarm category webpage. This category also includes information and relevant services.

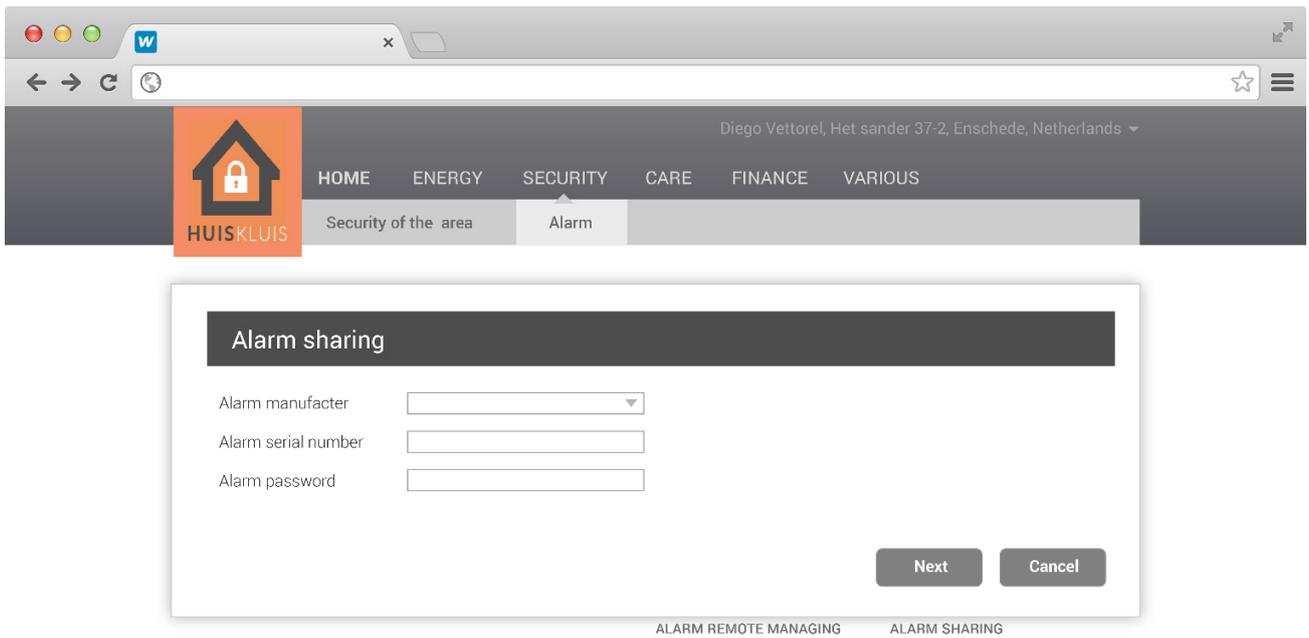


Figure 62: Mockup number one of the alarm sharing use case

In the previous figure is the interface of the first window of the “alarm sharing” use case (for further information about this use case please refer to section 3.4.1). As the reader can see the end-user can insert various pieces of information about his alarm.

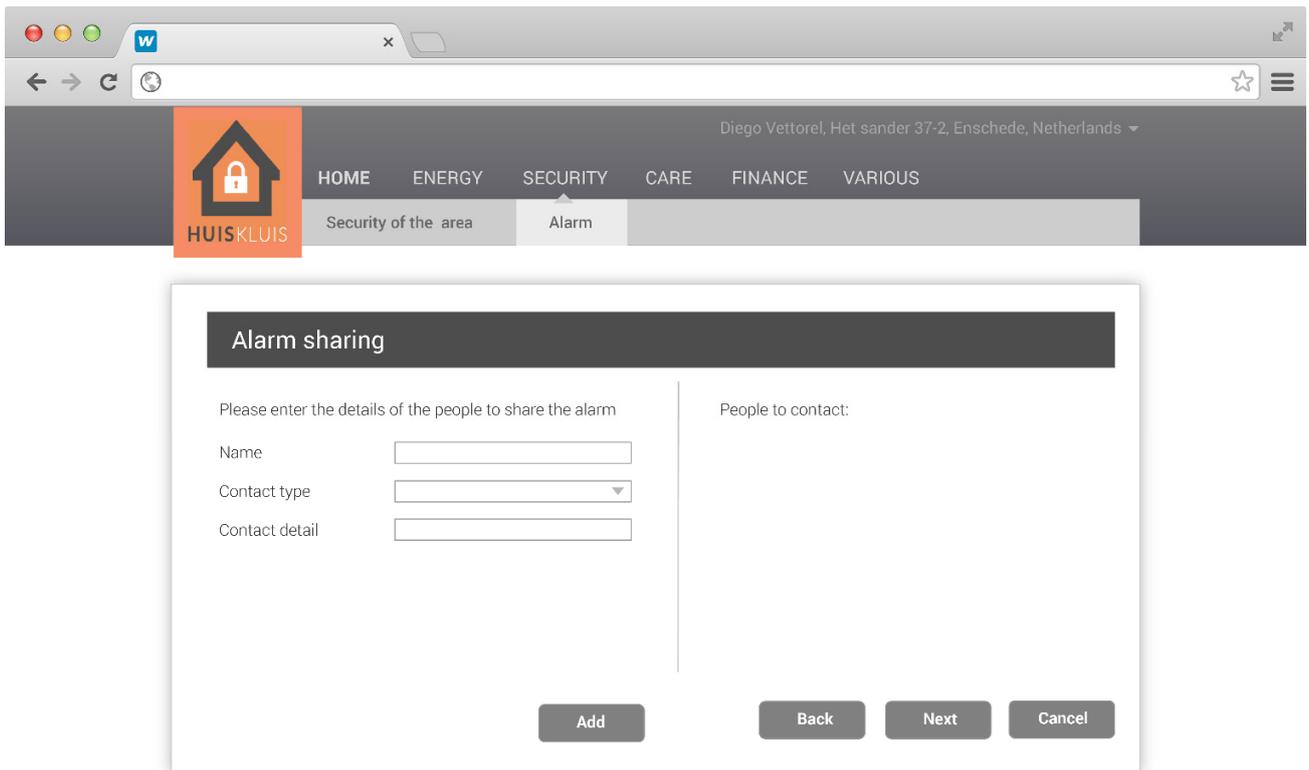


Figure 63: Mockup number two of the alarm sharing use case

Figure 62 is the interface of the second window of the “alarm sharing” use case (for further information about this use case please refer to section 3.4.1). As the reader can see the end-user can insert personal details of the end-users that need to be contacted with the information from the alarm.

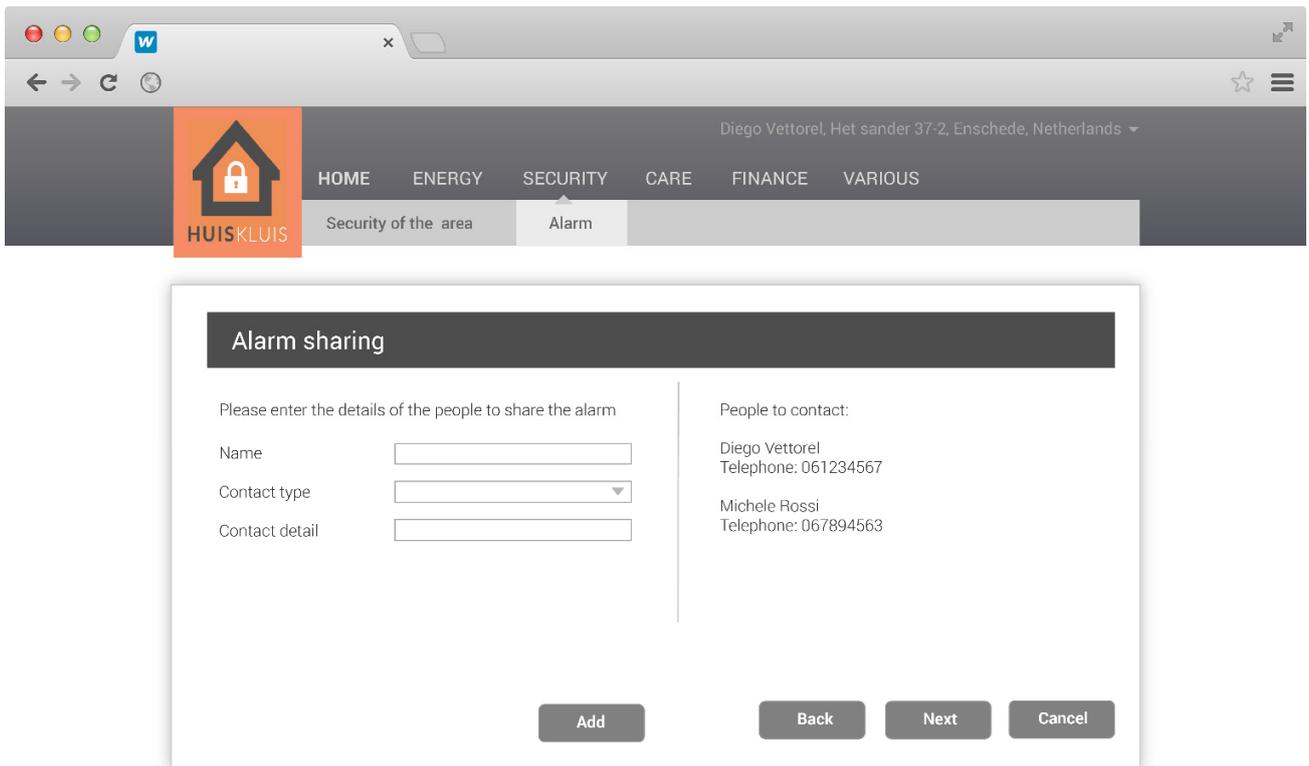


Figure 64: Mockup number three of the alarm sharing use case

The previous figure includes the interface of the third window of the “alarm sharing” use case (for further information about this use case please refer to section 3.4.1). The details of the end-users that need to be contacted with the alarm’s information are shown here.

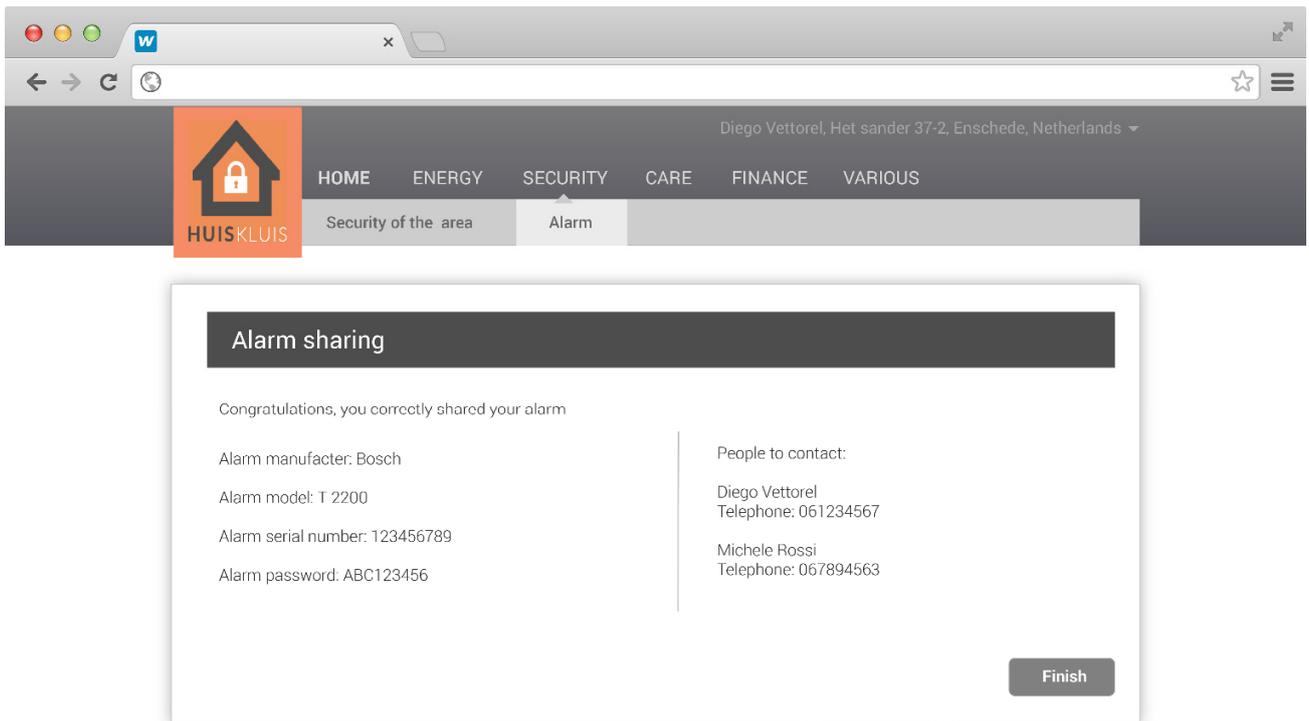


Figure 65: Mockup number four of the alarm sharing use case

Figure 64 presents the interface of the fourth window of the “alarm sharing” use case (for further information about this use case please refer to section 3.4.1). A summary of the information that the end-user inserted in the previous windows is shown here.