

# Improved UX of Network Device Monitoring

A Practical Example of Information Visualization

Lisa Claesson & Malin Tjärnemo



**LUNDS**  
UNIVERSITET



May 26, 2020

# Improved UX of Network Device Monitoring

A Practical Example of Information Visualization

©Copyright 2020 Lisa Claesson, Malin Tjärnemo

Published by  
Institution of Design Science  
Faculty of Engineering, Lund University  
Box 118, 221 00 Lund

Course: Interaction Design (MAMM01)  
Examinator Lund University, Faculty of Engineering: Christofer Rydenfält  
Supervisor Lund University, Faculty of Engineering: Kirsten Rasmussen-Gröhn  
Supervisors Advenica AB: Oskar Jönsson och Ardiana Osmani

# Abstract

Today, the possible amount of data to collect has grown with better computers and the internet. Therefore, the interest in how to visualize complex data has increased. In this thesis, a practical example of how to design an information visualization (IV) application has been conducted.

Advenica is a company located in Malmö that focuses on cyber-security. Advenica's product, ZoneGuard, works as a bridge between two security domains that blocks or allows network packages to transfer between the domains. Advenica saw a need to visualize the information exchange between the two domains and the status of a ZoneGuard.

To design an IV that is adapted to the users' needs, a user-centered design (UCD) approach was implemented. A UCD approach resulted in an IV application that Advenica was satisfied with. In that aspect, a UCD approach in this project was a success. However, with the limitation of access to real end-users, the end-users' user experience (UX) could not be fully tested. This was solved by instead testing on employees at Advenica. It is first when the application is tested on the end-users, that one can confirm if the visualization has a high usability or not.

Representative and interactive IV techniques were used to design the IV application. To represent information transferred through a ZoneGuard, vertical stacked bar graphs were used. In the application, there are examples of how bar graphs and line charts can be used to visualize data. There are also examples of how hovering over (mouse over), which is an interactive technique, can be used to enhance a user's understanding of the data.

**Keywords:** Information visualization (IV), user-centered design (UCD), user experience (UX)

# Sammanfattning

Idag är det möjligt att samla in större mängder data än det någonsin har varit möjligt att göra tidigare tack vare datorer och internet. Detta resulterar i ett ökat intresset för att visualisera komplex och stora mängder data. Detta examensarbete ger ett praktiskt exempel på hur en applikation för informationsvisualisering (IV) kan designas.

Applikationen som implementerades var en applikation för Advenica, ett företag i Malmö med fokus på cybersäkerhet. Advenicas produkt, ZoneGuard, fungerar som en bro mellan två säkerhetsdomäner för att blockera eller tillåta nätverkspaket att överföras mellan domänerna. Advenica såg ett behov av att visualisera informationsutbytet som sker mellan de två domänerna och statusen för en ZoneGuard.

För att designa en IV som är anpassad efter användarens behov användes en användarcentrerad designprocessen. En användarcentrerad designprocess resulterade i att Advenica var nöjda med applikationen. I den aspekten, var en användarcentrerad designprocess ett bra tillvägagångssätt. Däremot, kunde inte slutanvändarnas användarupplevelse testas på grund av att de inte var tillgängliga för detta projekt. Detta löstes genom att testa på Advenicas anställda. Det är först efter att användarupplevelsen testats på slutanvändarna som informationsvisualiseringen kan sägas ha hög användbarhet.

Representativa och interaktiva IV tekniker användes för att designa applikationen. För att representera informationsflödet genom en ZoneGuard användes bland annat vertikalt staplade stapeldiagram. I applikationen finns exempel på hur linjediagram och stapeldiagram kan användas för att visualisera information. Det finns också exempel på hur mouseover, vilket är en interaktiv teknik, kan förbättra användarens förståelse av data.

**Nyckelord:** Informationsvisualisering (IV), användarcentrerad design (UCD), användarupplevelse (UX)

# Acknowledgements

Heartfelt thanks to our supervisor at LTH, Kirre, to be with us throughout the thesis.

Heartfelt thanks to our supervisors at Advenica, Ardiana och Oskar, and the rest of the people at Advenica for helping us with our thesis and for the welcoming atmosphere at their office.

Heartfelt thanks to our friends and to our families for being there in ups and downs.

Lund, May 26, 2020

Lisa Claesson & Malin Tjærnemo

# Contents

<b>Acronyms and Abbreviations</b>	<b>9</b>
<b>1 Introduction</b>	<b>11</b>
1.1 Background . . . . .	11
1.1.1 Advenica AB . . . . .	12
1.1.2 Cross-Domain Solutions (CDS) . . . . .	12
1.2 Purpose . . . . .	13
1.3 Research Questions . . . . .	14
1.4 Work Load Distribution . . . . .	14
<b>2 Theory</b>	<b>15</b>
2.1 Interaction Design (ID) . . . . .	15
2.1.1 User-Centered Design (UCD) and User Experience (UX) . . . . .	15
2.1.2 Norman's Seven Fundamental Principles of Design . . . . .	16
2.2 Information Visualization (IV) . . . . .	17
2.2.1 Tufte's Criteria for a Good IV . . . . .	17
2.2.2 Representative Visualization Techniques . . . . .	18
2.2.3 Interactive Visualization Techniques . . . . .	19
2.3 Requirement Elicitation (RE) . . . . .	20
2.3.1 Interviews . . . . .	20
2.3.2 Affinity Diagram . . . . .	21
2.3.3 Brainstorming . . . . .	21
2.3.4 Design Workshop . . . . .	21
2.4 The Prototyping Process . . . . .	22
2.4.1 Deciding Fidelity-Level of the Prototype . . . . .	23
2.4.2 The Think-Aloud Protocol . . . . .	23
2.4.3 The Blank-Page Technique . . . . .	24
2.4.4 The "I like, I Wish, What if" Method . . . . .	24
<b>3 Phase 1 - Stakeholder Requirements</b>	<b>25</b>
3.1 Literature Search . . . . .	25
3.2 Unstructured Interview . . . . .	26
3.2.1 Method . . . . .	26

---

3.2.2	Result . . . . .	26
3.3	Requirement Specification (RS) . . . . .	27
<b>4</b>	<b>Phase 2 - Exploring User Needs</b>	<b>28</b>
4.1	The Prototyping Process . . . . .	28
4.2	Affinity Diagram . . . . .	29
4.2.1	Method . . . . .	29
4.2.2	Result . . . . .	29
4.3	Brainstorming . . . . .	31
4.3.1	Method . . . . .	31
4.3.2	Result . . . . .	31
4.4	Medium-Fidelity Prototype . . . . .	32
4.4.1	Method . . . . .	32
4.4.2	Result . . . . .	33
4.5	Stakeholder Validation . . . . .	37
4.5.1	Method . . . . .	37
4.5.2	Result . . . . .	37
4.6	Testing . . . . .	37
4.6.1	Method . . . . .	37
4.6.2	Result . . . . .	38
4.7	Analysis and Discussion . . . . .	39
4.7.1	Input to Next Iteration . . . . .	39
4.7.2	Analysis and Discussion of the Prototyping Process . . .	39
<b>5</b>	<b>Phase 3 - Visualizing Information</b>	<b>41</b>
5.1	The Prototyping Process . . . . .	41
5.2	Medium-Fidelity Prototype . . . . .	42
5.2.1	Method . . . . .	42
5.2.2	Result . . . . .	42
5.3	An IV Prototype of a CDS . . . . .	47
5.3.1	Method . . . . .	47
5.3.2	Result . . . . .	47
5.4	Stakeholder Validation . . . . .	48
5.4.1	Method . . . . .	48
5.4.2	Result . . . . .	48
5.5	Testing . . . . .	48
5.5.1	Method . . . . .	48
5.5.2	Result . . . . .	49
5.6	Analysis and Discussion . . . . .	50
5.6.1	Input to Next Iteration . . . . .	50
5.6.2	Analysis and Discussion of The Prototyping Process . . .	50

---

<b>6</b>	<b>Phase 4 - Refining the Visualization</b>	<b>52</b>
6.1	The Prototyping Process . . . . .	52
6.2	Design Workshop . . . . .	53
6.2.1	Method . . . . .	53
6.2.2	Result . . . . .	53
6.3	Medium-Fidelity Prototype . . . . .	55
6.3.1	Method . . . . .	55
6.3.2	Result . . . . .	55
6.4	Stakeholder Validation . . . . .	56
6.5	Analysis and Discussion . . . . .	58
6.5.1	Input to Next Iteration . . . . .	58
6.5.2	Analysis and Discussion of The Prototyping Process . . .	58
<b>7</b>	<b>Phase 5 - An Interactive Visualization</b>	<b>59</b>
7.1	The Prototyping Process . . . . .	59
7.2	Implementation . . . . .	60
7.2.1	Method . . . . .	60
7.2.2	Result . . . . .	60
7.3	Stakeholder Validation . . . . .	63
7.4	Analysis and Discussion . . . . .	63
<b>8</b>	<b>Discussion</b>	<b>64</b>
8.1	User-Centered Design . . . . .	64
8.2	Design Techniques . . . . .	64
8.3	Information Visualizations . . . . .	65
8.4	Constraints . . . . .	65
8.5	Future Work . . . . .	66
<b>9</b>	<b>Conclusion</b>	<b>67</b>
	<b>Bibliography</b>	<b>69</b>
	<b>Appendix A Requirement Specification</b>	<b>73</b>
	<b>Appendix B Cancelled Requirements</b>	<b>79</b>
	<b>Appendix C Prototype V.1 Test</b>	<b>80</b>
	C.1 Introduction Letter . . . . .	80
	C.2 Scenarios & Tasks . . . . .	81
	<b>Appendix D Prototype V.1 Result</b>	<b>83</b>
	<b>Appendix E ZoneGuard Visualization</b>	<b>87</b>



<b>Appendix F Prototype V.2 Test</b>	<b>89</b>
F.1 Introduction Letter . . . . .	89
F.2 Tasks . . . . .	90
F.3 Feedback Session . . . . .	90
<b>Appendix G Prototype V.2 Result</b>	<b>92</b>
<b>Appendix H Design Workshop</b>	<b>96</b>
H.1 Background . . . . .	96
H.2 Process . . . . .	96
H.2.1 Part 1: Prototype . . . . .	97
H.2.2 Part 2: Present and Critique . . . . .	97
H.2.3 Part 3: Converge . . . . .	98
H.2.4 Part 4: Prioritise . . . . .	98
<b>Appendix I Functional Prototype Result</b>	<b>100</b>

# Acronyms and Abbreviations

<b>CDS</b>	<b>C</b> ross <b>D</b> omain <b>S</b> olution
<b>GUI</b>	<b>G</b> raphical <b>U</b> ser <b>I</b> nterface
<b>HiFi</b>	<b>H</b> igh <b>F</b> idelity
<b>ID</b>	<b>I</b> nteraction <b>D</b> esign
<b>IV</b>	<b>I</b> nformation <b>V</b> isualization
<b>LoFi</b>	<b>L</b> ow <b>F</b> idelity
<b>RE</b>	<b>R</b> equirement <b>E</b> licitation
<b>RS</b>	<b>R</b> equirement <b>S</b> pecification
<b>UCD</b>	<b>U</b> ser <b>C</b> entered <b>D</b> esign
<b>UX</b>	<b>U</b> ser <b>E</b> xperience
<b>UI</b>	<b>U</b> ser <b>I</b> nterface

# Chapter 1

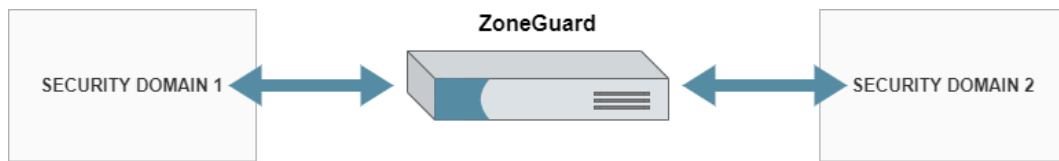
## Introduction

This chapter presents the background of This thesis and its purpose and research questions. The background gives an introduction to information visualization and introduces the company Advenica. It also contains an overall description of Advenica's cross domain solution(CDS) products: the ZoneGuard and the Data Diode.

### 1.1 Background

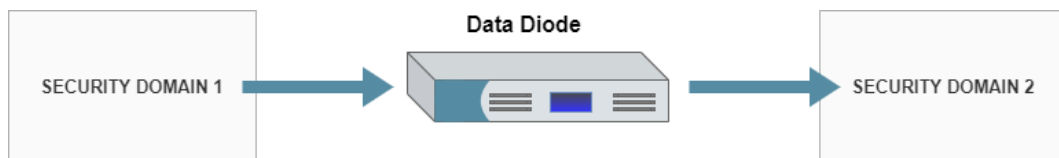
To visualize information is nothing new. Minard's famous visualization of Napoleon's 1812 march (Figure 1.1) is over 150 years old [1]. Yet, the field of information visualization as a standalone research area is only a little more than twenty years old. However, as the possible amount of data to collect has grown with better computers and the internet, the interest of how to visualize complex data increased[1]. Today, there are numerous books[1, 2, 3, 4] and research papers[5, 6, 7, 8] in the field of information visualization describing specific parts of the visualization and design process, for example visualization techniques and evaluation. As an alternative, this thesis demonstrates a practical example of information visualization. This thesis describes the process of designing an information visualization with high usability, from the start with only raw data to a finished visualization.





**Figure 1.2:** A ZoneGuard connected between two security domains. Information can be exchanged in both directions[9].

**Data Diode:** The second product is the Data Diode which is a one-way CDS. The key differences to the ZoneGuard are that the Data Diode makes two-directional information exchange impossible[10] and that does not filter the information. A Data Diode is used, for example, to guarantee that secret information in a security domain can not leak to a lower-level security domain by only making it possible to transfer data into the higher-level security domain. Figure 1.3 shows a Data Diode[10].



**Figure 1.3:** A Data Diode connected between two security domains. The information can only be transferred from security domain 1 to security domain 2[10].

The ZoneGuard and the Data Diode provide log control and can log any type of information. To do configuration on both the ZoneGuard and the Data Diode, Advenica provides an application where services and licenses can be modified.

## 1.2 Purpose

A CDS device contains a lot of information about data exchange. Today, this information is listed in text format. Advenica has found that there is a need to visualize this information to enhance the user experience. The visualization will make it easier for the user to understand and analyse the data exchange history and the status of the device to improve decision-making. Advenica have also found a need to handle multiple CDS devices at the same time. For example, to be able to monitor multiple CDS devices in a centralised view. Today, it is only possible to connect to one device at a time. Managing and monitoring multiple devices is therefore tedious work.

The goal of this thesis is to demonstrate how, for multiple devices, this can be visually represented and managed in a web application. The primary focus is to improve the user experience.

## 1.3 Research Questions

- RQ-1** How can network flow and information between two security domains be visually represented?
- RQ-2** How do users prefer to interact with multiple CDS devices in a network?
- RQ-3** What representative visualization techniques can be used to enhance the users' understanding of the network flow between two security domains and error search in a CDS device?
- RQ-4** What interactive visualization techniques can be used to enhance the users' understanding of the network flow between two security domains and error search in a CDS device?

## 1.4 Work Load Distribution

The work was mostly done at Advenica, but the functional prototype work was done remotely and both authors shared the workload equally through all phases.

# Chapter 2

## Theory

This chapter gives the most common definitions of Information Visualization, Interaction Design and User Experience relevant to this thesis. In the information visualization field, the chapter also describes the visualization process, representative and interactive visualization techniques. It also describes the prototyping process and the techniques that have been used in the process.

### 2.1 Interaction Design (ID)

Interaction design is described by Preece, Sharp and Rogers as *"Designing interactive products to support the way people communicate and interact in their every day and working lives"*[11]. One goal with ID is to address problems that occur, which can be perceived as annoying, confusing, ineffective, lacking in response or hard for the user. The earlier a problem is discovered, the less is the cost of fixing the problem. Therefore, it is essential to discover problems as early as possible in the design process.

A product developer should take these problems into account when designing a product. An approach to do this is by understanding who the user is and how the user will interact with the product. The user experience and how easy the product will be to use will depend on the type of user. That a product is easy to use can mean it has a low learning curve. It can also mean it is efficient to use or that the user gets a positive experience and to make the user want to use the product again[11].

#### 2.1.1 User-Centered Design (UCD) and User Experience (UX)

*"User-centered design is an iterative design process in which designers focus on the users and their needs in each phase of the design process"* according to

the Interaction Design Foundation[12]. The ISO definition of user experience (UX) 9241-210, is *"user's perceptions and responses that result from the use and/or anticipated use of a system, product or service"*[13]. Another common definition of UX is Nielsen och Norman's, *"all aspects of the end-user's interaction with the company, its services, and its products"*[14].

An approach to achieve a good UX is to use a UCD process[12]. In a UCD process, a combination of investigative and generative methods are used to involve the users[12]. Examples of investigative methods are interviews, surveys and observations. Examples of generative methods, are brainstorming, body-storming and role-playing. Which method to choose depends on the product itself, the users and the designers[15].

In a research phase, it can be better to go with interviews over surveys to receive quality over quantity. If there is a product available today, observation or a task demonstration may be used to gather information. In the end, there are many ways to come to a finished product[15].

### 2.1.2 Norman's Seven Fundamental Principles of Design

Don Norman's seven fundamental principles of design are used to create UI with good UX[16]. The following are the seven principles and how they can be applied in examples[16]:

1. **Discoverability:** Discoverability is about how easy it is for a user to recognise possible interactions with an object. If an action is not visible to a user, there is a need for a signifier.
2. **Feedback:** After the user has performed an action the user needs feedback that something happened. For example, if the user pressed on a "contact" button on a web page the feedback is given to the user by switching web page quickly to the contact web page. It is important that the feedback is not too much and annoying or too little to give the necessary information to the user.
3. **Conceptual Model:** A conceptual model is a representation of an object made to enhance users' understanding of the object. For example, today's computers have a desktop, folders and a trash bin to create a mental model of a physical desk.
4. **Affordance:** An affordance is a possible interaction between a user and an object. Affordance can, therefore, be described as a relationship between an user and an object. For example, if a chair is possible to lift by one person the chair has the affordance to be lifted. However, for a person that the chair is too heavy to lift, the chair does not have the affordance to



be lifted. An affordance does not have to be visible but for the designer, it is crucial that the users understand that the interaction is possible.

5. **Signifiers:** The possible interactions with an object need to be communicated to the user. This is achieved by using signifiers. A signifier can, for example, be a text, colour or sound. For example, the text "push" on a door signifies the door has the affordance to be pushed.
6. **Mapping:** A mapping is a relationship between two elements. For example, there is a relationship between a lamp and a lamp switch. The state of the lamp depends on the state of the lamp switch. Natural mapping is a mapping that is intuitive and obvious to the user. For example, ordering a set of lamp switches in the same order as the lamps in a room. Good mapping is a mapping that the user found easy to understand.
7. **Constraints:** Too many possible interactions can make the user confused. Therefore, it is necessary to limit the amount of action possible to the user. The interaction limitation is called constraints. Constraints can be logical, cultural and physical.

## 2.2 Information Visualization (IV)

Information Visualization is defined by Card, Mackinlay and Shneiderman as "*The use of computer supported, interactive, visual representations of abstract data to amplify cognition*"[3]. In the Figure 1.1 Minard's visualization of the Napoleons 1812 march is shown. The graph shows in 2D six different data; temperature, distance, latitude and longitude, size of Napoleon's army and direction. It also shows the army's placement relative to specific dates. This visualization is said to be the best statistical visualization according to Tufte's criteria described in the following section[2].

### 2.2.1 Tufte's Criteria for a Good IV

Edward Tufte's[1] definitions, graphical excellence and graphical integrity, are used to describe the difference between a good or a bad visualization:

**Graphical Excellence:** Tufte describes Minard's visualization as an example of an excellent IV. Graphical excellence is about presenting complicated data correctly and with a well-designed representation[2]. The user shall, with the help of the visualization, be able to make decisions based on facts by analysing the data. An excellent visualization of information is designed after the users' needs and increases the UX[2].

**Graphical Integrity** The graphical integrity is about telling the viewer the truth about the data. Tufte lists six principles that the visualization must follow to have graphical integrity[2]:

1. An object that represent the data must be proportional to the numerical quantities.
2. The visualization must be clear and have enough details to represent the information.
3. The visualization should visualize data variation and not design variation.
4. When visualizing time-series of money in economics, it is almost always better to display the real value of the money that has been adjusted for inflation instead of the nominal value.
5. A variable in n-dimensions should not be represented in more than n-dimensions.
6. The visualization should not take data out of context.

### 2.2.2 Representative Visualization Techniques

Fernandez and Fetais[6] lists six representative visualization techniques that can be used to represent data. These techniques are also described in[1, 17, 18, 4].

1. **Shape:** The shape of the visualization depends on the data structures. Some popular data structures are linear, temporal, spatial, network, hierarchical and geographic. Linear data structure can be represented as vectors, tables or pie charts. Temporal structures describe dynamic data that depends on time. Spatial and geographic structures are used to map data to a physical environment or map. Network and hierarchical structures are used to represent data relationships[1]
2. **Size:** Different sizes of an object should depend on the amount, the proportion or the importance of the object. A popular visualization, where the size is proportional to the data values, is the pie charts and side-by-side bars. Pie charts are a circular figure where the circle is divided into fractions to represent the data values[17].
3. **Colour:** Colours can be used for three different purposes, represent data values, highlights and categorise groups of data. The colour scheme used should depend on the purpose. If the purpose of adding colour is to

represent data values, a sequential or a divergent colour scale should be used[17]. For example, if the temperature in different countries should be visualized, a divergent colour pallet from blue to red could be used where the colour of a country depends on the temperature. To categorise groups of data, a qualitative colour scale should be used. A qualitative colour scale is a finite set of colours where the colours are easy to distinguish from each other and should not create an impression of order. Highlights should be used to point out specific data. The highlight colour should be from an accent colour scale and the baseline colour should not compete with it[17].

4. **Depth:** An IV can be in 2D or 3D. 3D adds a third dimension which gives the visualization a depth. This depth can be used to describe increasing and decreasing rates of values[6].
5. **Textures:** Textures is important for the visual perception of an object. Textures can be used to describe the depth or the surface of an object[4]. Textures are, for example, used in the medical field to describe anatomic structures in an information visualization[6].
6. **Opacity:** The opacity of a colour can be used instead of a sequential colour scheme to represent data values[18].
7. **Labelling:** Labels should be clear and correct and can improve the understanding of the visualization[6].

### 2.2.3 Interactive Visualization Techniques

Interactive visualization techniques are functionalities that enhance the UX and the understanding of the visualization. In accordance with Shneiderman's visualization task list[18], Fernandez and Fetais[6] have listed eight techniques that enhances the information quality for the user:

1. **Select:** The select functionality is used when the user should be able to interact with a specific object. For example, if the user should be able to move an object or get information about it[6].
2. **Filter:** The filter functionality is used when the user only wants objects with specific parameters/data to be shown[6].
3. **Zooming:** Zooming is, for example, useful in geographic data structures. When the user zooms in more details is shown about the object[6]. When the user zooms out more objects can be shown and a more complete overview is given. It is also used in graph structures with many nodes

and edges. The user should, for example, be able to zoom into a part of the graph to only see the relevant nodes and edges[1].

4. **Hovering over:** To get more information about an object, the user should be able to mouse over (also called hover over) the object[6].
5. **Reconfiguration:** Reconfiguration lets the user change the arrangement of the objects in the visualization. An example of a reconfiguration technique is to let the user sort the objects in a specific way[5].
6. **Blinking:** Blinking is useful in real-time application to get the user's attention by for example changing colors fast[6].
7. **Distortion:** Distortion is used to visualize time-oriented data by selecting a specific time-period[6].
8. **Customization:** Customization is an important feature to enhance UX for users with specific needs[1].

## 2.3 Requirement Elicitation (RE)

The techniques used to find and formulate requirements is called elicitation techniques[19]. The elicitation techniques used in this thesis is described in this section.

### 2.3.1 Interviews

An interview is a traditional qualitative data-gathering method[15]. It can be used to gather data about the present work and problems, goals and requirements[19]. It also gives the interviewer an insight into the interviewees' world around the subject, like opinions, thoughts and feelings[20]. An interview can be structured, semi-structured or unstructured. A structured interview consists of a set of pre-defined questions. An unstructured interview on the other-hand is more like a conversation where the interviewer decides the topic[15]. Many interviews are a combination of these two types, called a semi-structured interview or focused interview. These interviews use the combination of specific and open-ended questions used to elicit unexpected information that may be missed by only using specific questions[20].

How the interview is structured and who should be interviewed depends on the goal of the interview. The quality of the interview depends on the interviewers' skills. Some desired criteria brought up by Hove and Anda research group of what an interviewer need in part of skills include the following[20]:

- Encouraging the interviewees to talk freely.
- Asking relevant and insightful questions.
- Following up and exploring interesting topics.

### 2.3.2 Affinity Diagram

Affinity diagrams are about organising related facts into distinct groups[21]. This can help designers in making sense of the data collected in a data gathering phase. The data are categorised into smaller or bigger groups in relation to one another, and the process is usually done in the following fashion:

- Put the data on sticky notes, for easy transfer.
- Take one note at a time and compare it with the others already up, if they have a relation with any of them, put them in the same group, else form a new group. Repeat this until all the data is up.
- When all the data is up, name the groups according to the contents to get an overview.
- Some items may need to be reevaluated and moved between groups or create a new group during the process. And a big group may be broken down into subgroups.

### 2.3.3 Brainstorming

A brainstorming session is about generating ideas in a forum without judgement or criticism. Brainstorming creates a positive environment to generate ideas and when a session starts it may be slow, but can pick up speed quickly[15].

### 2.3.4 Design Workshop

A design workshop is used to brainstorm, prioritise and converge ideas in a group. A workshop can consists of the following steps[22]:

1. **Prototype:** The participants sketches and brainstorm a wide variety of ideas.
2. **Present and critique:** The participants discuss together what ideas they have come up with and give feedback to each other.

3. **Converge:** The participants converge on prototype ideas.
4. **Prioritise:** The participants prioritise the different ideas and in what order it should be prototyped and implemented.

## 2.4 The Prototyping Process

Prototyping is an effective and efficient method to use when developing a UI[23]. Richard Munoz defined prototyping as, "*Prototyping is externalizing and making concrete a design idea for the purpose of evaluation*"[24]. By creating prototypes, a visual representation of the written down software- and design-requirements is created. The prototype is then presented to stakeholders to find out if the prototype represents their requirements of the software to be developed. Errors from the RE, such as missing and misunderstood requirements, can be found by prototyping[25]. Prototyping can also be used to find usability flaws by performing usability tests. Prototyping can, in addition, be used to visualize different design ideas. To take advantage of the benefits of using prototypes and get closer to an optimal solution an iterative development process should be used[24].

The prototyping process consists of the following four steps[24].

1. **Plan:** Planning what should be prototyped is the first step in an effective prototyping process. This can, for example, be performed by choosing requirements, found from the requirement elicitation. Good requirements to prototype are requirements with high priority and that is suitable to prototype. Thereafter, the fidelity of the prototype should be chosen.
2. **Specification:** After deciding the fidelity of the prototype, the tools and methods used to design and create the prototype must be chosen.
3. **Design:** The design phase is where you create and design the prototype. Here, Norman's design principles and the theory about the IV can be used.
4. **Result:** In the result phase the prototype is tested and evaluated.

After the four steps, a prototype iteration is completed. If there is more time, and the requirements has change, a new prototype iteration can be performed to discover more RE errors and improve the usability[24].

### 2.4.1 Deciding Fidelity-Level of the Prototype

A prototype can be a low-fidelity (LoFi), a medium-fidelity, or a high-fidelity (HiFi) prototype. A medium-fidelity prototype is somewhere between a LoFi- and HiFi-prototype[23].

**LoFi-Prototype:** A LoFi-prototype is created quickly and can be done on paper with limited interaction and functionality. It can represent the general look and feel of the application but is far from the finished product. A LoFi-prototype is good to use in the early stages of a UCD process. It is efficient for RE when the customers and users don't know or have a hard time describing what they want from the application. With a LoFi-prototype, customers can focus on the fundamental design approaches instead of getting caught in details. The LoFi-prototype can then be used as a medium to improve communication and inform customers and users[23].

**HiFi-Prototype:** In contrast to a LoFi-prototype, a HiFi-prototype is fully interactive and is often developed either with code or with a design-application. A HiFi-prototype is more time consuming and expensive to develop but is more accurate and when usability testing more usability flaws are detected. A HiFi-prototype is ineffective to use in the early stages of the requirement gathering and should, therefore, be used when a requirement specification is produced[23].

### 2.4.2 The Think-Aloud Protocol

The think-aloud protocol is a technique used to get information on how test participants perceive information and how they process this information during problem solving[26]. The participants is encouraged to express any difficulties, emotions and thoughts while or after performing a task depending on if the concurrent or the retrospective think-aloud protocol is used. One wants however, that the thoughts are spoken as they occur when the thoughts are in the working memory, instead of having been processed into rationalised and transformed memories after some period of time. This can mean that the memories are filtered and not recalled accordingly when asked about in a questionnaire after the testing performed. There are some aspects to take into account when using this method. Those test participants who are unfamiliar with this method may found it difficult to use. It can be that the participants are having troubles expressing their thought with the right word, causing them somewhat staggering in their task, or that non-verbal thoughts are processed faster than speaking them out[26].

### **2.4.3 The Blank-Page Technique**

The blank-page technique is a technique that is used for coming up with ideas for contents and layouts, for example, a web page. Here the focus is on the users mental model of how they think something should look like after building up a mental model from exploring the rest of a web page. An example can be that during testing, one task could be to find the search feature to search for something. Where the users would look for this feature is an indication of how they have built their mental model. Another example is where users are encouraged to make sketches of what they think a layout of a web page should look like[27].

### **2.4.4 The "I like, I Wish, What if" Method**

"I like, I Wish, What if" is a method used to improve the feedback quality and variation when testing or discussing a prototype. The users have to present their feedback by starting a sentence with "I like", "I wish" or "What if". By dividing the feedback into three sections, the user expresses three types of feedback. An "I like" sentence results in positive feedback from the user where the user describes what is good with the prototype. An "I wish" sentence results in constructive feedback about what should be changed in the prototype and therefore prevent that the user only points out what is bad. Lastly, a user can express what could be added to the application in a "What if" sentence. The user can either write down the sentences or say it to the facilitator[28].



# Chapter 3

## Phase 1 - Stakeholder Requirements

This chapter describes how the literature search and how the interviews with the product stakeholders were performed. The requirement elicitation resulted in the first version of requirement specification, which can be read in its entirety in Appendix A.

### 3.1 Literature Search

The main databases used to find literature associated with the subjects of this thesis were LUBsearch and Google Scholar. Google Search was also used to find articles and web pages about the subjects. The literature search also consisted of reading Advenica's manuals for the ZoneGuard. This to obtain a deeper understanding of the device and what kind of data that can be extracted from it.

The literature search was primarily done at the beginning of this thesis to find tools and knowledge to have a foundation to build on. However, a continuous literature search was performed throughout the project to investigate areas further or to solve encountered problems. Employees at Advenica answered questions regarding their CDS products.

Two variants of search methods were used to find the literature used in this thesis. The first method was to search on keywords such as: *information visualization*, *data visualization*, *visualization techniques* and *user experience*. The second method was to look through the references used in the literature, found with the first method, to delve deeper into a subject.

## **3.2 Unstructured Interview**

### **3.2.1 Method**

An unstructured interview was performed at the beginning of the project with two of the project stakeholders and a third employee at Advenica. The participants had knowledge in Advenica's CDS products. The primary goal of the interview was to gather information about Advenica's requirements and expectations on the GUI. Another goal of the interview was to acquire an understanding of the present application that is used today.

### **3.2.2 Result**

From the interview, information about the present application used today was gathered. Today's application can only handle one CDS device and the log files are not presented in a GUI. The stakeholder requirements are summarised in the following list:

- The user shall be able to handle multiple CDS devices. For example, the user shall be able to update one or multiple CDS devices.
- Visualize the log file and data received from a chosen CDS device.
- Visualize the information about a CDS device that can be found in the configuration file.

### 3.3 Requirement Specification (RS)

From the interview and study of product manuals, the first draft of the RS was produced. The requirements were divided into eight sections. The draft was then verified by the stakeholders which resulted in the first version of the RS, found in Appendix A. In table 3.1 the type of requirements and the number of requirements in each section can be seen. The main points in the RS are the following:

- The user shall be able to do configurations on multiple CDS devices.
- The application shall visualize the data exchange between two domains of multiple CDS devices.
- The application shall present important information about the devices.

Requirement Type	Number of Requirement
Functional	8
Quality	1
Operating Environment	3
General	1
Visual	21
Data	3
Delivery	1
Prioritised	2

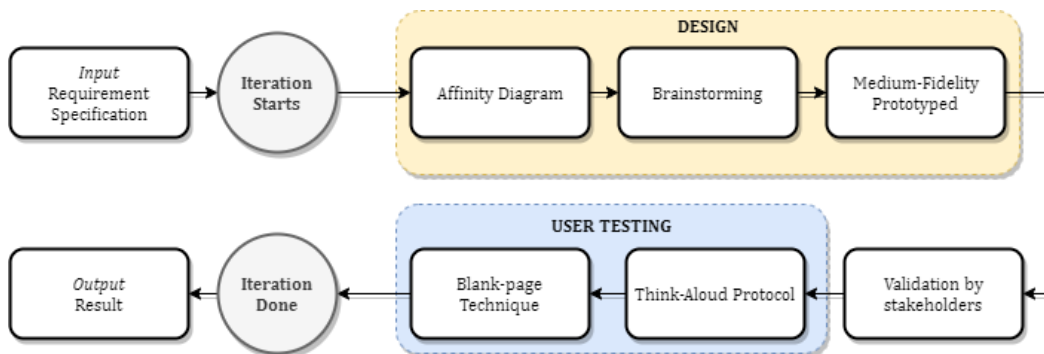
**Table 3.1:** Requirement type and the number of requirements of each type.

# Chapter 4

## Phase 2 - Exploring User Needs

### 4.1 The Prototyping Process

As mentioned in chapter 2.4 the prototyping process consists of four steps; *plan*, *specification*, *design* and *result*. What was performed in each step is described below and can be seen in Figure 4.1.



**Figure 4.1:** The prototyping process for the first iteration.

1. **Plan:** The requirements from the RS were planned to be prototyped. It was decided that the fidelity of the first prototype should be of low fidelity to enabling rapid development and testing.
2. **Specification:** In the Specification step, it was planned that the prototype should be created on paper. To generate ideas on how the application could be structured and designed, an affinity-diagram and a brainstorming session should be used at the beginning of the design step.
3. **Design:** In the Design step, an affinity diagram was created. The result from the affinity diagram was later used in the brainstorming session. Thereafter, the creation of the LoFi-prototype started. The fidelity of the

prototype was changed to medium-fidelity mid through. This decision is discussed in the medium-fidelity section.

4. **Result:** In the Result step, the test planning, testing, and evaluation took place. Before the user testing, the prototype was validated by the stakeholders. To test the prototype, the think-aloud protocol and the blank-page technique were used. After the iteration was completed the result from the iteration was used as input into the next iteration.

## 4.2 Affinity Diagram

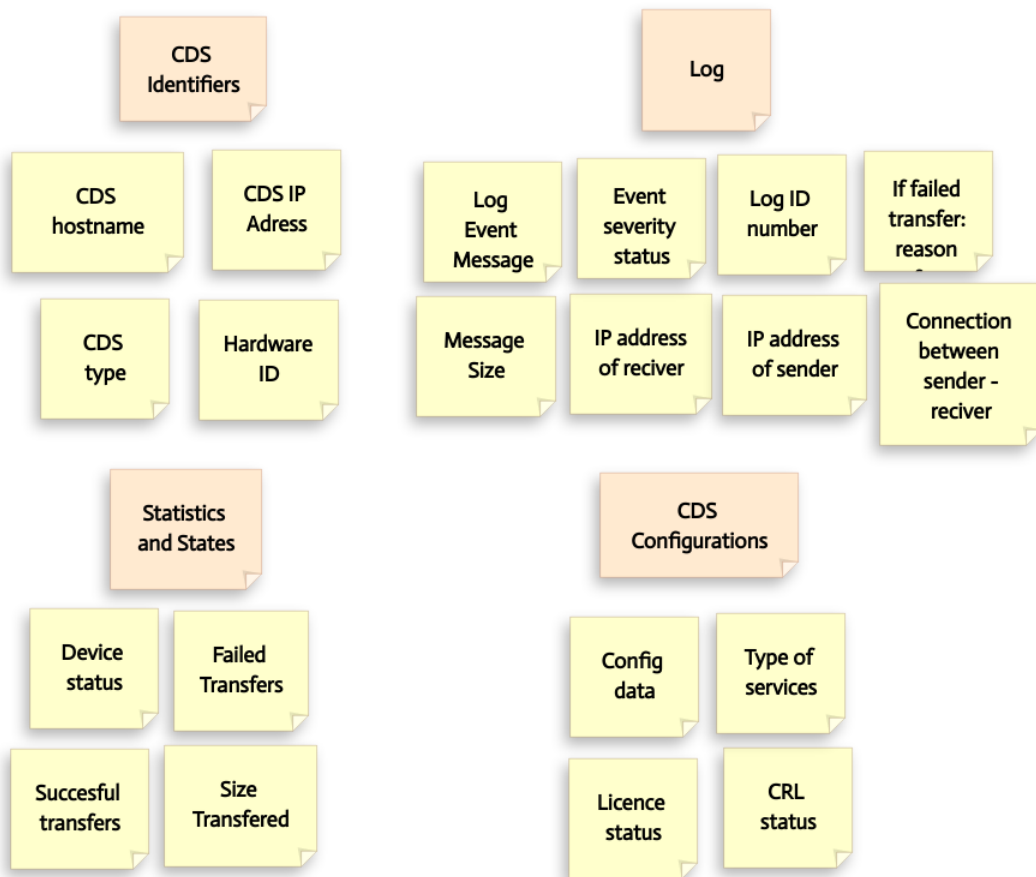
### 4.2.1 Method

After the first version of the RS was verified, an affinity diagram was produced. The affinity diagram method is described in chapter 2.3.2. The data were extracted from the RS document, see Appendix A, and only a few words were written per sticky-note. The sticky-notes were thereafter put up on a whiteboard in groups. After the groups were constructed, each group was labelled to describe the association between the data in a group.

### 4.2.2 Result

The affinity diagram can be seen in Figure 4.2. The affinity diagram consists of four groups; *CDS identifiers*, *Log*, *Statistics and States* and *CDS configuration*. The following groups are described below.

- **CDS Identifiers:** In this group the data that identifies a specific CDS, from multiple CDS devices, are listed. The data in the group are host-name, IP address, type and hardware ID.
- **Log:** In the log group, all information that can be extracted from one log event is listed.
- **Statistics and States:** In this group, the overall statistics and status of the CDS device are listed, for example, the total amount of data transferred.
- **CDS Configuration:** In the configuration group the configurable data is placed. For example, the config data can be changed by uploading a new configuration.



**Figure 4.2:** A readable version of the result from the affinity diagram session performed on a whiteboard with sticky-notes.

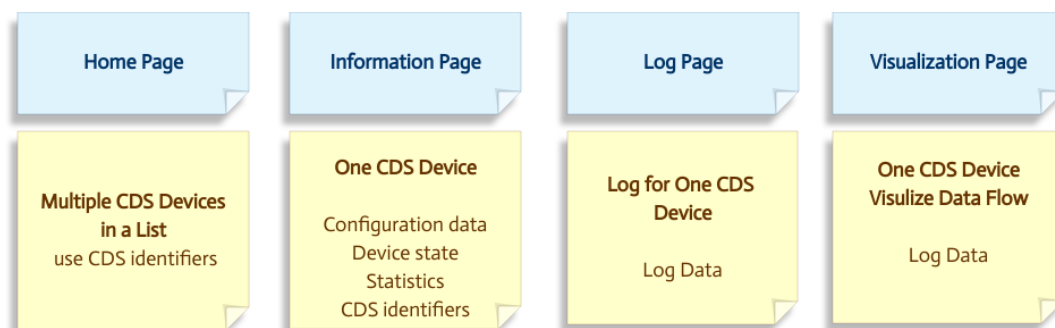
## 4.3 Brainstorming

### 4.3.1 Method

After the affinity diagram was created on a whiteboard, a brainstorming session started with only us two and lasted four hours. The brainstorming session consisted of discussing ideas and drawing ideas on paper and a whiteboard. In addition, ideas of how the user could interact and traverse between web pages were discussed.

### 4.3.2 Result

The brainstorming session generated ideas of what elements on the pages could possibly look like and where the data from the affinity diagram could be placed on a web page. From the session we found that the application could be designed with four web pages, a homepage showing multiple CDS devices and three pages showing information about a selected CDS device. The pages are represented in Figure 4.3. Some paper sketches from the brainstorming session can be seen in Figure 4.4. The paper sketch in the bottom left corner shows an idea of how the homepage could be designed, with a list of all CDS devices and ways to filter them. The right sketch demonstrates how pop-outs could be used to convey quick information about a specific CDS device in a list of CDS devices.



**Figure 4.3:** The primary pages that came from the brainstorming session and what general content to include on each page

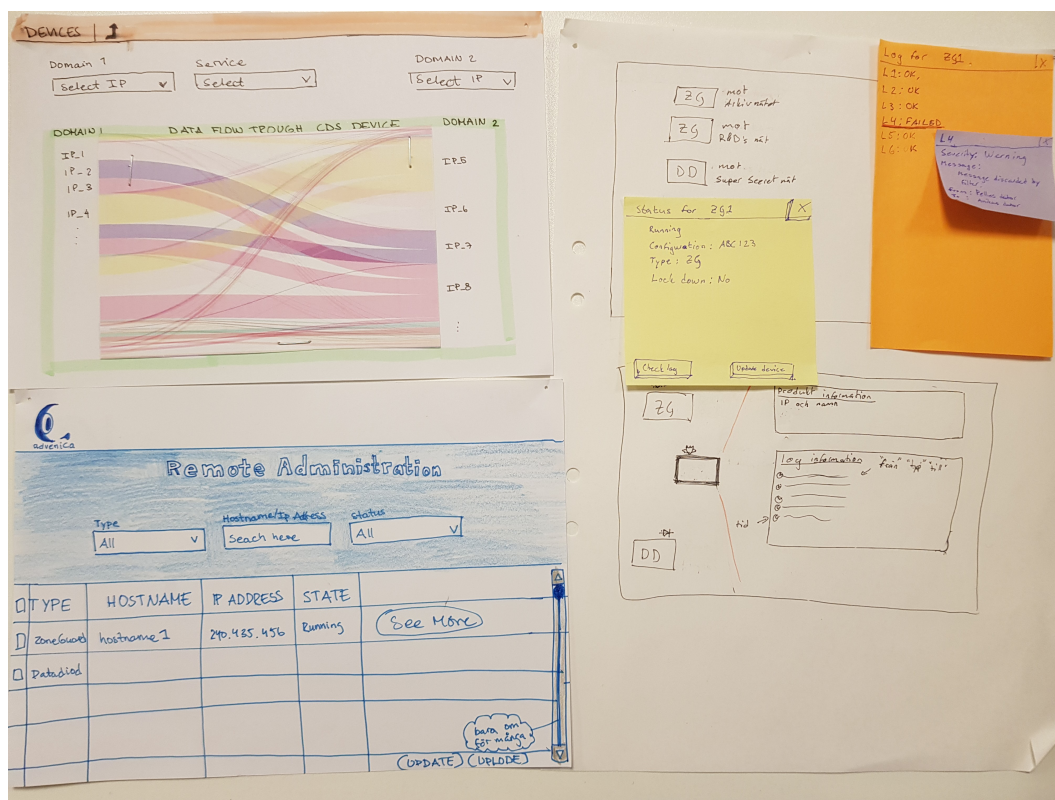


Figure 4.4: A handful of the sketches of elements to be considered for the first paper prototype.

## 4.4 Medium-Fidelity Prototype

### 4.4.1 Method

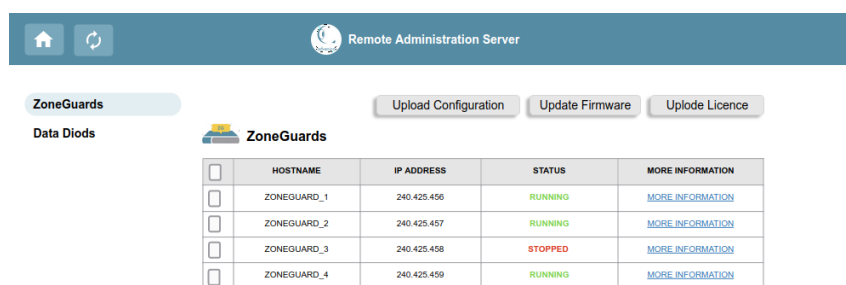
The plan was first to create a LoFi paper prototype. However, we found that it was too time-consuming to draw tables by hand and to make the modifications on paper. We both had experiences of using design-application tools to create prototypes, so we decided to switch to an online tool. A medium-fidelity prototype was designed with some interaction possibilities. The prototype was developed using Moqups[29] (Moqups is an online tool where the user can drag-and-drop user-interface components and add user interactions.)



## 4.4.2 Result

The result and ideas found in the brainstorming session were used to develop the prototype. A *ZoneGuard page* (homepage), a *Data Diode page*, an *information page*, a *configuration page*, a *log page*, and a *visualization page* were created. The pages can be seen in Figures 4.5-4.8. Norman's design principles were used in the development process to create the prototype. In each page description are examples of how Norman's design principles was used in the prototype. The following texts describe each page:

**Homepage:** The homepage, shown in Figure 4.5, is the first view a user sees. Here, the user can see all the ZoneGuards connected to the system. A similar view was produced for the Data Diodes. The user can click on the "Data Diode" text to be directed to the Data Diode view. The ZoneGuard can be separated by their unique IP address and hostname. The user could update configuration, update firmware and upload license to one or more ZoneGuards by clicking in a checkbox next to a ZoneGuard. However, these functionalities were never prototyped more than just with a button. To receive more information on a device, the user can click on the "MORE INFORMATION" text. To update the application, the user can press the second button in the blue-coloured header. The home-button next to the update-button directs the user to the homepage. The blue-coloured header is the same on all views. Norman's design principles were, for example, used at the more information-button. The discoverability is improved by applying blue colour to the text instead of the black colour used at the rest of the text. When clicking on the button the view changes and therefore, the user gains direct feedback.

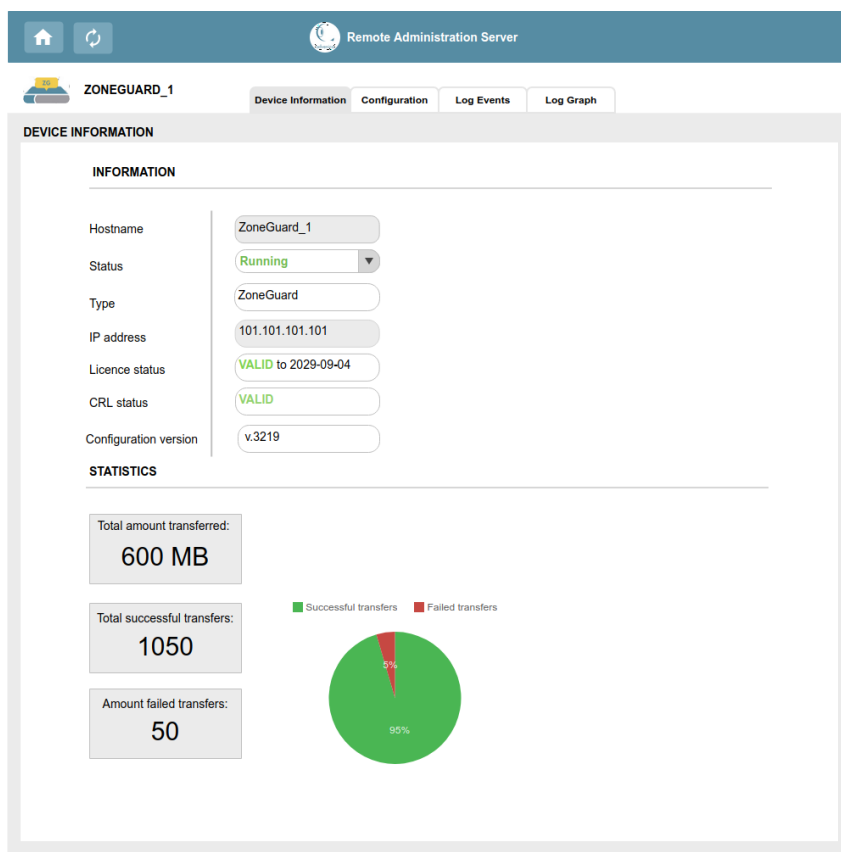


The screenshot shows the 'Remote Administration Server' interface. At the top, there is a blue header bar with a home icon, a refresh icon, and the text 'Remote Administration Server'. Below the header, there are three buttons: 'Upload Configuration', 'Update Firmware', and 'Uplode Licence'. The main content area is divided into two sections: 'ZoneGuards' (highlighted) and 'Data Diodes'. The 'ZoneGuards' section contains a table with the following data:

	HOSTNAME	IP ADDRESS	STATUS	MORE INFORMATION
<input type="checkbox"/>	ZONEGUARD_1	240.425.456	RUNNING	<a href="#">MORE INFORMATION</a>
<input type="checkbox"/>	ZONEGUARD_2	240.425.457	RUNNING	<a href="#">MORE INFORMATION</a>
<input type="checkbox"/>	ZONEGUARD_3	240.425.458	STOPPED	<a href="#">MORE INFORMATION</a>
<input type="checkbox"/>	ZONEGUARD_4	240.425.459	RUNNING	<a href="#">MORE INFORMATION</a>

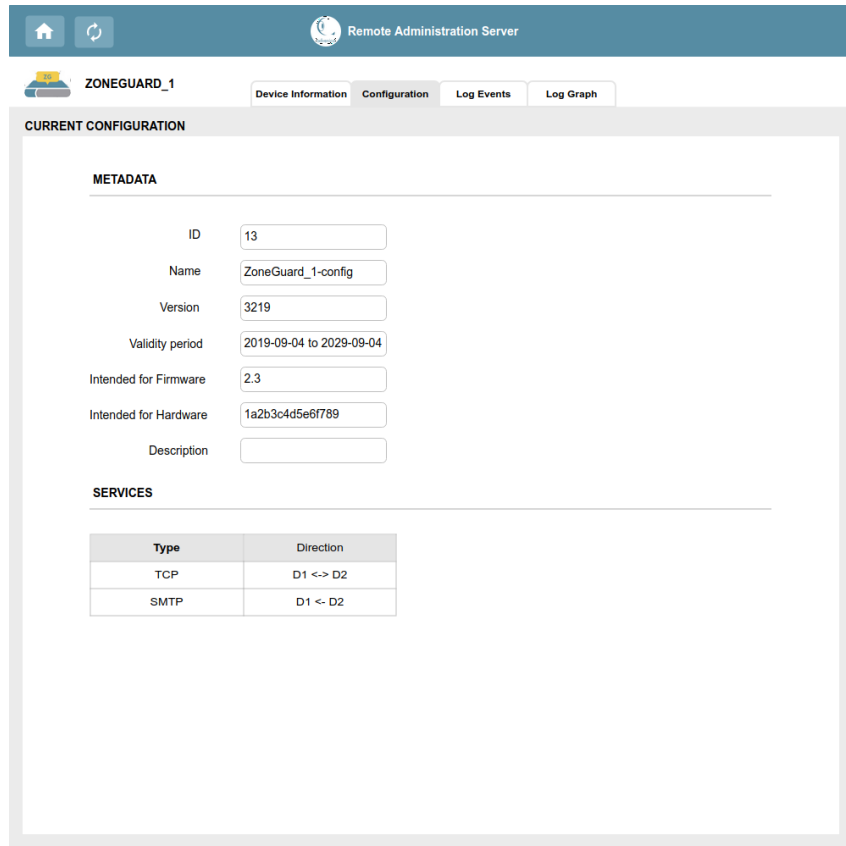
**Figure 4.5:** The first view of the prototype with a table of Zoneguards connected to the system, a similar view was made for the Data Diodes.

**Information Page:** After the user presses the "MORE INFORMATION"-text on a specific ZoneGuard or Data Diode the user is directed to the device information view. Here a user gets information on the specific device, such as the IP-address, the license, the CRL status, the configuration version, and the running status. At the bottom half of the view, a user can obtain statistical information. The statistical information that can be viewed is the amount of successful- and failed-transfers and the total amount of data transfers. The user can also see a pie chart of the numerical proportion of the successful- and failed-transfers. From here the mapping, in accordance with Norman's design principles, to other pages is through tabs at the top of the page. With the colour of the tab framing the active page to indicate what tab a user is on.



**Figure 4.6:** The information page of one CDS device with information about the CDS and statistics of the device.

**Configuration Page:** The configuration page can be seen in Figure 4.7. At the configuration view a user can see the current configuration metadata and the services running on the device. The metadata and services sections are not editable.



The screenshot displays the configuration page for a device named ZONEGUARD\_1. The page is titled "CURRENT CONFIGURATION" and is divided into two main sections: "METADATA" and "SERVICES".

**METADATA**

ID	13
Name	ZoneGuard_1-config
Version	3219
Validity period	2019-09-04 to 2029-09-04
Intended for Firmware	2.3
Intended for Hardware	1a2b3c4d5e6f789
Description	

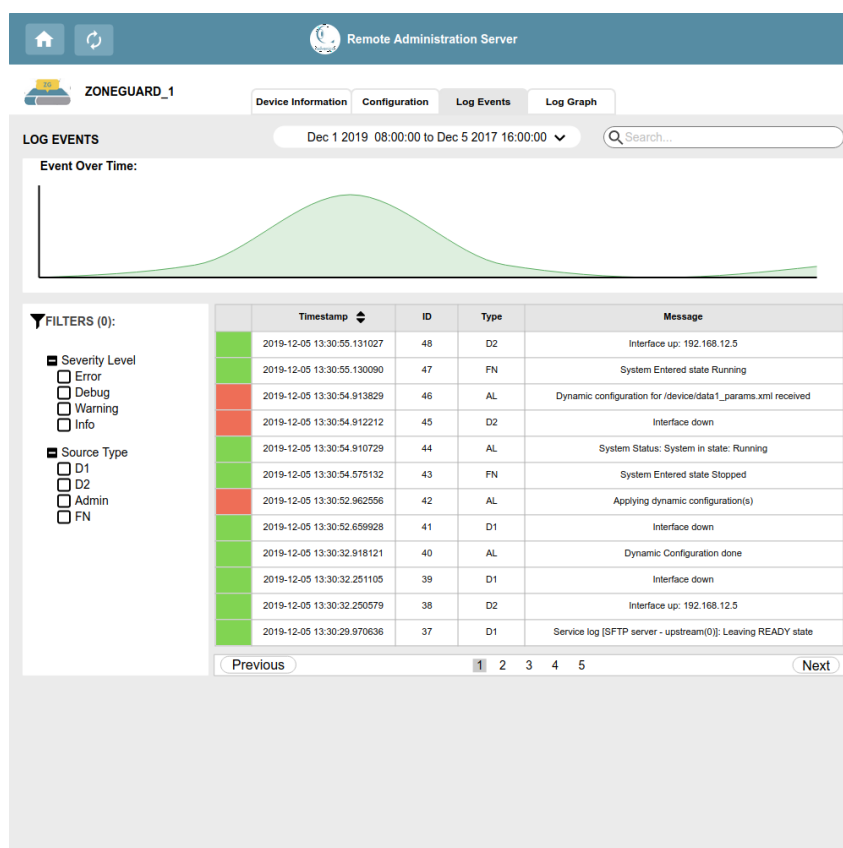
**SERVICES**

Type	Direction
TCP	D1 <-> D2
SMTP	D1 <- D2

**Figure 4.7:** The configuration page of one CDS device with the meta data from the CDS and which services that are running and to which directions.

**Log Page:** The log page can be seen in Figure 4.8. At the log page, a user can monitor the logs from a device. At the top of the log page is an "Events over time" graph. The graph represents the log events received at a specific time. The graph is missing labels and does not represent real data. It was decided that the graph should be prototyped and tested in later in the functional prototype. However, it was decided that the placement for the graph should be considered in the medium-fidelity-prototype. The logs can be filtered on severity level and source type in the filter section on the left side. A red-coloured log represents an error has occurred. The user can also search on an ID, Type or Message in the search field. Next to the search field is a date field where the user can choose logs in a specific time interval.

**Visualization Page:** The visualization page (called "Log Graph" in the tabs) was not prototyped. This was because we thought we needed more feedback from the users of what they wanted to be visualized.



**Figure 4.8:** The log page of one CDS device with a graphical timeline and a log that can be filtered.

## 4.5 Stakeholder Validation

### 4.5.1 Method

The prototype was presented to two stakeholders by showing the prototype and explaining the functionality and our design thoughts. The stakeholders could then express their thought on the design. This validation lasted for one hour.

### 4.5.2 Result

From the stakeholder validation, the stakeholders expressed some concerns about the functionality in the prototype. The main concern was that a lot of functionalities already existed in today's application. The conclusion was that the information page and the configuration page should not be in the next prototype iteration. The main focus should be:

- To create an overview of multiple CDS devices.
- Developing the log page.
- Find a way to visualize the data that can be received from a specific device.

The cancelled requirements can be viewed in Appendix B.

## 4.6 Testing

### 4.6.1 Method

The following step in the prototyping process was to test the prototype. The test session was performed at Advenica. We did not have access to the end-users because of security polices at Advenica. Therefore, with this limitation, the testing was instead performed on five employees with technical backgrounds from Advenica. One of the test participants works at Advenica's IT and Security Department and is used to monitor Advenica's devices. Therefore, this participant is the closest to a real end-users of the application.

Before the test session, an introduction letter describing the test session was sent to the participants. In the letter and at the start of the test-session the participant was informed that their comments would be used in this thesis. The

test session consisted of two parts. In the first part of the session, the user was asked to perform a number of tasks using the think-aloud protocol, described in chapter 2.4.2. In the second part, the user was asked to draw a visualization of the log page. The user could also choose to draw or write down what they thought should be visualized at the visualization page. This technique is called the blank-page technique and is described in chapter 2.4.3. The test session took between 20-40 min depending on the number of comments.

The introduction letter, the scenarios and tasks can be found in Appendix C.

### 4.6.2 Result

The complete test result can be found in Appendix D. The top usability problems on each side are described below.

**Homepage:** Four users wanted more information about the health and the status of a device. They also wanted to know from the homepage if there is a security issue with any of the devices. Three users had problems with their conceptual model because only one IP-address was shown when the ZoneGuard has four different IP-addresses.

**Information page:** All users had problems discovering which text fields that were editable. Three users requested a separate view when editing the text fields. They also wanted to have a save button.

**Configuration page:** The same usability problem with the editing in the information page was found on the configuration page.

**Log page:** To increase the discoverability of an event's source in the log, three users want to colour each source with a unique colour. They also wanted to be able to select an event to see similar events to the one selected. On the log page, many different kinds of feature requests were made. However, only three requests had more than one affected user.

**Visualization page:** Two users wanted the data flow to be visualized at this page.

## 4.7 Analysis and Discussion

### 4.7.1 Input to Next Iteration

The input to the next prototyping iteration was the feedback from the users and stakeholders. The goal of future development is described in the following list:

- To create an overview of both the ZoneGuards and Data Diodes. The overview should provide the user with important information regarding a device's security status and health status.
- Remove the configuration- and information-page and the functionality related to services, configuration and device information (see the requirements cancelled in Appendix B).
- Study the data and understand what kind of data that can be extracted from a device. Then, explore how the data can be visualized.

### 4.7.2 Analysis and Discussion of the Prototyping Process

There was a lot to be prototyped in the first iteration. The main problem we encountered was that we did not quite understand Advenica's products and the application that we were going to develop. From the RS we had requirements that the stakeholder thought that the end-user might want in the application. To categorise the data we did an affinity diagram and performed a brainstorming session.

The affinity diagram was quick to produce and provided an overview of the data. The brainstorming session was productive as it generated many ideas on how the data could be prototyped and categorised. It, in addition, gave us a mutual understanding and goal for the application. The disadvantage of the brainstorming session and affinity diagram was that we created assumptions about what we thought the stakeholders and users wanted. This was first tested in the stakeholder validation and the test sessions. Instead of starting the development of the prototype, we should have had a brainstorming session together with the users and stakeholders. This resulted in a loss of time as we were developing pages that the stakeholders did not want in the end. On top of this, we added interactions to the prototype which also was unnecessary because the main focus for the prototyping process was to find out more what the stakeholders and users wanted. This could have been prevented by using paper prototyping.

Because half of the pages were discarded (the information and configuration pages) we could have cut down on the number of tasks in the test sessions. However, the users did like the idea of having access to this information in the application. But because the functionality already existed in today's application the decision was that it should not be prototyped in future iterations. The think-aloud protocol was useful as it gave many comments on things that we did not notice. A problem we encountered at the test session was that many of the participants tried to think of how a possible end-user would think about the application. This resulted in feedback that started with the word "maybe". This feedback was hard to do something with in the end. We think a brainstorming session together with the participants would have resulted in more concrete feedback. The blank-page technique did not work in this prototyping process. This was because the participant did neither know what could be visualized and what data was available. They also thought it was hard to come up with an idea in such a short time. The ideas that were drawn we later found out was impossible to visualize because of lacking data.

From the test sessions, usability problems were found. The users had problems with discoverability and their conceptual model. To enhance the discoverability of an event's source on the Log page, a unique colour for each source can be added as a signifier[16]. To improve the conceptual model and enhance the user's understanding the table on the homepage needs to either include all the IP-addresses or none of them[16]. To improve the UX a number of affordances need to be added. For example, in the next prototype, a user shall be able to see the health of a CDS[16].

However, as a result of our first iteration, we obtained a more accurate understanding of what the stakeholders and users wanted. In addition, we acquired a better and more accurate understanding of Advenica's products and what kind of data that can be extracted from the devices.

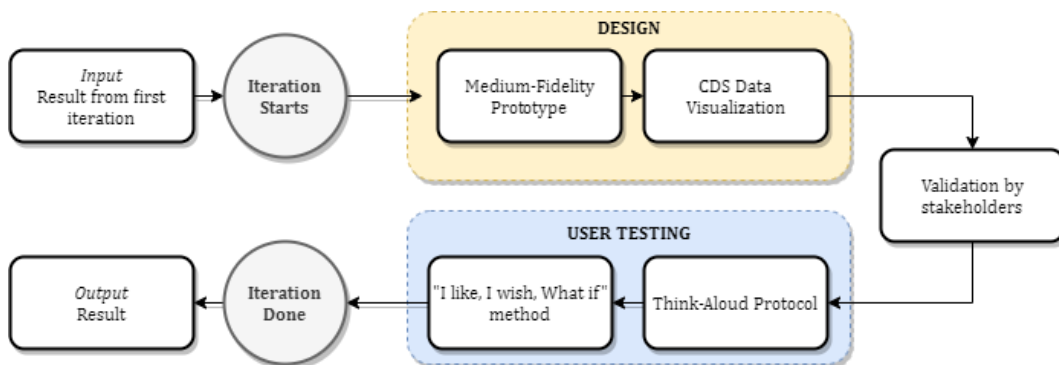


# Chapter 5

## Phase 3 - Visualizing Information

### 5.1 The Prototyping Process

The process for the second prototype is described in Figure 5.1 and in the text below.



**Figure 5.1:** The prototyping process for the second iteration.

1. **Plan:** In the Plan step, the data that can be received from a device was planned be prototyped. The fidelity of the second prototype was to be medium with no implemented interactions to be able to test it quickly.
2. **Specification:** In the Specification step, it was decided that the prototype should be created in Moqups. As seen in Figure 5.1, the input to the iteration was the result from the previous iteration.
3. **Design:** In the Design step, a data visualization of the ZoneGuard was created. The log page and homepage was also redesign and refined according to the user and stakeholder feedback from the previous iteration.
4. **Result:** After the prototype was finished, the Result step started. In the Result step, the test planning, testing and evaluation took place. Before

the user testing, the prototype was validated by the stakeholders similar to the first iteration. To test the prototype, the think-aloud protocol and the "I like, I wish, What if" method were used.

## 5.2 Medium-Fidelity Prototype

### 5.2.1 Method

This prototype was produced with the same Mock-up tool as the first iteration, Moqups. The prototyped was, unlike the first iteration, designed without interaction to reduce development time. This prototype focused foremost on the layout of the content, for example, grouping and placement of the graphs and cards. The representative visualization techniques that were explored in this prototype were shape, size and depth.

### 5.2.2 Result

The result of the prototyping can be seen in Figures 5.2-E.3. The pages prototyped were a new version of the *Homepage* and the *system log page* and a new page called the *device monitoring page*. The graphs were not prototyped with realistic data, instead, the focus was on the placement of the graphs and the types of graph designs.

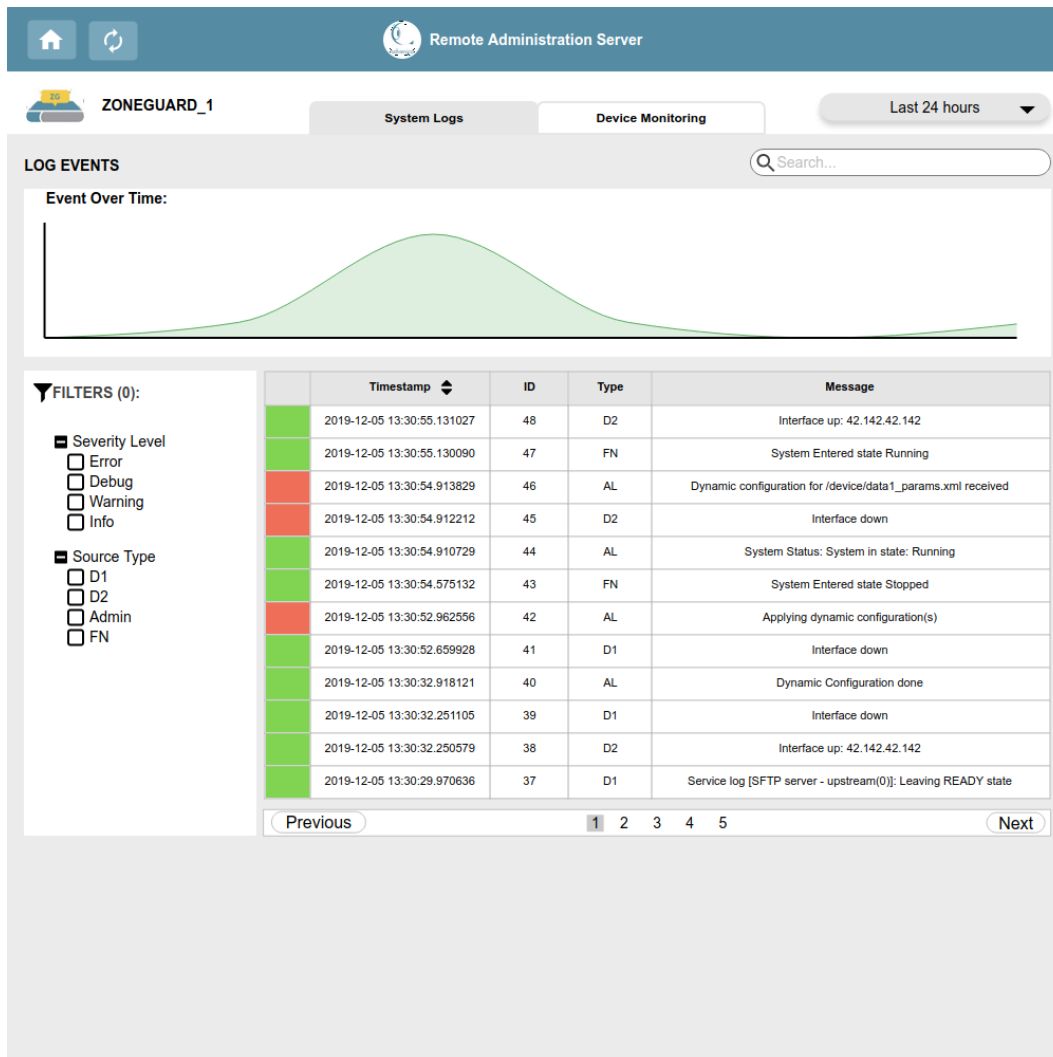
**Homepage:** From the first iteration the users requested an overview of the security state and hardware status of the devices at the homepage. The majority of the users also wanted ZoneGuards and Data Diodes on the same page. This resulted in the new homepage view that can be seen in Figure 5.2. The ZoneGuards and Data Diodes are both seen on the homepage. To enhance the users' understanding of the security state and hardware status of the devices a column called Health and a list of warning messages were added. The new column, Health, indicates the overall security state and hardware health. The warning messages notify a user about recent events that affect the health and status of the devices. The events are coloured in red and orange to indicate the severity level of the event. A user can delete a warning message by clicking the delete button on the right top corner on a message. To receive more information about a device, the user can click on the device's hostname.

The screenshot shows the 'Remote Administration Server' interface. At the top, there is a navigation bar with a home icon, a refresh icon, and the text 'Remote Administration Server'. Below this, the main content is divided into three sections:

- ZoneGuards:** A table with columns 'HOSTNAME', 'STATUS', and 'HEALTH'. It lists four ZoneGuard devices: ZONEGUARD\_1 (RUNNING, OKEY), ZONEGUARD\_2 (RUNNING, OKEY), ZONEGUARD\_3 (STOPPED, BAD), and ZONEGUARD\_4 (RUNNING, GOOD).
- Data Diodes:** A table with columns 'HOSTNAME', 'STATUS', and 'HEALTH'. It lists four Data Diode devices: DATADIOD\_1 (RUNNING, GOOD), DATADIOD\_2 (RUNNING, GOOD), DATADIOD\_3 (STOPPED, GOOD), and DATADIOD\_4 (RUNNING, GOOD).
- Warning Messages:** A list of three messages on the right side, each with a close button (X):
  - ZoneGuard\_3: STOPPED - SECURE STATE (Red message)
  - DataDiode\_3: STOPPED (Red message)
  - ZoneGuard\_1: CPU Load over 90 % (Yellow message)
  - ZoneGuard\_2: CPU Load over 90 % (Yellow message)

**Figure 5.2:** The homepage of the prototype with two tables, one for ZoneGuard devices and one for Data Diodes devices that are connected to the system. To the right side of the page is a list for warning messages.

**System Log Page:** This page is almost the same as the first prototype. The new log page can be seen in Figure 5.3. The only change was how the user changed the time span for the logs. The time span can now be changed by clicking on the drop-down menu at the top right corner, from both the log page and the device monitoring page. This change was done to imply that the time span would remain the same on both pages. For example, if the user selects "last hour" in the system log page and then switches tab to the device monitoring page the time span will be last hour for device monitoring page. The time span was also changed to specific time spans like one hour and 24 hours based on of the user feedback.



**Figure 5.3:** The view of the log page, mostly the same as the first prototype, see Figure 4.8, but with the time span menu at the top right corner.

**Device Monitoring Page:** The device monitoring page can be seen in Figure 5.4. This page presents the data that can be received from a CDS device. The figure in the top left corner represents a ZoneGuard connected between two security domains. The arrows, upstream and downstream, represent the data flow between the domains. The filter is represented with a dashed lined square. The figure is used to help the user understand the conceptual model of the page.

The area in the top right corner of the page is for warning messages. These messages also show up on the homepage.

The system overview card contains data about the hardware. Here there is information about the CPU-load over time, RAM usage over time, for how long the device has been up, the current temperature of the device and how much memory is free and being used relative to each other.

The downstream and upstream cards contain the data corresponding to the upstream and downstream arrows in the top left figure. The layout of the upstream card is the same as the downstream card. The downstream card contains the following information:

- The area graph to the left represent transferred and received packages over time.
- The line graph to the left represent blocked packages over time of a specific service.
- The first pie chart to the right represents the proportion of service type sent from D2.
- The second pie chart to the right represents the proportion of service type received at D1.
- The first box to the left presents the total number of packages to D1.
- The second box to the left presents the total number of blocked packages from D2 to D1.

The unit is in packages but can be changed to bytes in the gear icon at the top right corner in each card.



Figure 5.4: The view of the device monitoring page.

## 5.3 An IV Prototype of a CDS

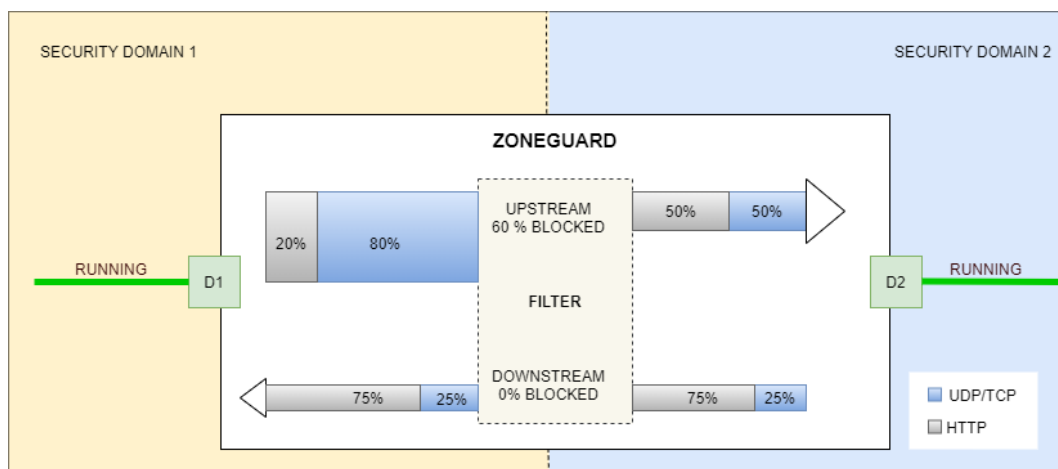
After the medium-fidelity prototype was created an attempt to make the data more understandable to the user was performed. Instead of using well-known graphs a customised figure of a ZoneGuard was prototyped. The idea was that the figure should visualize the data in the downstream and upstream cards in another way and create a conceptual model of a ZoneGuard.

### 5.3.1 Method

The prototype was developed using Google's Draw.io. The prototype would then be printed out on paper to be tested. The representative visualization techniques that were explored in this prototype were shape, size and depth.

### 5.3.2 Result

The result can be seen in Figure 5.5. The figure visualizes the data flow in a ZoneGuard connected between two security domains.



**Figure 5.5:** The figure represents the data flow in a running ZoneGuard.

In the figure, there are two arrows, an upstream arrow pointing from D1 to D2 and a downstream arrow pointing in the opposite direction. An arrow consists of one horizontal stacked bar graph. The packages transferred upstream to the filter consist of 80% UDP/TCP packages and 20% HTTP packages. At the filter, 60% of the upstream packages have been blocked by the filter. This is also represented by the width of the arrow, which is 60% smaller. The packages sent upstream from the filter to D2 consist of 50% UDP/TCP packages

and 50% HTTP packages. None of the packages sent downstream have been blocked. Therefore, the arrow's width was the same before and after the filter. Both interfaces, D1 and D2, are up and running which is represented by the green colour and the text "RUNNING". The security domains are separated by a dotted line and colour. How different states of a CDS device would change the visualization can be seen in Appendix E.

## 5.4 Stakeholder Validation

### 5.4.1 Method

Just like for the first prototype the stakeholder validation was performed with two stakeholders for half an hour. The prototype was presented to the stakeholders by showing the two prototypes and explaining the functionality and our design thoughts. The stakeholder could then express their thoughts on the design.

### 5.4.2 Result

The result of the stakeholder validation was that they liked the idea of placing graphs inside a ZoneGuard figure to potentially help users with their conceptual model. However, they thought that another iteration of this figure should be performed to improve the visualization based on the feedback from the upcoming user testing.

## 5.5 Testing

### 5.5.1 Method

The next step in the prototyping process was to test the prototypes. The test session was performed at Advenica with the same limitations as in the first iteration with no real end-users available. The tests were performed with four employees with technical backgrounds from Advenica. Three of them tested the first prototype. One of the test participants was from Advenica's IT and Security Department, the nearest to an end-user of the product.

Before the test session, an introduction letter describing the test session was sent to the participants, similar to the introduction letter for the first



iteration. In the letter and at the start of the test session the participant was informed that their comments would be used in this thesis. The introduction letter, tasks and the template for the "I like, I wish, What if" method can be found in Appendix F.

The test session consisted of two parts. In the first part of the session, the user was asked to perform a number of tasks using the think-aloud protocol, described in chapter 2.4.2. The prototypes that were done in Moqups and Google's Draw.io were printed out so that the testing was performed on paper. The interactions were made in accordance with paper-prototyping[30] where a human acts as the computer, giving feedback and changes the view for the user. In the second part, the user was introduced to the "I like, I wish, What if" method, described in chapter 2.4.4, and the different pages were analysed together with the user. The test session took between 15-40 min depending on the number of comments from the user.

### 5.5.2 Result

The result of the test session can be found in Appendix G. The main user feedback that was gathered was the following:

**Homepage:** An overall positive view of the homepage. One user worried of the ability to be able to delete warning messages and wanted that feature removed.

**Log page:** Three users liked the filter functionality on the system log page. One user had a problem to discover how to change the time span.

**Device Monitoring page:** The main feature request was that the users wanted labels and legends on the graphs. The users also wanted to be able to hover over to get more information and to be able to connect to a third-party program for more advanced features. One user had a problem with the conceptual model resulted in confusion over the names D1, D2, upstream and downstream.

**ZoneGuard figure:** All users had difficulties with their conceptual model of the horizontal stacked bar graph. They did not understand that the data flowed through the filter. They also had a problem understanding that the different height of the bar graphs represented the number of packages.

## 5.6 Analysis and Discussion

### 5.6.1 Input to Next Iteration

The input to the next prototyping iteration was the feedback from the users and stakeholders. The goal of the upcoming development was the following:

- Prototype more ideas on how the data from a CDS can be visualized.

The homepage and log page had a few usability problems but were otherwise liked. Therefore, those pages are ready for the functional iteration. The usability problems and the other feedback related to the CDS data visualization figures will be input to the next iteration.

### 5.6.2 Analysis and Discussion of The Prototyping Process

This prototype iteration was shorter than the first iteration. Unlike the previous iteration, we had gained more knowledge about what the stakeholders and users wanted from the application. This resulted in a much more time-efficient prototyping where fewer unnecessary functionalities were prototyped. The prototyping process was more time-efficient than the previous iteration because no user interaction was implemented in Moqups. The test session gave us more valuable feedback by using the "I Like, I Wish, What if" method. The method produced both positive and constructive feedback compare to the previous iteration which did not necessarily give us positive feedback. The positive feedback gave us information about what the users liked and therefore made it easier for us to decide what should be kept and what to improve.

The prototype had several usability problems. On the homepage the affordance, remove a warning message, should be removed[16]. To improve the discoverability of the time span on the pages the colour of the time span could be changed to a more dramatic colour as a signifier[16]. To improve the UX, the representative visualization technique, labelling, needs to be added to the graphs on the device monitoring page[6]. The names on the device monitoring page, D1, D2, upstream and downstream, need to be explained to improve the conception model[16]. This can, for example, be achieved by only using arrows instead of writing upstream and downstream. The arrows in the ZoneGuard figure can then be mapped to the arrows in the cards on the Device Monitoring page[16]. The users had problems understanding the ZoneGuard figure. To improve the conceptual model other representative visualization techniques regarding the shape and size of the visualization should be explored[1, 17].

The RS was not updated in this iteration and was not updated later in this project. This is because it was time-consuming to update and the stakeholders

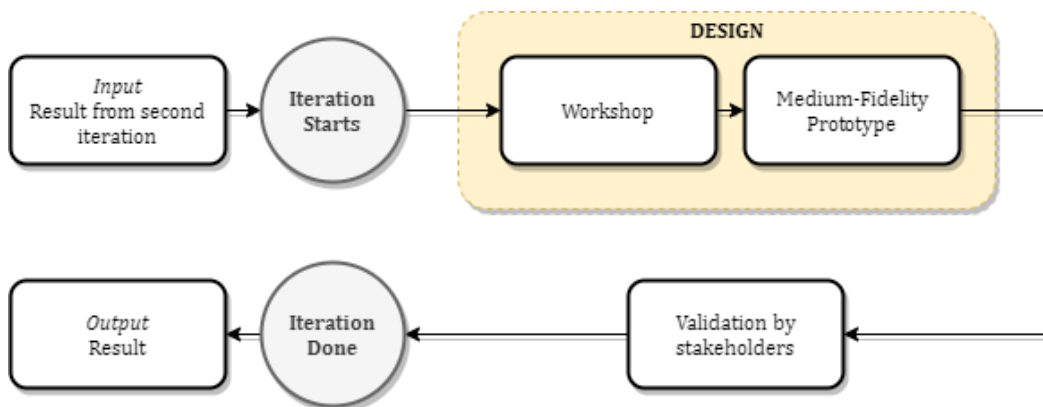
felt it was not necessary to have a RS. In the Stakeholder requirement phase, it was practical for us to gather all the stakeholder expectations in a structural way. However, it is no longer necessary to have a RS as we have user feedback from test sessions (documented in Appendix G) and have continuous stakeholder validations.

# Chapter 6

## Phase 4 - Refining the Visualization

### 6.1 The Prototyping Process

An overview of the process for the forth phase in this thesis can be seen in Figure 6.1.



**Figure 6.1:** The prototyping process for the third prototype iteration.

1. **Plan:** The plan for the third prototyping iteration was to develop a medium-fidelity prototype of the device monitoring page with a focus on developing and integrating the ZoneGuard visualization figure.
2. **Specification:** The methods used for this iteration was design workshop and the tools Draw.io and Moqups to produce the prototype.
3. **Design:** A medium-fidelity prototype of the device monitoring page was produced.
4. **Result:** The prototype was only validated by the stakeholders as the workshop provided both user and stakeholder feedback.

## 6.2 Design Workshop

### 6.2.1 Method

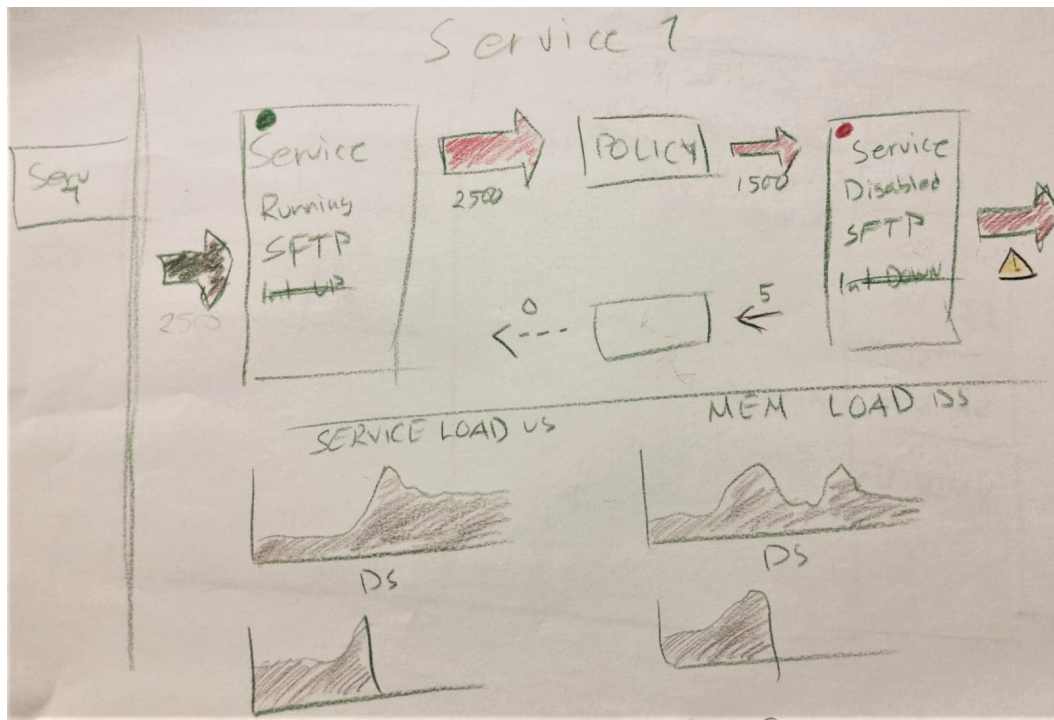
The goal of the design workshop was to generate ideas on how the data from a ZoneGuard could be visualized. The workshop was two hours long and consisted of six participants. The six participants all worked at Advenica and had various relations to the ZoneGuard, from tester to product owner. The workshop consisted of the following four parts:

1. **Prototype:** The workshop began with an introduction about the workshop and to the theme of this thesis, information visualization. Thereafter two sketch sessions were performed, first individually and then in pairs.
2. **Present and critique:** All pairs presented their ideas to the group and gave feedback and discussed the sketches made.
3. **Converge:** The group then decided on what parts of each pair's sketches that they wanted to combine into one prototype.
4. **Prioritise:** The last part consisted of presenting the prototype from the second iteration to the group and having them prioritise changes to be made in the next prototype iteration.

A more detail description of the workshop can be found in Appendix H.

### 6.2.2 Result

The sketch in Figure 6.2 was made at the workshop by two participants. The two participants explored how a service can be visualized. The arrows represent the data flow direction and the height of an arrow represents the amount of data transferred. At the bottom of the sketch, area charts are used to represent loads over time.



**Figure 6.2:** One of the sketches made at the workshop.

The conclusion of what should be changed in the prototype from the workshop was the following:

- The ZoneGuard consists of three zones. These zones have their own CPU and RAM usage data. This should be visualized and showed in the prototype.
- A way to identify if a service is up or down.
- An average graph and an over time graph for both the CPU and the RAM of the three zones should be added.
- Instead of visualizing the percentage of the different services in the ZoneGuard figure with a vertical line, a horizontal line should be used.

## 6.3 Medium-Fidelity Prototype

### 6.3.1 Method

The prototype was made using Draw.io and Moqups like the previous prototype iterations. The focus was on visualizing the data received from a ZoneGuard. The prototype was developed without Moqups's interactions. The representative visualization techniques that were used in this prototype were shape, size, colour and depth.

### 6.3.2 Result

The data visualization on the device monitoring page can be seen in Figure 6.3. The page consists of the following:

- **Warning Messages:** The warning message card is moved to the top left corner and is therefore easy for the user to discover.
- **Temperature and Uptime:** Is moved to its own card under the warning messages.
- **ZoneGuard Figure:** A more advanced ZoneGuard figure is integrated into the monitoring page. The figure visualizes the data transferred and received with different services. In this example, the services used are UDP/TCP and HTTP. The dotted line represents that no data have been sent the last 24 hours. The shape used for the visualization is a vertical stacked bar graph instead of using a horizontal stacked bar graph. The ZoneGuard is divided into three zones. The colour on the zone text is used in the RAM and CPU graphs.
- **Current Service Status:** The current status for individual services.
- **CPU Average:** Represents the average of the CPUs for the three zones.
- **CPU over time:** Represents the CPU values over time of the three zones.
- **RAM Average:** Represents the average of the RAM for the three zones.
- **RAM over time:** Represents the RAM values over time of the three zones.
- **Blocked by Filter over time:** Has now its own card and contains information of blocked data from both upstream and downstream in the same graph.

- **Transferred and Received over time:** Has now its own card and contains information of transferred and received data from both upstream and downstream in the same graph.

## 6.4 Stakeholder Validation

The stakeholder validation was made by having two stakeholders explore the prototype and lasted 20 minutes. The stakeholders were satisfied with the prototype and thought that a functional prototype could now be implemented.



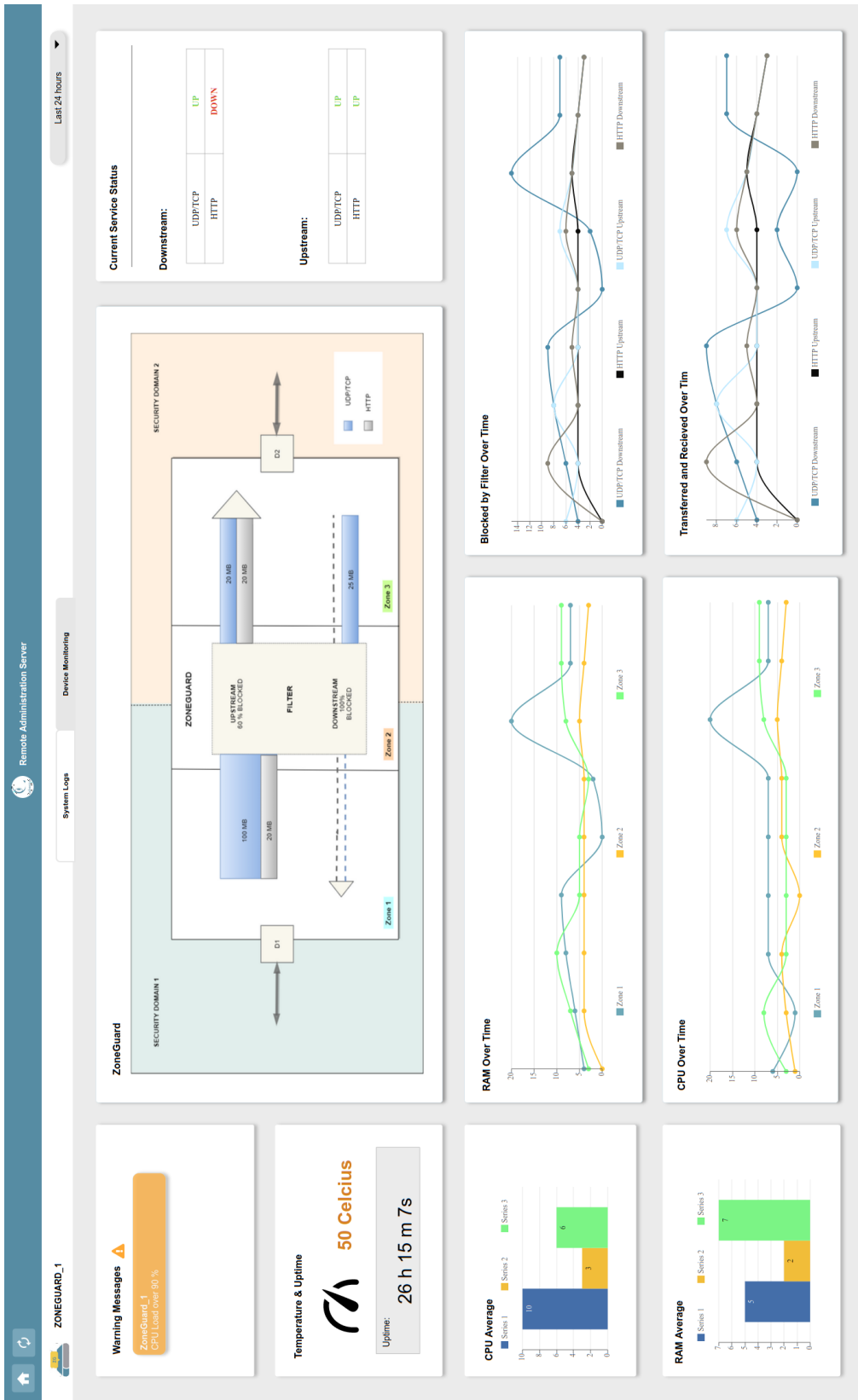


Figure 6.3: The third prototype's version of the device monitoring page.

## 6.5 Analysis and Discussion

### 6.5.1 Input to Next Iteration

The Figure 6.3 will be implemented in the functional prototype.

### 6.5.2 Analysis and Discussion of The Prototyping Process

A workshop was needed in this iteration as we had difficulties coming up with ideas that could improve the visualization. The main advantage of the workshop was that the participants could have a discussion with each other of what kind of functionalities that were important. This was impossible with the think-aloud protocol because it was performed with one user at a time. Therefore, from the experience with the workshop, we should have had a workshop earlier in the project's timeline.

We also did not feel the need for a test session during this iteration as with previous iterations. This was because the workshop resulted in both ideas and user feedback on how to change the previous prototype.

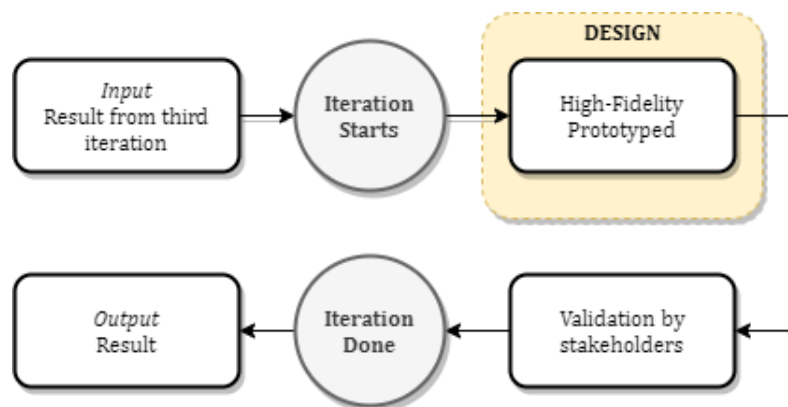
In the ZoneGuard figure, the shape of the visualization, horizontal stacked bar graph, used in the previous iteration was replaced with a vertical stacked bar graph[1]. This will help the user with their conceptual model of the visualization[16]. By using a vertical stacked bar graph the user gets a better perception that the data flows from one security domain to another through the filter. Unique colours are used to represent zones and services to clarify that there is a mapping between the graphs[16, 17]. For example, the colour orange is used to represent Zone 2 in both the ZoneGuard figure and CPU average.

# Chapter 7

## Phase 5 - An Interactive Visualization

### 7.1 The Prototyping Process

An overview of the process of the final phase can be seen in Figure 7.1.



**Figure 7.1:** The prototyping process for the functional prototype iteration.

1. **Plan:** For the last iteration the plan was to develop a high-fidelity prototype of the device monitoring page with a focus on user interactions with the data.
2. **Specification:** No elicitation method was going to be used before the implementation.
3. **Design:** In the Design phase, IV methods were used to create interactive graphs.
4. **Result:** The prototype was validated by the stakeholders.

## 7.2 Implementation

### 7.2.1 Method

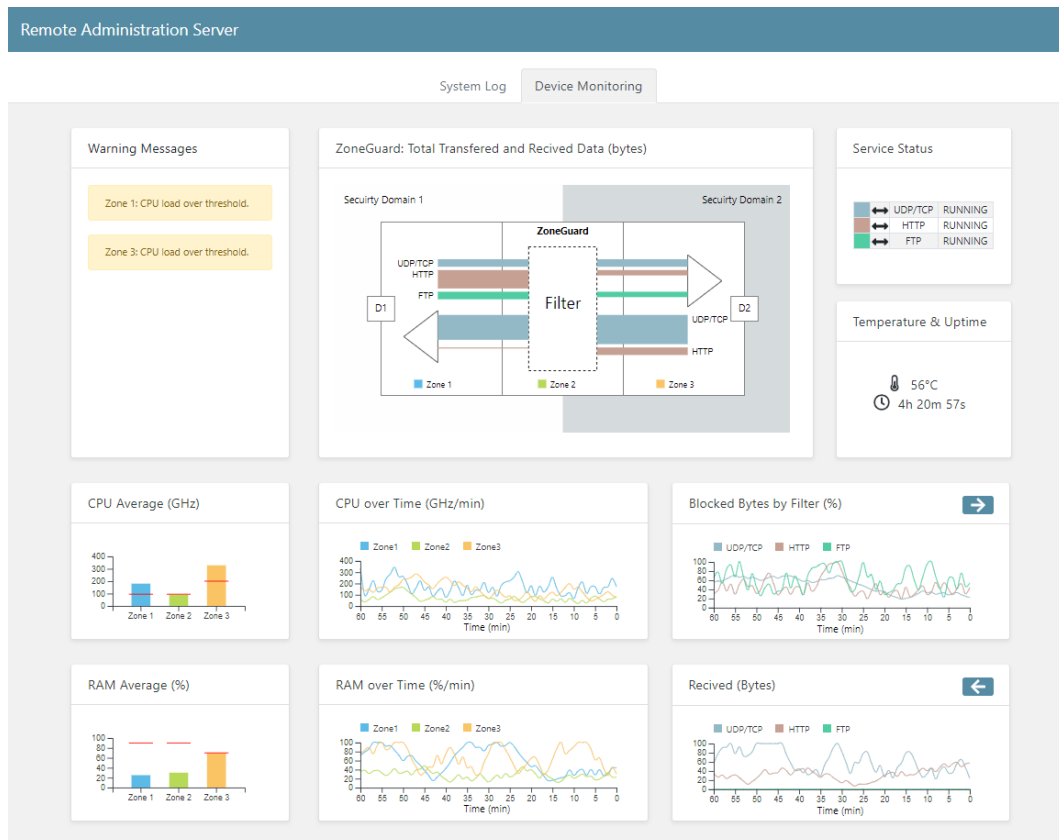
The representative visualization techniques that were explored in this prototype were colour and labelling and the interactive technique that was used was hovering over an element. The web application was developed using the following JavaScript tools:

- **React.js** is a JavaScript library for building UI[31].
- **D3.js** is a JavaScript library for building interactive customised graphs. All graphs in the web application was created using d3.js[32].
- **React-Bootstrap** is a front-end framework with ready-to-use react components[33].
- **Node.js** is a JavaScript runtime-environment[34].
- **Express.js** is a Node.js application framework that was used in the application to set up the server[35].

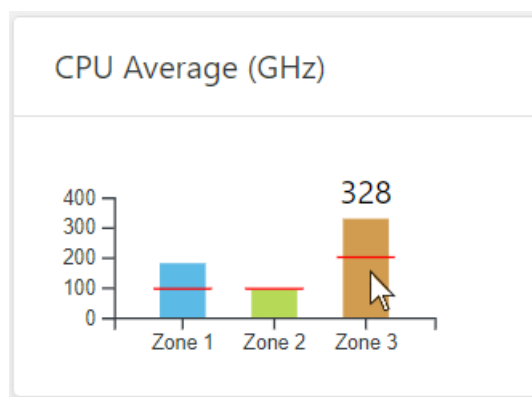
### 7.2.2 Result

A static screenshot of the prototype can be found in Figure 7.2, and more detailed figures of the individual cards on the application can be found in Appendix I. The four cards on the bottom right of the web page, *CPU over Time*, *RAM over Time*, *Blocked Bytes by Filter* and *Received* updated every time new data are sent to the system. The *ZoneGuard* figure, the *CPU Average* and the *RAM average* represent the average of the data since the *ZoneGuard* was turned on (see the uptime in the *Temperature Uptime* card in Figure 7.2).

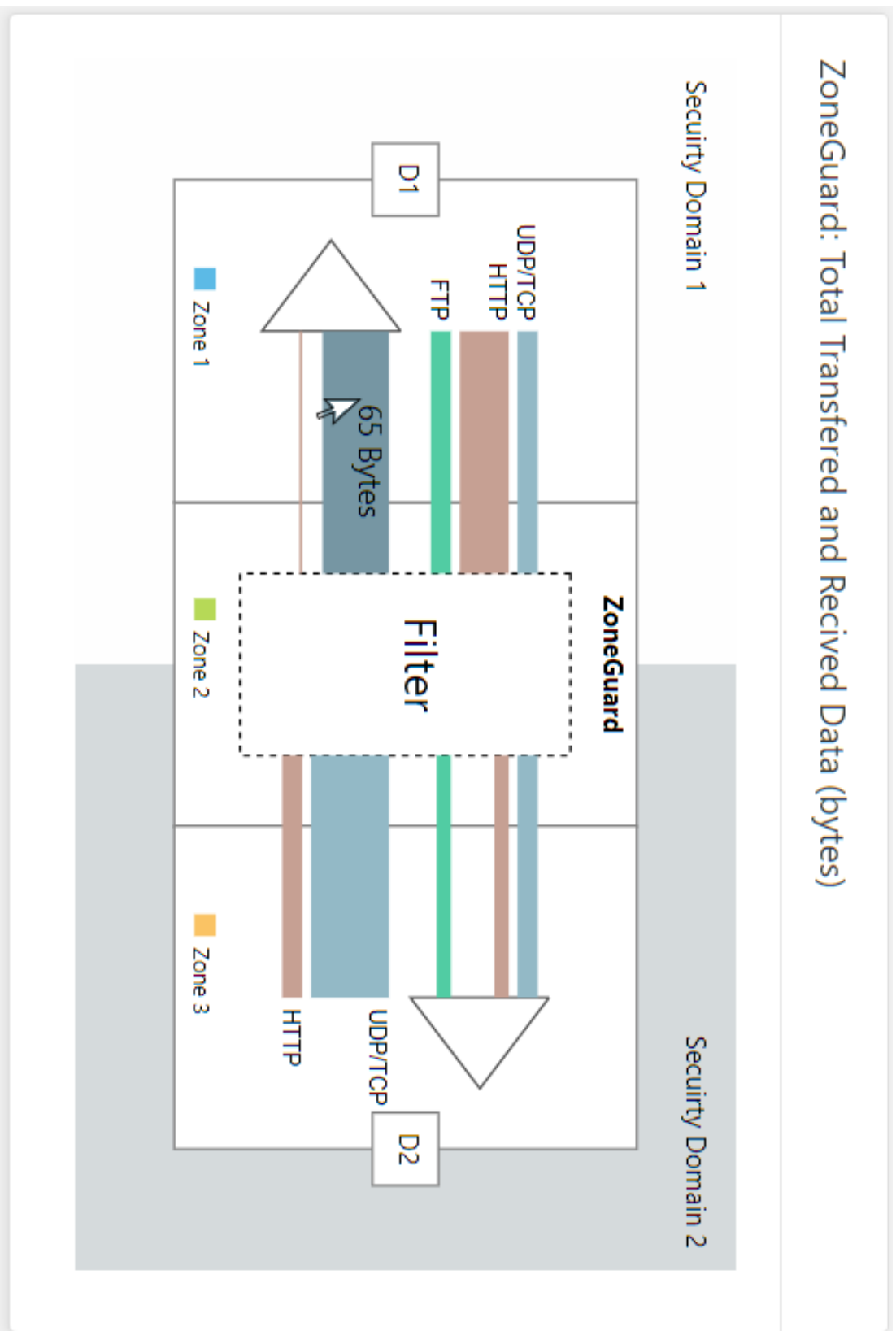
Interactions were added to the *ZoneGuard* figure and the average graphs. In the average graphs, Figure 7.3, a user can hover over a bar to see the data value. In the *ZoneGuard* figure, Figure 7.4, a user can hover over each selection of the bars to see the data values. The colour scheme used in the application was a qualitative colour scheme to more easily be able to distinguish each colour. Each colour represents a specific zone or service.



**Figure 7.2:** A screenshot of the functional prototype.



**Figure 7.3:** A card with the average of the CPU for the different zones of the ZoneGuard over the whole time the device has been running. Here the user can get the value by hovering the mouse over the individual bars in the graph. The red line represents the CPU threshold for each zone.



**Figure 7.4:** The ZoneGuard card with an overview of the total amount of transferred and received data that can be seen when hovering the mouse over the individual sidebars.

## 7.3 Stakeholder Validation

Two stakeholders explored the application and the design and functionality were discussed and lasted 30 minutes. The stakeholders were satisfied with the application. They thought that the qualitative colour schemes used throughout the graphs resulted in clear graphs. They thought the labels were correct and easy to understand.

## 7.4 Analysis and Discussion

We did not have knowledge beforehand about the different JavaScript libraries used to implement the application. Therefore, two weeks were invested in this phase just to explore the libraries to see what could work in this project. React and React-Bootstrap were relatively easy to learn and use. D3.js had a high learning curve compared to other visualization libraries. However, with D3.js it is possible to create customised graphs and interactions. With Node.js and Express.js it was effortless to set up servers.

In conclusion, the libraries were easy to use and suited the functionality in the application.

No user testing was performed in this iteration, therefore, we can not exactly estimate how the UX is in the application. If we would have had more time, user tests would have been performed. However, we did receive feedback from the stakeholders.

To improve the UX, more interactive visualization techniques could be explored. For example, distortion could be added to the line charts[1]. With distortion, the user would be able to select a specific time interval that they would like to analyse further. The interactive visualization technique, customisation, could be used in the graphs visualizing the services[1]. For example, to be able to switch the unit between bytes and data packages in the visualization.

# Chapter 8

## Discussion

In this chapter we discuss some factors that affected the result of this thesis and what could be done in future work to the application.

### 8.1 User-Centered Design

The most important in UCD is to have access to the users. In this project, the lack of access to end-users was a major problem when developing an application using UCD. There was no way to evaluate how the end-users thought about the application. The user tests and the gathering of user needs were performed on employees at Advenica. From Advenica's point of view, the application did satisfy their needs and expectations and in that meaning a UCD approach was successful. However, it is first proven that a UCD approach was successful when the real end-users have tested it and they think the UX is good.

### 8.2 Design Techniques

In this project, several design techniques were used. The design workshop was the most valuable technique for the project. The advantage of the design workshop was that we gathered a group of users that combined their ideas and knowledge. If we could start the project all over again, we would have held the design workshop much earlier in the timeline. We also used techniques that did not result in any progression in the project. For example, the blank-page techniques did not result in valuable sketches because they needed more time and knowledge from the participants. The participants also felt uncomfortable



sketching on their own and on-demand.

## 8.3 Information Visualizations

Tufte's criteria for a good IV are graphical excellence and graphical integrity, see section 2.2.1. The visualization created in the project is designed to satisfy Tufte's criteria.

**Graphical Excellence:** The visualization is created to satisfy the users' needs to the extent that we could. The stakeholder thought the visualization was easy to use when analysing the data. However, a testing of the real end-users needs to be performed to evaluate if the visualization satisfies this criterion.

**Graphical Integrity:** To satisfy graphical integrity, the data have to be proportional to the numerical quantities. In the application, all graphs fulfil this criterion. For example, in the ZoneGuard figure, the height in a vertical stacked bar graph is proportional to the number of bytes.

The visualization must also be clear and have enough details to represent the information. The stakeholders thought the graphs were clear and detailed enough to fulfil this criterion. For example, they thought it was easy to distinguish the different services and zones throughout the graphs when using a qualitative colour scheme. In the final visualization labels and interactions were also added to make it more clear what the graph represents. For example, hovering over a bar graph gives the data value.

The visualization of a variable in n-dimensions must also be represented in no more than n-dimensions. This is fulfilled in all graphs in the application. For example, the CPU over time graph has the two dimensions time and GHz and is represented in a 2D-graph. The data sent to the system have a timestamp and a value.

## 8.4 Constraints

During this thesis, some limitations were established to limit the scope. The following list lists the limitations and other factors from outside of this thesis that limited the work.

- Because of the time limitation of this thesis, this thesis primarily focuses on the ZoneGuard. However, when designing the GUI we kept in mind that it should also be applicable for the Data Diode.

- Because of the COVID-19 outbreak, a test phase in the final iteration of the functional prototype could not be executed properly. Instead, only a stakeholder validation was performed.
- We did not have access to the end-users of the application. How this affected this thesis is discussed in section 8.1.

## 8.5 Future Work

In the application, only the device monitoring page was implemented. So an implementation of the rest of the medium-fidelity prototype such as the homepage and the system log page can be made in the future.

For the monitoring page, the following can be explored in future work:

- Explore how the visualization would work for a Data Diode.
- Do user testing on real end-users and analyse if users can quickly find security issues.
- Explore if the interactive visualization technique, distortion of time periods, in the CPU, RAM, Blocked data and Received data will improve the UX. This would let a user zoom in on a specific time period.
- Let the user customise the units on graphs, for example, change bytes to the number of data packages.
- Explore if an interaction with one graph that affects another graph can improve a user's understanding. For example, if a user selects the bar of Zone 1 in the CPU average graph, then Zone 1 in the ZoneGuard figure is highlighted.
- Connect the web application to a real running ZoneGuard to receive real data and do user testing on real data.

# Chapter 9

## Conclusion

In this chapter, we return to the research questions in 1.3 and answer them with the results made during this thesis.

**RQ-1: How can network flow and information between two security domains be visually represented?**

The final prototype in Chapter 7 demonstrates one way of how the network flow can be visually represented.

**RQ-2 How do users prefer to interact with multiple CDS devices in a network?**

During the first and second prototype iterations, we found that users preferred to interact with multiple CDS devices in a list where security risks are notified to the user. From the list, the user can select a CDS device to obtain more information about that specific device. The users did not prefer the idea of presenting network flow from multiple devices on a single page.

**RQ-3: What representative visualization techniques can be used to enhance the users' understanding of the network flow between two security domains and error search in a CDS device?**

The following representative visualization techniques were used in the visualizations:

- **Shape:** Temporal line graphs were used to represent CPU, RAM, Blocked data and Received data over time. Bar graphs were used to represent CPU and RAM average. To represent the data flow between the two security domains four vertical stacked bar graphs were used in the Zone-Guard figure.

- **Size:** The height of the bar graphs and stacked bar graphs depend on the data value.
- **Colour:** A qualitative colour scheme was used to be able to clearly distinguish data in all graphs. The same colour was used to represent the same object in all graphs in the application. For example, a specific service is represented by the same colour in both the ZoneGuard figure and in the Blocked bytes over time graph.
- **Depth:** The visualization is created in 2D.
- **Labelling:** Labels and legends were used in the graphs to enhance the user's understanding.

**RQ-4: What interactive visualization techniques can be used to enhance the users' understanding of the network flow between two security domains and error search in a CDS device?**

The following interactive visualization techniques were used in the visualizations:

- **Filter:** On the system log page (this was not implemented in the functional prototype) the user is able to filter the log on sources and severity level.
- **Hovering over:** In the web application the user can hover over the bar graphs and the stacked bar graphs to see the exact data value.

# Bibliography

- [1] R. Mazza, *Introduction to Information Visualization*. London: Springer-Verlog, 2009.
- [2] E. R. Tufte, *The visual display of quantitative information*. Graphics Press, 2001.
- [3] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.
- [4] C. Ware, *Information visualization. Perception for design*. Interactive technologies, Elsevier/MK, 2013.
- [5] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko, “Toward a deeper understanding of the role of interaction in information visualization.,” *IEEE Transactions on Visualization and Computer Graphics, Visualization and Computer Graphics, IEEE Transactions on, IEEE Trans. Visual. Comput. Graphics*, no. 6, p. 1224, 2007.
- [6] R. Fernandez and N. Fetais, “Survey of information visualization techniques for enhancing visual analysis.,” *2017 International Conference on Computer and Applications (ICCA), Computer and Applications (ICCA), 2017 International Conference on*, pp. 360 – 363, 2017.
- [7] J. Xiaoxiao and G. Ian, “Research on information visualization based on users’ demands.,” *2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Computer Technology, Electronics and Communication (ICCTEC), 2017 International Conference on*, p. 103, 2017.
- [8] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, “Seven guiding scenarios for information visualization evaluation,” 2011.
- [9] Advenica, “Zoneguard.” Available at: <https://www.advenica.com/sv/cds/zoneguard>, Accessed: 2020-04-28.

- [10] Advenica, “Data Diode.” Available at: <https://www.advenica.com/sv/cds/data-diodes>, Accessed: 2020-04-28.
- [11] H. Sharp, J. Preece, and Y. Rogers, *Interaction design : beyond human-computer interaction*. Wiley, 2015.
- [12] Interaction Design Foundation, “User-centered design,” Available at: <https://www.interaction-design.org/literature/topics/user-centered-design>. Accessed: 2019-11-26.
- [13] ISO, “Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems,” standard, International Organization for Standardization, 2019.
- [14] J. Nielsen and D. Norman, “The definition of user experience (ux).” Available at: <https://www.nngroup.com/articles/definition-user-experience/>, Accessed: 2019-11-07.
- [15] B. Martin and B. M. Hanington, *Universal methods of design : 100 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Publishers, 2012.
- [16] D. Norman, *The design of everyday things: Revised and expanded edition*. Basic books, 2013.
- [17] C. O. Wilke, *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. O’Reilly Media, 2019.
- [18] B. Shneiderman, “The eyes have it: a task by data type taxonomy for information visualizations.,” *Proceedings 1996 IEEE Symposium on Visual Languages, Visual Languages, 1996. Proceedings., IEEE Symposium on*, p. 336, 1996.
- [19] S. Lauesen, *Software requirements : styles and techniques*. Addison-Wesley, 2002.
- [20] S. Hove and B. Anda, “Experiences from conducting semi-structured interviews in empirical software engineering research.,” *11th IEEE International Software Metrics Symposium (METRICS’05), Software Metrics, 2005. 11th IEEE International Symposium, Software Metrics, 2005*.
- [21] R. Dam and T. Siang, “Affinity diagrams – learn how to cluster and bundle ideas and facts,” *The Interaction Design Foundation*, Nov 2019.

- [22] K. Kaplan, “Facilitating an effective design studio workshop,” Jul 2017. Available at: <https://www.nngroup.com/articles/facilitating-design-studio-workshop/>, Accessed: 2020-03-05.
- [23] J. Rudd, K. Stern, and S. Isensee, “Low vs. high-fidelity prototyping debate.,” *Interactions: New Visions of Human-Computer Interaction*, vol. 3, no. 1, p. 76, 1996.
- [24] J. Arnowitz, M. Arent, and N. Berger, *Effective prototyping for software makers*. Morgan Kaufmann, 2007.
- [25] Z. Shakeri Hossein Abad, S. Moazzam, C. Lo, T. Lan, E. Frroku, and H. Kim, “Loud and interactive paper prototyping in requirements elicitation: What is it good for?.,” *2018 IEEE 7th International Workshop on Empirical Requirements Engineering (EmpiRE), Empirical Requirements Engineering (EmpiRE), 2018 IEEE 7th International Workshop on, EMPIRE*, pp. 16 – 23, 2018.
- [26] J. Cowan, “The potential of cognitive think-aloud protocols for educational action-research.,” *Active Learning in Higher Education*, vol. 20, no. 3, pp. 219 – 232, 2019.
- [27] B. Still and J. Morris, “The blank-page technique: Reinvigorating paper prototyping in usability testing.,” *IEEE Transactions on Professional Communication, Professional Communication, IEEE Transactions on, IEEE Trans. Profess. Commun*, vol. 53, no. 2, pp. 144 – 157, 2010.
- [28] Interaction Design Foundation, “Test your prototypes: How to gather feedback and maximise learning.” Available at: <https://www.interaction-design.org/literature/article/test-your-prototypes-how-to-gather-feedback-and-maximise-learning>, Accessed: 2020-02-06.
- [29] Moqups. <https://moqups.com/>, Accessed: 2020-02-05.
- [30] C. Snyder, *Paper Prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann series in interactive technologies, Morgan Kaufmann, 2003.
- [31] Facebook Open Source, “React – a javascript library for building user interfaces.” Available at: <https://reactjs.org/>, Accessed: 2020-04-28.
- [32] M. Bostock, “D3 - data-driven documents.” Available at: <https://d3js.org/>, Accessed: 2020-04-28.
- [33] “React bootstrap.” Available at: <https://react-bootstrap.github.io/>, Accessed: 2020-04-28.

## BIBLIOGRAPHY

---

- [34] OpenJS Foundation, “Node.js.” Available at: <https://nodejs.org/en/>, Accessed: 2020-04-28.
- [35] StrongLoop, IBM, and other expressjs.com contributors, “Express4.17.1 fast, unopinionated, minimalist web framework for node.js.” Available at: <https://expressjs.com/>, Accessed: 2020-04-28.



Appendix A

Requirement Specification

**Software Requirements Specification**  
for  
Advenica's Log Visualisation Application

Lisa Claesson and Malin Tjärnemo

November 20, 2019

## Chapter 1

# Requirements for Advenica's Log Visualisation Application

### 1.1 Customer

Advenica AB

### 1.2 Background

Advenica is a company that works with cyber-security and provide software, hardware and services for secure information exchange. Advenica was founded in 1993 in Lund and is specialised in cross-domain solutions (CDS). Advenica uses the concept of whitelisting in their products, meaning only explicitly allowed information may pass through. A domain (also called segment) in a network can be separated and secured from the rest of the network with a CDS device. Advenica has three CDS products. The first one, is the ZoneGuard which works by allowing only whitelisted information to pass between connected networks. The second product is the Data Diodes which is a one-way CDS. The key difference between the ZoneGuard and the Data Diodes is that Data Diodes makes two directional information exchange impossible. The last CDS product is SecuriRAM which is a self-erasing USB drive that works as a manual CDS. This requirements specification uses only the first two products mentioned.

ZoneGuard and Data Diod provide log control and audit trails and can log any type of information. To do configuration on both the ZoneGuard and the Data Diod Advenica provides a platform where services and licences can be modified. This service is only available for one device at a time today and this specification is for a service handling one or more devices at the same time.

### 1.3 Purpose

The log from a ZoneGuard or a Data Diod contains a lot of information about the data exchange history. Today, this information is listed in a text format. Advenica has found that there is a need to visualise this information to enhance the user experience. The visualisation will make it easier for the user to understand the information and improve decision-making.

They have also found a need to handle multiple CDS devices at the same time. For example, be able to update all CDS devices from one view.

A web application shall be developed and have a visualisation of the information in the log file. The user shall also be able to handle multiple CDS devices in the application. The primary focus shall be to improve the

user experience.

## 1.4 Glossary

Application	The web application that shall be delivered.
CDS	Cross Domain Solution
CRL	Certificate revocation list
Device	Can be either a ZoneGuard or a Data Diod.
Filter	A message format definition used in the communication between filters and services
Configuration	Consists of services and policies as well as device and environment parameters.
Policy	Used to define schemas and filters. contains a schema for incoming messages, a filter and a schema for outgoing messages.
Receiver	A network component that receives the data from a CDS device.
Schema	A definition of allowed contents of traffic data that can be transported through the system.
Sender	A network component that sends the data to a CDS device.
Service	Enables communication between network components and CDS device.
System	The logic behind the application.

## 1.5 Functional Requirements

- (F-1) **The user shall be able to update the firmware in one or more devices.**
- (F-2) **The user shall be able to upload a licence to one or more devices.**
- (F-3) **The user shall be able to upload a configuration to one or more devices.**
- (F-4) **The user shall be able to change the hostname of a device.**
- (F-5) **The user shall be able to change the IP address and netmask on a device.**
- (F-6) **The user shall be able to start a stopped device.**
- (F-7) **The user shall be able to stop a running device.**
- (F-8) **The user shall be able to reboot a device.**

## 1.6 Quality Requirements

	Critical	Important	As usual	Unimportant	Ignore
Response time			X		
Data volume					X
Usability		X			
Maintainability				X	
Reliability			X		
Fault tolerance			X		
Safety and security			X		
Reusability				X	

### 1.6.1 Usability

- (Q-1) **During the design of the application, at least three iterations have to be made and after each iteration a usability test has to be performed and the most important defects corrected.**

## 1.7 Operating Environment

- (O-1) **The system shall work in the following browsers:**
  - (a) Chrome (version 60.0.3112.113)
  - (b) Firefox (version 69.0.1)
  - (c) Edge (version 44.18362.449.0)
- (O-2) **The system shall be developed using the React framework.**
- (O-3) **The system shall use Node.js as runtime environment.**

## 1.8 General Requirements

- (G-1) **The language of the application shall be in English.**

## 1.9 Visual Requirements

The application shall visualise the following:

- (V-1) **The CRL status of a device.**
  - (V-2) **The license status on a device.**
  - (V-3) **The configuration meta data which gives the following information:**
    - (a) ID
    - (b) Name
    - (c) Version
    - (d) Validity period
    - (e) Description
    - (f) Intended for firmware
  - (V-4) **The IP address of a device.**
  - (V-5) **The hardware ID of a device.**
  - (V-6) **The hostname of a device.**
  - (V-7) **The type of services that is sent through a device.**
  - (V-8) **The connection between a sender and receiver.**
  - (V-8) **The IP address of a sender.**
  - (V-8) **The IP address of a receiver.**
  - (V-9) **Number of transfers.**
  - (V-10) **Size of the transfers.**
  - (V-11) **Successful transfers.**
  - (V-12) **Failed transfers.**
-

- (V-13) Reason for failing of a failed transfer.
- (V-14) Up to a maximum of twenty devices.
- (V-15) The type of a device.
- (V-16) The status of a device, which can be the following:
  - (a) Unconfigured
  - (b) Running
  - (c) Stopped
  - (d) Secure
- (V-18) The log id number.
- (V-19) The severity level of an event. The different levels are:
  - (a) 1 - Error
  - (b) 2 - Warning
  - (c) 3 - Info
  - (d) 4 - Debug
- (V-20) The Log message text.

## 1.10 Data Requirements

- (DA-1) The system shall handle XML files from input.
- (DA-2) The system shall handle CVS files from input.
- (DA-2) The system shall receive information from a device in a SNMPv3 protocol.

## 1.11 Delivery Requirements

- (DE-1) The system shall be delivered to the customer 2020-03-20.

## 1.12 Prioritised requirements

- (P-1) The customer shall be able to prioritise the functional and visual requirements.
- (P-2) A requirement with higher priority shall be implemented before a requirement with lower priority.

# Appendix B

## Cancelled Requirements

The following requirements were cancelled after the stakeholder validation in Chapter 4:

- (F-1) **The user shall be able to update the firmware in one or more devices.**
- (F-2) **The user shall be able to upload a licence to one or more devices.**
- (F-3) **The user shall be able to upload a configuration to one or more devices.**
- (F-4) **The user shall be able to change the hostname of a device.**
- (F-5) **The user shall be able to change the IP address and netmask on a device.**
- (V-1) **The CRL status of a device.**
- (V-2) **The license status on a device.**
- (V-3) **The configuration meta data(...).**
- (V-5) **The hardware ID of a device.**

# Appendix C

## Prototype V.1 Test

### C.1 Introduction Letter

#### **Welcome to our test session!**

Today, you will help us test a paper prototype of Advenica's future Remote Administration Server Application. The interface to be tested is a web application that handles multiple CDS-devices, it contains both the administration side and the log management side of a CDS device. The end-user can, for example, be a configurator or an administrator working at a company that owns one or multiple CDS products from Advenica. The main goal of the session is to get feedback on the interface from a user's perspective.

The prototype is meant to be unfinished, for example, only a limitation of the functionality is implemented. Therefore, the prototype is open for changes, and ideas are welcomed. Remember, we are testing the interface and not you so any problems encountered are valuable feedback.

We would like you to perform 13 tasks listed in the table below. When performing a task, we encourage you to express any difficulties, emotions, and thoughts. Afterwards, we will have a discussion of what troubles you may have encountered, what went well and anything else that may come up about the prototype.

**Note that your feedback on the prototype will be used in our master thesis report.**

The test session will be performed in the following order:

1. The facilitator will introduce the starting state of your test.



2. The facilitator will present a scenario with a corresponding task. While performing the task, you are encouraged to express any difficulties, emotions, and thoughts.
3. After all tasks are completed an open discussion about your experience will be held. Here we would also love to hear if you have any new ideas.
4. After the questions, you are asked to draw what you think should be on the visualization page. This is further explained in the test session.
5. Now you are finished.

**Thank you for your contribution!**

## C.2 Scenarios & Tasks

	Scenarios	Tasks
1	You heard that there are Data Diodes that are in a stopped state in the network. You would like to check this and count how many Data Diodes that are stopped.	Locate and count the stopped Data Diodes.
2	It was some time ago a configuration was uploaded to ZoneGuard_1, you would like to find out for how long the configuration is valid before you need to upload a new configuration file.	Find out for how long the configuration for ZoneGuard_1 is still valid.
3	You would like to change the IP address and hostname for ZoneGuard_1.	Locate where the IP address and hostname can be changed. Then, describe what you would do to change them.

4	You would like to stop ZoneGuard_1.	Find where you change the running state and describe how you would change the state of ZoneGuard_1 to stop.
5	You are curious about how much data is successfully being transferred over ZoneGuard_1.	Locate where this data is.
6	You would like to see what services there are on ZoneGuard_1.	Find out if a SMTP transfer is allowed from D1 to D2 in ZoneGuard_1.
7	There are some error events in the log of ZoneGuard_1, you would like to see only the errors for Data2.	Go to the log of ZoneGuard_1 and filter out all the errors for type D2.
8	You would like to find the version of the current configuration on ZoneGuard_1.	Locate the configuration version of ZoneGuard_1.
9	You would like to know if the percentage of failed transfers is less than 2%.	Locate the statistics of successful and failed transfers.
10	You would like to know if the CRL status and the license status are valid.	First locate the CRL status. Then, locate the license status.
11	You would like to find the metadata for the current configuration on ZoneGuard_1.	Locate the metadata for ZoneGuard_1.
12	You would like to find out the time period of the log events that is shown on the log page. You would also like to change this time period.	Locate the current time period for the log events. Then, describe how you would do to change the time period.
13	You would like to search for a specific log event with the message "Interface down".	Describe how you would search for the log event with the message "Interface down".

# Appendix D

## Prototype V.1 Result

The result from the test session can be found in tables D.1-D.5. Each table consists of found usability problems, the total number of affected users and if the end-user was affected. The end-user was the person that worked at the IT and Security Department described in the section 4.6.

<b>Homepage</b>		
<b>Usability problems</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Feature request:</i> Device health	4/5	1/1
<i>Feature request:</i> Notification about abnormal number of blocked transfers.	4/5	1/1
<i>Feature request:</i> Be able to stop and start device.	1/5	1/1
<i>Feature request:</i> Filter devices	1/5	0/1
<i>Layout:</i> Both the ZoneGuards and the Data Diodes in the same view	2/5	1/1
<i>Mental model:</i> The user is confused over which IP-addresses is showed. Interface D1 or D2?	3/5	1/1
<i>Mental model:</i> The user wonders if there might be a better word than "more information".	3/5	0/1

**Table D.1:** The usability problems that was found at the homepage page.

<b>Information Page</b>		
<b>Usability problems</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Discoverability:</i> It was not intuitive how to edit the fields.	5/5	1/1
<i>Feature request:</i> Another view for editing the fields, with buttons for canceling and for saving the changes made.	3/5	1/1
<i>Feature request:</i> Download a list for the CRL-license.	2/5	0/1
<i>Mental model:</i> The user is confused if fail means blocked transfers.	1/5	0/1
<i>Mental model:</i> The user is confused over if the CRL status should be placed here.	1/5	0/1
<i>Mental model:</i> The user is confused over why the start date is shown.	1/5	1/1

**Table D.2:** The usability problems that was found at the information page.

<b>Configuration Page</b>		
<b>Usability problems</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Discoverability:</i> Hard to see what is editable.	2/5	0/1
<i>Layout:</i> Let services have it's own tab.	2/5	0/1
<i>Mental model:</i> Does not understand what metadata is.	2/5	0/1
<i>Mental model:</i> Users are confused about that it is not possible to configure the device at the configuration page.	1/5	0/1

**Table D.3:** The usability problems that was found at the configuration page.

<b>Log Page</b>		
<b>Usability problems</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Aesthetics:</i> Let each source have it's own color and then color the logs.	3/5	0/1
<i>Feature request:</i> A graph for each device when mulitple devices are shown	1/5	1/1
<i>Feature request:</i> Be able to change the amount of logs per page, between 10/50/100 per page	1/5	0/1
<i>Feature request:</i> Be able to see similar logs by selecting a log.	3/5	1/1
<i>Feature request:</i> Export logs.	1/5	0/1
<i>Feature request:</i> Filter on services	1/5	0/1
<i>Feature request:</i> Filter on specific time interval, such as, last hour, last 24 hours, last week.	2/5	0/1
<i>Feature request:</i> In the graph, see errors over time instead of events over time.	1/5	0/1
<i>Feature request:</i> Save red logs longer than green logs.	1/5	0/1
<i>Feature request:</i> See logs from all CDS devices and be able to filter on device.	1/5	0/1
<i>Feature request:</i> See sudden spikes of high load in the system log.	1/5	0/1
<i>Mental model:</i> The user is confused about what search words can be used in the search field.	1/5	0/1
<i>Mental model:</i> The user is confused if the graph would change if the user uses the filter function.	1/5	1/1
<i>Mental model:</i> The user is confused over what "type" is.	1/5	1/1
<i>Mental model:</i> The user is confused over when the log is updated.	1/5	1/1

**Table D.4:** The usability problems that was found at the log page.

<b>Visualization Page</b>		
<b>Usability problems</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Feature request:</i> See dataflow information, who talks to who?	1/5	0/1
<i>Feature request:</i> Visualize the data gathered from the SNMP-file	2/5	1/1

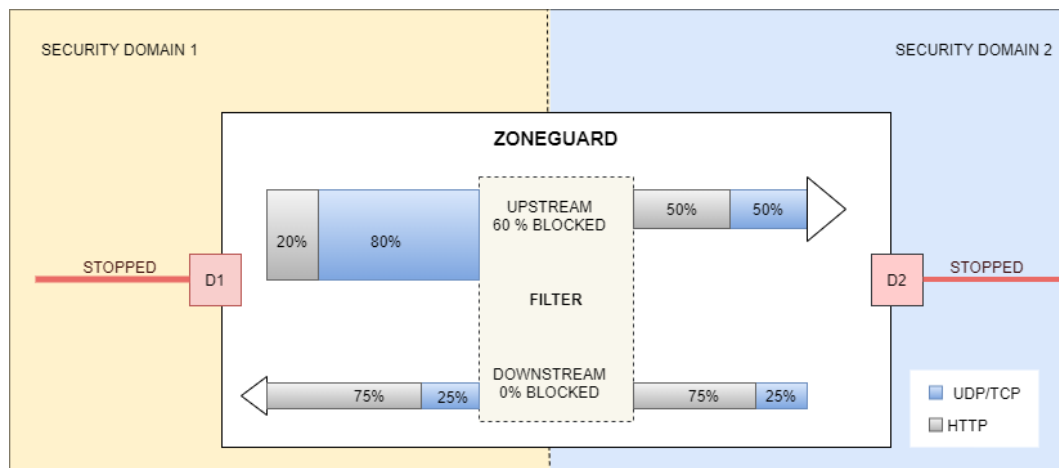
**Table D.5:** The usability problems that was found at the visualization page.

# Appendix E

## The ZoneGuard Visualization

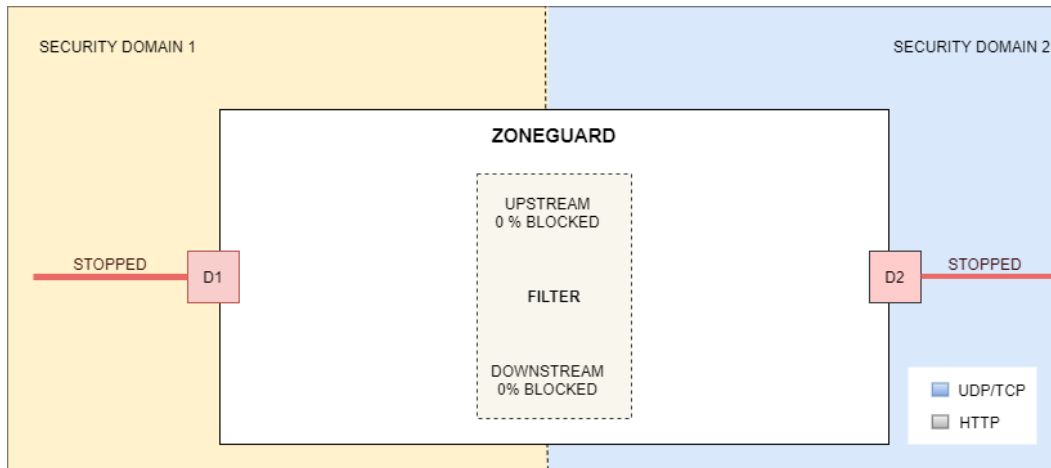
The Appendix describes how different states of a CDS device can be visualized in the prototype from Chapter 5.

In Figure E.1 the ZoneGuard has been running for a while but has entered a stopped state. The stopped state is represented by the change of colour in the D1 and D2 squares and the lines out from the squares to the colour red. The text "RUNNING" has also been changed to "STOPPED". Before the ZoneGuard was stopped, data were transferred. Therefore, there are still arrows representing data in the figure.



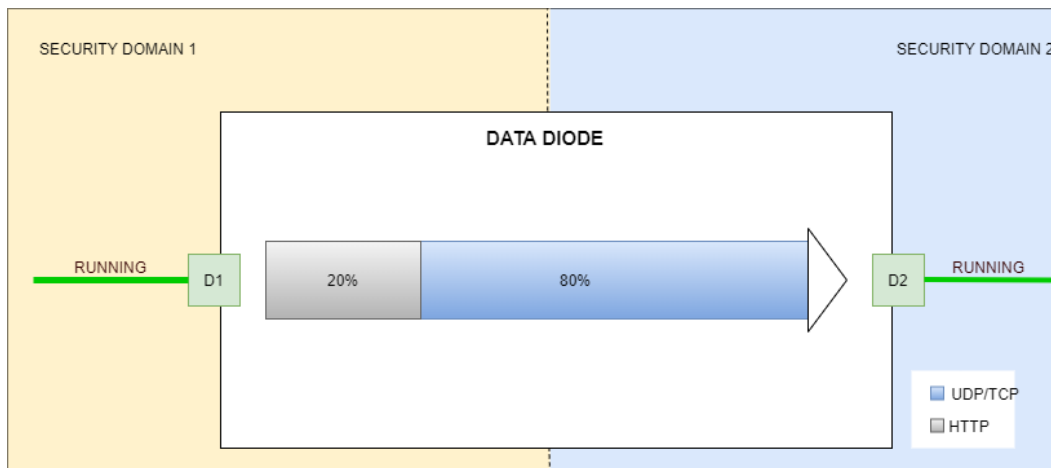
**Figure E.1:** The figure represents the data flow in a stopped ZoneGuard.

In Figure E.2 the ZoneGuard has been in a stopped state from the start of the chosen time span or no data has been sent before the stopped state. This is represented by no visible arrows.



**Figure E.2:** The figure represents a stopped ZoneGuard with no data transfers.

Figure E.3 shows how a Data Diode could be represented in a similar visualization. Because the Data Diode does not have a filter and data can only be transmitted from D1 to D2 there is no filter and only one arrow.



**Figure E.3:** The figure represents the data flow in a running Data Diode.



# Appendix F

## Prototype V.2 Test

### F.1 Introduction Letter

#### **Welcome to our test session!**

We would like you to help us test a paper prototype of Advenica's future Remote Administration Server application. The interface to be tested is a web application that enables the user to monitor multiple CDS-devices. It contains system log management, data flow statistics, and the statuses of CDS devices. The end-user can, for example, be an administrator working at a company that owns one or multiple CDS products from Advenica. The main goal of the session is to get feedback on the interface from a user's perspective.

The prototype is a paper prototype. It is meant to be unfinished and open for changes, so ideas are welcomed. We will act as the computer by changing the views and give feedback as you explore the prototype. Remember, we are testing the interface and not you so any problems encountered are valuable feedback.

We would like you to perform seven tasks. When performing a task, we encourage you to express any difficulties, emotions, and thoughts. Afterwards, we will have a feedback session where we discuss what troubles you may have encountered, what went well and anything else that may come up about the prototype.

The test session will be performed in the following order:

1. The facilitator will introduce the starting state of your test and the think-aloud protocol.
2. The facilitator will present a list of tasks for you to perform.

3. After all tasks are completed the feedback session starts there you will do something called “I Like, I Wish, What if” and a discussion.
4. After the feedback session, you are finished.

Your feedback on the prototype will be used in our master thesis report.

**Thank you for your contribution!**

## F.2 Tasks

Following is a set of task we would like you to complete.

1. Which device do you want as an administrator get more information on?
2. Go to system logs for ZoneGuard\_1.
3. How would you do to see the logs from the last two days?
4. Go to device monitoring for ZoneGuard\_1.
5. Analyze the information found for the system overview.
6. Analyze the information found for the Downstream.
7. Analyze the figures of the ZoneGuard.

## F.3 Feedback Session

The “I Like, I Wish, What if” template can be found on the next page.

## “I Like, I Wish, What If” Template

I Like	I Wish	What If

 INTERACTION DESIGN FOUNDATION | [INTERACT\(ON-DESIGN\).ORG](http://INTERACT(ON-DESIGN).ORG)

CC BY-NC-SA Creative Commons BY-SA license: You are free to edit and redistribute this template, even for commercial use, as long as you give credit to the Interaction Design Foundation. Also, if you remix, transform, or build upon this template, you must distribute it under the same CC BY-SA license.

# Appendix G

## Prototype V.2 Result

The result from the test session can be found in tables G.1-G.5. Each table consists of found usability problems, feedback, the total number of affected users and if the end-user was affected. The end-user was the person that worked at the IT and Security Department described in the method.

<b>Home Page</b>		
<b>Usability problems and Feedback</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Feature Request:</i> Time line on the warning messages.	1/4	0/1
<i>Positive Feedback:</i> The user liked the warning messages.	1/4	1/1
<i>Positive Feedback:</i> The user liked the device health column.	2/4	1/1
<i>Positive Feedback:</i> The user liked the device status column.	2/4	1/1
<i>Positive Feedback:</i> The user liked to see all devices.	2/4	0/1
<i>Positive Feedback:</i> The user liked the simplicity of the tables.	1/4	0/1
<i>Security:</i> The user is concerned over that a user can delete warning messages.	1/4	0/1
<i>User Suggestion:</i> Change the name of "bad" and "okey" to "failure" and "warning"	1/4	1/1

**Table G.1:** Feedback and usability problems on the home page.

---

<b>Log Page</b>		
<b>Usability problems and Feedback</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Discoverability:</i> The user has problem finding how to change time span.	1/4	0/1
<i>Feature Request:</i> Be able to change the time span by interacting with the graph.	1/4	1/1
<i>Positive Feedback:</i> The user liked the layout of the page.	1/4	0/1
<i>Positive Feedback:</i> The user liked the filter functionality.	3/4	1/1
<i>Positive Feedback:</i> The user liked the specified time spans.	1/4	0/1
<i>Positive Feedback:</i> The user liked the search functionality.	1/4	0/1

**Table G.2:** Feedback and usability problems on the log page.

<b>CDS Data Visualization Figures</b>		
<b>Usability problems and Feedback</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Feature Request:</i> Show the data in the figure in real-time.	1/4	1/1
<i>Feature Request:</i> To get an introduction to the figures.	1/4	0/1
<i>Mental Model:</i> The user is confused over the different width of the arrows in the figures.	4/4	1/1
<i>User Suggestions:</i> Instead of representing no data with no arrows, represent it with a very thin arrow line.	1/4	0/1
<i>User Suggestions:</i> Instead of arrows, use time line to see transfers over time.	1/4	0/1

**Table G.3:** The usability problems that was found at the homepage page.

<b>Device Monitoring Page</b>		
<b>Usability problems and Feedback</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Discoverability:</i> The user has problem finding how to change unit.	1/4	0/1
<i>Feature Request:</i> Be able to see the values and scales on the axes.	3/4	0/1
<i>Feature Request:</i> To get more information by hover over the graph.	1/4	0/1
<i>Feature Request:</i> Be able to see the disc status of the Data Diodes.	1/4	1/1
<i>Feature Request:</i> Be able to open a more advanced third party monitoring program when necessary.	1/4	1/1
<i>Layout:</i> The user think it would be hard to compare downstream and upstream with this layout.	1/4	0/1
<i>Mental Model:</i> The user is confused over if the values in the area graph is stacked.	2/4	0/1
<i>Mental Model:</i> The user is confused over why the time drop-down menu is outside tab page area.	2/4	0/1
<i>Mental Model:</i> The user is confused over the names; D1, D2, downstream and upstream.	1/4	0/1
<i>Mental Model:</i> The user is confused over when the logs are updated.	1/4	0/1
<i>Positive Feedback:</i> The user liked the layout of the page.	3/4	1/1
<i>Positive Feedback:</i> The user liked the simplicity of the page.	1/4	1/1
<i>User Suggestion:</i> The user is think that a pie chart is unnecessary for the memory. Instead, show only free memory.	1/4	0/1

**Table G.4:** Feedback and usability problems on the device monitoring page.

---

<b>Other Comments</b>		
<b>Usability problems and Feedback</b>	<b>Affected users</b>	<b>Affected end-user</b>
<i>Feedback Request:</i> To be able to administrate services, configurations and firmware version.	1/4	0/1
<i>Feedback Request:</i> To be able to start and stop the device.	1/4	0/1

**Table G.5:** Other feedback received on the prototype.

# Appendix H

## Design Workshop Plan

### **H.1 Background**

The purpose of the workshop is to generate ideas on how the data from the SNMP file can be visualized.

### **H.2 Process**

The workshop will be two hours long. The time schedule can be seen in Table H.1. A design workshop exits of the following steps[22]:

1. Prototype
2. Present and critique
3. Converge
4. Prioritise



Time (min)	Activity
13	Introduction to the workshop
25	Sketch individually
5	Discuss in pairs
15	Sketch in pairs
10	Break
20	Six thinking hats method
15	Sketch in group
15	Prioritise features

**Table H.1:** Timetable for the design workshop.

### H.2.1 Part 1: Prototype

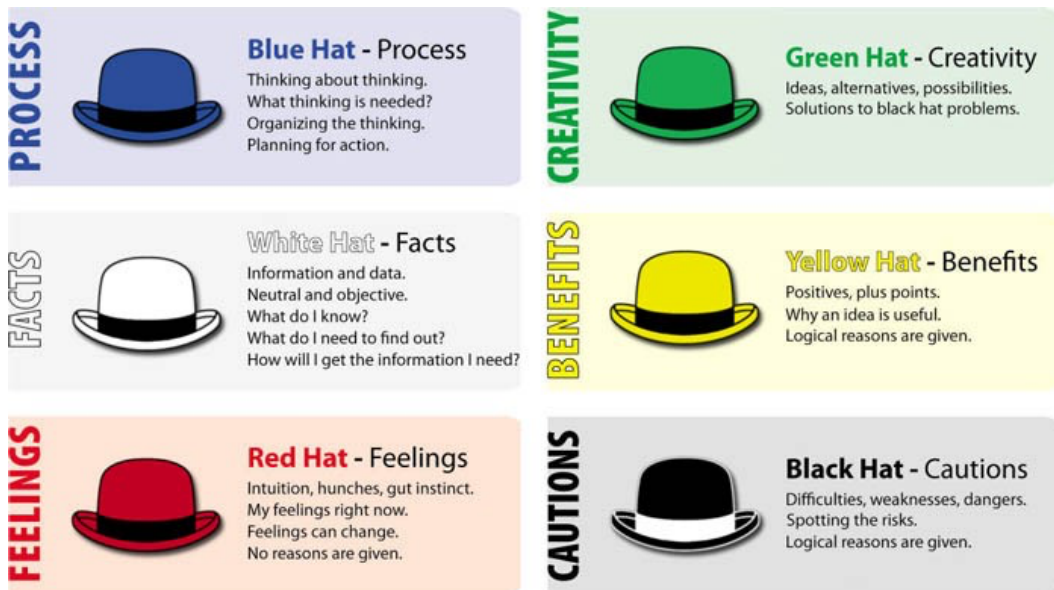
**Goal:** The primary goal of the prototype step is to generate a wide set of ideas.

**Method:** The participants brainstorm and sketch ideas on paper individually. After some time the participants divide into pairs and present and discuss their sketches to each other. After a short discussion another round of brainstorming and sketching is performed but this time in pairs.

### H.2.2 Part 2: Present and Critique

**Goal:** The goal of the present and critique step is to share and discuss the ideas in the group. The goal is, in addition, to generate feedback to discover each sketch's weaknesses and strengths.

**Method:** The facilitators present the method, the six thinking hats, to the participants. The six thinking hats is a technique used to generate various kinds of feedback and discussion. The hats are distributed to each participant and each hat represents a specific way of thinking[22]. The hats that are going to be used in the workshop are described in Figure H.1.



**Figure H.1:** Six hats representing different ways of thinking. (Figure is from: [www.onedaydesignchallenge.net](http://www.onedaydesignchallenge.net).)

### H.2.3 Part 3: Converge

**Goal:** The goal of the converge part is that the group together agrees on a design sketch.

**Method:** The group brainstorms and together sketches their design on paper. The feedback from the last step should help the participant sketch and decide on a design.

### H.2.4 Part 4: Prioritise

**Goal:** In the prioritise part, the goal is to decide on a direction for the application. The following questions should be answered:

- Should the sketch be implemented?
- Which functionality should be implemented?
- What priority has each design or functionality?

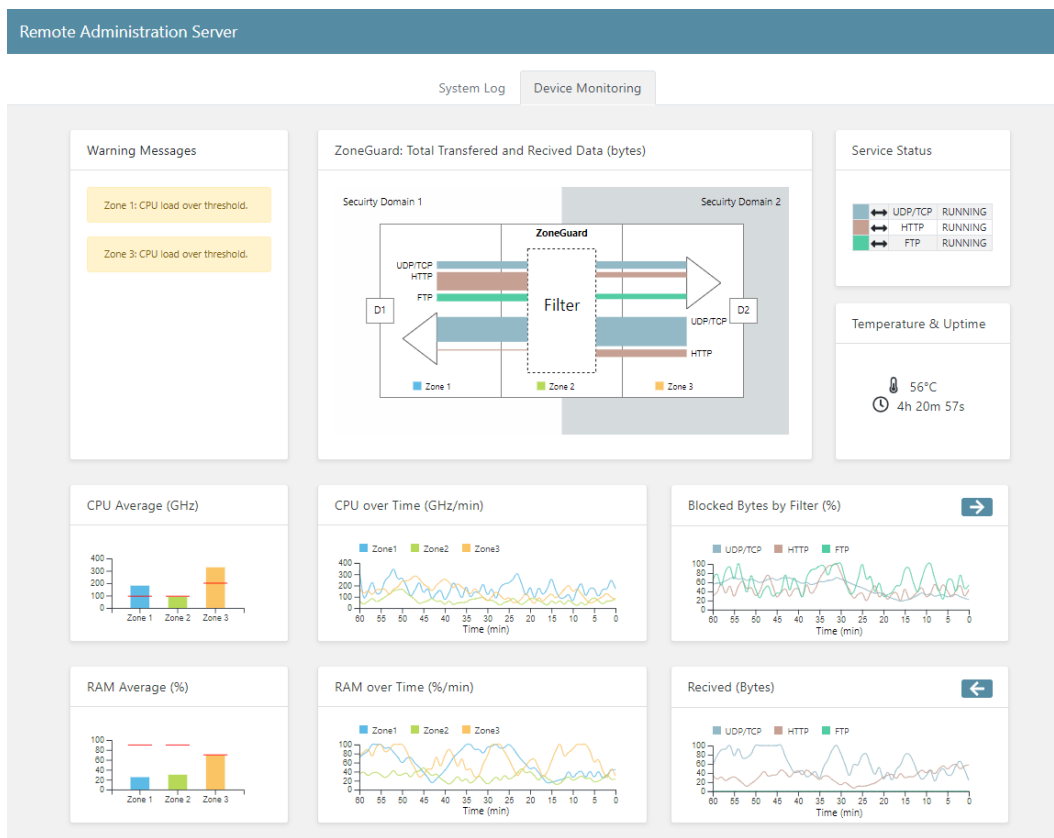
**Method:** The facilitators present the questions that should be answered (see the list above) and describe the prioritising technique. The prioritising should be performed by giving each design or functionality a priority. The following priorities can be given:

1. Low priority
2. Medium priority
3. High priority
4. Super high priority

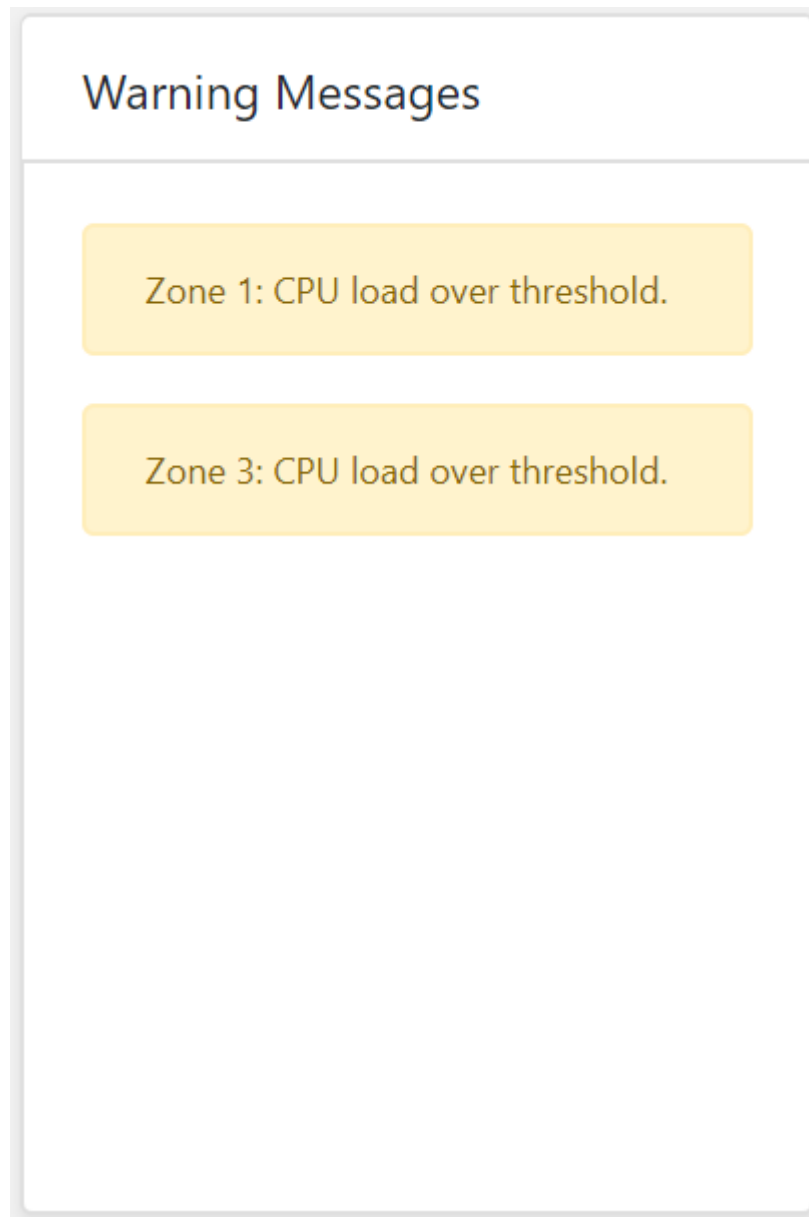
# Appendix I

## Functional Prototype Result

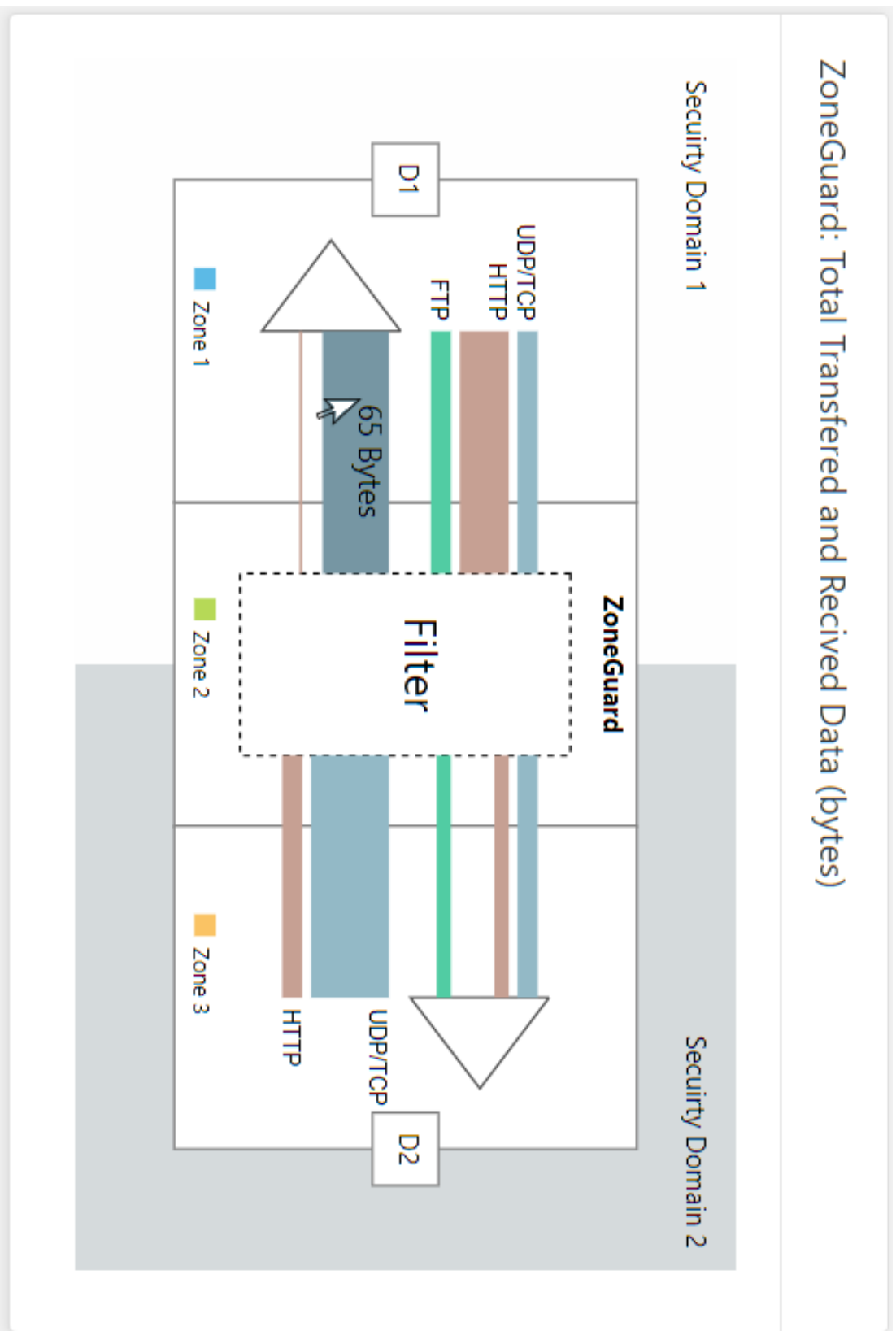
In this appendix one can find the functional prototype and close ups for the individual cards for more details.



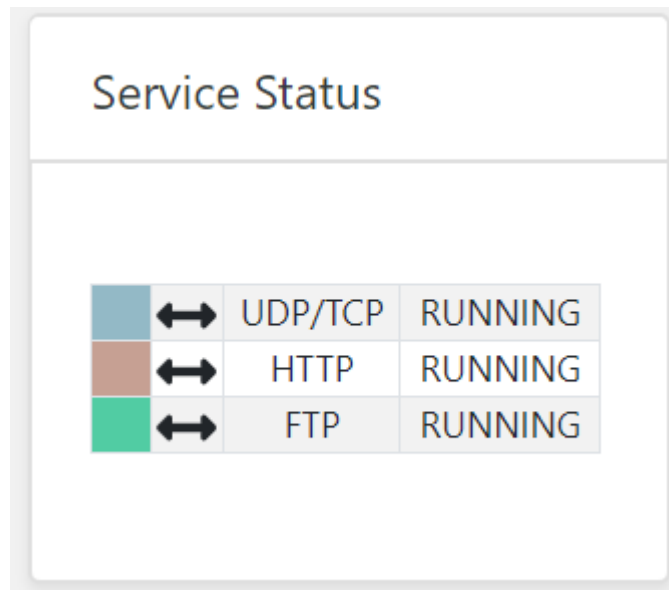
**Figure I.1:** The functional prototype iteration as a whole.



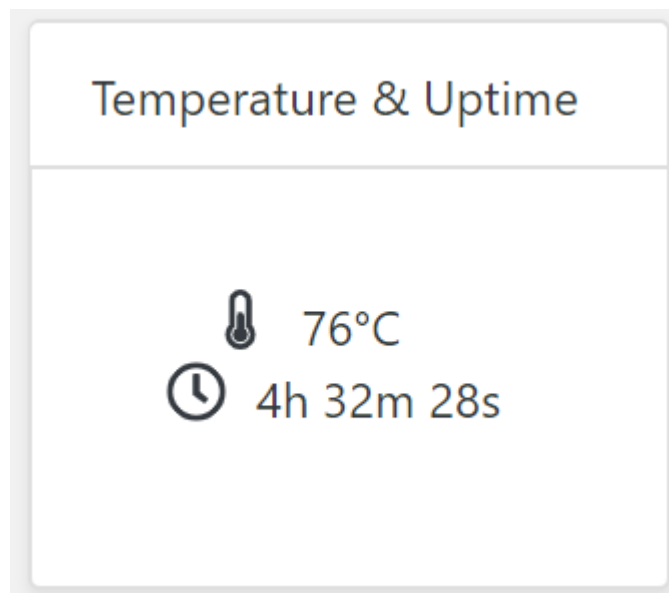
**Figure I.2:** The cards for warning messages.



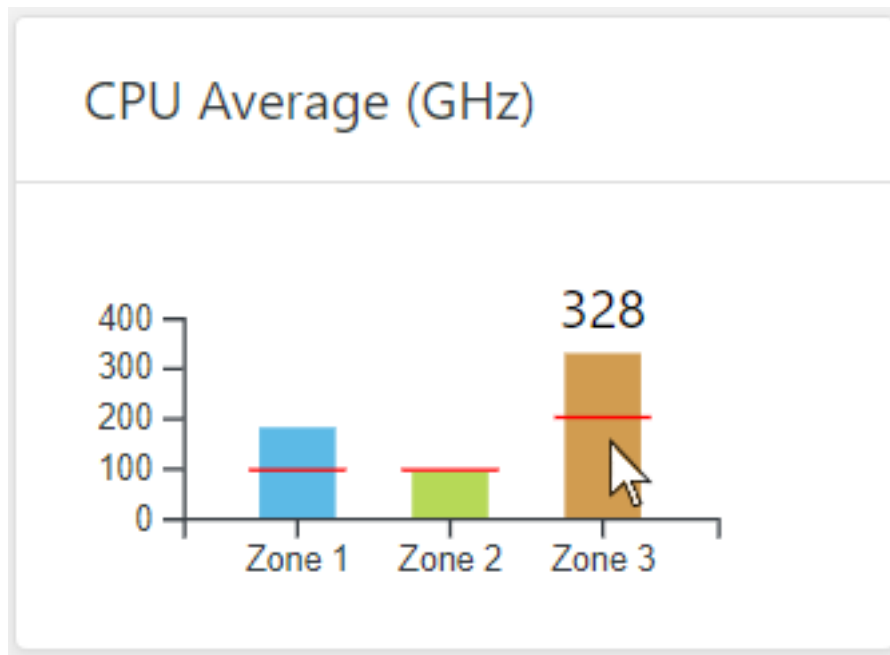
**Figure I.3:** The ZoneGuard card with an overview of the total amount of transferred and received data that can be seen when hovering the mouse over the individual sidebars.



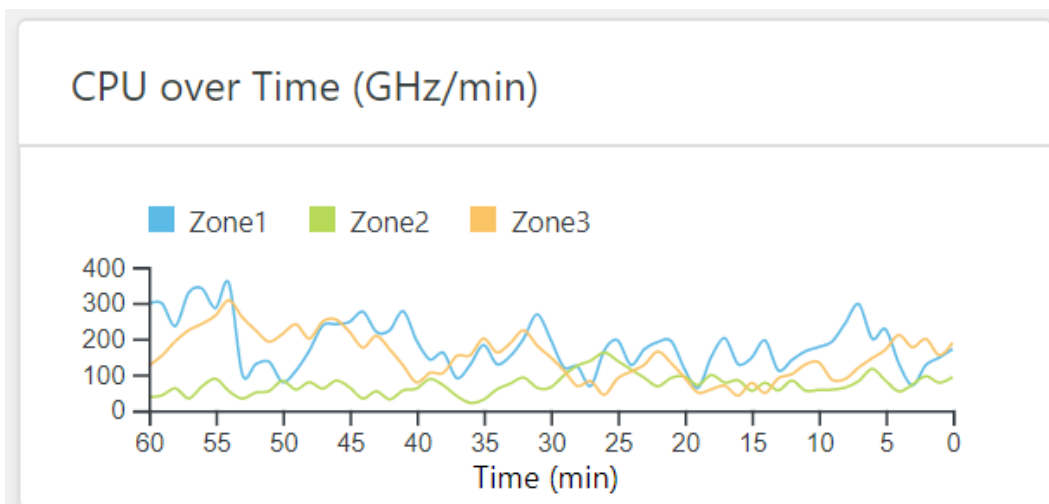
**Figure I.4:** The service status card with information about what service is up and running or down and for which way the data of a service can flow.



**Figure I.5:** A card with information of the devices temperature and for how long the device has been running.

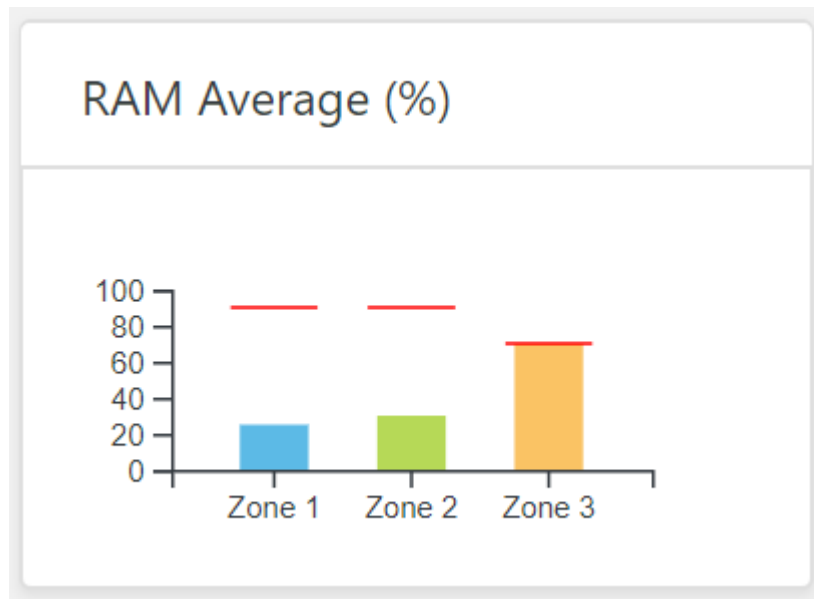


**Figure I.6:** A card with the average of the CPU for the different zones of the ZoneGuard over the whole time the device has been running. Here the user can get the value by hovering the mouse over the individual bars in the graph. The red line represents the CPU threshold for each zone.

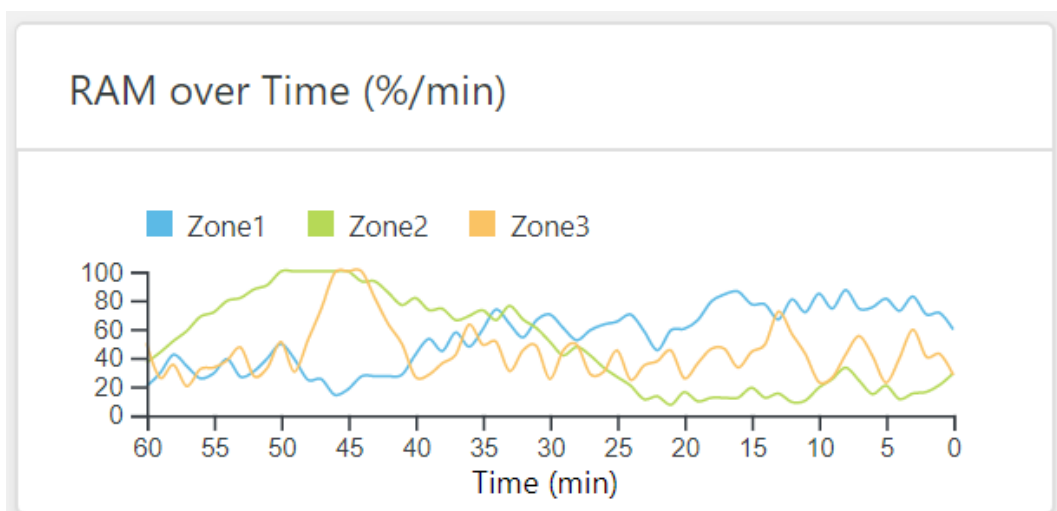


**Figure I.7:** A card with the CPU over the latest hour for the different zones of the ZoneGuard.

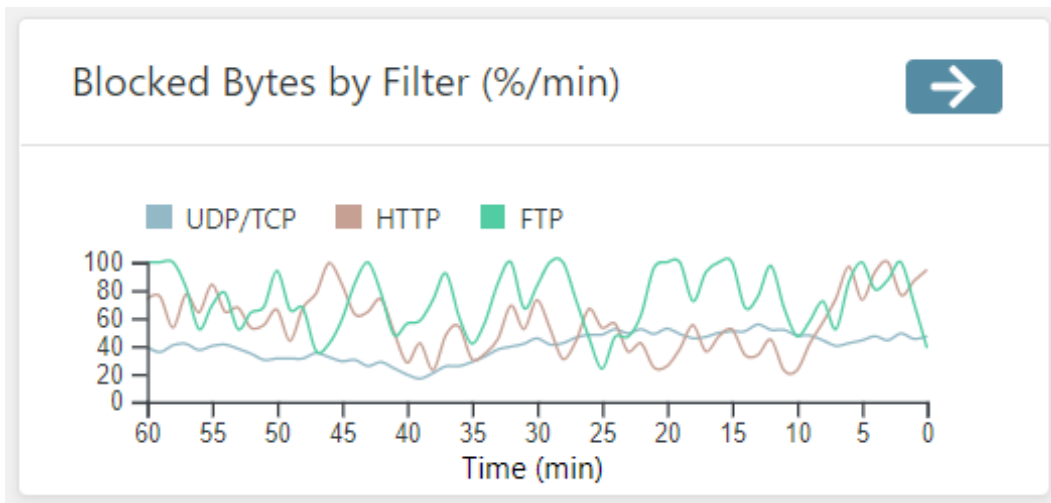




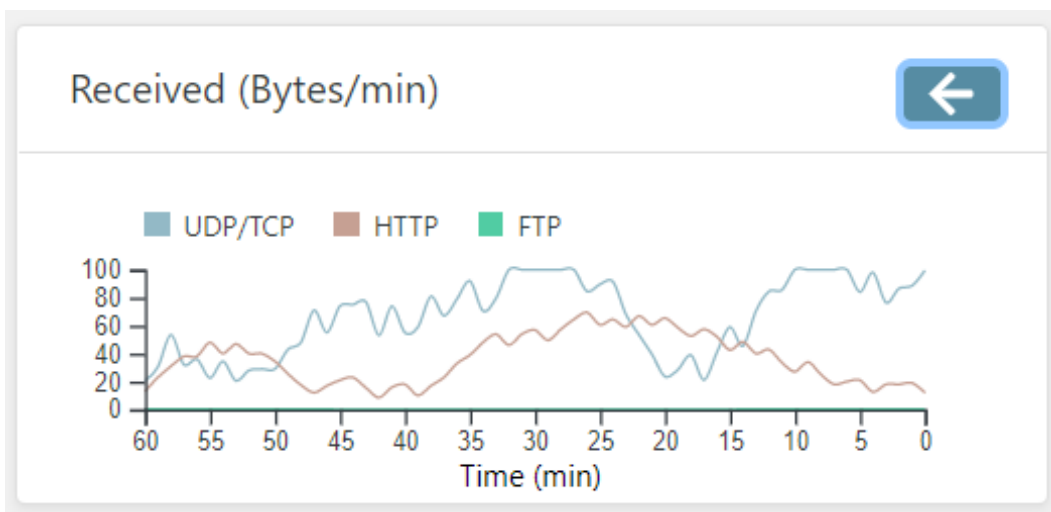
**Figure I.8:** A card with the average of the RAM for the different zones of the ZoneGuard over the whole time the device has been running. Here the user can get the value by hovering the mouse over the individual bars in the graph. The red line represents the RAM threshold for each zone.



**Figure I.9:** A card with the RAM over the latest hour for the different zones of the ZoneGuard.



**Figure I.10:** A card with a view of how much data in percent that is blocked over time for the different services in the direction security domain 1 to security domain 2.



**Figure I.11:** A card with how much data for the different services that passed the filter in the direction Security Domain 2 to Security Domain 1.