

# Optimization Swisscom Design System Chart Components

Department of Computer Science  
OST - University of Applied Sciences  
Campus Rapperswil-Jona

Spring Term 2023  
Version 8de42c0

Authors	Florian Blum & Niklas Kaesler
Advisor	Prof. Dr. Markus Stolze
Project Partner	Swisscom AG
Project Partner Representative	Alex Kühne
External Co-Examiner	Markus Flückiger
Internal Co-Examiner	Ivan Bütler, Docent OST Cyber Security

# Abstract

Swisscom AG uses the Swisscom Digital Experience (SDX) design system, which provides reusable components and guidelines for web development. Within SDX, chart components (pie chart, horizontal bar chart, vertical bar chart) are migrated to modern web components to create a unified design and user experience. While the pie chart is optimized, the other charts rely on rudimentary implementations of basic technologies, resulting in inconsistent design and a complex usage. This thesis aims to reimplement the legacy chart components as web components using the Stencil library and Scalable Vector Graphics (SVG).

A comprehensive analysis clarified the functionality and identified areas for improvement, focusing on horizontal and vertical bar charts. The aim was to achieve a consistent design and improved usability. Key technologies used included Stencil for web components, SVG for scalable graphics and anime.js for animations. A prototype was developed for testing and familiarization. The implementation followed an agile approach. Usability and user experience testing validated the usability for developers and an enhanced experience for users.

To ensure timely delivery, only the horizontal bar chart component was implemented. It handles input attributes, displays error messages for input data related issues and improves the user experience across desktop and mobile devices. Using SVG, it displays colored, stacked bars and labels while adhering to SDX design constraints. Rigorous testing verified correct behavior through manual and code testing. The component is delivered as a web component, enhancing its usability within HTML. Usability testing provided positive feedback, with improved readability and comprehension compared to the legacy chart component.

# Management Summary

## Initial Situation

Swisscom AG builds its websites using its own design system called Swisscom Digital Experience (SDX). SDX provides various types of reusable elements for application and web development across Swisscom AG. These components are built with the brand design in mind and adhere to defined Swisscom AG design guidelines. Within SDX there are three chart components: a pie chart, a horizontal bar chart and a vertical bar chart. The horizontal and vertical bar charts are built using simple technologies and are implemented in a way that doesn't enforce a consistent design across all chart instances. As a result, the initialization and usage of the charts is confusing and complex. In contrast, the pie chart is implemented using a new approach called a web component, which enforces a consistent design and usability across all chart instances.

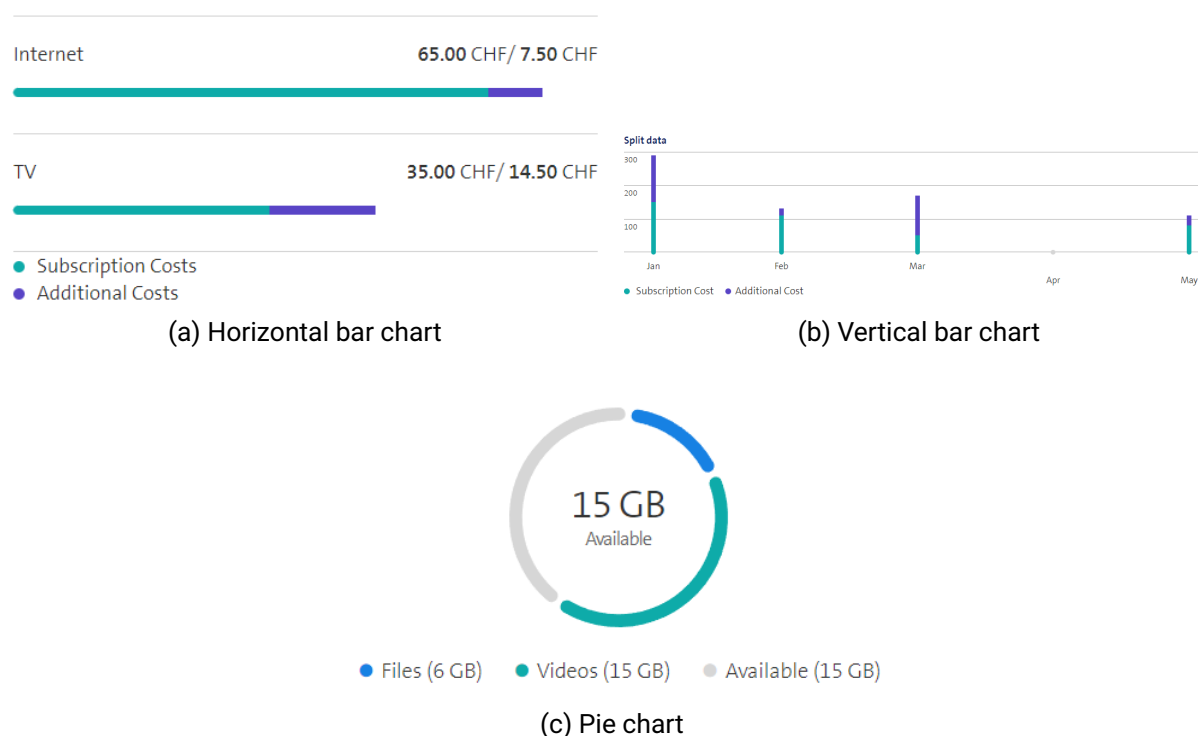


Figure 1: Chart variations

## Purpose

The aim of this thesis is to optimize the horizontal and vertical bar charts to comply with the web component standard. It also focuses on improving the design and usability to achieve consistency across all chart instances. To make the chart more attractive to developers the usability and documentation need to be improved.

## Approach

An analysis of the business requirements, existing chart functionality and design was undertaken. Constraints were assessed, resulting in the use of technologies that can handle web components. These technologies were then evaluated to identify the most suitable options to achieve the desired objectives. Based on the analysis, a well-structured solution approach was formulated, ensuring modularity and reusability through the use of web components. In

addition, design choices were made to establish a consistent design approach across all chart instances. To validate the proposed solution, a prototype and design blueprint were created, to allow stakeholders to visualize and provide feedback on the intended outcome. During implementation, the selected technologies were used to implement the charts as web components to meet the requirements. To ensure a positive user experience (UX) and developer usability, various testing processes were followed. These processes aimed to identify and resolve any potential problems and ensure that the final product met the desired UX and usability standards.

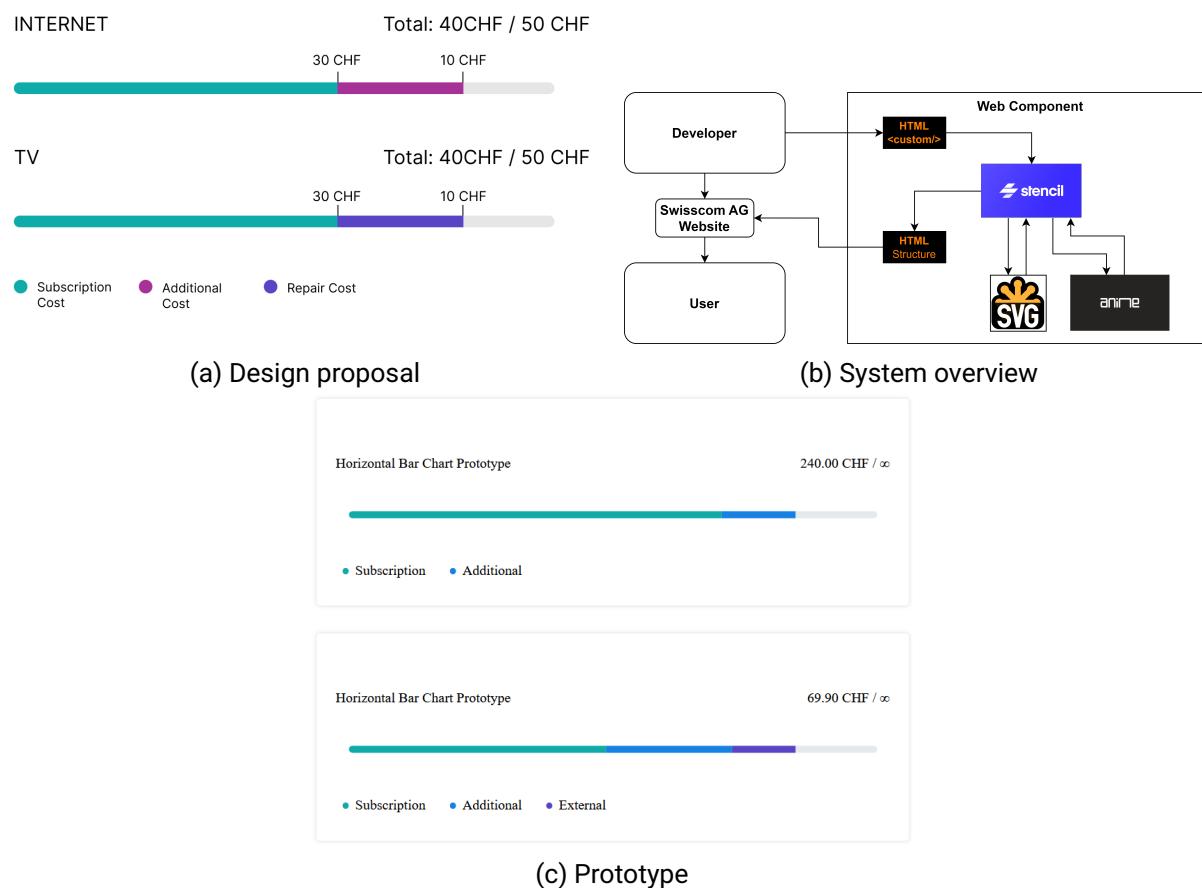


Figure 2: Approach overview

## Result

The horizontal bar chart has been optimized using the web component standard. This implementation ensures unified usability and design across all chart instances, promoting a consistent and seamless user experience. The vertical bar chart was not implemented due to time constraints. The functionality of the original horizontal bar chart has been retained and enhanced with robust error handling mechanisms. This enhancement ensures the reliability and stability of the chart, minimizing potential problems associated with incorrect or inconsistent data entry. Additionally, the new web component has been made easier to initialize and use. Supplementary to this a guide for developers has been created. The implemented solution underwent rigorous UX and usability testing, which yielded positive results. The tests demonstrated improved user experience and usability. This outcome validates the effectiveness of the design decisions and development efforts in improving the overall usability and user satisfaction.

```

<div
  id="legacy-horizontal-bar-chart-1"
  class="bar-chart-horizontal bar-chart-horizontal--split"
  data-max="80"
  data-unit="CHF"
  data-precision="2"
  data-legend="#legend"
>
  <div class="js-data-list">
    <div class="js-data" data-value="65" data-color="turquoise">
      Subscription Costs
    </div>
    <div class="js-data" data-value="7.5" data-color="iris">
      Additional Costs
    </div>
  </div>
  <div class="bar-chart__detail">
    <div class="detail-left">Internet</div>
    <div class="detail-right"></div>
  </div>
  <div class="bar-chart__progress"></div>
</div>
<div
  id="legacy-horizontal-bar-chart-2"
  class="bar-chart-horizontal bar-chart-horizontal--split"
  data-max="80"
  data-unit="CHF"
  data-precision="2"
>
  <div class="js-data-list">
    <div class="js-data" data-value="35" data-color="turquoise">
      Subscription Costs
    </div>
    <div class="js-data" data-value="14.5" data-color="iris">
      Additional Costs
    </div>
  </div>
  <div class="bar-chart__detail">
    <div class="detail-left">TV</div>
    <div class="detail-right"></div>
  </div>
  <div class="bar-chart__progress"></div>
</div>
<ul class="bar-chart__legend" id="legend"></ul>

```

```

<sdx-horizontal-bar-chart
  unit="CHF"
  precision="2"
  data='[{
    "value": 65,
    "category": "Internet",
    "subcategory": "Subscription costs",
    "maxValue": 80
  }, {
    "value": 7.5,
    "category": "Internet",
    "subcategory": "Additional costs"
  }, {
    "value": 35,
    "category": "TV",
    "subcategory": "Subscription costs",
    "maxValue": 80
  }, {
    "value": 14.5,
    "category": "TV",
    "subcategory": "Additional costs"
  }]'
></sdx-horizontal-bar-chart>

```

(a) Old chart initialization

(b) New chart initialization

Figure 3: Chart initialization comparison

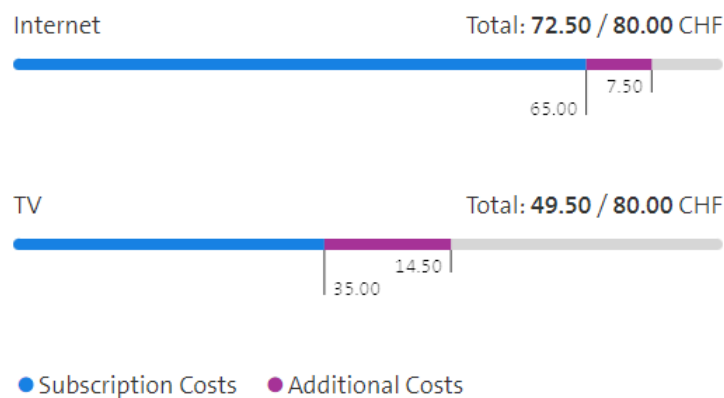


Figure 4: Horizontal bar chart web component

**Outlook**

Optimization of the vertical bar chart component is recommended so that all chart components comply with the web component standard. It is also possible to add other chart variations such as a line chart. The new horizontal bar chart can be extended with more features such as localization, improved accessibility and interactive features.

# Acknowledgements

We would like to thank the following people for their support in our thesis.

**Prof. Dr. Markus Stolze** We would like to express our special thanks to Professor Stolze for his competent and high quality supervision of our thesis.

**Markus Flückiger** We would like to express our thanks to Mr. Flückiger for his participation in our presentation and for his valuable expertise.

**Ivan Bütler, Docent OST Cyber Security** We would like to express our thanks to Mr. Bütler for his participation in our presentation and for his valuable feedback.

**Alex Kühne** We would like to express our thanks to Mr. Kühne for the pleasant cooperation and his valuable feedback.

**Michael Gfeller, Research Associate** We would like to express our thanks to Mr. Gfeller for the valuable reviews.

**AnneMarie O'Neill** We would like to express our thanks to Mrs. O'Neill for the review of the documentation and her valuable feedback.

**Test candidates** We would like to express our thanks to all test candidates for their participation and providing valuable feedback.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Management Summary</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>Contents</b>	<b>ix</b>
<b>Glossary</b>	<b>x</b>
<b>I Product Documentation</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Task Scope . . . . .	2
1.2 Vision . . . . .	3
1.3 Problem Identification . . . . .	3
1.4 Status Quo . . . . .	4
<b>2 Requirements</b>	<b>8</b>
2.1 Business Requirements . . . . .	9
2.2 Personas . . . . .	10
2.3 Functional Requirements . . . . .	11
2.4 Non-Functional Requirements . . . . .	13
2.5 Minimum Viable Product . . . . .	15
<b>3 Architecture</b>	<b>16</b>
3.1 High-Level Overview . . . . .	16
3.2 Domain Model . . . . .	17
3.3 C4 Model . . . . .	17
3.4 Architectural Constraints . . . . .	18
3.5 Architectural Decisions . . . . .	19
3.6 Chart Component Structure . . . . .	21
3.7 UX Design . . . . .	22
3.8 Libraries & Frameworks . . . . .	26
3.9 Technologies . . . . .	37
3.10 Prototype . . . . .	37
3.11 Patterns & Standards . . . . .	41
<b>4 Quality Measures</b>	<b>42</b>
4.1 Coding Guidelines . . . . .	42
4.2 UX Guidelines . . . . .	43
4.3 Definition of Done . . . . .	43
4.4 Code Metrics . . . . .	43
4.5 Test Concept . . . . .	44
4.6 Code Reviews . . . . .	50
<b>5 Deployment</b>	<b>52</b>
5.1 Production Environment Requirements . . . . .	52
5.2 Production Deployment . . . . .	52



---

<b>6</b>	<b>Results</b>	<b>53</b>
6.1	Functional Requirements Evaluation . . . . .	53
6.2	Non-Functional Requirements Evaluation . . . . .	62
6.3	Limitations . . . . .	64
6.4	Expansion Possibilities . . . . .	64
<b>II</b>	<b>Project Documentation</b>	<b>67</b>
<b>7</b>	<b>Project Plan</b>	<b>68</b>
7.1	Project Schedule . . . . .	68
7.2	Milestones . . . . .	69
<b>8</b>	<b>Project Organization</b>	<b>70</b>
8.1	Involved Parties . . . . .	70
8.2	Meeting Organization . . . . .	71
<b>9</b>	<b>Management Tools</b>	<b>72</b>
9.1	Issue Management . . . . .	72
9.2	Version Control . . . . .	72
9.3	Other Tools . . . . .	73
<b>10</b>	<b>Risk Management</b>	<b>74</b>
10.1	Risk Identification . . . . .	74
10.2	Risk Matrix . . . . .	74
10.3	Risk Evolution . . . . .	75
10.4	Risk Mitigation . . . . .	76
<b>11</b>	<b>Time Tracking</b>	<b>77</b>
<b>III</b>	<b>Appendix</b>	<b>79</b>
<b>A</b>	<b>License Agreement</b>	<b>80</b>
<b>B</b>	<b>Developer Guide</b>	<b>81</b>
<b>C</b>	<b>Manual Testing Protocol</b>	<b>90</b>
<b>D</b>	<b>UX Testing Protocols</b>	<b>95</b>
D.1	UX Test on May 23, 2023 . . . . .	95
D.2	UX Test on May 31, 2023 . . . . .	95
D.3	UX Test on June 02, 2023 . . . . .	96
D.4	UX Test on June 09, 2023 . . . . .	97
<b>E</b>	<b>Usability Testing Protocols</b>	<b>99</b>
E.1	Usability Test on May 23, 2023 . . . . .	99
E.2	Usability Test on May 31, 2023 . . . . .	99
E.3	Usability Test on June 02, 2023 . . . . .	100
E.4	Usability Test on June 09, 2023 . . . . .	101
<b>F</b>	<b>Product Screenshots</b>	<b>102</b>

---

<b>G Licenses</b>	<b>106</b>
G.1 Animate.css License . . . . .	106
G.2 Anime.js License . . . . .	109
G.3 Beautifier License . . . . .	110
G.4 Chai License . . . . .	111
G.5 ESLint License . . . . .	112
G.6 Jest License . . . . .	113
G.7 JSLint License . . . . .	114
G.8 Less License . . . . .	115
G.9 Lit License . . . . .	119
G.10 Mocha License . . . . .	120
G.11 Polymer License . . . . .	121
G.12 Prettier License . . . . .	122
G.13 Puppeteer License . . . . .	123
G.14 Sass License . . . . .	127
G.15 Sinon.JS License . . . . .	128
G.16 Slim.js License . . . . .	129
G.17 Stencil License . . . . .	130
G.18 Stylus License . . . . .	131
<b>List of Tables</b>	<b>133</b>
<b>List of Figures</b>	<b>134</b>
<b>List of Listings</b>	<b>135</b>
<b>Bibliography</b>	<b>136</b>

# Glossary

- API** Application Programming Interface: A set of rules and protocols that allows different software applications to communicate and interact with each other.. [21](#), [28](#), [30](#)
- CMS** Content Management System: A software application that allows users to create, manage, and modify digital content, enabling easy content editing, publishing and organization.. [4](#)
- CSP** Content Security Policy: A security mechanism implemented in web applications to mitigate cross-site scripting and other code injection attacks.. [21](#)
- DoD** Definition of Done: A set of agreed-upon criteria that defines when a software development task or project is considered complete.. [43](#)
- LTS** Long-Term Support: A software release or version that is supported with bug fixes, security updates, and maintenance for an extended period.. [52](#)
- MVP** Minimum Viable Product: The most basic version of a product that includes only the essential features and functionality.. [15](#)
- RUP** Rational Unified Process: A software development framework that provides a structured approach to the entire software development life cycle.. [68](#)
- SDX** Swisscom Digital Experience: Swisscom's design system which consists of a component libraries, UX principles, design guidelines, documentations and rules.. [2](#), [7](#), [8](#), [18](#), [21](#), [22](#), [25](#), [43](#)
- US** User Story: A brief, user-centric description of a desired functionality or feature.. [11](#)
- UX** User Experience: The overall impression and satisfaction a user has when interacting with a product or system.. [6](#)
- WCAG** Web Content Accessibility Guidelines: A set of internationally recognized guidelines that provide recommendations for making web content more accessible.. [14](#), [25](#), [64](#)

**Part I**

**Product Documentation**

# Introduction

Bar charts are widely used in data visualization, providing a clear and intuitive way to present information. Together with Swisscom AG as an industrial partner, this thesis aims to improve their existing chart implementations. The current implementations suffer from suboptimal usage experience and lacks consistency in design, posing significant limitations to users. This bachelor thesis aims to address these limitations by reimagining and optimizing bar charts as web components that are not restricted by the browser environment or user implementation choices.

## 1.1 Task Scope

Swisscom AG is in the process of converting legacy components from its own Swisscom Digital Experience (SDX) design system into web components. The project scope includes the existing chart components. There are three chart components:

- Horizontal bar chart
- Vertical bar chart
- Pie chart

The optimization of the pie chart component has already been successfully completed by Swisscom AG. The optimization and reimplementations of the horizontal and vertical bar charts is still pending. The aim of this thesis is to optimize these two bar charts. The minimum requirement is the implementation of one chart variant.

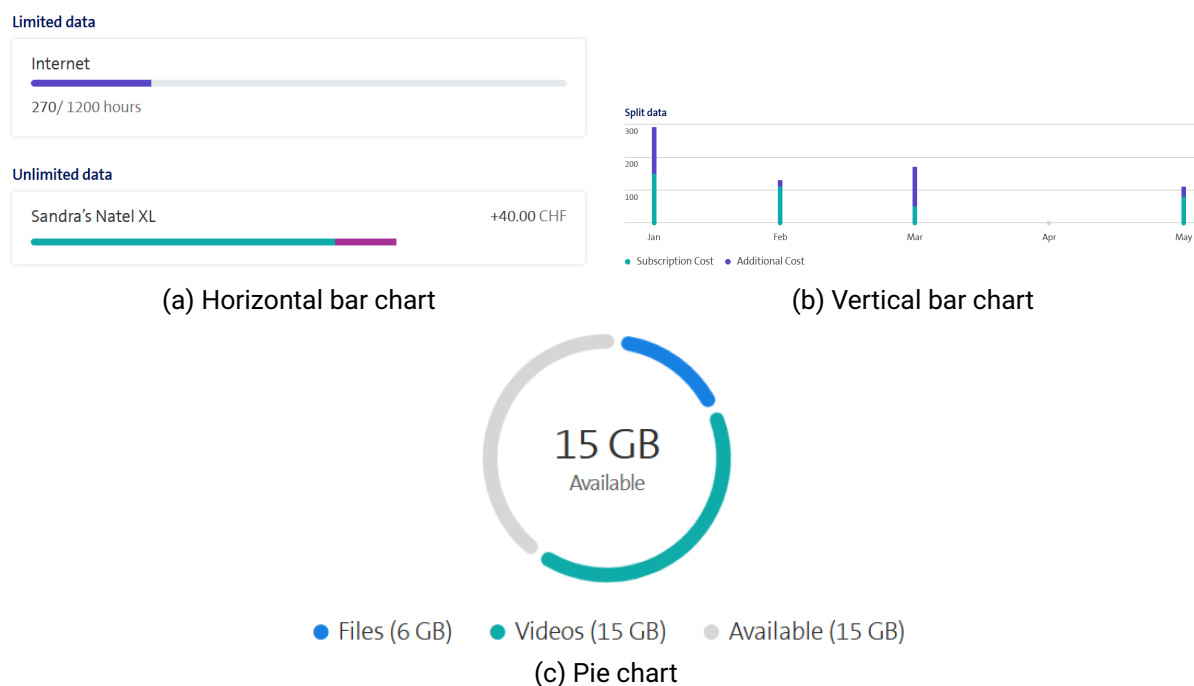


Figure 1.1: Existing chart variations

## 1.2 Vision

This bachelor thesis strives to reimplement and optimize the existing bar charts by developing flexible web components. The project involves a thorough analysis of the current charts and their functionalities to ensure a seamless integration of all existing features, while focusing on improving usability and design. The primary goal of this thesis is to implement at least one of the two proposed charts. However, efforts will be made to implement both charts, accompanied by additional features such as improved accessibility.

The development of bar chart web components follows predefined design principles aimed at unifying the existing design landscape. By adopting the web component paradigm, users will benefit from simplified usability and navigation. In addition, the implemented solution will facilitate a streamlined update cycle, allowing for rapid deployment and visibility of changes made to the bar chart to all users.

## 1.3 Problem Identification

The legacy chart components are causing problems for Swisscom AG's internal and external user groups. Internal users face problems with the usability of the components and the ability to maintain the components. Design and functionality adjustments are almost impossible due to the unknown number of chart instances across Swisscom AG. External users cannot be provided with a consistent experience. This scenario primarily affects end users. Each chart has a different style. Labels, colors, positioning, typography and shape can vary from chart to chart. All of this contributes to a confusing user experience, when confronted with legacy chart components.

The main user groups<sup>1</sup> affected are:

- [Developers](#) (subsection 1.3.1)
- [CMS users](#) (subsection 1.3.2)
- [End users](#) (subsection 1.3.3)

### 1.3.1 Developer Problems

The main issues for the developer user group are:

- Limited or insufficient documentation
- Legacy code
- Maintainability of existing code
- Usage of components is not transparent and difficult

Swisscom AG developers face complications when using existing chart components in HTML code. The parameters and their types are not documented. A developer has to refer to existing examples for more information. Even then, a developer still has to figure out which parameter causes which effect. As can be seen from the [Listing 1.1](#) and the resulting [Figure 1.2](#), there is no further guidance provided on the Swisscom AG developer site. It is not clear which attributes provide what information in the final visual of the chart.

---

<sup>1</sup> [Internal users](#) [External users](#)

```
1 <div id="my-chart" class="bar-chart-horizontal bar-chart-horizontal--limited"  
    data-max="1200" data-unit="hours">  
2   <div class="js-data-list">  
3     <div class="js-data" data-value="270" data-color="iris">Internet</div>  
4   </div>  
5   <div class="detail-left">Internet</div>  
6   <div class="bar-chart__progress"></div>  
7   <div class="detail-bottom"></div>  
8 </div>
```

Listing 1.1: Limited horizontal bar chart HTML initialization

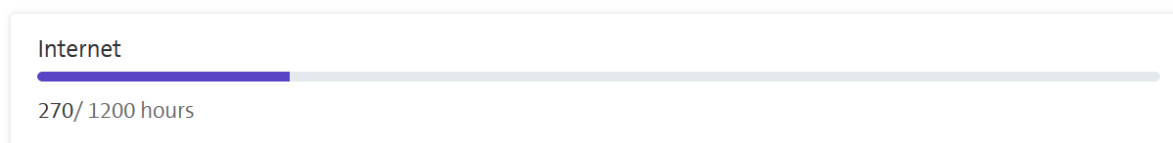


Figure 1.2: Limited horizontal bar chart produced by Listing 1.1

Due to the large number of HTML tags, it is almost impossible to modify code fragments of these web components in production. Maintaining existing code fragments using the legacy chart components is therefore problematic.

### 1.3.2 CMS User Problems

CMS users usually cannot code. Minimal coding skills are often expected. Data is difficult to render within chart components if their functionality is not documented. It can be even more difficult if the user is not familiar with HTML code. CMS users rely on the examples provided. Examples are inconsistent and take time to understand. Considering the variation of all legacy components, it is challenging to set up a chart as a CMS user.

### 1.3.3 End User Problems

Inconsistent designs confuse end users. The visuals produced by different chart versions are not consistent. Labels, typography, positioning and coloring vary from chart to chart. End users may find it difficult to process information from charts if they are repeatedly confronted with inconsistent design approaches. Legacy components provide inadequate support for disabled users. These users are prevented from easily accessing chart information using a screen reader. People with poor eyesight may find it difficult to read legacy charts and distinguish between graphical elements and colors.

## 1.4 Status Quo

The legacy bar chart solutions make use of standard web technologies. JavaScript is used as the main programming language. A Microsoft JavaScript superset called TypeScript is used to ensure type safety. In addition, HTML elements and CSS rules are used for correct web page presentation and graphical design aspects. CSS is enhanced with the Sass extension to allow for more development options and to avoid boilerplate code. Animations are mainly done using the anime.js library, with the exception of hover and focus actions.

The newer implementation of the pie chart uses the Stencil library. The core technologies used have not changed. TypeScript, HTML and Sass are still being used. Stencil is the library that helps to combine all these different technologies for a web component implementation.

The legacy chart components are invoked by specifying multiple HTML tags. A combination of CSS classes and HTML attributes provide the rendering data for the component implementation. The JavaScript code processes all the input and builds the chart component. The implementation uses JavaScript classes and defines behavior for updating data and unmounting objects.

```
1 <h4 class="h4">Vertical bar chart title</h4>
2 <div id="my-chart" class="bar-chart-vertical" data-max="300" data-unit="CHF">
3   <div class="js-data-list">
4     <div class="js-category" data-color="turquoise">Subscription Cost</div>
5     <div class="js-category" data-color="iris">Additional Cost</div>
6     <div class="js-data" data-value="150, 140">Jan</div>
7     <div class="js-data" data-value="110, 20">Feb</div>
8     <div class="js-data" data-value="50, 120">Mar</div>
9     <div class="js-data" data-value="0, 0" data-class="hidden-phone-down">Apr</div>
10    <div class="js-data" data-value="80, 30"
11      data-class="hidden-phone-down">May</div>
12  </div>
13  <div class="chart-wrapper">
14    <ul class="bar-chart__grid">
15      <li><div class="axis-y-label">300</div></li>
16      <li><div class="axis-y-label">200</div></li>
17      <li><div class="axis-y-label">100</div></li>
18    </ul>
19    <ul class="chart js-chart"></ul>
20  </div>
21  <ul class="bar-chart__legend"></ul>
22 </div>
```

Listing 1.2: Vertical bar chart HTML initialization

### Vertical bar chart title

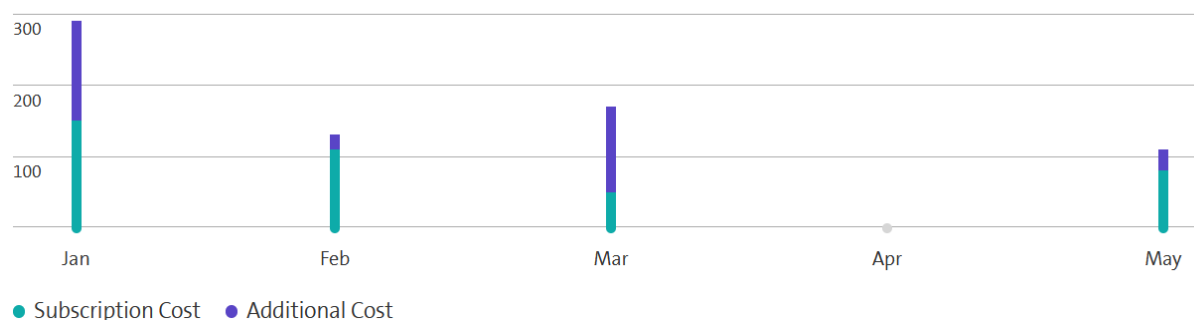
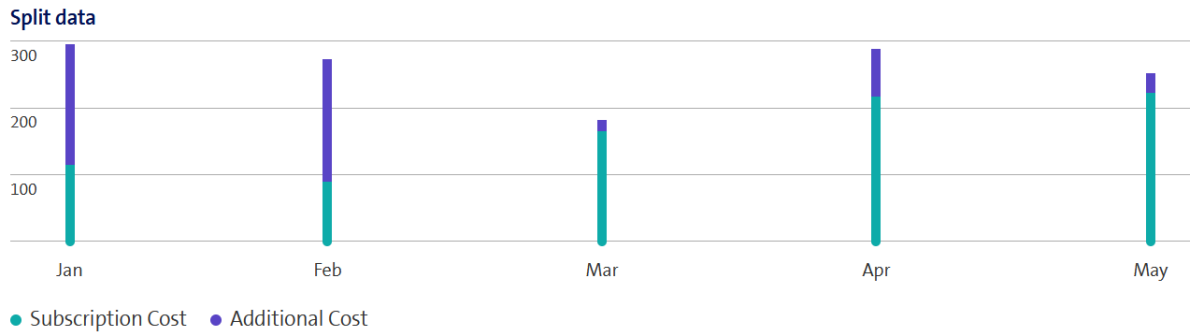


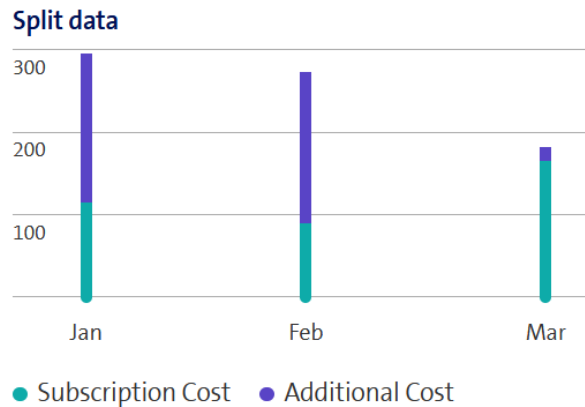
Figure 1.3: Vertical bar chart from Listing 1.2

The behavior for mobile devices is solved by a responsive approach. All chart components use this approach. The vertical bar chart displays fewer bars in mobile view to save space and avoid overcrowding on small displays.





(a) Vertical bar chart desktop view

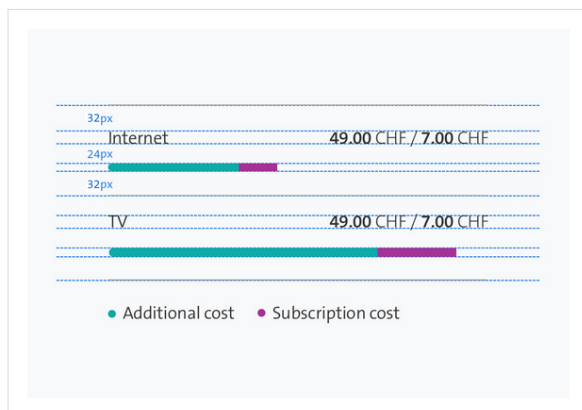


(b) Vertical bar chart mobile view

Figure 1.4: Vertical bar chart desktop vs. mobile view

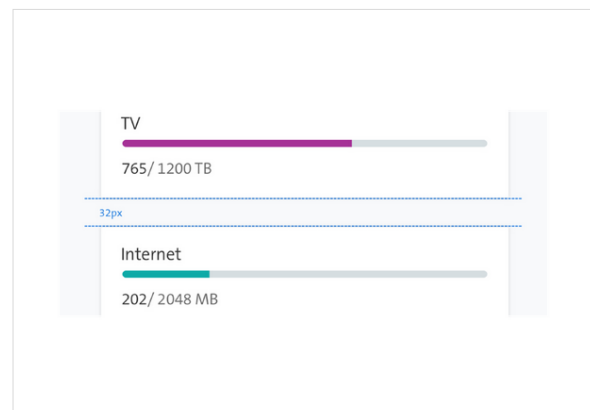
UX design guidelines are provided by Swisscom AG. Each component follows rules defined by the Swisscom AG designers. This includes coloring, spacing, positioning and designs for chart variations. The style guides are not complete, but give a general idea of how the design rules are applied.

### Minimum height



The bar chart always requires a minimum padding between elements to ensure a visual relationship between data and labels.

### Relative positioning



The bar chart should always have 32px space between itself and other elements.

Figure 1.5: Excerpt from horizontal bar chart designer rules

---

Code testing is not available, because errors can be easily detected in the final rendering of a component. Swisscom AG does not plan to implement testing for chart components, although a testing solution for the new web components would not be rejected.

Documentation is available in the form of code comments and on the Swisscom AG [SDX](#) website. HTML code snippets show how the components are used. The bar chart lacks parameter descriptions. Documentation is generally usable, but lacking in specific use cases.

# Requirements

The following chapter specifies the elaborated functional and the non-functional requirements of the web components. These requirements are the foundation for the architecture. A verification of each requirement is done at the end of the project. For the functional and non-functional requirements the ISO/IEC 9126 standard is used [1]. This standard divides the software quality into 6 categories:

- **Functionality:** Is covered through functional and non-functional requirements.
- **Usability:** Is covered through functional and non-functional requirements.
- **Maintainability:** Is covered through non-functional requirements.
- **Efficiency:** Is covered through non-functional requirements.
- **Portability:** Is covered through non-functional requirements.
- **Reliability:** Is not covered as the chart for this category is dependent on the reliability of the [SDX](#) platform where it will be used in production.



Figure 2.1: ISO 9126 [2]

## 2.1 Business Requirements

This bachelor thesis, conducted in collaboration with Swisscom AG, aims to develop improved implementations of web-based bar chart components. The aim is to develop these implementations using the web component paradigm, Stencil and SVG. The new codebase should adhere to the established concept and style of the previously converted pie chart component. In addition, the new implementations should provide at least the same level of functionality as their predecessors. Furthermore, the new charts should also have a consistent user experience and design across Swisscom AG. Justifications for introducing new functionality and design changes must be adequately supported. The decision to adopt the web component style is driven by the desire to address maintainability challenges in the production code. Swisscom AG places a high value on the production readiness and usability of the code implementations.

## 2.2 Personas

On closer inspection, it becomes clear that the implementation of this solution affects two groups of users: developers and end users. To better understand their needs, two distinct personas were created using an online persona generator<sup>1</sup>. These personas were developed using insights from Matthew J. Hamm's "Wireframing Essentials" book [3], to ensure a comprehensive understanding of the target audience.

Both Figure 2.2 and Figure 2.3 were created with HubSpot. [4]



Figure 2.2: Persona developer



Figure 2.3: Persona end-user

<sup>1</sup><https://www.hubspot.com/make-my-persona>

## 2.3 Functional Requirements

In this section, the functional requirements are presented as User Stories (US) derived from the analysis and elaboration of the existing bar chart code and the assignment in consultation with the stakeholders. These user stories capture the desired functionalities and interactions that serve as a foundation for the development of the new web component implementations. In the context of this project, a use case diagram is not included, as it does not provide any additional relevant information.

The functional requirements cover the functionality of the ISO/IEC 9126 standard to ensure that the web component can meet its requirements. This section also covers the usability of the standard through a set of useful functionalities that ensure a fluid and intuitive usage of the chart.

### 2.3.1 General Functional Requirements

The following requirements apply to both the required horizontal chart and the optional vertical bar chart.

**GNL-US1** As a user, I want to be able to view a legend for all charts so that I can easily understand the data being presented.

**GNL-US2** As a user, I want to see gray as a default background bar color if there is no data, so that I can comprehend and compare the existing data better.

**GNL-US3** As a user, I want all elements to be consistently labeled across all charts for easier comprehension.

**GNL-US4** As a user, I want the colors to be shown in a logical order within the graph, so that I can differentiate between them easily.

**GNL-US5** As a user, I want to have similar input data across all charts, so that I can easily switch between the graphs.

**GNL-US6** As a user, I want to add negative data, so that I have more flexibility and options with my data.

**GNL-US7** As a user, I want to know which chart is the best option for certain data sets in form of a quick guide, so that I can choose the best chart for my data.

**GNL-US8** As a user, I want to have an update function that updates the chart with an animation, to enhance my experience using the chart component and removing the need to reload the site.

**GNL-US9** As a user, I want to have a consistent font and spacing, so that I can easily compare graphs with each other.

**GNL-US10** As a user, I want to see a gray dot if there is no data or the data is zero, so that I am not confused as to why there is no bar.

### 2.3.2 Horizontal Bar Chart Functional Requirements

These requirements apply for the horizontal bar chart type.

#### General:

**HCG-US1** As a user, I want to see a horizontal bar for my data, so that I can quickly have an overview.

**HCG-US2** As a user, I want to define the unit of the data so that I can display the data in a meaningful way.

**HCG-US3** As a user, I want to display detailed information about the data, so that I can gain quick access to more information about the chart and the data.

**HCG-US4** As a user, I want to be able to see all the data points, even if they are very small compared to the largest, so that I can still recognize even the smallest values.

**HCG-US5** As a user, I want to adjust the precision of the value displayed so that I can control the level of detail in the chart.

#### **Infinity data version:**

**HCI-US1** As a user, I want to display data without any upper limit so that I can visualize large amounts of data in a single chart.

**HCI-US2** As a user, I want to display multiple different values in a single bar so that I can compare different values side by side.

**HCI-US3** As a user, I want input values to be displayed proportional so that I can see the relative size of each value.

#### **Limited data version:**

**HCL-US1** As a user, I want to display data in proportion to a set limited, so that I know when the limit is reached.

**HCL-US2** As a user, I want to define the maximum value to set the scale, so that I can have more fine control over that data.

#### **Stacked data version:**

**HCS-US1** As a user, I want to display multiple categories in one chart split over multiple bars, so that I can easily compare the categories.

**HCS-US2** As a user, I want to see different colors for the different subcategories, so that I can easily distinguish between them.

**HCS-US3** As a user, I want to have a simple design without gray dividing lines, so that I have a compact view of the data.

### **2.3.3 Optional Functional Requirements**

#### **Vertical bar chart:**

**OPT-US1** As a user, I want to display different types of data in a chart of two dimensions, so that I can easily compare them.

**OPT-US2** As a user, I want to label the axis, so that I have a better understanding of the displayed data.

**OPT-US3** As a user, I want to have the option to set a maximum unit, so that I can insert data without too much overhead.

**OPT-US4** As a user, I want to have a consistent design for the displayed bars, so that I am not confused where the bars start and end.

**OPT-US5** As a user, I want to show multiple values in one bar which are distinguished with colors, to display more information within one chart.

**Miscellaneous:**

**OPT-US6** As a user, I want to click on one part of a bar to highlight it and display additional information to help me understand the chart better.

**OPT-US7** As a vision impaired user, I want to have better accessibility, so that I can easily differentiate between the colors.

**OPT-US8** As a developer, I want to be able to run tests, so that I can prove the features are working as intended.

**OPT-US9** As a user, I want to display my data in a line chart, so that I can show my data over a period of time or other dimensions.

**OPT-US10** As a user, I want to be able to view hover annotations to display additional information and aid my understanding of the chart.

## 2.4 Non-Functional Requirements

This section covers the maintainability, efficiency, portability and usability categories of the ISO/IEC 9126 standard to ensure smooth and continuous use of the chart.

NFR-M01		Maintainability
Requirement	The generated code is maintainable and adheres to a professional production standard.	
Measure	Two code reviews are performed by a consulting code reviewer.	
Dependencies	-	
Urgency	Mandatory	
Priority	High	

Table 2.1: NFR maintainability 01

NFR-M02		Maintainability
Requirement	Code complies with coding guidelines.	
Measure	Stakeholder inspects the code in an acceptance process. Revisions are realized by the developers. Provided linter and code formatter are applied.	
Dependencies	-	
Urgency	Mandatory	
Priority	High	

Table 2.2: NFR maintainability 02



NFR-M03		Maintainability
Requirement	Test coverage is at least 75% on all written code pieces.	
Measure	Automated test suite and code coverage tool.	
Dependencies	<a href="#">OPT-US8</a>	
Urgency	Optional	
Priority	Medium	

Table 2.3: NFR maintainability 03

NFR-E01		Efficiency
Requirement	Loading times of the new implementations do not increase more than 50%.	
Measure	Compare loading times.	
Dependencies	-	
Urgency	Mandatory	
Priority	High	

Table 2.4: NFR efficiency 01

NFR-P01		Portability
Requirement	Cross browsing capabilities are fulfilled for the latest versions (defined on 21.03.2023) of: <ul style="list-style-type: none"> <li>• Google Chrome 111.0.5563.65</li> <li>• Mozilla Firefox 111.0</li> <li>• Apple Safari 16.3</li> <li>• Microsoft Edge 111.0.1661.44</li> </ul>	
Measure	Validate requirements in each browser.	
Dependencies	<a href="#">GNL-US3</a> , <a href="#">GNL-US8</a> , <a href="#">GNL-US10</a>	
Urgency	Mandatory	
Priority	High	

Table 2.5: NFR portability 01

NFR-U01		Usability
Requirement	Component implementation satisfy the needs of the <a href="#">WCAG</a> level 2 AA standard.	
Measure	Manual tests according to the <a href="#">WCAG</a> standard and usage of Google Lighthouse.	
Dependencies	<a href="#">OPT-US7</a>	
Urgency	Optional	
Priority	Low	

Table 2.6: NFR usability 01

## 2.5 Minimum Viable Product

The Minimum Viable Product (MVP) consists of all non-optional functional and non-functional requirements. The MVP includes the complete realization of the horizontal bar chart using Stencil and SVG technologies, following the principles of web component development. The implementation will include the adoption of all existing functionality with the exception of the hover feature, which will be replaced with a direct visible labelling approach. The design and usage of the chart component has to follow the approach used for the pie chart.

# Architecture

## 3.1 High-Level Overview

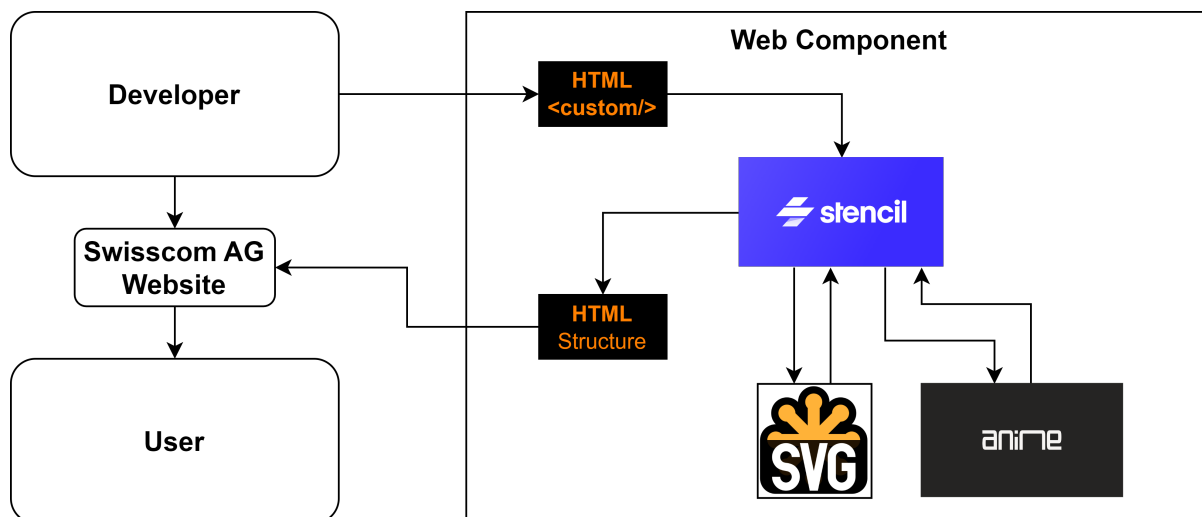


Figure 3.1: High level overview

**HTML custom tag:** The developer can use a defined simple custom tag to create the web component.

**Stencil:** Stencil is used to configure, build and render the web component.

**SVG:** SVG is used to display responsive graphics.

**anime.js:** anime.js is used to animate the graphics.

## 3.2 Domain Model

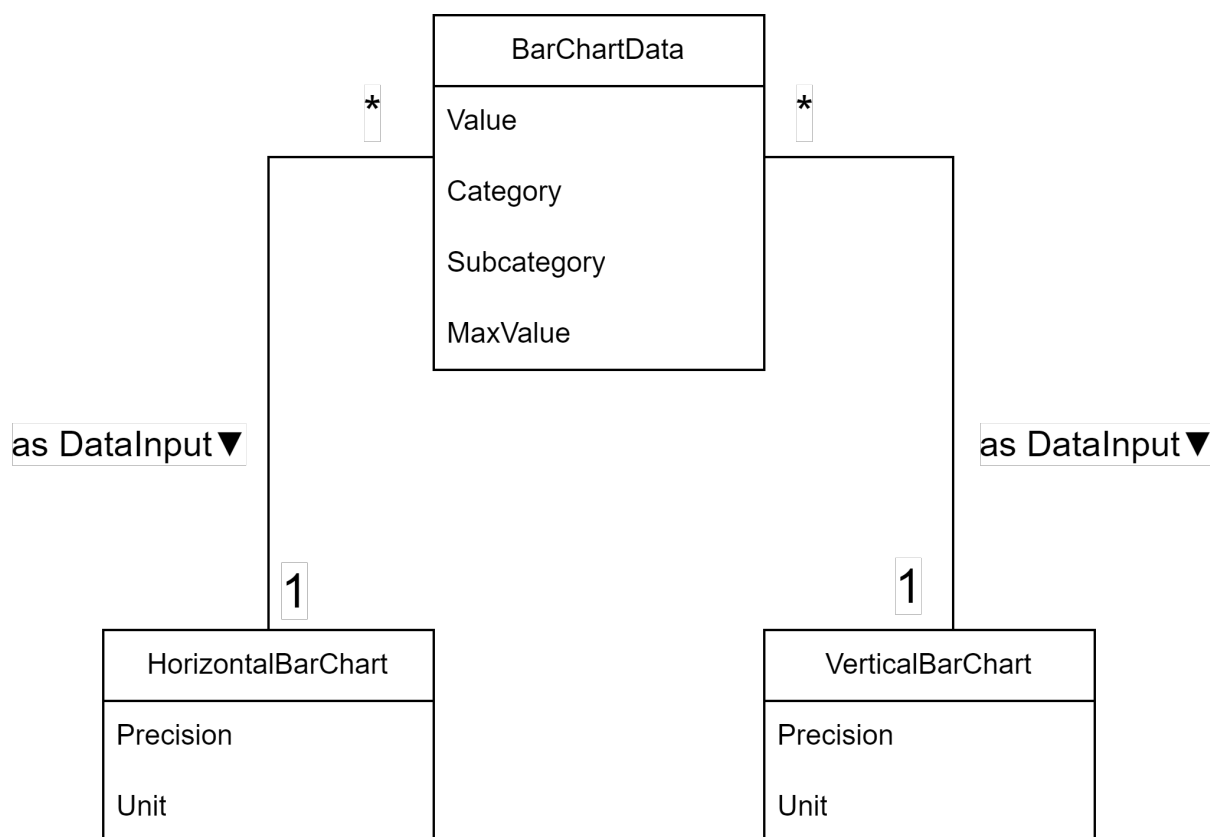


Figure 3.2: Domain model

## 3.3 C4 Model

From an architectural point of view, web components have the inherent qualities of being self-contained and reusable components, establishing themselves as a low-level system. The term "component" itself implies a certain architectural level of abstraction. While web components can be integrated into larger systems, this thesis focuses specifically on web components and their reimplementation. As a result, the architectural level chosen for this analysis is the third level of the C4 model, as the higher two levels would not provide significant additional insight or information.

The web components are designed to be used by developers who are responsible for building their own websites or specific sections within them. These developers integrate the web components into their projects and are able to configure the charts using the parameters provided. The charts themselves rely on data provided by the user to generate and display the appropriate graphs. A further explanation of the components can be found in the [section 3.6](#).

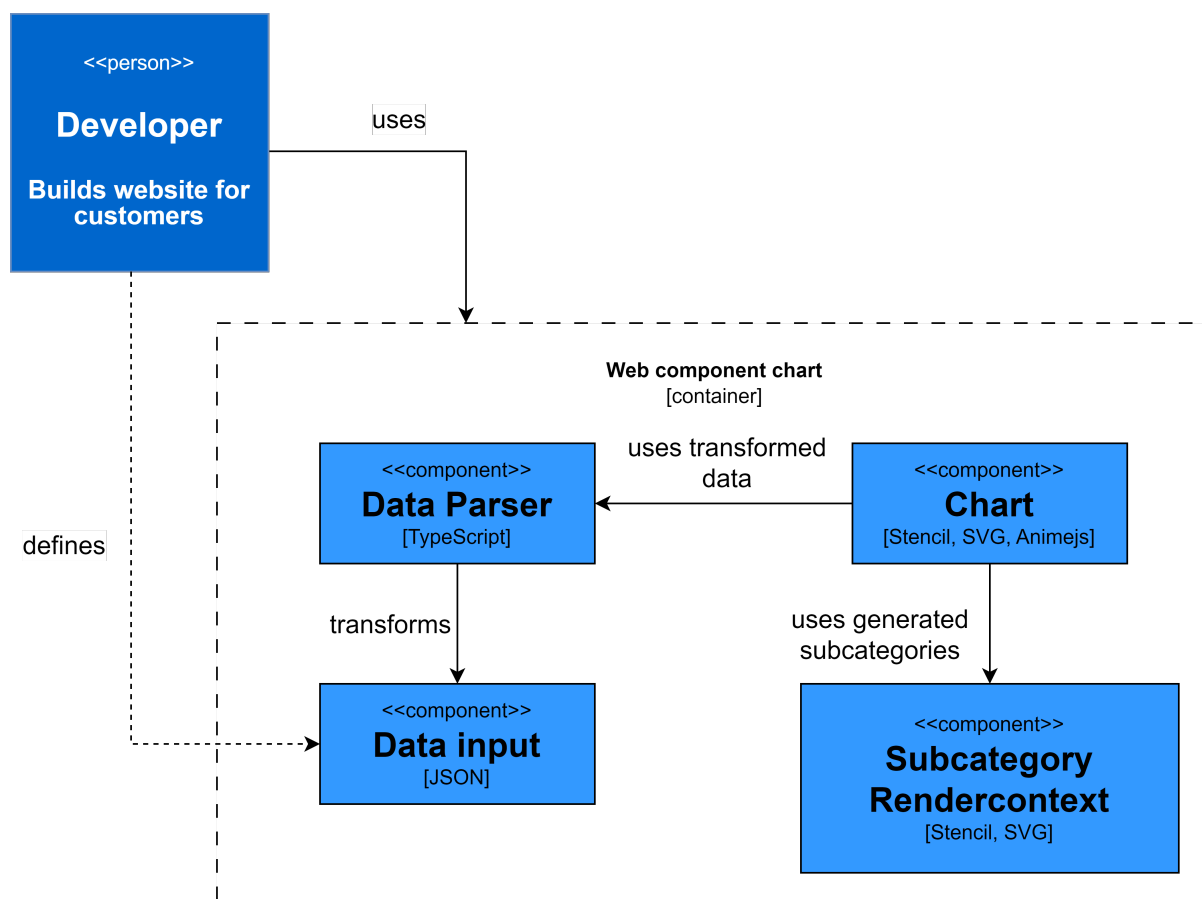


Figure 3.3: C4 model level 3 container

## 3.4 Architectural Constraints

This section provides a detailed explanation of the architectural constraints, which were derived from a number of sources, including the task requirements, evaluation of existing charts and collaborative discussions with Swisscom AG.

### Constraint 1: Use of Stencil

Stencil is a library designed for quick and easy development of web components. It has been used to create the pie chart in the [SDX](#) platform. With Stencil, developers can quickly configure components using annotations for components and props, providing efficient setup and flexibility for further logic integration.

#### Constraint 1.1: Use of the web component standard

The use of the web component standard provides the benefits of encapsulation and interoperability, facilitating the seamless deployment of updates across all instances of the charts. This constraint works in conjunction with Stencil, a purpose-built framework for implementing and deploying web components.

### Constraint 2: Use of SVG

SVG, an XML-based standard for scalable vector graphics, is used to render complex graphics within the web component. It is widely supported by all major web browsers, making it a good choice for consistent, high quality visualizations. The existing pie chart also uses SVG for its graphical representations.

## 3.5 Architectural Decisions

The following architectural decisions are made to enable a modular software application and an easy use for the users and developers. They are explained in detail in the following subsections using the format of the Y-template for architectural decision records. [5]

- [Chart types](#)
- [Data structure](#)
- [Value ranges](#)
- [Other chart decisions](#)
- [Animation](#)
- [Security](#)

### 3.5.1 Chart Types

In the context of the different chart types by name horizontal and vertical chart, facing the need to display data with the same structure it was decided to use both chart types and it was decided against using only one chart type to be able to use the advantages of each chart type for different types of data accepting that some data could be displayed equally with each chart, making one redundant. [6] [7] [8]

#### Advantages of horizontal chart:

- Allows better presentation of long labels
- Better presentation of rankings
- Better comparison over different categories
- Better comparison for proportions between subcategories

#### Advantages of a vertical chart:

- Presenting sequential data (time-range, age-range, ...)
- Comparing negative and positive data
- Suited for histograms
- Suited for trend analysis

### 3.5.2 Data Structure

In the context of the data structure requirement for the bar charts, facing the need to input user data in a simple way similar to the pie chart structure it was decided to use a one-dimensional JSON format and a multidimensional JSON format was neglected to achieve consistent user experience across all charts accepting that a one-dimensional structure can lead to duplicated data.

The naming of the key/value pairs differs from the pie chart data structure to suit the use case of a bar chart, which can have multiple categories with maximum values.

Key/Value structure with example values:

```
1  {
2    "value": 10,
3    "category": "Internet",
4    "subcategory": "Subscription",
5    "maxValue": 100
6  }
```

Listing 3.1: JSON data structure

Explanation of the JSON key/value pairs:

**"value"**: Defines a number that represents the value of the subcategory.

**"category"**: Defines the parent category to which many subcategories can belong.

**"subcategory"**: Defines a specific subcategory to which the value belongs.

**"maxValue"**: Defines the maximum value a category can have.

### 3.5.3 Value Ranges

#### Negative and positive values

In the context of the value ranges, facing the need to display positive and negative values, it was decided to allow negative and positive value ranges within a chart, but to allow only positive or negative values for a category, and it was decided against a combination of positive and negative values for a category to achieve a consistent user experience and visual presentation across all charts accepting that the need to display positive and negative data has to be split across different categories.

#### Limited and infinity maximum

In the context of the value ranges, facing the need to display limited and infinite values, it was decided to allow limited and infinite maxima within a chart, but to allow only limited or infinite maxima for a category, and it was decided against to always require a limited maximum for a category to give the user more fine-grained control over the data and the ability to display different types of data accepting that a combination of infinite maxima and limited maxima may cause some initial confusion in the visual presentation of the charts.

### 3.5.4 Other chart decisions

In the context of the visual presentation of the chart, facing the need to display multiple categories within one chart, it was decided to allow up to 6 categories per chart and up to 3 subcategories per category and not to allow more than this to achieve a balance between having the ability to display many categories and having a good user experience that allows for easy comparison between a few categories accepting that some data cannot be displayed with only one chart.

### 3.5.5 Animation

In the context of displaying the chart, facing the need to animate the SVG bars, it was decided to use an animation library and to neglect a custom implementation with CSS to use a standardized implementation for animation across all charts accepting that some specific animation use cases may not be implemented in the future.

A further analysis of different libraries can be found in the [section 3.8](#).

### 3.5.6 Security

In the context of Content Security Policies (CSP) in web components, facing the need to be compatible with possible future CSP changes of Swisscom AG, it was decided to use only the same technologies as in the pie chart and not to use any other technologies to accommodate the current implementation of the SDX platform used by Swisscom AG accepting that the web component may need to be adapted for future changes.

## 3.6 Chart Component Structure

The chart component is built upon the web component standard and uses the capabilities provided by Stencil for easy creation and configuration. A web component consists of three main elements: custom elements, shadow DOM, and HTML templates [9]. Stencil simplifies the implementation of these elements and provides an API as an abstraction layer for web component development [10].

For the chart component, the following decorators have been defined:

**@Component:** This decorator initializes the chart component and sets its properties.

```
1  Component({
2    tag: "bar-chart",
3    styleUrls: "bar-chart.scss",
4    shadow: true
5  })
```

Listing 3.2: Component decorator

**@Prop:** This decorator defines the data attributes that users can provide to the chart.

```
1  @Prop() public data
2  @Prop() public unit
3  @Prop() public precision
```

Listing 3.3: Prop decorators

**@Watch:** This decorator enables automatic updates when a property changes.

```
1  @Watch("data")
```

Listing 3.4: Watch decorator

The chart component also includes the following lifecycle hooks:

#### **render():**

This hook returns the complete HTML structure of the web component, which is then rendered by the browser.

#### **componentDidLoad():**

After the rendering process is complete, this hook is used to handle the animation.



## 3.7 UX Design

### 3.7.1 UX Design Constraints

The design of the web components follows the design rules outlined on the official Swisscom AG SDX website [11]. The aim is to maintain the overall look and feel of the existing web components, while allowing for improvements to enhance the user experience. The design constraints presented in this section are derived from various sources, ranging from the task requirements to the evaluation of existing charts and collaborative discussions with Swisscom AG.

The following list shows the defined constraints:

- Smallest font size: 14 pixel
- Color scheme: Defined by the accent color palette on the [SDX](#) website. Only the non-"Rollover" colors can be used. Furthermore, the color "gray-tint-8" from the support color scheme can be used for the background bars. In the [subsection 3.7.4](#) the color scheme is analyzed in detail.
- Smallest compatible view: Mobile 360 pixel width

### 3.7.2 UX Design Decisions

The following design decisions are made to ensure a fluid and intuitive user experience. [12]

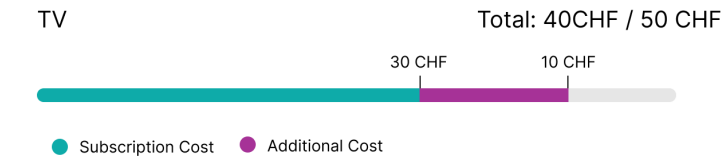
#### Design Decisions

Chart Type	Decision	Reasoning
General	The legend is displayed at the bottom left of each chart.	A legend is needed to map and understand the colors with the subcategories.
General	Each subcategory has a label indicator displaying the value at the end of the bar.	The label is needed so that the user has all the necessary information at a glance. This feature replaces the need for a hover, which was originally implemented in the non-web component charts.
General	The name of the category is at the upper left of each category segment.	The name makes it possible to put the displayed subcategories into context.
General	The total and max value label of the category is at the upper right of each category segment.	The sum and maximum value are helpful to gain a quick overview.
General	The colors have always the same ordering.	This ensures a unified experience across all usages of the charts.
Horizontal	The chart always has a gray background bar.	This ensures an easier comparability between subcategories and categories because it helps with visualizing the maximum length of the bar.
Horizontal	The chart displays negative data from the right side.	This ensures a clear distinction between positive and negative data.
Horizontal	A negative category always has a "0"-label where the start of the bar is.	This helps the user to distinguish easier between positive and negative data.
Horizontal	The chart displays no gray dividing lines anymore.	This ensures a more compact view of the chart.
Vertical	The vertical bar chart displays a gray dot if there is no data.	This ensures a clear communication with the user that there is no data to be shown.
Vertical	The axis are not labeled.	The title should always provide information about what the chart is about.
Vertical	Y-Axis number of lines is automatically derived from the data, but the user has still the option to set one manually.	This ensures a unified experience across all usages of the chart.

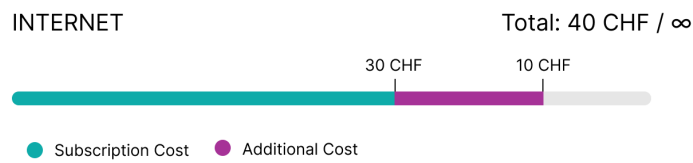
Table 3.1: Design decisions

### 3.7.3 UX Mockups

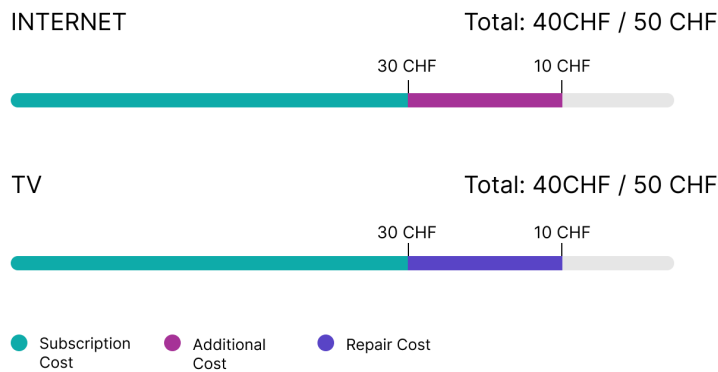
Mockups were chosen as a fundamental element of the design process because they provide a more detailed representation of the user interface than wireframes. They are used to illustrate how the new design incorporates and enhances the existing elements. The images below show the three primary iterations of the horizontal bar chart and one for the vertical bar chart.



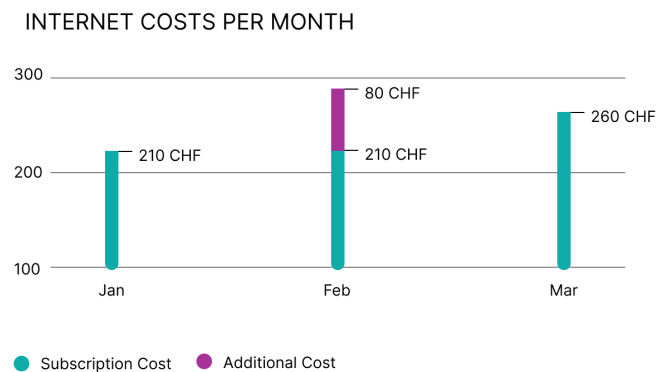
(a) Horizontal bar chart with limit



(b) Horizontal bar chart with no limit



(c) Horizontal bar chart stacked



(d) Vertical bar chart stacked

Figure 3.4: UX mockups

### 3.7.4 UX Accessibility

#### Color scheme

The color scheme defined by SDX is the following:

#### Accent colours

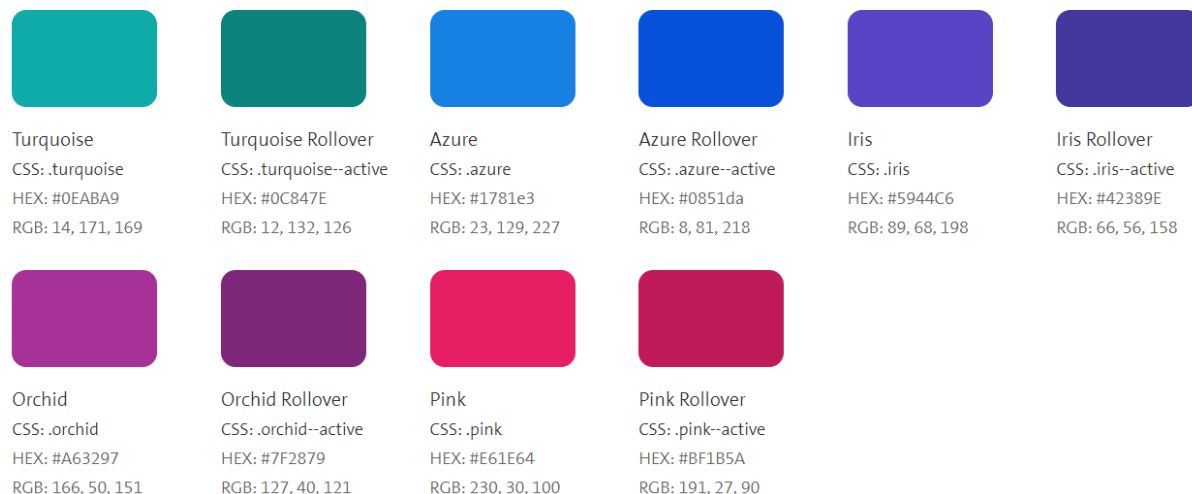


Figure 3.5: Color scheme

The WCAG AA standard (AA stands for the second level of three) requires a minimum contrast ratio of 4.5:1 between the background color and the foreground color, as defined by the Web Content Accessibility Guidelines (WCAG) 2.1. However, the standard does not explicitly address color-to-color contrast between two colors at the same level. Exceptions are made for larger text (at least 18 point or 14 point bold), for which only a 3:1 ratio is required. Incidental elements and logotypes are exempt from contrast requirements. More detailed information on minimum contrast requirements can be found in the WCAG 2.1 documentation<sup>1</sup>.

#### Contrast analysis

To analyze color combinations, it is important to designate one color as the background color and another as the foreground or text color. The contrast checker tool available at <https://webaim.org/resources/contrastchecker/> is used to perform the analysis. Note that the contrast ratio remains the same regardless of which color is used as the background or foreground color.

Background color	Front/Text color	Contrast ratio	Result
Turquoise	Azure	1.4:1	Failed
Turquoise	Iris	2.41:1	Failed
Turquoise	Orchid	2.1:1	Failed
Turquoise	Pink	1.56:1	Failed
Azure	Iris	1.72:1	Failed
Azure	Orchid	1.49:1	Failed
Azure	Pink	1.11:1	Failed
Iris	Orchid	1.14:1	Failed
Iris	Pink	1.54:1	Failed
Orchid	Pink	1.34:1	Failed

Table 3.2: Contrasts

<sup>1</sup><https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html>

**Decision**

None of the combinations meet the requirements of the AA standard. However, the color scheme is a constraint and the colors won't be changed. It is recommended that the scheme be revised in the future to meet the requirements of the AA standard.

**3.8 Libraries & Frameworks****3.8.1 Utility Analysis****Web Component Libraries**

Library	Version	Usage Status
Stencil	3.2.1	in-use
Reference	<a href="https://stenciljs.com">https://stenciljs.com</a>	
License	MIT License	
Description		
Stencil is a compiler library which allows the generation of web components by integrating the most effective concepts from popular frameworks. TypeScript, JSX and CSS are leveraged to build web components according to the best practices. [13]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Integration support for many popular libraries</li> <li>• Extensive documentation</li> <li>• Available tutorials</li> <li>• Active community</li> <li>• JSX is supported</li> <li>• Usage of the web component standard</li> <li>• Backed by a company</li> </ul>		<ul style="list-style-type: none"> <li>• Very new (First release 2019)</li> <li>• No specialist literature available yet</li> <li>• Version update breaking changes</li> </ul>

Table 3.3: Utility analysis Stencil

Library	Version	Usage Status
slim.js	5.0.10	rejected
Reference	<a href="https://slimjs.com">https://slimjs.com</a>	
License	MIT License	
Description		
Slim.js is an extensible small web component library. Using the web component standards the library is fast and resourceful. With a declarative style approach the code remains maintainable and easy to read. [14]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Flexible integration options</li> <li>• Very lightweight</li> <li>• Fast and resourceful</li> <li>• Custom implementation extensions</li> </ul>		<ul style="list-style-type: none"> <li>• Small community</li> <li>• Not widely used</li> <li>• Inactive development</li> <li>• Lacking documentation</li> </ul>

Table 3.4: Utility analysis slim.js

Library	Version	Usage Status
Polymer	3.5.1	rejected
Reference	<a href="https://polymer-library.polymer-project.org">https://polymer-library.polymer-project.org</a>	
License	<a href="#">BSD-3-Clause License</a>	
Description		
Allows construction of reusable and encapsulated custom HTML elements. Is meant for building web applications and known for its effortless import and usage capabilities. A powerful developer utility backed by the company Google. [15]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Guides available</li> <li>• Extensive documentation</li> <li>• Easy integration and usage</li> <li>• Active community</li> <li>• Many possibilities</li> <li>• Backed by a tech giant</li> </ul>		<ul style="list-style-type: none"> <li>• Deprecated (Superseded by Lit (<a href="#">Table 3.6</a>))</li> <li>• Probably end of life</li> <li>• Rather high learning curve</li> </ul>

Table 3.5: Utility analysis Polymer

Library	Version	Usage Status
Lit	2.7.2	rejected
Reference	<a href="https://lit.dev">https://lit.dev</a>	
License	<a href="#">BSD-3-Clause License</a>	
Description		
The successor of the Polymer ( <a href="#">Table 3.5</a> ) library. Making use of custom elements, scoped styles, reactive properties and declarative templates. Avoiding boilerplate code and a big footprint with thought through features and library size. [16]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Extensive documentation</li> <li>• Tutorials available</li> <li>• Active community</li> <li>• Field tested</li> <li>• Backed by Google</li> <li>• Lightweight</li> </ul>		<ul style="list-style-type: none"> <li>• Learning curve</li> <li>• No specialist literature available yet</li> </ul>

Table 3.6: Utility analysis Lit

### Testing Libraries

Library	Version	Usage Status
Jest	29.5.0	in-use
Reference	<a href="https://jestjs.io">https://jestjs.io</a>	
License	<a href="#">MIT License</a>	
Description		
Jest is JavaScript testing framework that works out of the box. No complex configuration is needed. Simplicity is the main focus. It supports common industry technologies. [17]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• TypeScript compatible</li> <li>• Industry standard</li> <li>• Good documentation</li> <li>• Simple configuration</li> <li>• No additional dependencies needed</li> </ul>		<ul style="list-style-type: none"> <li>• Learning curve</li> <li>• Immature support for external libraries</li> </ul>

Table 3.7: Utility analysis Jest

Library	Version	Usage Status
Puppeteer	19.11.1	in-use
Reference	<a href="https://pptr.dev">https://pptr.dev</a>	
License	<a href="#">Apache-2.0 License</a>	
Description		
A Node.js library for end-to-end testing with Google Chrome. Puppeteer provides most manual available functionalities of a browser as an API. [18]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Big community</li> <li>• Backed by a company</li> <li>• Performant</li> <li>• Easy integration</li> </ul>		<ul style="list-style-type: none"> <li>• Browser limitation</li> <li>• Not very mature library</li> </ul>

Table 3.8: Utility analysis Puppeteer

Library	Version	Usage Status
Mocha	10.2.0	rejected
Reference	<a href="https://mochajs.org">https://mochajs.org</a>	
License	<a href="#">MIT License</a>	
Description		
A feature rich JavaScript testing framework running on Node.js. Allows asynchronous testing and runs tests in a serial mode. The code is maintained by an open source community. [19]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Simple usage</li> <li>• Asynchronous testing</li> <li>• Fast</li> <li>• Flexible with library integrations</li> </ul>		<ul style="list-style-type: none"> <li>• Complex configuration setup</li> <li>• Needs additional dependencies</li> <li>• No mock, stub, spy function</li> </ul>

Table 3.9: Utility analysis Mocha

Library	Version	Usage Status
Sinon.JS	15.0.4	rejected
Reference	<a href="https://sinonjs.org">https://sinonjs.org</a>	
License	<a href="#">BSD-3-Clause License</a>	
Description		
A standalone library providing stubs, spies and mocks for JavaScript. Can be integrated into any testing framework. [20]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Easy to use</li> <li>• Works with any testing framework</li> <li>• Available guides</li> </ul>		<ul style="list-style-type: none"> <li>• Needs a testing framework</li> <li>• Limited functionality</li> <li>• Rather inactive community</li> </ul>

Table 3.10: Utility analysis Sinon.JS



Library	Version	Usage Status
Chai	4.3.7	rejected
Reference	<a href="https://www.chaijs.com">https://www.chaijs.com</a>	
License	MIT License	
Description		
Chai is a simple library for Node.js. It allows better assertions and can be paired with any JavaScript testing framework. [21]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Easy usage</li> <li>• Very lightweight</li> <li>• Many available plugins</li> <li>• Clear documentation</li> </ul>		<ul style="list-style-type: none"> <li>• Needs a testing framework</li> <li>• Limited functionality</li> <li>• Small community</li> </ul>

Table 3.11: Utility analysis Chai

### Animation Libraries

Library	Version	Usage Status
anime.js	3.2.1	in-use
Reference	<a href="https://animejs.com">https://animejs.com</a>	
License	MIT License	
Description		
A lightweight JavaScript animation library with an easy-to-use API. Supports common JavaScript functionalities and vector graphics. [22]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Detailed documentation</li> <li>• Big community</li> <li>• Good browser compatibility</li> <li>• Minimizes CSS code</li> <li>• Provides template animations</li> </ul>		<ul style="list-style-type: none"> <li>• Inactive code repository</li> <li>• Unpractical documentation</li> <li>• Learning curve</li> </ul>

Table 3.12: Utility analysis anime.js

Library	Version	Usage Status
Animate.css	4.1.1	rejected
Reference	<a href="https://animate.style">https://animate.style</a>	
License	<a href="#">Hippocratic License</a>	
Description		
Animate.css is a library which provides basic animations for HTML elements. Animations are controlled via the class attribute. The library is published under an open source license. [23]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Big community</li> <li>• Accessibility support</li> <li>• Reduces styling code</li> </ul>		<ul style="list-style-type: none"> <li>• Limited functionality</li> <li>• No animation of vector graphics</li> <li>• Limited documentation</li> <li>• Version breaking changes</li> </ul>

Table 3.13: Utility analysis Animate.css

### Styling Libraries

Library	Version	Usage Status
Sass	1.62.0	in-use
Reference	<a href="https://sass-lang.com">https://sass-lang.com</a>	
License	<a href="#">MIT License</a>	
Description		
Sass is an extension library for native CSS. It provides new functionalities for efficient coding and avoids boilerplate code. [24]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Good documentation</li> <li>• Big community</li> <li>• Well maintained</li> <li>• Many features</li> <li>• No compatibility problems</li> </ul>		<ul style="list-style-type: none"> <li>• Learning curve</li> <li>• Slower compiling</li> </ul>

Table 3.14: Utility analysis Sass

Library	Version	Usage Status
Stylus	0.59-0	rejected
Reference	<a href="https://stylus-lang.com">https://stylus-lang.com</a>	
License	<a href="#">MIT License</a>	
Description		
A dynamic CSS preprocessor language. It generates native CSS code using the loose stylus syntax. [25]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Compact code</li> <li>• Cross browser compatibility</li> <li>• Loose syntax restrictions</li> </ul>		<ul style="list-style-type: none"> <li>• Learning curve</li> <li>• Difficult to debug</li> <li>• Hard to read</li> <li>• Small community</li> </ul>

Table 3.15: Utility analysis Stylus

Library	Version	Usage Status
Less	4.1.3	rejected
Reference	<a href="https://lesscss.org">https://lesscss.org</a>	
License	<a href="#">Apache-2.0 License</a>	
Description		
A backwards compatible extension tool for CSS. Less converts styles to native CSS and provides a whole new set of features. [26]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Cross browser compatibility</li> <li>• New styling features</li> <li>• Reuse code easily</li> </ul>		<ul style="list-style-type: none"> <li>• Learning curve</li> <li>• Provides fewer framework integration</li> <li>• Debugging can be difficult</li> </ul>

Table 3.16: Utility analysis Less

### Linting Libraries

Library	Version	Usage Status
ESLint	8.39.0	in-use
Reference	<a href="https://eslint.org">https://eslint.org</a>	
License	<a href="#">MIT License</a>	
Description		
A static code analysis tool for JavaScript supported by many text editors. With custom configuration options and an automatic fixing feature, it is definitely an industry standard. [27]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Custom configuration</li> <li>• Big community</li> <li>• Good documentation</li> <li>• Auto fix feature</li> <li>• Extensible with plugins</li> </ul>		<ul style="list-style-type: none"> <li>• Setup is bothersome</li> <li>• Requires more computing power</li> </ul>

Table 3.17: Utility analysis ESLint

Library	Version	Usage Status
JSLint	2023.1.29	rejected
Reference	<a href="https://www.jshint.com">https://www.jshint.com</a>	
License	<a href="#">Unlicense License</a>	
Description		
JSLint is a small code quality and code coverage tool for JavaScript. By scanning an existing source, semantic and syntactic imperfections can be corrected. [28]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Lightweight</li> <li>• Easy setup</li> <li>• Active maintenance</li> <li>• Available editor plugins</li> </ul>		<ul style="list-style-type: none"> <li>• Minimal documentation</li> <li>• Small community</li> </ul>

Table 3.18: Utility analysis JSLint

**Formatting Libraries**

Library	Version	Usage Status
Prettier	2.8.8	in-use
Reference	<a href="https://prettier.io">https://prettier.io</a>	
License	<a href="#">MIT License</a>	
Description		
Prettier is a code formatter for multiple coding languages. It integrates into most editors seamlessly. Code style can be automated. [29]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Big community</li> <li>• Active maintenance</li> <li>• Good documentation</li> <li>• Multiple language support</li> <li>• Editor integration</li> <li>• Identical code style</li> </ul>		<ul style="list-style-type: none"> <li>• Setup can be tedious</li> <li>• Many configuration options</li> </ul>

Table 3.19: Utility analysis Prettier

Library	Version	Usage Status
js-beautify	1.14.6	rejected
Reference	<a href="https://beautifier.io">https://beautifier.io</a>	
License	<a href="#">MIT License</a>	
Description		
A reformatting tool for JavaScript code. Made for stylistic code and uniform indenting. Implemented by developers for developers within the web community. [30]		
Advantages		Disadvantages
<ul style="list-style-type: none"> <li>• Active maintenance</li> <li>• Easy setup</li> <li>• Lightweight</li> <li>• Performant</li> </ul>		<ul style="list-style-type: none"> <li>• Limited documentation</li> <li>• Limited language support</li> <li>• Small community</li> </ul>

Table 3.20: Utility analysis js-beautify

### 3.8.2 Selection & Alternatives

#### Web Component

Selection	Alternatives
<ul style="list-style-type: none"> <li>• <a href="#">Stencil</a></li> </ul>	<ol style="list-style-type: none"> <li>1. <a href="#">Lit</a></li> <li>2. <a href="#">Polymer</a></li> <li>3. <a href="#">slim.js</a></li> </ol>

Table 3.21: Web component technology evaluation

The Stencil library has been selected in regard to the architectural constraints ([section 3.4](#)). This technology acts as the core of the whole system. Other selected technologies depend on this library and are directly integrated into its development process.

#### Testing

Selection	Alternatives
<ul style="list-style-type: none"> <li>• <a href="#">Jest</a></li> <li>• <a href="#">Puppeteer</a></li> </ul>	A combination of the following technologies: <ul style="list-style-type: none"> <li>• <a href="#">Mocha</a></li> <li>• <a href="#">Sinon.JS</a></li> <li>• <a href="#">Chai</a></li> </ul>

Table 3.22: Testing technology evaluation

Jest and Puppeteer were selected due to their good integration with the Stencil library. Moreover, both technologies are needed to fulfill the non-functional requirement [NFR-M03](#). Jest specializes in unit tests and Puppeteer focuses on end-to-end testing.

#### Animation

Selection	Alternatives
<ul style="list-style-type: none"> <li>• <a href="#">anime.js</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">Animate.css</a></li> </ul>

Table 3.23: Animation technology evaluation

The selected library anime.js was already used by the legacy chart components for animations. In the newer pie chart implementation, anime.js is still utilized. The usage of the library is not essential, but it simplifies the implementation of the requirement [GNL-US8](#).

#### Styling

Selection	Alternatives
<ul style="list-style-type: none"> <li>• <a href="#">Sass</a></li> </ul>	<ol style="list-style-type: none"> <li>1. <a href="#">Stylus</a></li> <li>2. <a href="#">Less</a></li> </ol>

Table 3.24: Styling technology evaluation

Sass ships by default with Stencil and integrates seamlessly with other libraries. The library was selected to avoid boilerplate code and make the produced code more maintainable. Those aspects are important to successfully accomplish [NFR-M01](#). Further it can be seen as an indirect architectural constraint ([section 3.4](#)) due to the fact that Stencil uses Sass by default.

### Linting

Selection	Alternatives
<ul style="list-style-type: none"><li>• <a href="#">ESLint</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">JSLint</a></li></ul>

Table 3.25: Linting technology evaluation

To identify code inconsistencies and enforce a uniform syntax ESLint was selected. The stakeholder provided a code linter configuration for ESLint. This made the library more advantageous to use. The library will facilitate fulfillment of [NFR-M01](#) and [NFR-M02](#).

### Formatting

Selection	Alternatives
<ul style="list-style-type: none"><li>• <a href="#">Prettier</a></li></ul>	<ul style="list-style-type: none"><li>• <a href="#">js-beautify</a></li></ul>

Table 3.26: Formatting technology evaluation

Prettier allows to reliably format code in an automated procedure. The stakeholder provided existing Prettier configuration. Similar to the linting technologies, this library will ease the fulfillment of [NFR-M01](#) and [NFR-M02](#).

## 3.9 Technologies

Category	Technologies
Runtime environment	<ul style="list-style-type: none"><li>• Node.js</li></ul>
Languages	<ul style="list-style-type: none"><li>• CSS</li><li>• HTML</li><li>• JavaScript</li><li>• JSON</li><li>• SVG</li></ul>
Libraries	<ul style="list-style-type: none"><li>• <a href="#">anime.js</a></li><li>• <a href="#">ESLint</a></li><li>• <a href="#">Jest</a></li><li>• <a href="#">Prettier</a></li><li>• <a href="#">Sass</a></li><li>• <a href="#">Stencil</a></li><li>• <a href="#">TypeScript</a></li></ul>

Table 3.27: Technologies

## 3.10 Prototype

### 3.10.1 Realization

The prototype implementation is available as a separate branch named "prototype" in the project git repository<sup>1</sup>.

Establishing a proof of concept for a chart web component is done with a prototype implementation. A simplified representation of a horizontal bar chart must be realized to proof the feasibility of the evaluated technologies and implementation ideas. Laying focus on the following aspects for the prototype application:

- Using the assessed technologies
- Functionality and simplicity over aesthetics
- SVG incorporation
- Test animation possibilities

The primary objective of the prototype is to identify potential complications and explore implementation alternatives. Moreover, an exercise scenario is established to get the development team accommodated with the evaluated technologies. It is important to note that the resulting prototype application will not be used as the underlying basis for subsequent versions of the implementation. Nonetheless, it can serve as a reference and source of inspiration. The prototype emphasizes acclimating the development team to the technologies and demonstrating the ability to visualize chart elements using animated SVG utilities.

<sup>1</sup><https://gitlab.ost.ch/florian.blum/chart-components-code/-/tree/prototype>



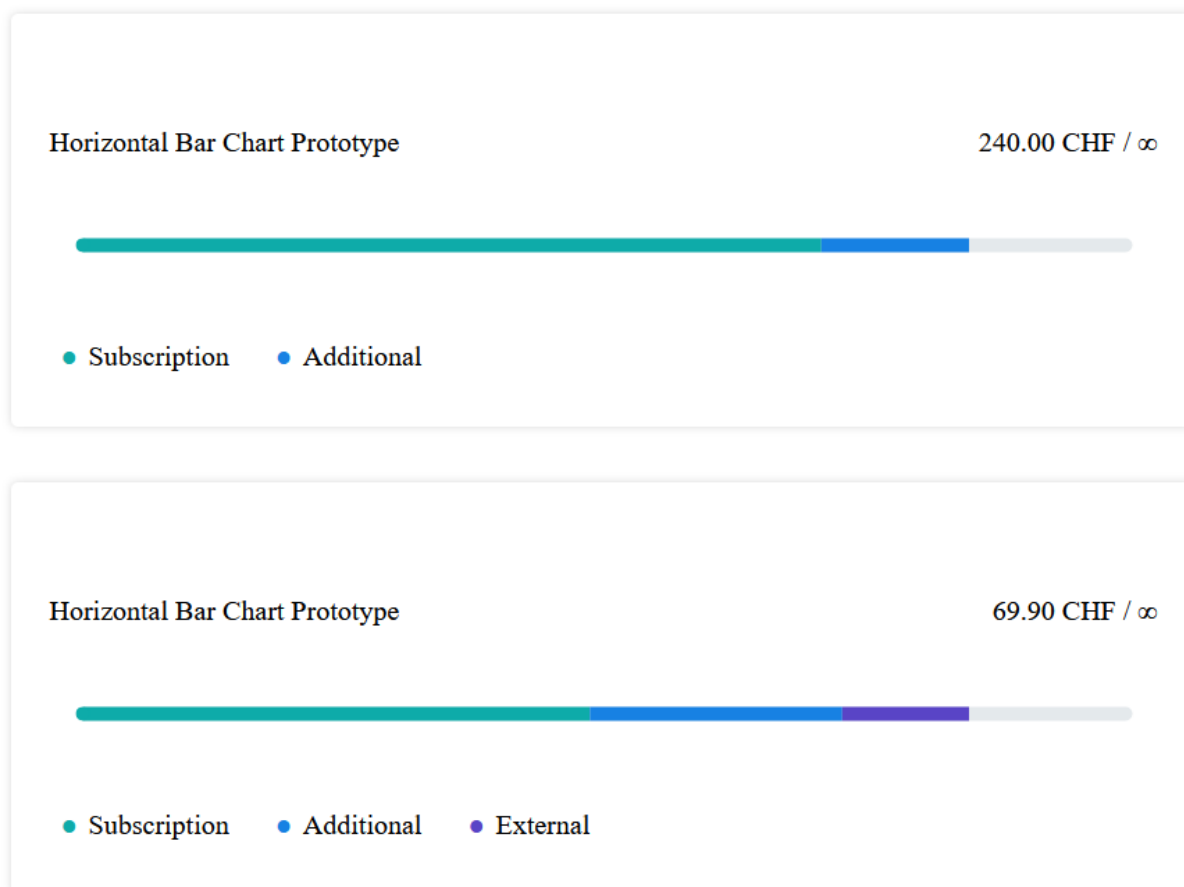


Figure 3.6: Prototype overview

The developed prototype implementation enables rendering of a single chart in the form of a web component. Each chart consists of a title text, a value tracker, a stacked bar visualization and a legend. The rendering of the stacked bar utilizes available SVG tools. Positioning and box sizing are primarily achieved through the utilization of the CSS flexbox property. Animations are applied to every legend item and each segment of the stacked bar visualization. During the initial render cycle, the animations play in a left-to-right manner.

Due to the limited color options, the implementation supports only up to five stacked bars and consequently only five types in the legend. The bar scaling will always reach 80% of the bar length regardless of the set values. However, this can be disregarded as the prototype does not incorporate any logic pertaining upper limits.

### 3.10.2 Issues & Complications

Finding potential problem factors is the primary goal of the prototype. In this case, the issues arise from the utilization of the animation library [anime.js](#) and the incorporation of SVG utilities within the web component. All of the problems are summarized in the table below.

Issue / Complication	Description
Animation of SVG	The animation library <a href="#">anime.js</a> fails to correctly animate an SVG line or path element if the property <code>vector-effect="non-scaling-stroke"</code> is set.
SVG scaling	The SVG bar elements are to be scaled only on the x-axis. This proves to be difficult due to the fact that SVG is made to naturally scale both on x and y-axis.
SVG box sizing	The SVG path element appears bigger than defined in most browsers. This can create additional margins which are unwanted. The <a href="#">Figure 3.7</a> displays the problem.
SVG marker	Defined markers in SVG are not rendered the same in all browsers. The <a href="#">Figure 3.8</a> of a render in Google Chrome depicts that the markers are displayed incorrectly.
SVG viewBox	The SVG property <code>viewBox</code> interprets units differently from browser to browser when combined with a <code>vector-effect</code> and the animation library <a href="#">anime.js</a> .

Table 3.28: Prototype problem factors

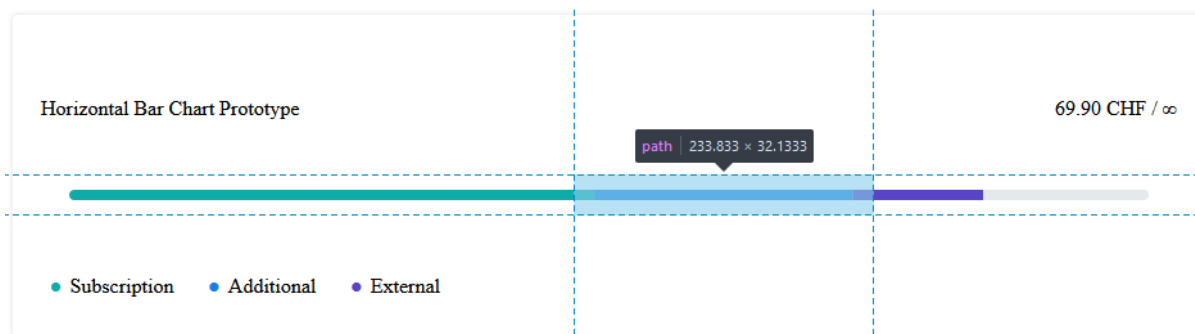


Figure 3.7: Prototype SVG box

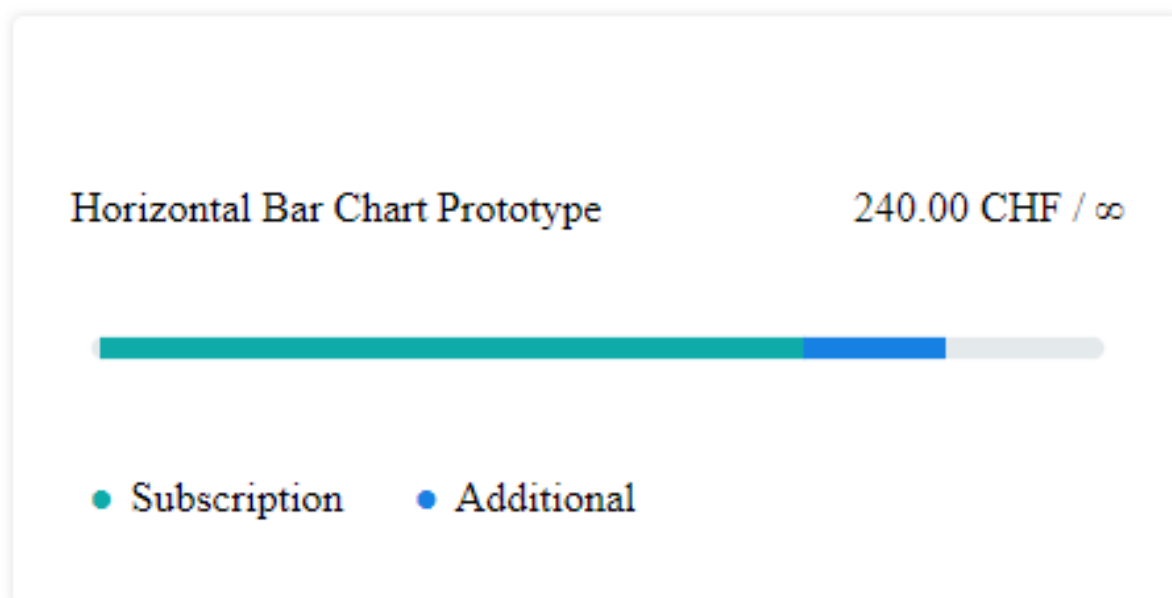


Figure 3.8: Prototype marker in Chrome

### 3.10.3 Learnings

After the completion of the prototype, the following aspects are to be considered for subsequent versions of the application.

- SVG is inconsistent over multiple web browsers.
- SVG markers are not a viable solution in terms of compatibility.
- SVG vector-effects break the animation library calculations.
- SVG viewBox combined with vector-effect can lead to unexpected behavior.
- SVG animated by [anime.js](#) breaks if a vector-effect is set.

In conclusion, it is recommended to minimize the use of the named SVG attributes and to exercise caution when using [anime.js](#). It is important to read the documentation of the animation library before implementing the methods it provides, to avoid inadvertently breaking the SVG metrics calculations.

Furthermore, the usage of CSS flexbox<sup>1</sup> is deemed to produce more complex code. Given that the web component does not heavily depend on a responsive layout, the usage of CSS grid<sup>2</sup> can be beneficial due to the components static layout.

<sup>1</sup><https://css-tricks.com/snippets/css/a-guide-to-flexbox>

<sup>2</sup><https://css-tricks.com/snippets/css/complete-guide-grid>

## 3.11 Patterns & Standards

While the current implementation does not directly use software design patterns, it is built as a web component using the [Stencil](#) library. Stencil allows for pattern integration and adherence to common web standards. The implementation implicitly incorporates the following design patterns and web standards:

- Shadow DOM
- Custom elements
- HTML template

By using Stencil, these concepts are applied seamlessly, resulting in a robust and compliant web component.

The shadow DOM is a feature that allows a component to have its own isolated DOM tree that is rendered independently of the main DOM. This isolation ensures that the component's structure, styles, and behavior remain encapsulated and do not interfere with other components or the surrounding environment. By using the shadow DOM, components can have localized styles and encapsulated functionality, enhancing modularity and preventing unintended style conflicts or interactions with other elements on the page.

Custom elements in HTML allow for the definition and usage of new element options within the markup. These custom elements can encapsulate their own functionality and behavior. Typically, a custom element represents a composition of other HTML tags, forming a distinct and reusable web page feature with its own set of properties and behaviors. Custom elements provide a powerful mechanism for creating modular and reusable components within HTML-based applications.

HTML templates provide the ability to define content that is not immediately rendered in the DOM, but can be referenced from other sources. When parsing HTML, the browser ignores the content within the template element and only checks for syntax errors. By using HTML templates, developers can reduce the amount of boilerplate code and avoid unnecessary repetition of markup. These templates provide a dynamic and flexible approach to defining reusable content that can be cloned, manipulated, and inserted into the DOM as needed. [31] [9]

# Quality Measures

## 4.1 Coding Guidelines

This section presents the coding guidelines that have been followed in the development of the web bar chart components. The coding guidelines provide a set of standards and best practices that ensure code readability, maintainability and adherence to Swisscom AG standards. The motto of the Swisscom AG is "No code is best code".

### 4.1.1 Clean Code

Clean code is an essential software development principle that focuses on writing code that is easy to read, understand and maintain. It promotes code readability, reduces complexity and improves collaboration between developers. The following is a list of the main practices used:

- Clear and concise naming of classes, methods, variables and files
- Modularity and single responsibility of units
- Comments and documentation within the code only if necessary
- Similar coding structure across charts
- Avoiding code smells

### 4.1.2 Version Control

The Git version control system is used to ensure collaborative development, code history management and continuous integration. The platform <https://gitlab.ost.ch> is used to host the repository.

### 4.1.3 Folder structure

The folder structure is divided into the chart components and utils. The chart component folder contains all the logic, behavior and tests that are directly related to the chart. The utils folder contains helper functions and logic that can be used by every chart.

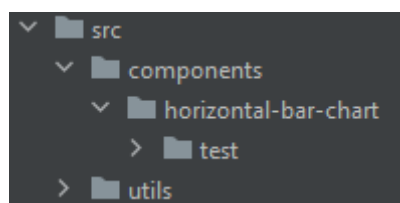


Figure 4.1: Folder structure

### 4.1.4 Continuous integration and delivery

#### CI/CD documentation:

This pipeline consist of one stage "build\_doc" which is responsible for building a PDF from the LaTeX source code.

**CI code:**

The code is not built by the Git repository, but has to be built locally by the developer in the process of fulfilling the Definition of Done for each task to ensure the code is functional. The repository then is only used as a version control system and backup system for the code for collaborative development.

**4.1.5 Workflow Feature Implementation**

Each requirement is broken down into small, implementable user stories. A new branch is created for each user story. Once the implementation is complete and any merge conflicts have been resolved, a merge request is created and a reviewer must check that the Definition of Done is met. If the reviewer approves, the branch is merged into the main branch and the user story is closed.

**4.2 UX Guidelines**

The guidelines are based on the guidelines and design principles of the [SDX](#) platform. More information can be found at <https://sdx.swisscom.ch/principles.html>.

**Principle 1: Visual**

The visual presentation is light, with a purposeful visual focus that feels understated.

**Principle 2: Interaction**

The interaction should feel rewarding, effortless and stimulate discovery.

**Principle 3: Motion**

The motion displayed should make the interaction feel more meaningful and authentic.

**4.3 Definition of Done**

The Definition of Done ([DoD](#)) ensures that the software product will be of high quality. The [DoD](#) applies to every user stories and has to be verified upon completion. A user story has to meet the following criteria:

- The functional requirement is met.
- The code compiles.
- Testing of the user story and/or overarching epic is defined and fulfilled.
- Coding guideline and best practices are fulfilled.
- The task has been reviewed by another developer.

**4.4 Code Metrics****4.4.1 Test Coverage**

The whole data processing logic is tested and covered with unit testing. The other classes and function are tested through manual testing as explained in [section 4.5](#).

**All files**

100% Statements 92/92 100% Branches 28/28 100% Functions 22/22 100% Lines 87/87

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File ▲	Statements ⇅	Branches ⇅	Functions ⇅	Lines ⇅
<a href="#">components/horizontal-bar-chart</a>	100%	87/87	100%	28/28
<a href="#">utils</a>	100%	5/5	100%	0/0

Figure 4.2: Test coverage report

#### 4.4.2 Lines of Code

The JSON parser is not included in this list as it was provided by Swisscom AG.

File	Line Count
bar-chart-data-parser.spec.ts	479
bar-chart-data-parser.tsx	201
horizontal-bar-chart.tsx	429
subcategory-render-context.tsx	309
svg-config.ts	21
color-type.ts	23
data-point.ts	6
horizontal-bar-chart.scss	103
utils.ts	15
<b>Sum</b>	<b>1586</b>
<b>Sum without testing</b>	<b>1107</b>

Table 4.1: Line count

#### 4.4.3 Class Count

The following classes exist:

- "BarChartDataParser"
- "HorizontalBarChart"
- "SubcategoryRenderContext"

**Sum: 3**

## 4.5 Test Concept

Testing is an important step in the software development process to ensure functional and quality needs are met. The following sections list various testing approaches to ensure that functional and non-functional requirements have been fulfilled.

### 4.5.1 Manual System Testing

These system tests aim to validate the full functionality of the web component against the functional and non-functional requirements. System testing and end-to-end testing are very similar in this context, because to perform a manual system test, the component is tested in the same way that a user would use the component. Test cases are deliberately kept small to focus on specific edge cases and different behaviors. All test scenarios are always validated in all four browsers (Chrome, Edge, Firefox, Safari). If a browser has a special/unexpected result, it is noted separately. If no special action is taken after a test, it means that the expected result is the same as the actual result. The results can be found in the [Appendix C](#).

#### Error behavior

No.	Starting Position	Expected Result
1	A category has a negative value and a positive maximum defined.	An error message is displayed instead of the chart showing that minus and positive cannot be mixed in a category.
2	A category has a positive value and a negative maximum defined.	An error message is displayed instead of the chart showing that minus and positive cannot be mixed in a category.
3	A category has a positive value and a negative value defined.	An error message is displayed instead of the chart showing that minus and positive cannot be mixed in a category.
4	More than 6 different categories are defined in the input data.	An error message is displayed instead of the chart showing that the maximum limit of categories has been exceeded.
5	More than 5 different types are defined in the input data.	An error message is displayed instead of the chart showing that the maximum limit of types has been exceeded.
6	More than 3 different types are defined for a single category.	An error message is displayed instead of the chart showing that the maximum limit of types for a single category has been exceeded.

Table 4.2: Error behavior

#### Missing attribute and zero values behavior

No.	Starting Position	Expected Result
1	The value for a type is set to zero.	The type will not be displayed.
2	All types of a category are zero.	A zero label at the beginning of the bar indicates the value zero.
3	The maximum is set to zero.	The maximum for this category is set to zero.
4	One or all attributes are missing inside the data input JSON.	An error message is displayed showing that the data input isn't correct.
5	The category and the type attributes are set to empty strings.	The categories and types are rendered with no labels.
6	The value attribute is not set.	An error message is displayed showing that the data input isn't correct.

Table 4.3: Missing attributes and zero values



**Maximum and infinity behavior**

No.	Starting Position	Expected Result
1	No maximum is not set.	The maximum for this category is set to infinity.
2	The maximum is set to no-number type value (A string for this case).	An error message is displayed showing that the data input is not correct.
3	The total sum of the category exceeds the maximum.	The bar takes up all the space but does not overflow.
4	The total sum is the same or higher(lower if negative maximum) than the maximum.	The end of the bar is rounded.
5	The category is an infinity category.	The bar is always 80% full.

Table 4.4: Maximum and infinity behavior

**Negative behavior**

No.	Starting Position	Expected Result
1	The category has negative values.	The bar begins to draw on the right side.
2	The category has negative values.	The bar has always a zero label on the right side.
3	The category has negative values.	The labels have a minus sign.

Table 4.5: Negative behavior

**Other**

No.	Starting Position	Expected Result
1	A category has two entries of the same type.	The values get summed up and will be displayed as one type.
2	The data is changed ad-hoc.	The chart will update without the need to reload the site.
3	The chart is being displayed by accessing a website that renders it.	The animation is played correctly for each category regarding negative/positive/time/speed in relation to each other.
4	The chart is getting resized.	The bar and labels are responsive and are not overlapping.
5	The chart is being displayed by accessing a website that renders it.	The types and colors match for each category and are displayed in the correct order.
6	The chart is being displayed by accessing a website that renders it.	The legend and colors match and the legend is animated.
7	The chart is getting resized.	Legend labels do not wrap around but the title and total label wraps around if they are too long
8	The chart is being displayed by accessing a website that renders it.	Each label is capitalized at the beginning.
9	The chart is being displayed by accessing a website that renders it.	Labelling is correct for all types and for full and empty bars.

Table 4.6: Other

**Normal behavior**

No.	Starting Position	Expected Result
1	A chart is initialized with two categories and four different types (2 types per category).	The chart shows two segments with two bars each being separated into two sub bars for the two types. The legend shows the four types with the correct colors.
2	A chart is initialized with one category and one type.	The chart shows one segment with one bar. The legend shows the one type.
3	A chart is initialized with six categories and five different types.	The chart shows six segments with six bars. The bars are separated into sub bars according to the category types. The legend shows the five types with the correct colors.
4	A chart is initialized with two categories(one with a maximum value the other is infinity) and one type. The categories both have the same total sum value. The maximum for the first chart equals the sum.	The chart shows two segments each with one bar. Both have the same color. The infinity category bar is only 80% full. The other bar is full.

Table 4.7: Normal behavior

**4.5.2 Usability Tests**

Usability test are a way to ensure the use of the web component is intuitive, fast and simple to understand. This allows to verify a coherent and fluid experience according to the functional and non-functional requirements. To conduct these tests and gather important feedback end users such as developers will be asked to run through these test cases. Down below the test scenarios are specified. The tests candidates will be given instruction on how to use the web component similar to the final developer documentation. The instruction will explain how to initialize and use the component and what each attribute does. Then they are given different scenarios to solve, where they can use these instructions, but are otherwise on their own. The result of the sessions with the candidates can be found in the [Appendix E](#).

#	Scenario Steps
1	<ul style="list-style-type: none"> <li>a. You want to build a horizontal bar chart. Initialize the component.</li> <li>b. You want to display Swiss francs. How do you do that?</li> <li>c. You want to display up to two decimals. What is your next step?</li> <li>d. You would like to show the Internet cost per month of your customer. How can you achieve this?</li> <li>e. Internet costs consist of two types: Subscription and Roaming How would you display this?</li> <li>f. Your customer has a subscription that costs him 20 Fr. per month. What is your next step?</li> <li>g. The customer also has used 5 Fr. of roaming this month for his trip to Australia. Again, how would you do it?</li> <li>h. Your customer has a limit of 50 Fr. per month. How would you set this limit?</li> <li>i. You always want to show the subscription cost first. What is your next step?</li> </ul>
2	<ul style="list-style-type: none"> <li>a. You would like to create a horizontal bar chart. Initialize the component tag in HTML.</li> <li>b. The bar should display all values in Megabytes. How do you proceed?</li> <li>c. Numbers should display exactly one decimal. What do you do/not do?</li> <li>d. You would like to display the roaming data usage of the Mobile plan. Currently, 70 Megabytes have been used. What are your steps?</li> <li>e. The mobile plan does not have an upper limit for data usage. How can you display this?</li> </ul>
3	<ul style="list-style-type: none"> <li>a. You would like to create a horizontal bar chart. Initialize the component tag in HTML.</li> <li>b. You want to display the current debt of the customer in Fr. How do you do that?</li> <li>c. The debts are always whole numbers, so no precision is needed. How can you achieve this?</li> <li>d. Your customer has a maximum allowed debt of -200 Fr. What will you do?</li> <li>e. The current accumulated debt is -113 Fr. for medical bills How would you show this?</li> <li>f. f. The customer also has -73 Fr. debt for mobile payments. How would you display this?</li> </ul>
4	<ul style="list-style-type: none"> <li>a. You would like to create a horizontal bar chart. Initialize the component tag in HTML.</li> <li>b. You want to display the financial circumstances of a German family, so the unit is euros. How do you do this?</li> <li>c. You want to show up to two decimals to display cents. What is your next step?</li> <li>d. There are 3 Family members: Sandra - the mother, Alex - the father and Anna - the daughter. Each of them have as a difference balance. How would you show this?</li> <li>e. Alex has a good job, so he has no limit on his expenses and his current balance is -5000 EUR because he recently bought a new car and took a debt for it. What is your next step?</li> <li>f. Sandra has a monthly limit of 1000 euro. She already spent 200 Euros on new clothes and bought food for 100 this month, so her new balance is 700. How would you do this?</li> <li>g. Anna has 50 Euro available per month, but she already spent everything and has nothing left. How would you display this?</li> </ul>

Table 4.8: Usability tests scenarios

### 4.5.3 User Experience Tests

The user experience tests aim to provide meaningful feedback of the visual chart representation. The test scenarios are done in a peer-to-peer manner with end users such as potential customers to observe their behavior and interaction with the chart. The feedback is used to validate the design approach for easy usability, intuitiveness and comprehensibility. The following test cases challenge the test candidate with various scenarios that could occur in the real world. The scenarios are ranging from normal cases to various edge cases. The results of the session with the test candidate can be found in the [Appendix D](#).

The following questions apply to each scenario and will be asked of the test candidate:

- What is this chart about?
- What costs are being displayed?
- How much did you spend already?
- What is your spending limit?
- Which colors correspond to which costs?

No.	Scenario
1	A single category with two subcategories is shown in the positive range. In addition, the category has no maximum.
2	There are two categories with four subcategories, and one subcategory spans the two categories.
3	There are two categories with four subcategories, and one subcategory spans both categories. In addition, the sum of the second category exceeds the maximum.
4	A single category with two subcategories is shown in the negative range.
5	There are two categories with four subcategories. In addition, the sum of the two categories exceeds their respective maximums and the second category is in the negative range.
6	There are 6 categories with five subcategories. One subcategory spans three categories.

Table 4.9: UX tests scenarios

### 4.5.4 Unit Tests

Unit tests are being used to test the functional correctness. Automatic as well as manual testing is used for this approach.

### 4.5.5 Performance Testing

To validate the load time of the web components, Google's Lighthouse tool is used to generate reports on various performance measurements. The reports can be found in the [section 6.2](#).

## 4.6 Code Reviews

### 4.6.1 Code Review 1

On the 10th May 2023 the first code review was conducted by Michael Gfeller. The review involves all changes made up to commit [9dbb3f950edbf1fd29d071690d0a03c3492f2714](https://gitlab.ost.ch/florian.blum/chart-components-code/-/commit/9dbb3f950edbf1fd29d071690d0a03c3492f2714) in the development branch<sup>1</sup> of the code repository.

In general, the code is comprehensible and contains no major code smells. However, there exist opportunities to revise the code through refactoring in the following areas:

- Splitting long methods
- Move business logic to separate files
- Outsource functional utility methods
- Place guards before loops if possible

Splitting long methods is challenging due to code formatting constraints but necessary for improved readability and a future-proof code foundation.

The absence of testing in the codebase represents a limitation for future maintainers. It hinders their ability to ensure the reliability and correctness of the system through integrated tests. Notably, the code incorporates numerous private methods that are typically excluded from testing procedures. In view of this, it is advisable to cast tested classes to an "any" type and conduct comprehensive testing of private functions as well.

In addition, the absence of error messages can result in confusion among users. It is therefore highly recommended that a mechanism for reporting errors is implemented. This can be achieved through console logging or by displaying error messages directly on the web interface. This will improve both usability and debugging capabilities.

The negative chart version uses negative values which although intuitive add complexity to the code. Normalizing negative values could resolve this issue.

Lastly the security of the system is addressed. The web component does not make use of security compromising utilities like an "eval" or "innerHTML". In general security can be neglected because the web component does not introduce client-server interactions.

---

<sup>1</sup><https://gitlab.ost.ch/florian.blum/chart-components-code/-/tree/development>

### 4.6.2 Code Review 2

On May 31, 2023 the second code review was conducted by Michael Gfeller. The review involves all changes made up to commit [8a44a2839df8adb413ec8f5be9ec7daff7019121](https://gitlab.ost.ch/florian.blum/chart-components-code/-/tree/development) in the development branch<sup>1</sup> of the code repository.

The code shows improved effectiveness in handling input data and operates seamlessly without encountering issues. Notably the validation of the input data represents the most complex module within the codebase. In addition to improved input handling, error messages are displayed which is useful and expedites the integration of the web component into a web page. No major code smells are present. Refactoring in the following areas is advised:

- Splitting long methods
- Outsource constants
- Dividing logic into more classes

Many utility functions are still present in the business logic. They can be moved to a separate file to avoid an oversized class.

The code exhibits deficiencies in certain areas related to quality. Specifically the presence of duplicated code and undeclared constants creates confusion and warrants attention. Recognizing this issue the development team has already initiated efforts to optimize these specific areas. The order of the methods is not very logical and it is recommended to improve this aspect in a future version.

Another area of contention in the code are undeclared types of variables and functions that return an implicit "any" type. For readability and debugging purposes, it is recommended to always declare types of variables and functions.

Finally, it would make sense to move certain business logic files into the appropriate folder within the codebase. Some of these are still located in the utility tool area.

---

<sup>1</sup><https://gitlab.ost.ch/florian.blum/chart-components-code/-/tree/development>

# Deployment

## 5.1 Production Environment Requirements

This application uses Node.js as its runtime environment. This application also requires the use of npm, the official package manager for Node.js.

An installation of a recent npm and Node.js [LTS](#) version is required to run the implementation. At this point in time, the most recent [LTS](#) version of Node.js is 18.16.0<sup>1</sup>. The latest npm version is 9.7.1<sup>2</sup>

## 5.2 Production Deployment

No special precautions need to be taken. The implementation is ready to run after installation. The implementation provides the ability to construct a streamlined build suitable for production purposes. Further details can be found in the README markdown file in the code repository.

---

<sup>1</sup><https://nodejs.org/dist/v18.16.0>

<sup>2</sup><https://docs.npmjs.com/cli/v9>

# Results

## 6.1 Functional Requirements Evaluation

### 6.1.1 GNL-US1 Legend

#### GNL-US1

Each chart has a legend that allows for a mapping between colors and subcategory labels. If a subcategory has a value of zero, it will not be displayed, and therefore no legend icon will be displayed for that particular subcategory, to avoid confusion. In the following example, a subcategory "Roaming" with a value of "0" exists but is not displayed because it has a value of "0". There is no icon in the legend because the color shown in the legend cannot be mapped to a color in the bar.

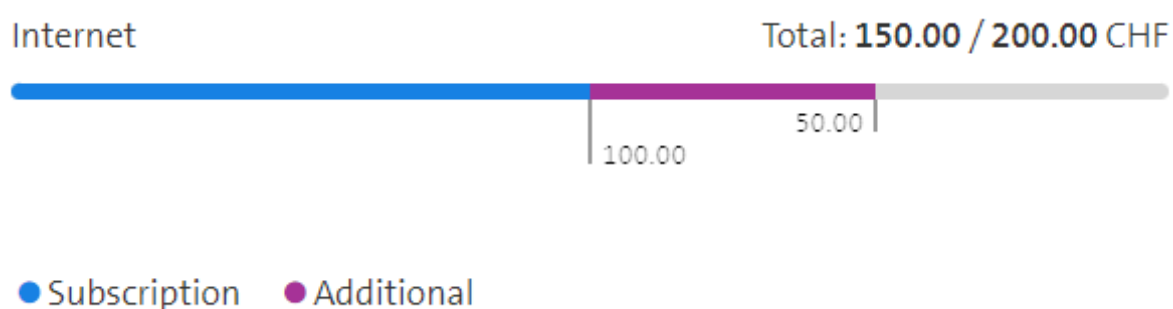


Figure 6.1: Result legend

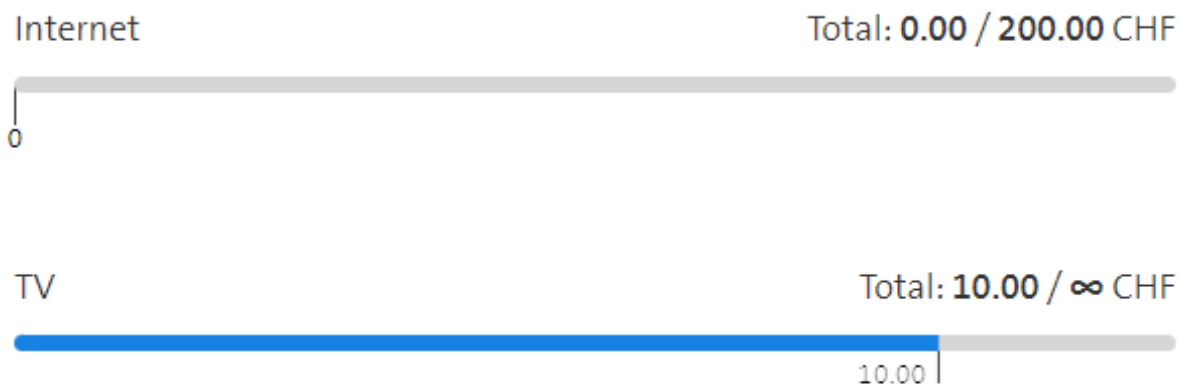
### 6.1.2 GNL-US2 Background Bar & GNL-US10 Missing Data

#### GNL-US2

#### GNL-US10

Each bar that displays the category has a gray background bar that is always visible to indicate how long the colored bars for subcategories above might be. This gray bar is also visible when there is no data to display for subcategories.





- Subscription

Figure 6.2: Result gray background bar

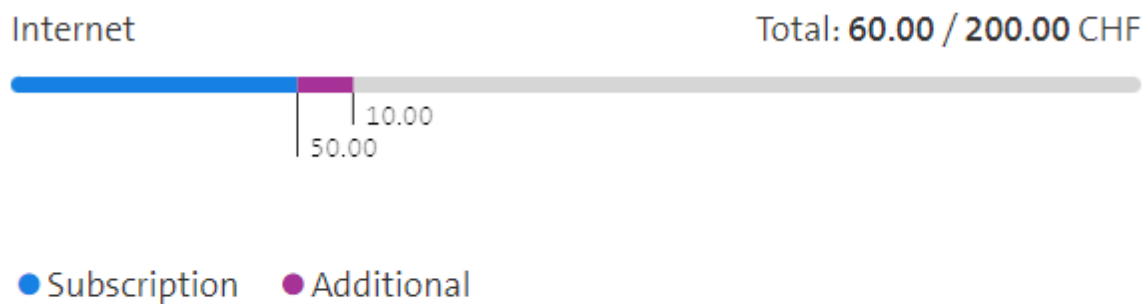
### 6.1.3 GNL-US3 Labelling & GNL-US9 Font/Spacing

GNL-US3

GNL-US9

Each chart follows the same labelling design approach to ensure a consistent design and user experience. Each chart always displays the category and subcategory names, as well as all value labels and legends. The font and spacing of each label type is the same for all categories and charts, making them easier to identify and compare.

## Chart 1



## Chart 2

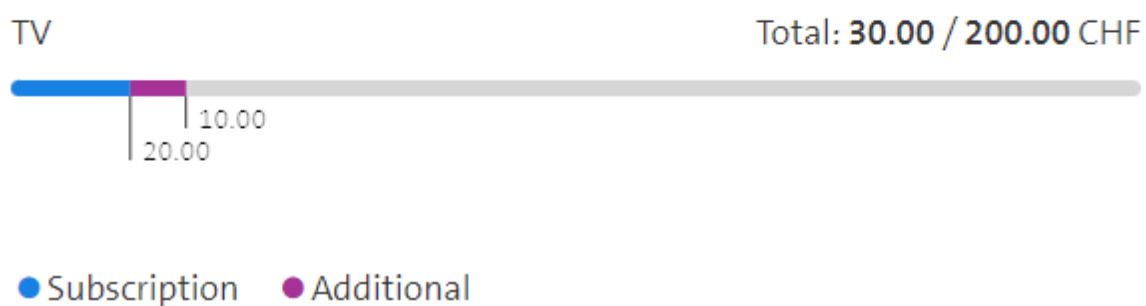


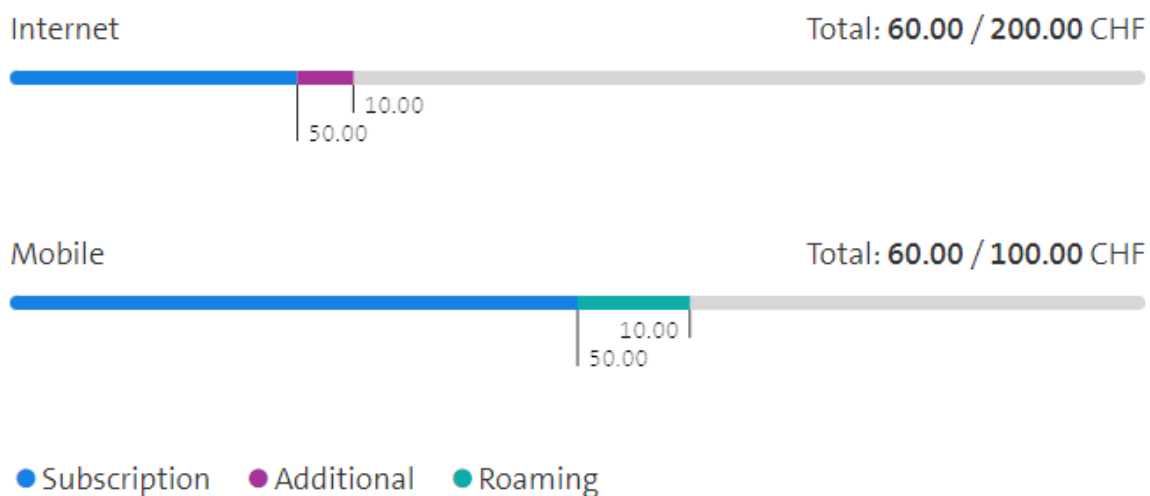
Figure 6.3: Result labelling across charts

### 6.1.4 GNL-US4 Color Order

#### GNL-US4

The colors are always in the same order in all categories and charts.

## Chart 1



## Chart 2

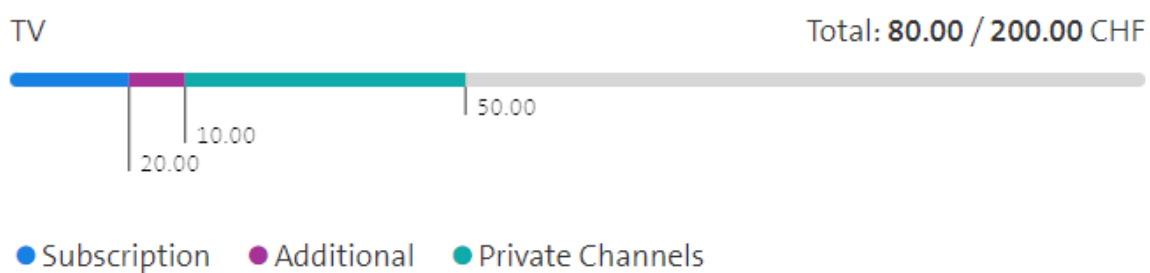


Figure 6.4: Result color order

### 6.1.5 GNL-US5 Data Structure & HCI-US2 Multiple Categories/Subcategories

GNL-US5

HCI-US2

The chart uses an array of one-dimensional JSON objects. This structure is based on the structure of the pie chart, which has a similar one-dimensional JSON structure. The user can set a category, a subcategory, a maximum value for the category and a value for the subcategory. Multiple entries can be added to add more categories and subcategories.

```
data='[{
  "value": 50,
  "category": "Internet",
  "subcategory": "Subscription",
  "maxValue": 200
},{
  "value": 10,
  "category": "Internet",
  "subcategory": "Additional"
},{
  "value": 50,
  "category": "Mobile",
  "subcategory": "Subscription",
  "maxValue": 100
},{
  "value": 10,
  "category": "Mobile",
  "subcategory": "Roaming"
}]'
```

Figure 6.5: Result data structure

### 6.1.6 GNL-US6 Negative Data

#### GNL-US6

The chart supports negative values. These are animated starting from the right side of the chart, and the bar always displays a zero, further enhancing the user experience by indicating where the bar starts.



Figure 6.6: Result negative values

### 6.1.7 GNL-US7 Chart Type Recommendation

#### GNL-US7

The developer's guide has a chapter that explains the different types of data and how best to use a horizontal bar chart with which data. The guide can be found in the [Appendix B](#).

### 6.1.8 GNL-US8 Update Function

#### GNL-US8

It is possible to update the chart without reloading the site by simply changing the data within the data attribute. The chart recognizes the change by using a decorator called "@Watch()". This calls the render method and displays the new data.

### 6.1.9 HCG-US1 Horizontal Bar Chart & HCI-US3 Proportions Infinity & HCL-US1 Proportions Maximum Value

#### HCG-US1

#### HCI-US3

#### HCL-US1

The chart shows a horizontal bar with sub-bars. The sub-bars represent sub-categories. The length of the sub-bars represents their values in relation to the maximum value. If no maximum value has been set for the chart, the bar is always filled to 80%.

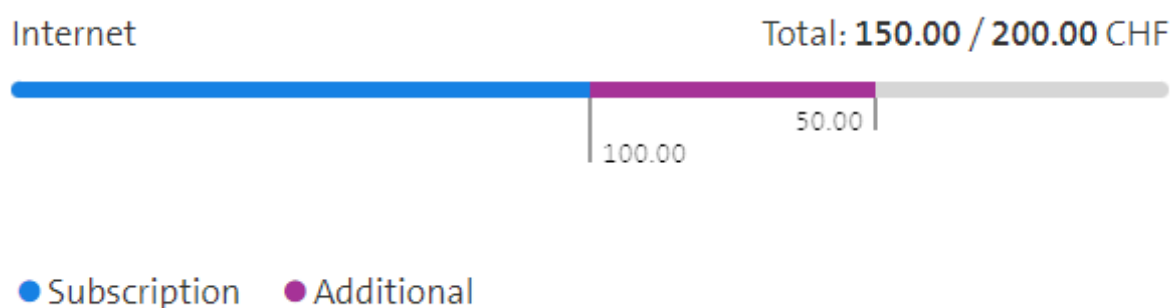


Figure 6.7: Result horizontal bar chart

### 6.1.10 HCG-US2 Unit & HCG-US5 Precision

#### HCG-US2

#### HCG-US5

The user can define a unit and the precision via an attribute which then will be displayed within the chart.

```
unit="CHF"  
precision="2"
```

Figure 6.8: Result unit and precision

### 6.1.11 HCG-US3 Detailed Information

#### HCG-US3

The user can define a category and subcategories within the data structure. This information is then displayed on the graph. See [Figure 6.5](#) and [Figure 6.7](#).

### 6.1.12 HCG-US4 Small Data Points

#### HCG-US4

Small values, which would take up almost no space in the bar if the maximum was set to a high value, are displayed with a minimum length to ensure comprehension and readability.

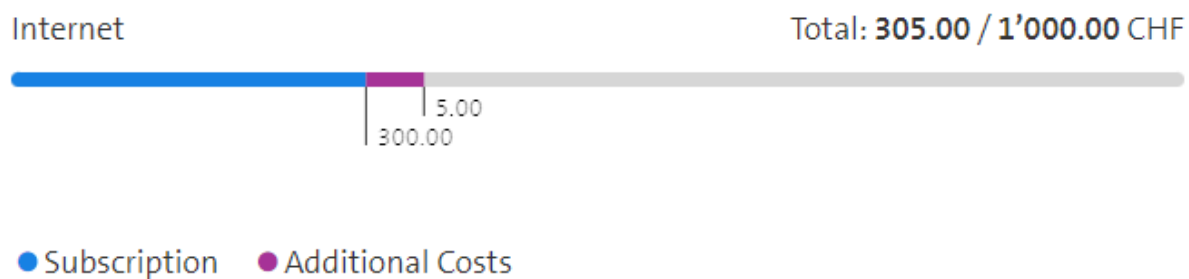
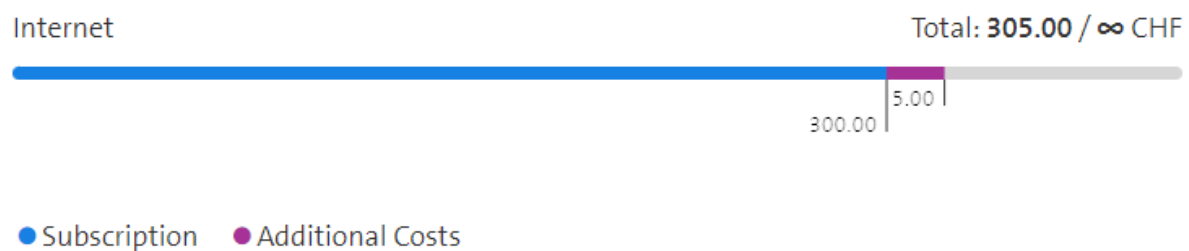


Figure 6.9: Result minimum length

### 6.1.13 HCI-US1 Infinity

#### HCI-US1

The "maxValue" attribute in the data structure can be omitted for the first entry of a category to set the maximum value of that category to infinity.



(a) Infinity chart

```
data='[{
  "value": 300,
  "category": "Internet",
  "subcategory": "Subscription"
},{
  "value": 5,
  "category": "Internet",
  "subcategory": "Additional Costs"
}]'
```

(b) No maximum value data structure

Figure 6.10: Result infinity behavior

### 6.1.14 HCL-US2 Maximum Value

#### HCL-US2

The "maxValue" attribute can be set for the first entry in a category. All subsequent definitions of "maxValue" for that category will be ignored.



(a) Maximum value chart

```
data='[{
  "value": 300,
  "category": "Internet",
  "subcategory": "Subscription",
  "maxValue": 400
},{
  "value": 5,
  "category": "Internet",
  "subcategory": "Additional Costs"
}]'
```

(b) Maximum value data structure

Figure 6.11: Result maximum value behavior

### 6.1.15 HCS-US1 Multiple Categories & HCS-US2 Color Distribution & HCS-US3 Gray Dividing Lines

#### HCS-US1

#### HCS-US2

#### HCS-US3

A chart can display up to six different categories. Each category is represented by a bar. A category can display up to 3 different subcategories. The whole chart is limited to 5 different subcategories across all categories. Each of the subcategories has a different color with a fixed order. There are no gray dividing lines between the categories.

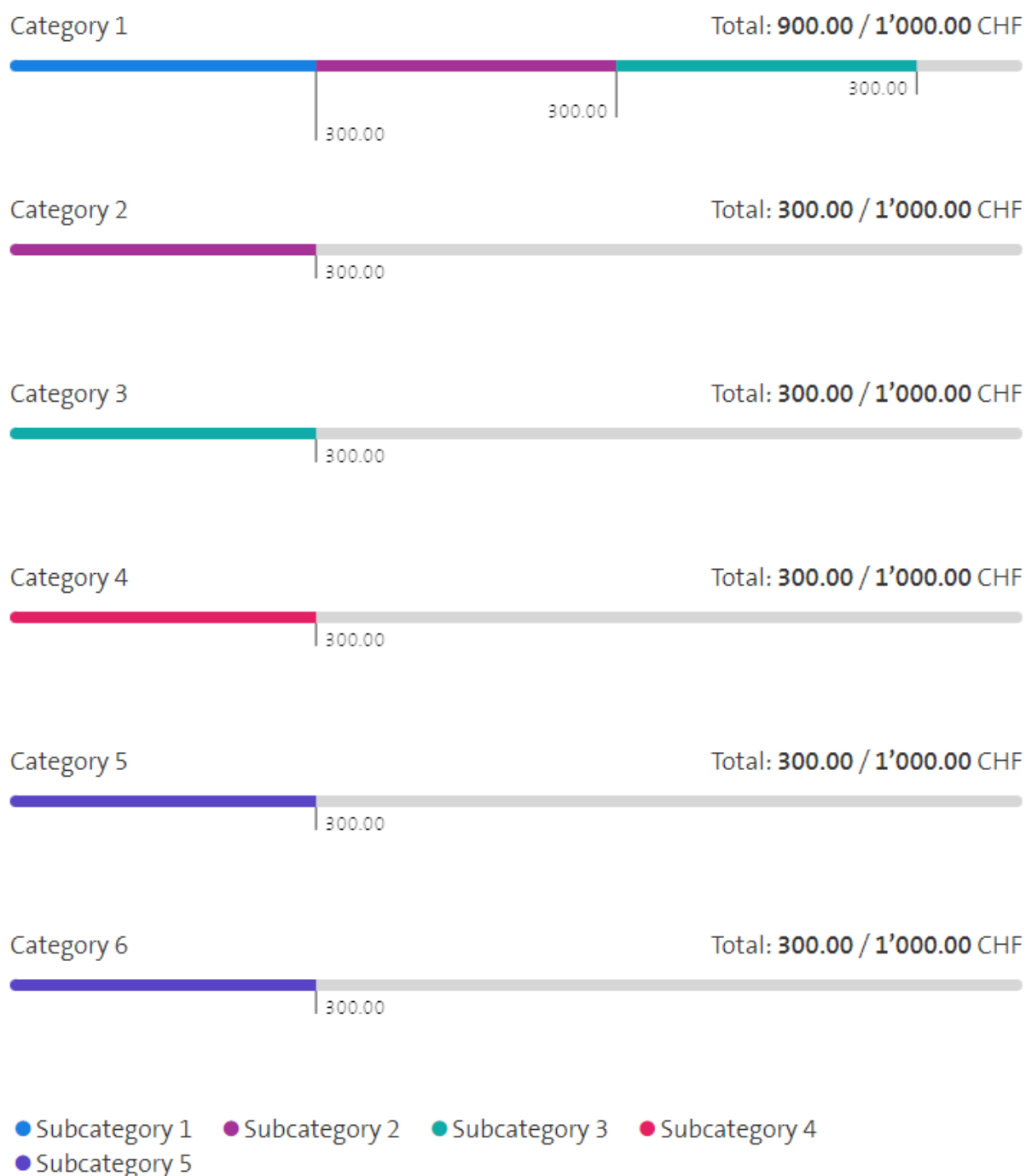


Figure 6.12: Result stacked chart

### 6.1.16 Optional Functional Requirements

#### Vertical Bar Chart:

The vertical bar chart has not been implemented to allow time for the horizontal bar chart to be implemented. Therefore, all functional requirements from OPT-US1 to OPT-US5 have not been implemented.

#### Miscellaneous:

The functional requirement [OPT-US8](#) has been implemented. Unit tests for the data parser and the utility functions have been added. The other optional functional requirements were not implemented.



## 6.2 Non-Functional Requirements Evaluation

### 6.2.1 NFR-M01 Maintainability

#### NFR-M01

Two code reviews were conducted with the result that the code is maintainable and adheres to the defined standards. Improvements suggested by the Code Review have been implemented.

**Result:** Fulfilled.

### 6.2.2 NFR-M02 Code Guidelines

#### NFR-M02

Defined coding guidelines and standards were followed during development. The provided linter and formatter were used. Feedback from the code review has been taken into account. The feedback from the stakeholder was positive.

**Result:** Fulfilled.

### 6.2.3 NFR-M03 Test Coverage

#### NFR-M03

Unit testing achieved 100% coverage of the business logic. The rendering logic has been thoroughly tested with manual system testing of various edge cases and normal cases.

**Result:** Fulfilled.

### 6.2.4 NFR-E01 Performance

#### NFR-E01

Performance of the legacy non-web component:

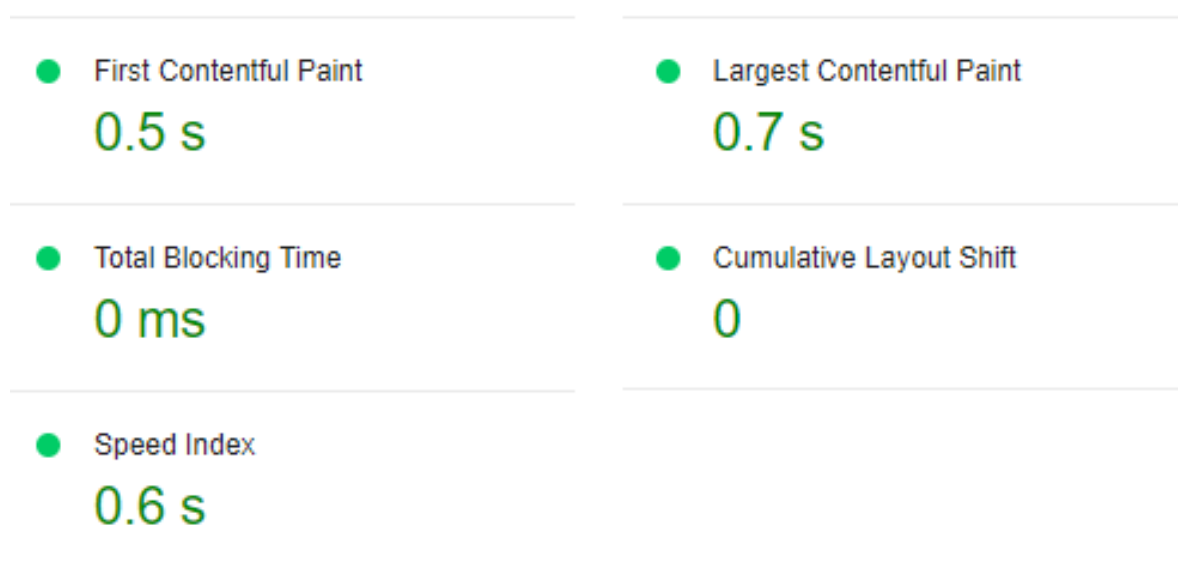


Figure 6.13: Performance legacy horizontal bar chart component

Performance of the web component:

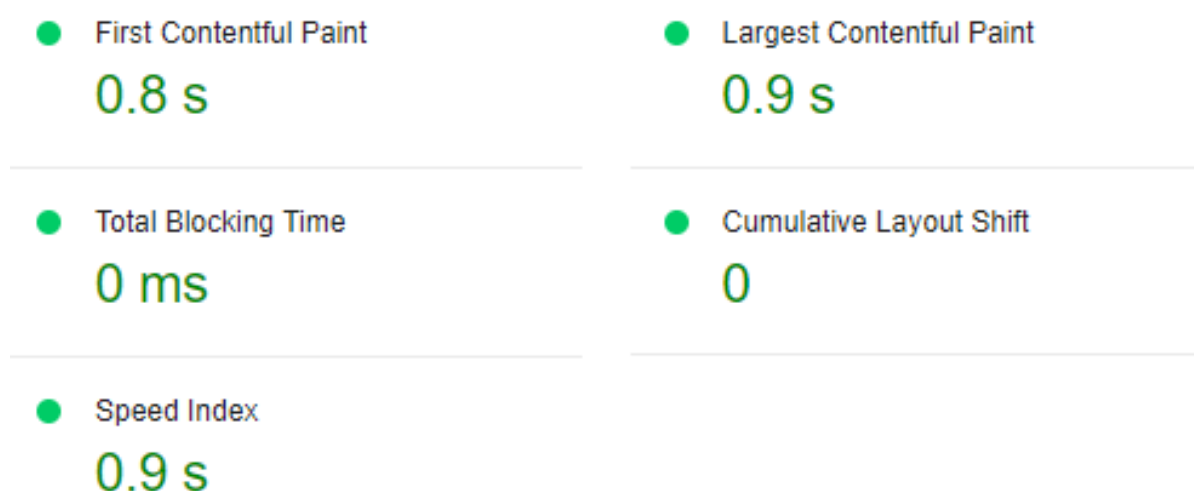


Figure 6.14: Performance horizontal bar chart web component

The "First Contentful Paint" has increased by an average of 62%. The "Largest Contentful Paint" has increased by an average of 28%. The speed index has increased by an average of 50%. This results in an overall increase of 42% which means a slight loss in performance. This is because the web component uses more layers of abstraction and relies on libraries such as Stencil, which adds more complexity. The performance loss is still within the defined limits.

**Result:** Fulfilled.

### 6.2.5 NFR-P01 Browser Compatibility

#### NFR-P01

All manual system tests were performed on each browser. Each browser passed all test cases.

**Result:** Fulfilled.

### 6.2.6 NFR-U01 Accessibility

#### NFR-U01

Google Lighthouse reports an accessibility of 100%.



## Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

Figure 6.15: Accessibility horizontal bar chart web component

However, an important part of the component is the color scheme and as explained in [subsection 3.7.4](#) the color scheme doesn't meet the AA standard. In addition, many other criteria of the [WCAG](#) standard have not been implemented, due to design and time constraints.

**Result:** Not fulfilled.

## 6.3 Limitations

### 6.3.1 Scope

The optional vertical bar chart was not implemented in this study as the available time and resources were prioritized for the implementation and analysis of the required horizontal bar chart. This means that the final design and implementation of this work will need to be revised if it is to be used to implement the vertical bar chart.

### 6.3.2 Design

Currently, a maximum of six categories with three subcategories per category is allowed. If more than this number is required in the future, the chart may become difficult to read due to its extensive design.

### 6.3.3 Technology

The current implementation of the subcategory value labels is based on SVG technology. SVG text is very limited in its responsive design. If a more responsive design is required in the future, SVG text may not be sufficient and will need to be replaced.

## 6.4 Expansion Possibilities

### 6.4.1 Accessibility Expansion

The accessibility of the component has been improved. Though it fails to meet the [WCAG](#) (Web Content Accessibility Guidelines)<sup>1</sup> Level 2.1 AA standard. This enhancement, originally specified in the optional non-functional requirement [NFR-U01](#), would provide significant benefits to users with disabilities.

The implementation of this standard is challenging due to the existing design constraints imposed by Swisscom AG. However, the component does not preclude the inclusion of additional accessible elements. It should be noted that the [WCAG](#) standard is comprehensive and requires careful testing. Testing with real users can provide valuable insights for both developers and end users, albeit at a cost in time. Taking all these factors into account, the estimated minimum effort to implement this enhancement is 80 hours.

Expansion	Effort Estimation	Task Summary
Accessibility	100h	Improving the web components accessibility to a Level 2.1 AA <a href="#">WCAG</a> standard.

Table 6.1: Accessibility expansion overview

<sup>1</sup><https://www.w3.org/TR/WCAG21>

### 6.4.2 Data Sorting Expansion

The current component lacks the ability to facilitate data sorting for both developers and end users. To address this limitation, an optional sorting property could be introduced to allow developers to incorporate sorting functionality. For end users an interactive sorting element could be implemented within the component, allowing them to select sorting options based on their preferences. These enhancements would provide both developers and end users with the flexibility and convenience of sorting the data displayed.

The existing implementation already includes a data input parser specifically tailored to the requirements of the horizontal bar chart. The parser itself is complex, which presents a challenge when attempting to incorporate data sorting functionality within it. It is generally recommended that sorting operations are applied after the data has been parsed. Although this approach may not optimize performance, it is considered feasible given the expected amount of input data.

Expansion	Effort Estimation	Task Summary
Data Sorting	15h	Providing a sorting functionality for both end-users and developers.

Table 6.2: Data sorting expansion overview

### 6.4.3 Filter & Search Expansion

The current state of the component lacks filtering and searching options for charts containing multiple segments. The addition of filtering and searching functionality can significantly improve the overall user experience. These features would allow users to view only the relevant data or search for specific chart elements based on desired values. By introducing filtering and searching, the component can give users more precise control over the data they interact with, resulting in an improved user experience.

Efforts should be made to ensure that filtering remains accessible and user-friendly, without being unnecessarily complicated. Given the availability of numerous filtering options and diverse data types within the component, careful consideration should be given to provide a streamlined filtering experience. It is important to design a filtering system that effectively handles the different data types and allows users to easily apply filters based on their requirements.

Expansion	Effort Estimation	Task Summary
Filter and Search	25h	Provide a filter and search function within the web component.

Table 6.3: Filter and search expansion overview

### 6.4.4 End-to-End Test Expansion

The current implementation of the component lacks an automated end-to-end test suite and relies on manual testing methods. Manual testing can be time-consuming and labor-intensive. However, the [Stencil](#) library provides support for various testing frameworks, including Google's [Puppeteer](#), which can be used for end-to-end testing. Implementing an automated end-to-end test suite streamlines the testing process, improves efficiency and ensures comprehensive functionality test coverage. Automated testing offers the benefits of faster execution and the ability to repeat tests consistently, resulting in more reliable and scalable testing practices.

End-to-end testing is highly recommended for this web component, as there are numerous edge cases that can be effectively tested using automated procedures. In particular, visual testing

is recommended for validating SVG renders. By implementing visual testing techniques, developers can compare the expected visual output of the component with the actual rendered output.

Expansion	Effort Estimation	Task Summary
End-to-End Testing	15h	Introduce an end-to-end test suit using the desired technology.

Table 6.4: End-to-End test expansion overview

### 6.4.5 Internationalization Expansion

The existing component implementation has a deficiency in its automatic lexical adaptation capabilities. Swisscom AG has a customer base consisting of people who communicate in different languages. Implementing internationalization in accordance with the I18n<sup>1</sup> standard would improve the operational functionality of the component by adapting it to local standards and meeting user expectations in specific regions.

The diagram component conveys predominantly visual information with minimal reliance on textual elements, which limits the interpretability of the content for translation purposes. The inclusion of an automatic translation feature for the input data would be beneficial, although the option to disable this functionality should also be provided to accommodate user preferences.

Expansion	Effort Estimation	Task Summary
Internationalization	10h	Provide internationalization functionality within the component.

Table 6.5: Internationalization expansion overview

### 6.4.6 Line Chart Expansion

An omission in Swisscom AG's design system is the lack of a dedicated component for visualizing line charts. Line charts are a valuable tool for effectively presenting data sets that may not be adequately represented by pie or bar charts.

The newly proposed chart component should adhere to the design system guidelines set by Swisscom AG, while taking advantage of the benefits offered by web components. The use of a web component makes the chart more user-friendly and facilitates a standardized design approach throughout the component's usage. Vector graphics should be used to ensure optimal visual representation of the line chart.

Expansion	Effort Estimation	Task Summary
Line Chart Component	360h	Implement a line chart web component.

Table 6.6: Line Chart expansion overview

<sup>1</sup><https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/i18n>

## **Part II**

# **Project Documentation**

# Project Plan

The Rational Unified Process (RUP) is used as a framework for planning the project with an agile approach. The project is broken down into four phases:

- Inception
- Elaboration
- Construction
- Transition

A lightweight version of Scrum, tailored to the needs of the project team, is being used. The parts of Scrum that the team is incorporating into the project activities are:

- Sprint period of one week
- Product backlog/Sprint backlog
- Sprint planning/Sprint review
- Agile assignment of work items

## 7.1 Project Schedule

The following figure shows the project plan. The four RUP phases are visualized together with the corresponding milestones.

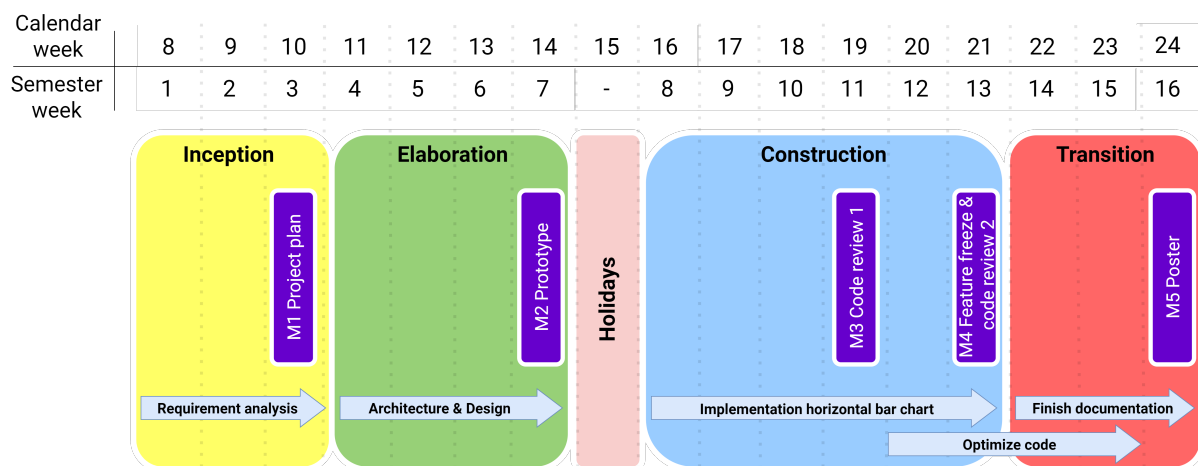


Figure 7.1: Project plan

## 7.2 Milestones

A milestone indicates the end of a project stage and can act as a verification utility to check if all defined goals have been reached. There are 5 milestones defined in this project plan.

M#	Milestone Name	Artifacts	Due within week
M1	Project Plan	<ul style="list-style-type: none"> <li>• Repositories set up</li> <li>• FR / NFR / constraints</li> <li>• Project plan</li> </ul>	3
M2	Prototype	<ul style="list-style-type: none"> <li>• Architecture</li> <li>• Updated project plan</li> <li>• Prototype</li> </ul>	7
M3	Code review 1	<ul style="list-style-type: none"> <li>• Chart implementation</li> <li>• Test suit</li> <li>• Code review 1</li> </ul>	11
M4	Feature freeze & code review 2	<ul style="list-style-type: none"> <li>• Updated documentation</li> <li>• Code review 2</li> <li>• No new features</li> </ul>	13
M5	Poster	<ul style="list-style-type: none"> <li>• Finished documentation</li> <li>• Poster finished</li> </ul>	16

Table 7.1: Milestone definitions



# Project Organization

## 8.1 Involved Parties

The roles in the project team are as follows:

<b>Role</b>	<b>Assigned person(s)</b>
Design	Florian Blum
Business logic	Niklas Kaesler
Frontend	Florian Blum
Minutes Secretary	Niklas Kaesler
Meeting organization	Florian Blum
Scrum Master	Niklas Kaesler
Product Owner	Florian Blum
Developer	Florian Blum, Niklas Kaesler
Documentation	Florian Blum, Niklas Kaesler

Table 8.1: Role assignment

The roles determine which person has the main responsibility. In case of absence, the other team member takes over.

### **Role description:**

- Design: Responsibility for the visual design
- Business logic: Responsibility for the logical aspect of the application
- Frontend: Responsibility for the visual implementation of the application
- Minutes Secretary: Responsibility for taking notes during meetings
- Meeting organization: Responsibility to organize meetings
- Scrum Master: Responsibility for following the Scrum principles
- Product Owner: Responsibility for tracking product progress
- Developer: Responsibility for implementing features
- Documentation: Responsibility for product documentation

## 8.2 Meeting Organization

<b>Meeting</b>	<b>Frequency/Date</b>
Sprint Planning	Weekly
Sprint Review	Weekly
Advisor Meeting	Weekly/Bi-weekly
Code Review 1	10.05.2023
Code Review 2	31.05.2023
Interim Presentation	20.04.2023
Final Presentation	22.08.2023
Stakeholder meeting	At least once in each RUP phase except for Transition

Table 8.2: Meetings

There is no daily scrum, as the whole team is in constant communication with each other due to the fact that all team members are full-time students.

# Management Tools

## 9.1 Issue Management

The issues are managed via Jira. Time is also managed using an add-on in Jira.

There are four different types of issues:

- **Story:** Represents a feature that is directly connected to a requirement.
- **Sub-Task:** A user story can be broken down into smaller units, making it easier to manage development.
- **Task:** Represents a task that is not directly related to a requirement, but needs to be done for the project, such as documenting a feature.
- **Bug:** For unexpected issues with the codebase directly related to features.

There are two types of backlogs:

- **Sprint backlog:** Contains all the refined issues that are planned to be worked on during a sprint.
- **Product backlog:** Contains all the issues required to complete the project in a prioritized order. The top issues are transferred to the Sprint Backlog in the next Sprint iteration.

Within a sprint there are three stages for the issues:

- **To-Do:** Issues that still need to be done.
- **In progress:** Issues currently being worked on.
- **Done:** Finished

## 9.2 Version Control

### Documentation

The documentation is version controlled using GitLab on the OST infrastructure. Link to the GitLab repository:

<https://gitlab.ost.ch/florian.blum/chart-components-documentation>.

### Source code

The source code is version controlled using GitLab on the OST infrastructure. Link to the GitLab repository:

<https://gitlab.ost.ch/florian.blum/chart-components-code>.

## 9.3 Other Tools

### 9.3.1 Documentation Tools

<b>Purpose</b>	<b>Tools</b>
Documentation IDE/Environment	Visual Studio Code with LaTeX
Mockups	Figma
Personas	HubSpot
Other Graphics	draw.io
Risk Evaluation	Excel
Spelling, Grammar, Sentence structure	DeepL(-Write), ChatGPT, LaTeX extensions (LTeX, Spell Right)

Table 9.1: Documentation tools

### 9.3.2 Code Tools

<b>Purpose</b>	<b>Tools</b>
Code IDE/Environment	WebStorm with NPM

Table 9.2: Code tools

# Risk Management

## 10.1 Risk Identification

Risk ID	Title	Description
R1	Technologies	Technologies that we use are constantly evolving. The project and application needs to adapt to such changes. Rebuilding knowledge and dealing with new bugs can influence the project timeline.
R2	Health & Safety	One or more team members get ill or have an accident. The tasks of that member are not being worked on or will be delayed.
R3	Communication	A team member does not openly discuss issues in the meetings / chats. This would lead to problems that are going to be discovered at a later stage.
R4	Scope	The scope of the project is too big, and the targeted goals are unrealistic and not achievable in the given time.
R5	Knowledge	Lack of acknowledgement of a team member or the whole team concerning technical aspects of the project. This could lead to a time delay or a shortening of the scope.
R6	External hazards	Global conflicts, weather disasters or other crises make it impossible to work on the project.
R7	Time management	Due to poor time planning, the project cannot be finished in time.
R8	Requirements	Missing or incorrect functional requirements lead to an unsatisfying end product.
R9	Software quality	A lacking strategy to ensure the requested strategy leads to an unsatisfying product.
R10	Absences	A stakeholder is not available during a critical time period.

Table 10.1: Risk identification

## 10.2 Risk Matrix

Probability	Severity			
	Negligible	Marginal	Critical	Catastrophic
Certain				
Likely				
Possible		R2, R4, R5		
Unlikely		R8, R9, R10	R7	
Rare		R1		R6
Eliminated	Eliminated: R3			

Impact: Low Medium High Very High

Table 10.2: Risk matrix

### 10.3 Risk Evolution

Each risk gets a risk score assigned. The score is calculated as follows: Probability \* Severity \* Impact = Risk score.

For the probability the values are: Rare=1, Unlikely=2, Possible=3, Likely=4, Certain=5.

For the severity the values are: Negligible=1, Marginal=2, Critical=3, Catastrophic=4.

For the impact the values are: Low=1, Medium=2, High=3, Very High=4.

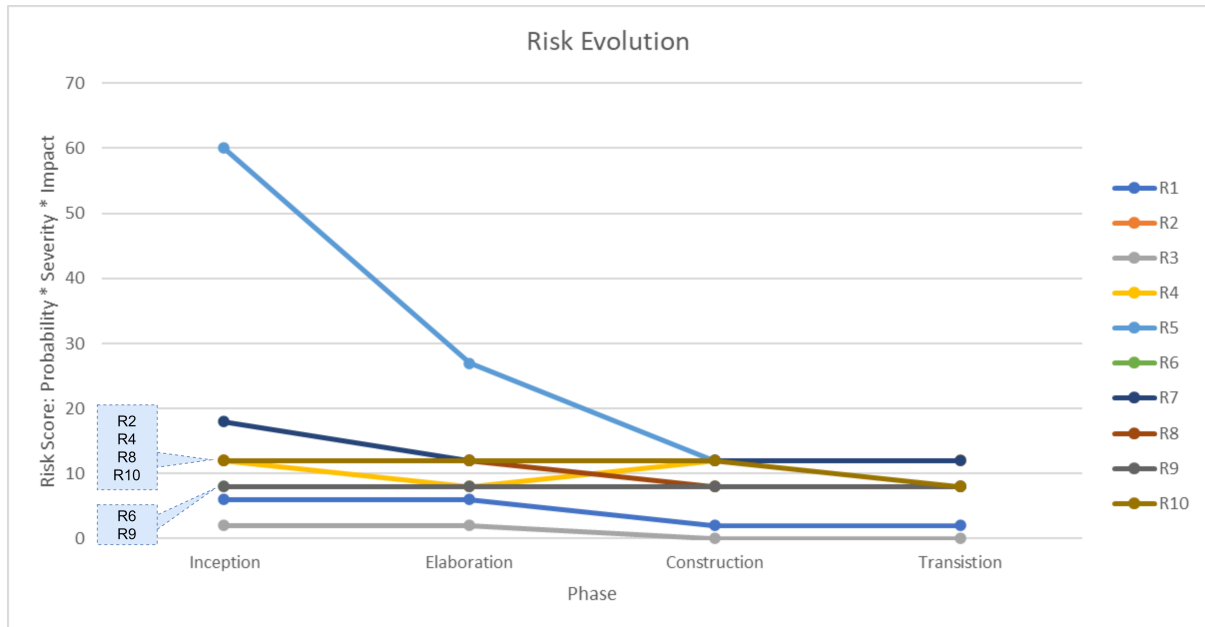


Figure 10.1: Risk Evolution Graph

## 10.4 Risk Mitigation

Affected Risks	Measures
R1	Stable versions of the libraries are being used and the software will be built modularly, where parts can easily be changed without impacting the whole application.
R2, R3, R7, R10	Frequent meetings and open communication are being practiced. In weekly and/or daily syncs knowledge is shared so no knowledge silos arise.
R4, R7, R8	The planning phase is carried out properly and industry practices are used for the project management. Frequent reoccurring update meetings with the stakeholders are being held to reach a consensus.
R6	Code and documentation is backed up. Apart from that, the risk is being accepted.
R5	In the Inception and Elaboration phase tutorials will be worked through and the code and its structure will be analyzed. The team will train itself through research session that are being taken into account when planning the project.
R9	Code review are being held to check the quality together with the stakeholder. The requirements have to fulfill their Definition of Done and have to pass the defined tests.

Table 10.3: Risk Mitigation and Prevention

# Time Tracking

For the time tracking a combination of the standard tool from Jira and the extension "Timesheet Reports and Gadgets for JIRA Cloud" is used. The standard tool allows to report time spent on each task and the extension can generate various reports with different filters such as per week, per team member or per type of task. These are the following types of task defined:

- Documentation
- Code
- Management
- Research

## Time tracking by labels

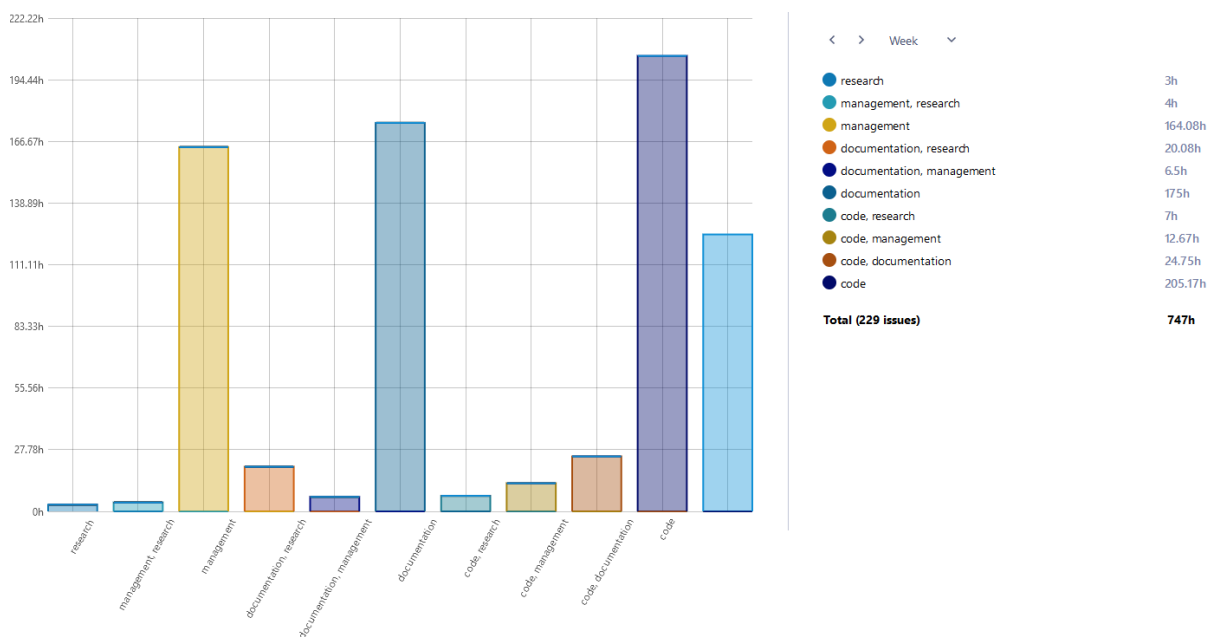


Figure 11.1: Time tracking by labels



### Time tracking by semester week

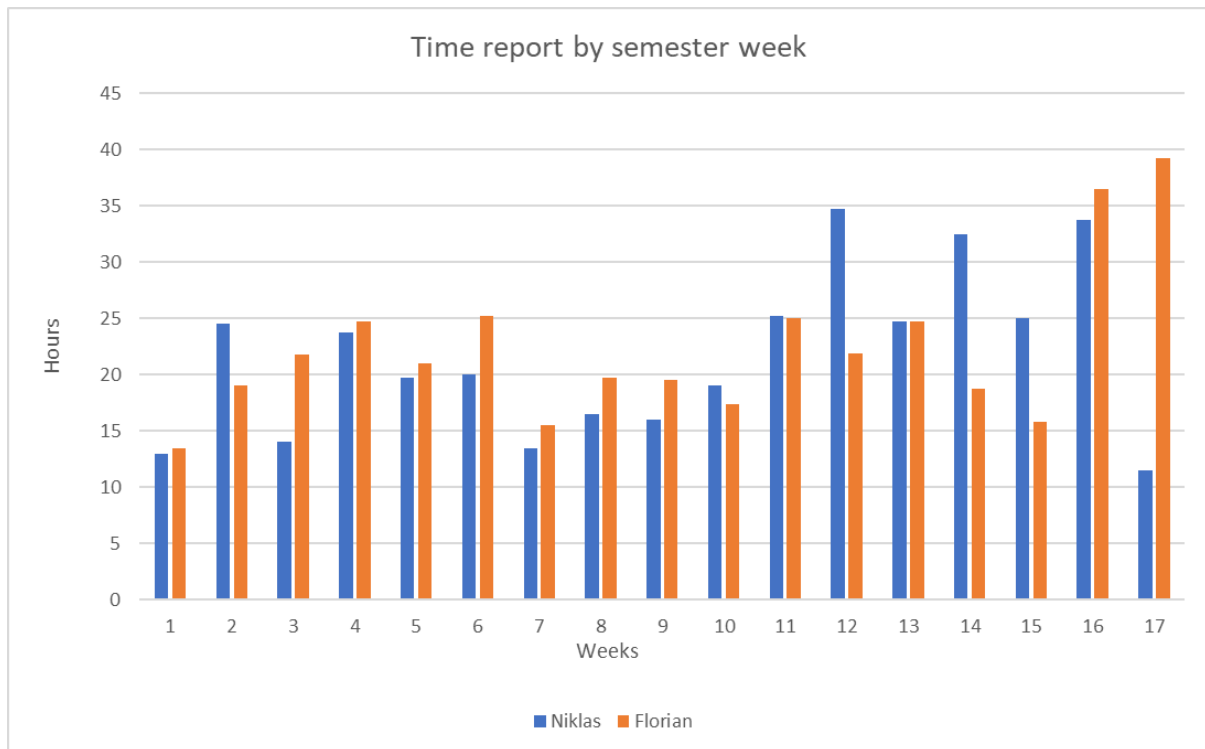


Figure 11.2: Time tracking by semester week

**Part III**

**Appendix**

# License Agreement

MIT License

Copyright (c) 2018

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Developer Guide

DEVELOPER-GUIDE.md

6/13/2023

## Intro

---

### When to Use

This horizontal bar chart is most effective for displaying graphs that represent any of the following:

#### Nominal Comparisons

The data present matches the following criteria:

- No specific order
- Categorical subdivisions present
- For each subdivision one or more measures available

#### Rankings

The data present matches the following criteria:

- Data ordered ascending or descending
- Categorical subdivisions present
- For each subdivision one measure available

If low values should be highlighted align the data in ascending order. For high value highlighting align the data in descending order.

#### Part-to-Whole Visualizations

The data present matches the following criteria:

- Individual categorical subdivisions available
- Measures in a ratio to the whole

Stacked bars can be used if the parts of a whole need to be displayed.

---

## Designers

---

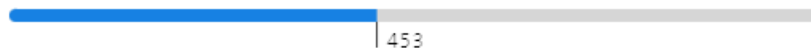
### Variations

#### Limited

Limited charts display datasets which have an upper limit.

### Mobile Plan

Total: **453 / 1'000**



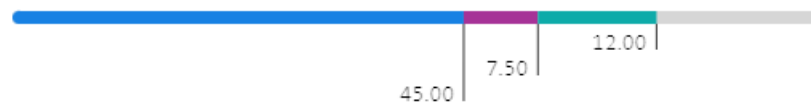
- SMS

### Infinite

Infinite charts show data that can not reach an upper limit. The bars will always extend to 80% of the chart width no matter the defined values.

### TV

Total: **64.50 / ∞ CHF**



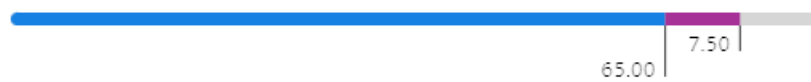
- Subscription Cost
- Movie Rentals
- Additional Channels

### Multi

Multi charts display multiple bars which represent different categories. These categories often share the same legend types.

### Internet

Total: **72.50 / 80.00 CHF**



### TV

Total: **49.50 / 80.00 CHF**



- Subscription Costs
- Additional Costs

### Negative

Negative charts display their bar in a right to left manner. Combining any other chart variation with the negative chart variation is not recommended.

DEVELOPER-GUIDE.md

6/13/2023



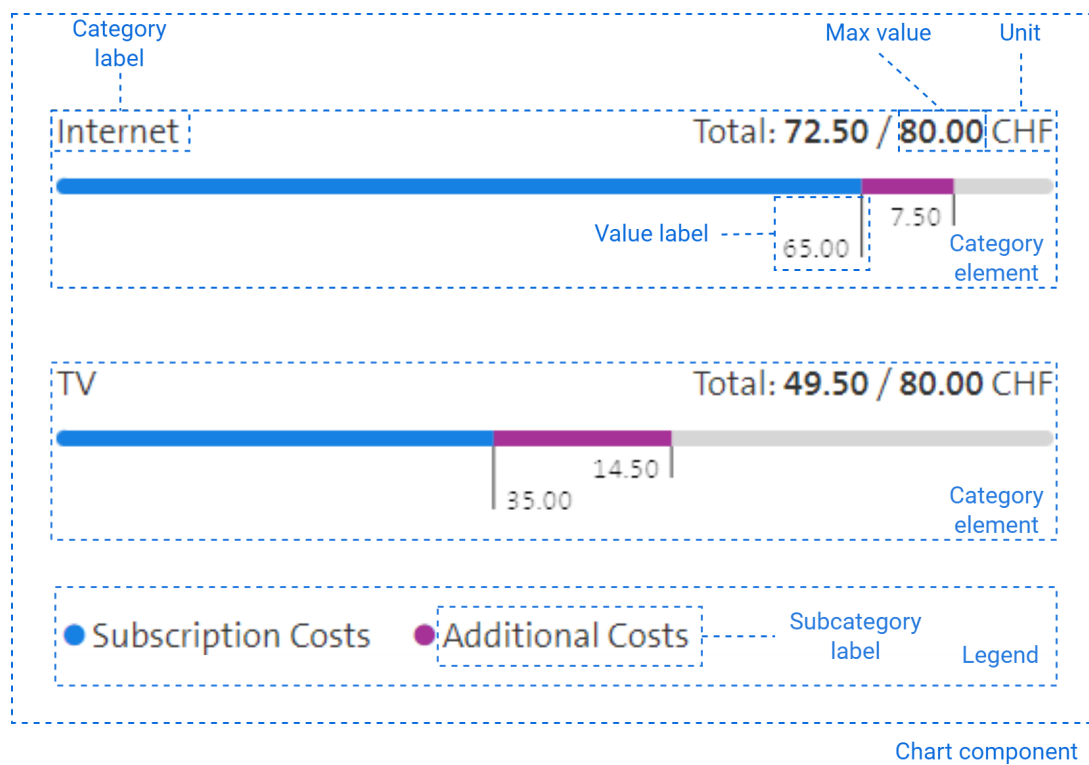
● Balance

Empty

Empty charts indicate with a zero label their empty state. No legend is displayed.



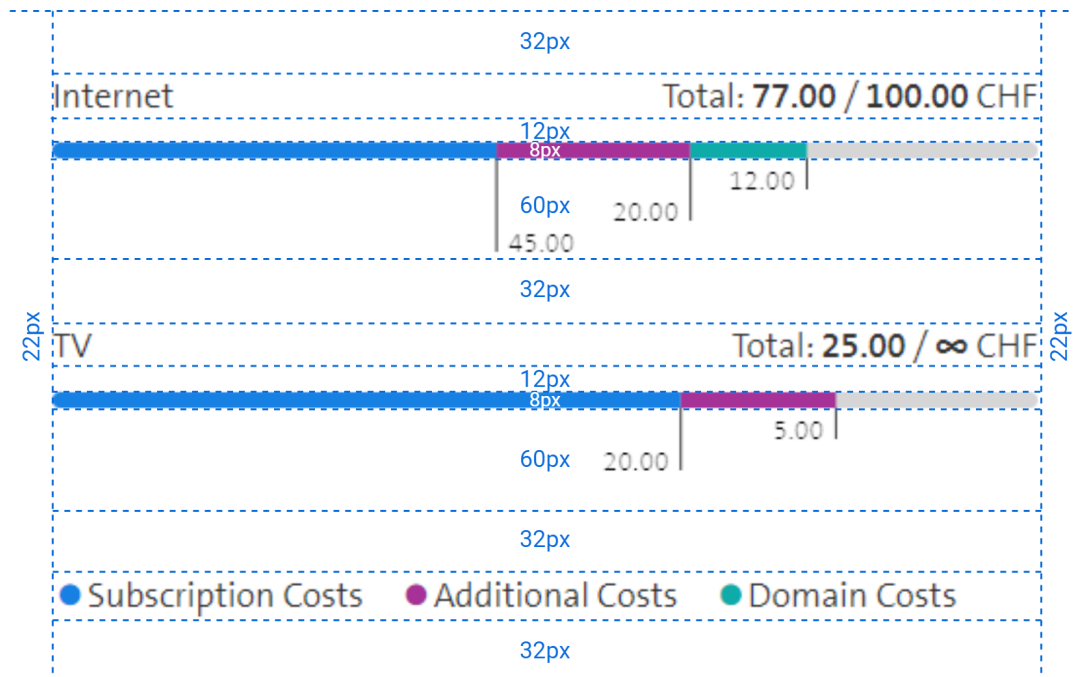
## Chart Elements



## Constraints

- Max 6 category elements per chart component
- Max 5 subcategories per chart component
- Max 3 subcategories per category element

## Spacing



## Colors

Colors used are the SDX accent colors. The order of color selection is:

1. Azure
2. Orchid
3. Turquoise
4. Pink
5. Iris

Note that this order only strictly applies to the legend. This is due to the fact that categories can consist of non-consecutive subcategory entries.

## Bar Label Behavior

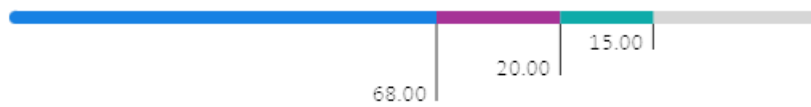
Labels are displayed in a stair format and adjust their height according to their ordering.

## Positive Chart

DEVELOPER-GUIDE.md

6/13/2023

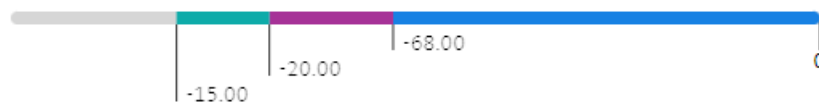
## Mobile Plan

Total: **103.00** /  $\infty$  CHF

● Subscription ● Additional ● Special

## Negative Chart

## Mobile Plan

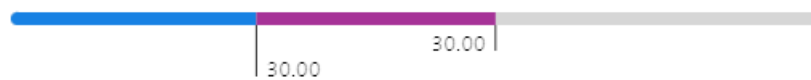
Total: **-103.00** /  $-\infty$  CHF

● Subscription ● Additional ● Special

## Orientation &amp; Overflow

To avoid overflows on the outer edges, labels switch their orientation depending on which chart half they are located on.

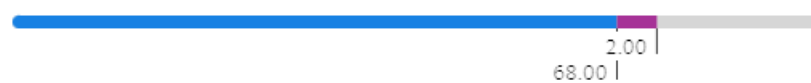
## Mobile Plan

Total: **60.00** / **100.00** CHF

● Subscription ● Additional

value labels possess a background color to remain readable when covering a label indicator line.

## Mobile Plan

Total: **70.00** /  $\infty$  CHF

● Subscription Cost ● Additional Cost

## Bar Sizing Behavior

If a subcategory value represents less than 5% of the sum of all subcategory values within a category, it will be displayed with a width of exactly 5%.

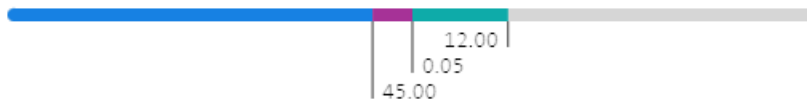


DEVELOPER-GUIDE.md

6/13/2023

Internet

Total: 57.05 / 100.00 CHF



● Subscription Costs ● Additional Costs ● Domain Costs

## Developers

### Properties

Documentation of all component properties are available as comments within the code:

</src/components/horizontal-bar-chart/horizontal-bar-chart.tsx>

### Unit

```
unit: string = ""
/**
 * Defines the unit of the displayed values:
 * unit="CHF"
 * --> 3 CHF
 * The default is "" (empty)
 */
```

### Precision

```
precision: number = 0
/**
 * Defines the number of decimal places of the displayed values:
 * precision="3"
 * --> 3.000 CHF
 * The default is "0"
 */
```

### Data

```
data: [] = []
/**
 * The data input as an array in a one dimensional JSON format:
 * '[{
 *   "value": 100,
 *   "category": "Internet",
```

DEVELOPER-GUIDE.md

6/13/2023

```
*   "subcategory": "Subscription",
*   "maxValue": 300
*   }]'
*/
```

A single JSON object and its preferred types

```
{
  "value": "number",
  "category": "string",
  "subcategory": "string",
  "maxValue": "number"
}
```

Object Key	Description
<code>value*</code>	Numerical value assigned to the defined subcategory
<code>category*</code>	Category name
<code>subcategory*</code>	Subcategory name (will also show up in the legend)
<code>maxValue</code>	Define the numeric limit for the sum of all subcategory values within a category. Must be declared on the first JSON object of a category. If omitted an infinite chart variation is created.

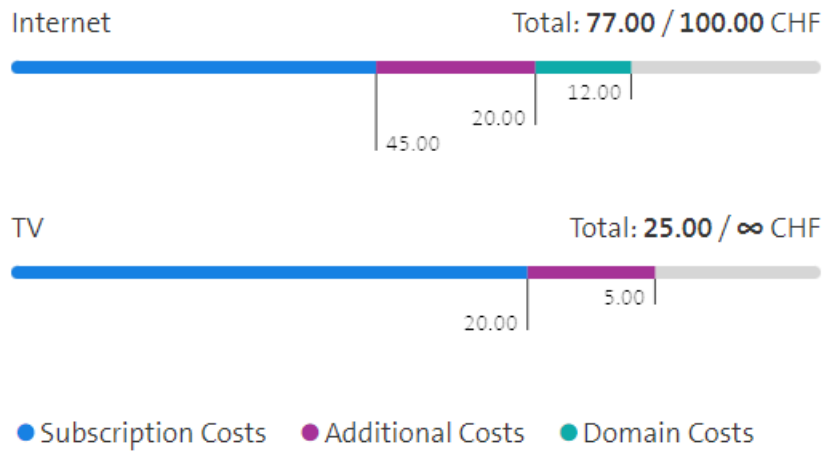
\* = mandatory

## Errors

The component is able to scan component properties for false types and input errors. A basic error message if an error is detected. The error is displayed in the web browser instead of a rendered output.

## Examples

Example 1

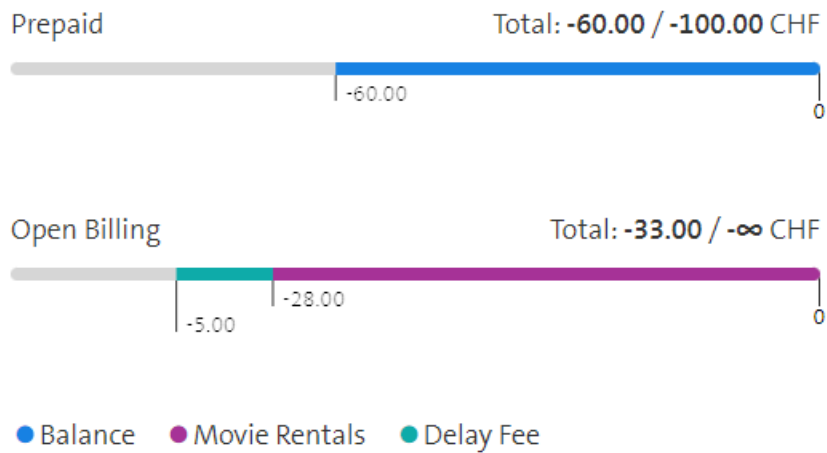


```

<sdx-horizontal-bar-chart
  unit="CHF"
  precision="2"
  data='[{
    "value": 45,
    "category": "Internet",
    "subcategory": "Subscription costs",
    "maxValue": 100
  }, {
    "value": 20,
    "category": "Internet",
    "subcategory": "Additional costs"
  }, {
    "value": 12,
    "category": "Internet",
    "subcategory": "Domain costs"
  }, {
    "value": 20,
    "category": "TV",
    "subcategory": "Subscription costs"
  }, {
    "value": 5,
    "category": "TV",
    "subcategory": "Additional costs"
  }]'
></sdx-horizontal-bar-chart>

```

Example 2



```
<sdx-horizontal-bar-chart
  unit="CHF"
  precision="2"
  data='[{
    "value": -60,
    "category": "Prepaid",
    "subcategory": "Balance",
    "maxValue": -100
  }, {
    "value": -28,
    "category": "Open billing",
    "subcategory": "Movie rentals"
  }, {
    "value": -5,
    "category": "Open billing",
    "subcategory": "Delay fee"
  }]'
>>/sdx-horizontal-bar-chart>
```

# Manual Testing Protocol

## Error behavior

Ref No.	Date	Actual Result	Planned Measures
2	25.05.23	An error message was displayed explaining the error.	none
3	25.05.23	An error message was displayed explaining the error.	none
4	25.05.23	An error message was displayed explaining the error.	none
5	25.05.23	An error message was displayed explaining the error.	none
6	25.05.23	An error message was displayed explaining the error.	none
1	11.06.23	An error message was displayed explaining the error.	none
2	11.06.23	An error message was displayed explaining the error.	none
3	11.06.23	An error message was displayed explaining the error.	none
4	11.06.23	An error message was displayed explaining the error.	none
5	11.06.23	An error message was displayed explaining the error.	none
6	11.06.23	An error message was displayed explaining the error.	none
1	25.05.23	An error message was displayed explaining the error.	none

Table C.1: Error behavior - Results

## Missing attributes/ Zero values behavior

Ref No.	Date	Actual Result	Planned Measures
1	25.05.23	As intended, the subcategory is not displayed within the bar, but the legend still lists the subcategory.	Evaluate if this is helpful or confusing.
2	25.05.23	As intended, the subcategories are not displayed within the bar, but the legend still lists the subcategories.	Evaluate if this is helpful or confusing.
3	25.05.23	The max value of the category is set to infinity.	none (This test was done before the Infinity/Zero change, therefore it was correct behavior)
4	25.05.23	A chart is shown with an empty category and sub category label. The chart crashed due to "Expected length got NaN%" error.	Evaluate and implement a better input data validation.
5	25.05.23	A chart is shown with an empty category and sub category label.	none
6	25.05.23	A chart is shown with an empty category and sub category label.	Evaluate and implement a better input data validation.
1	11.06.23	As intended, the subcategory is not displayed within the bar and the legend doesn't display it.	none
2	11.06.23	As intended, the subcategories are not displayed within the bar and the legend doesn't display them.	none
3	11.06.23	The max value of the category is set to zero.	none
4	11.06.23	An error message is shown displaying that the data isn't valid.	none
5	11.06.23	A chart is shown with an empty category and sub category label.	none
6	11.06.23	An error message is shown displaying that the data isn't valid.	none

Table C.2: Zero Values - Results

**Maximum/Infinity behavior**

Ref No.	Date	Actual Result	Planned Measures
1	25.05.23	The max value of the category is set to infinity	none
2	25.05.23	The chart doesn't render. The chart crashes with the error "Expected length got NaN%"	Evaluate and implement a better input data validation.
3	25.05.23	The total is colored pink and the bar is filled.	none
4	25.05.23	The end of the bar is round.	none
5	25.05.23	The bar is not exactly at 80%.	Correct it to exactly 80%.
1	11.06.23	The max value of the category is set to infinity	none
2	11.06.23	An error message is shown displaying that the data isn't valid.	none
3	11.06.23	The bar is filled.	none
4	11.06.23	The end of the bar is round.	none
5	11.06.23	The bar is exactly at 80%.	none

Table C.3: Maximum/Infinity behavior - Results

**Negative behavior**

Ref No.	Date	Actual Result	Planned Measures
1	25.05.23	The bar animation starts at the right side.	none
2	25.05.23	The zero label is visible.	none
3	25.05.23	The label values have a minus sign.	none
1	11.06.23	The bar animation starts at the right side.	none
2	11.06.23	The zero label is visible.	none
3	11.06.23	The label values have a minus sign.	none

Table C.4: Negative behavior - Results

**Other**

Ref No.	Date	Actual Result	Planned Measures
1	25.05.23	The two subcategories are shown as one and the values are summed.	none
2	25.05.23	The chart doesn't update and doesn't get re-rendered.	Fix update functionality
3	25.05.23	The animation plays for each category/bar correctly.	none
4	25.05.23	The bar is responsive, but the labels cut off at the edges.	Fix labels
5	25.05.23	The colors are matching with the subcategories.	none
6	25.05.23	The legend gets animated and shows the correct colors for the subcategories.	none
7	25.05.23	The legend sticks together and the title wraps around. The total label doesn't wrap around.	Fix total label
8	25.05.23	All the labels are capitalized at the beginning of each word.	none
9	25.05.23	Title, legend, values and total are correctly labeled.	none
1	11.06.23	The two subcategories are shown as one and the values are summed.	none
2	11.06.23	The chart updates and re-renders without reloading the site.	none
3	11.06.23	The animation plays for each category/bar correctly.	none
4	11.06.23	The bar is responsive and all the labels are visible.	none
5	11.06.23	The colors are matching with the subcategories.	none
6	11.06.23	The legend gets animated and shows the correct colors for the subcategories.	none
7	11.06.23	The legend sticks together and the title wraps around. The total label wraps around.	none
8	11.06.23	All the labels are capitalized at the beginning of each word.	none
9	11.06.23	Title, legend, values and total are correctly labeled.	none

Table C.5: Other - Results

**Normal behavior**



<b>Ref No.</b>	<b>Date</b>	<b>Actual Result</b>	<b>Planned Measures</b>
1	25.05.23	Chart is shown correctly with 2 categories and 4 subcategories.	none
2	25.05.23	Chart is shown correctly with 1 category and 1 subcategory.	none
3	25.05.23	Chart is shown correctly with 6 categories and 5 subcategories.	none
4	25.05.23	Chart is shown correctly, but the infinity category bar is completely filled.	Fix infinity display behavior
1	11.06.23	Chart is shown correctly with 2 categories and 4 subcategories.	none
2	11.06.23	Chart is shown correctly with 1 category and 1 subcategory.	none
3	11.06.23	Chart is shown correctly with 6 categories and 5 subcategories.	none
4	11.06.23	Chart is shown correctly and the infinity bar is only at 80%.	none

Table C.6: Normal behavior - Results

# UX Testing Protocols

## D.1 UX Test on May 23, 2023

The scenarios have been played through by a candidate representing a potential user.

Ref No.	Result	Special Observations	Planned Measures
1	The user answered all questions correctly.	The user was confused why the infinity bar was not completely filled with color.	None
2	The user answered all questions correctly.	The user tried to hover over the bar with the mouse.	Evaluate simple tooltip / hover option
3	The user answered all questions correctly.	The user suggested that an exceeded maximum could be displayed more clearly.	Evaluate exceeded limit with stakeholder
4	The user answered all questions correctly.	None	None
5	The user answered all questions correctly.	The labels at the edges were not fully visible.	Correct label displaying behavior
6	The user answered all questions correctly.	The user had to scroll back and forth to connect subcategories to their colors.	Evaluate simple hover or sticky legend

Table D.1: UX test results May, 23 2023

**Overall Feedback:** Overall, the user had a good understanding of what each chart was showing. The user was able to identify all the important information. A few small visual things were not clear at first glance.

## D.2 UX Test on May 31, 2023

The scenarios have been played through by a candidate representing a potential user.

Ref No.	Result	Special Observations	Planned Measures
1	The user answered all questions correctly.	The user had to check twice to understand that the bar is an infinity bar.	None
2	The user answered all questions correctly.	None	None
3	The user answered all questions correctly.	Exceeded maximum was not clear for the user.	In agreement with the stakeholder, it is not the task of the bar chart to notify the user for an exceeded limit.
4	The user answered all questions correctly.	None	None
5	The user answered all questions correctly.	The different label orientations were not intuitive at first sight.	None - the swapping orientation is a trade-off for visibility.
6	The user answered all questions correctly.	Some color combinations were hard to distinguish.	None - the color scheme is a constraint.

Table D.2: UX test results May 31, 2023

**Overall Feedback:** The user quickly understood each chart and its purpose. The user was able to gather all the necessary information. Only a few visual things were not clear at the beginning.

### D.3 UX Test on June 02, 2023

The scenarios have been played through by a candidate representing a potential user.

Ref No.	Result	Special Observations	Planned Measures
1	The user answered all questions correctly.	User confused by due to the fact that the chart is not finite. Infinity can usually not be displayed.	Infinite charts are not intuitive but needed in this use case.
2	The user answered all questions correctly.	-	-
3	The user answered all questions correctly.	If the max value is breached, there is no indication.	This is expected behavior. The developer is responsible to handle overflows outside the component.
4	The user answered all questions correctly.	The negative chart is not intuitive. Hard to read for the user. Why would one need such a chart.	Is a functional requirement of the stakeholder. For readability purposes a zero label and negative characters were added.
5	The user answered all questions correctly.	Combination of negative and positive elements is confusing. Zero is only displayed on the negative version. Value label indicators switch sides, which can lead to wrong value to sub-bar assignments.	The developer guide advises to not mix negative and positive charts. Regarding the label indicators the text orientation is a necessary precaution to avoid overflows near the component edge.
6	The user answered all questions correctly.	-	-

Table D.3: UX test results June 02, 2023

**Overall Feedback:** The test user is able to read information from the chart. The negative and infinite chart variations seem unintuitive. If a combination of negative and positive chart variations is used reading the data becomes difficult.

## D.4 UX Test on June 09, 2023

The scenarios have been played through by a candidate representing a potential user.

Ref No.	Result	Special Observations	Planned Measures
1	The user answered all questions correctly.	-	-
2	The user answered all questions correctly.	-	-
3	The user answered all questions correctly.	Overflow is noticed after further inspection.	-
4	The user answered all questions correctly.	-	-
5	The user answered all questions correctly.	Mix of negative and positive chart variations is confusing.	The developer guide specifies to not mix negative and positive chart variations.
6	The user answered all questions correctly.	Infinity charts stop at approximately 80%, which does not represent an infinite value. The pink color leads to the assumption of an error.	The color pink is a design standard of Swisscom AG. There is no feasible method to represent infinity in a bar chart.

Table D.4: UX test results June 09, 2023

**Overall Feedback:** All data is displayed and can be read with ease. The pink color can lead to false assumptions about the chart. No needed interaction with the chart improves the user experience.

# Usability Testing Protocols

## E.1 Usability Test on May 23, 2023

The scenarios have been played through by a developer.

**Developer Guide Results:** The attribute description for the "value" attribute was not clear, and the developer still was not sure what the "value" attribute was after reading it several times. The different terminology for bars, sub-bars, categories and sub-categories was very confusing to the developer. The order of categories and subcategories was not explained.

**Planned Measures:** Update guide with clear terminology and descriptions.

**Scenario Results:**

Ref No.	Result	Special Observations	Planned Measures
1	The candidate successfully completed the scenario.	At first, the developer had some problems figuring out how it all worked just from the manual. Then the developer used the example provided in the developer's guide as a starting point.	Rework guide.
2	The candidate successfully completed the scenario.	None	None
3	The candidate successfully completed the scenario.	The developer tested with negative/positive values and the corresponding error message appeared. It was clear to the developer what had gone wrong.	None
4	The candidate successfully completed the scenario.	Labels were not fully visible for the developer.	Fix label alignment

Table E.1: Usability test results May, 23 2023

**Overall Feedback:** The candidate completed each task with success. The developer guide was used as an auxiliary tool.

## E.2 Usability Test on May 31, 2023

The scenarios have been played through by a developer.

**Developer Guide Results:** The hierarchy for category and subcategory and the implicit ordering was not clear to the developer. The described precision behavior differed from the actual behavior, as did the behavior for the maximum value.

**Scenario Results:**

Ref No.	Result	Special Observations	Planned Measures
1	The candidate successfully completed the scenario.	The developer used the guide as an entry point and playground to understand the chart.	None
2	The candidate successfully completed the scenario.	None	None
3	The candidate successfully completed the scenario.	The developer defined a maximum value for the second subcategory in a category and was confused as to why the maximum value was set to infinity.	Rework guide with clearer descriptions.
4	The candidate successfully completed the scenario.	The developer was confused as to why the precision behavior was rounding its precision instead of just cutting off the decimal places.	Rework guide to describe the intended behavior.

Table E.2: Usability test results May, 31 2023

**Overall Feedback:** The candidate successfully completed each task. The developer's guide was used as an auxiliary tool. The candidate quickly understood how the chart worked and completed each successive task more quickly.

### E.3 Usability Test on June 02, 2023

The scenarios have been played through by a developer.

**Developer Guide Results:** Property description is sufficient. The data attribute is difficult to understand and needs detailed description. Examples with code templates are helpful and enable the user to work more efficiently.

**Scenario Results:**

Ref No.	Result	Special Observations	Planned Measures
1	The candidate successfully completed the scenario.	The JSON format can be difficult to handle.	Provide templates for users.
2	The candidate successfully completed the scenario.	-	-
3	The candidate successfully completed the scenario.	After mishaps the user debugged with the help of the error code.	-
4	The candidate successfully completed the scenario.	Some scenario steps are complicated and too long.	Break scenario steps into smaller tasks.

Table E.3: Usability test results June, 02 2023

**Overall Feedback:** Each task was completed with success. The developer guide is still not complete and needs further revision. In general, the component is easy to operate after a certain period of familiarization.

## E.4 Usability Test on June 09, 2023

The scenarios have been played through by a developer.

**Developer Guide Results:** The difference between the category and subcategory is confusing and needs further explanation. Code templates and images with details about component element names and position is appreciated and useful.

### Scenario Results:

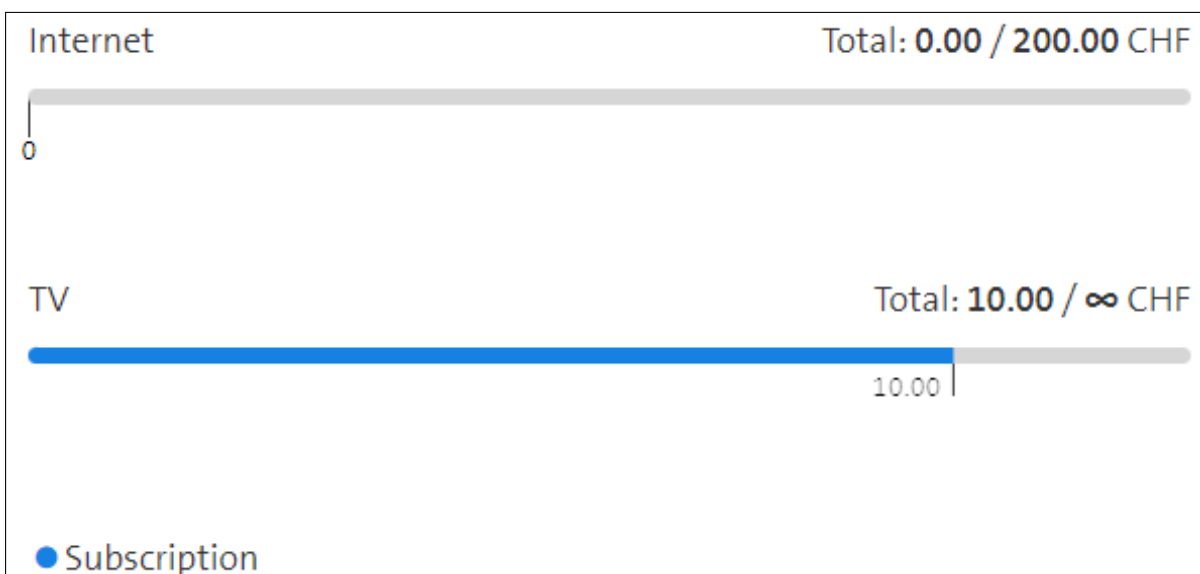
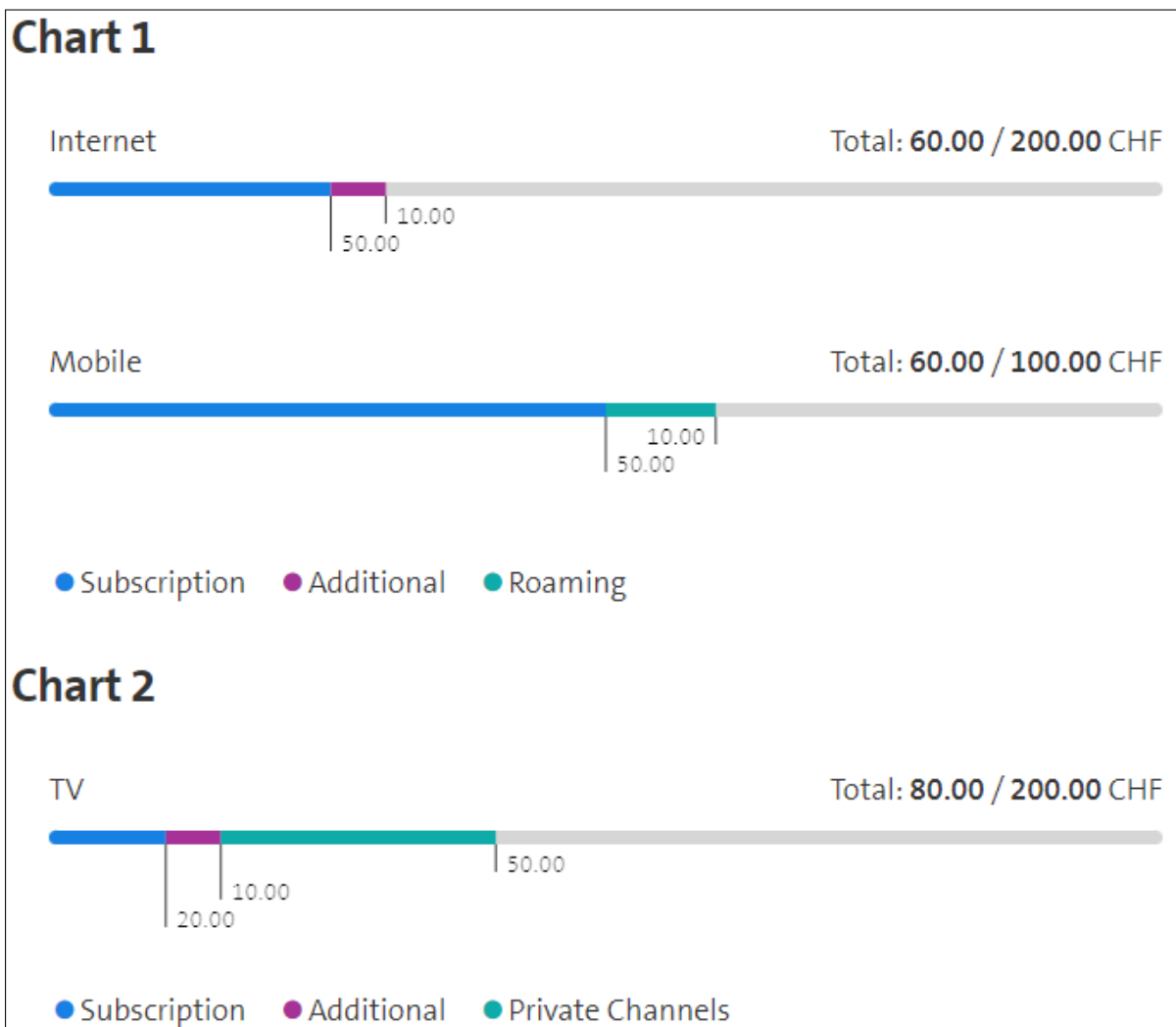
Ref No.	Result	Special Observations	Planned Measures
1	The candidate successfully completed the scenario.	Some properties were written into the data object array.	False attributes are discarded in the data object array. An improved explanation in the developer guide is advised.
2	The candidate successfully completed the scenario.	-	-
3	The candidate successfully completed the scenario.	Applied the negative chart mechanism with ease.	-
4	The candidate successfully completed the scenario.	-	-

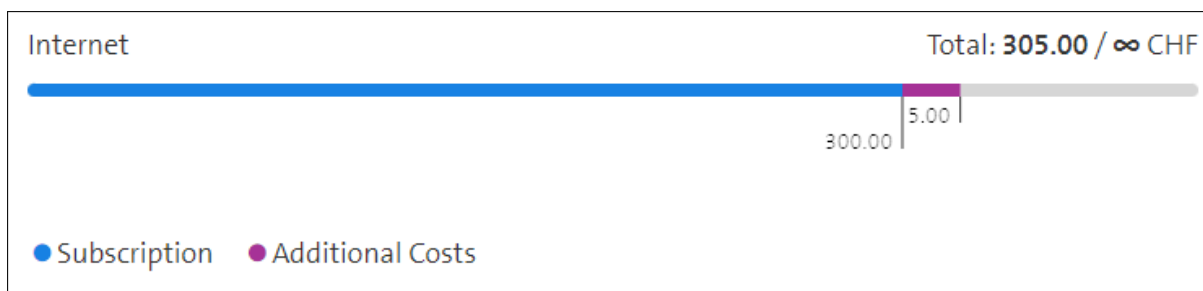
Table E.4: Usability test results June, 09 2023

**Overall Feedback:** The component is easy to use due to the fact that properties are explained well. The JSON format is handy, but it is advised to provide input templates.

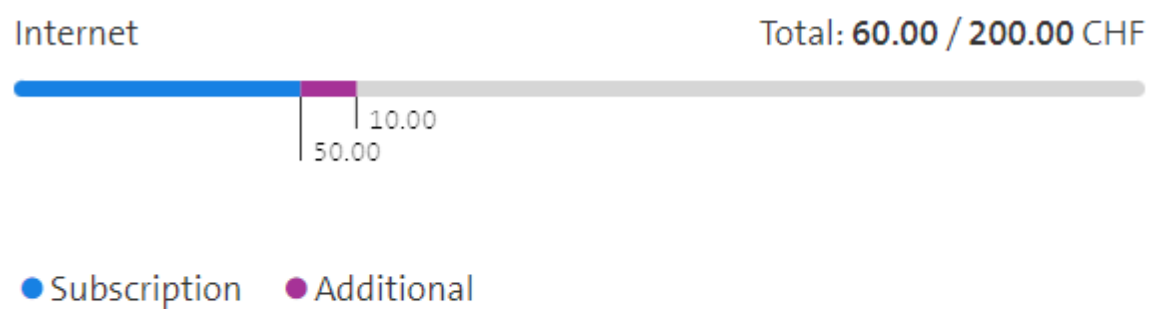


# Product Screenshots

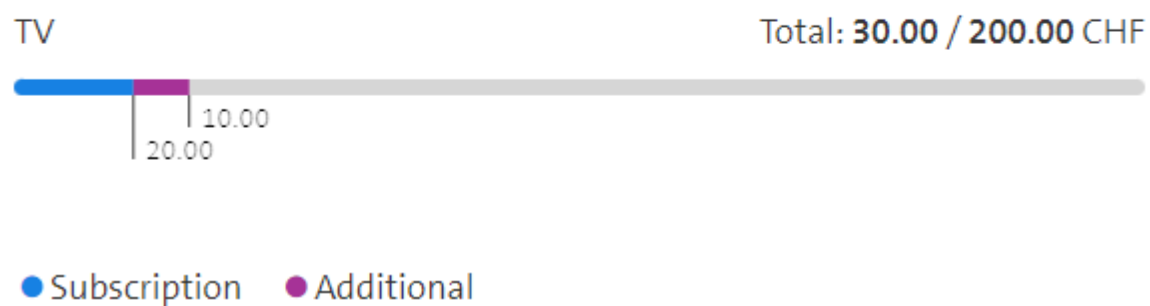


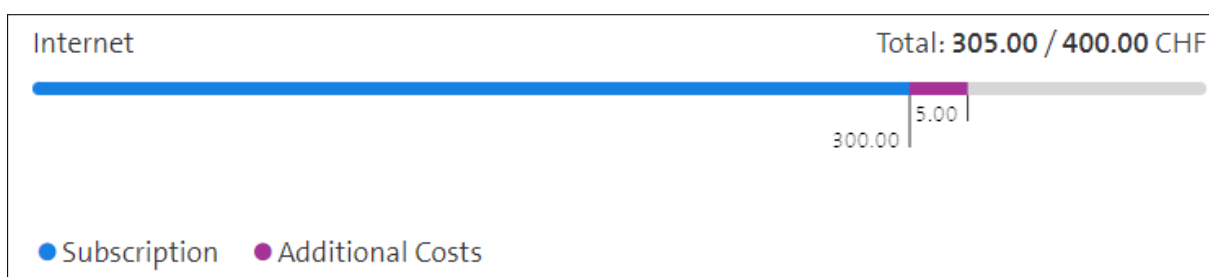
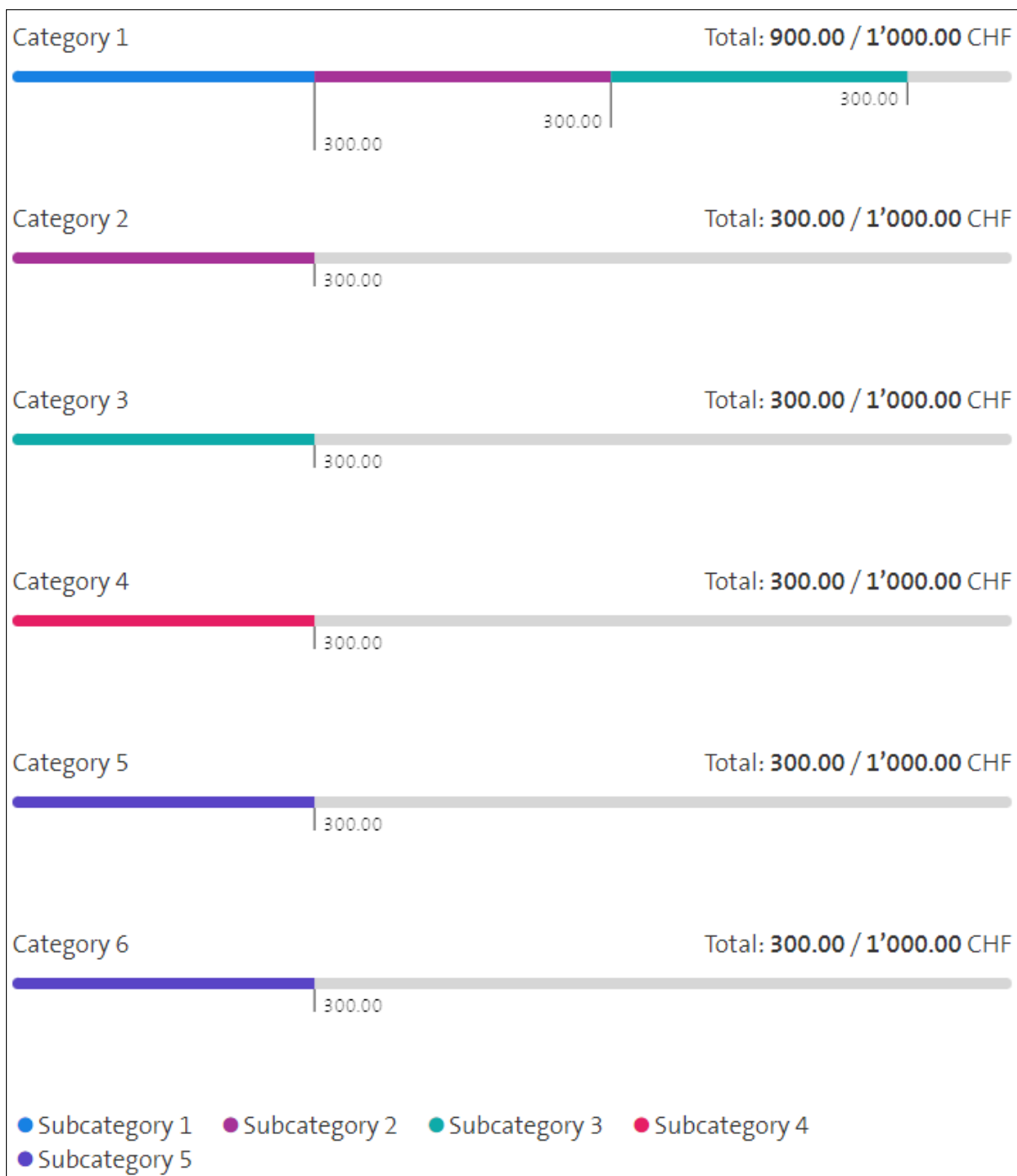


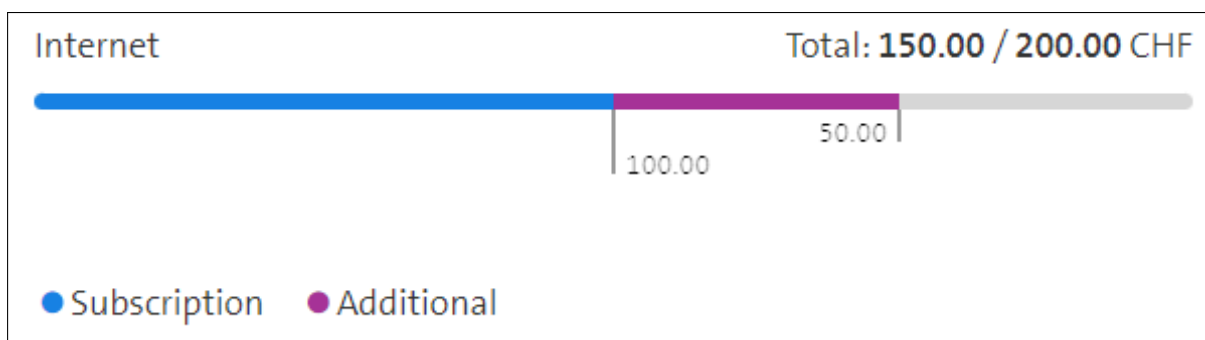
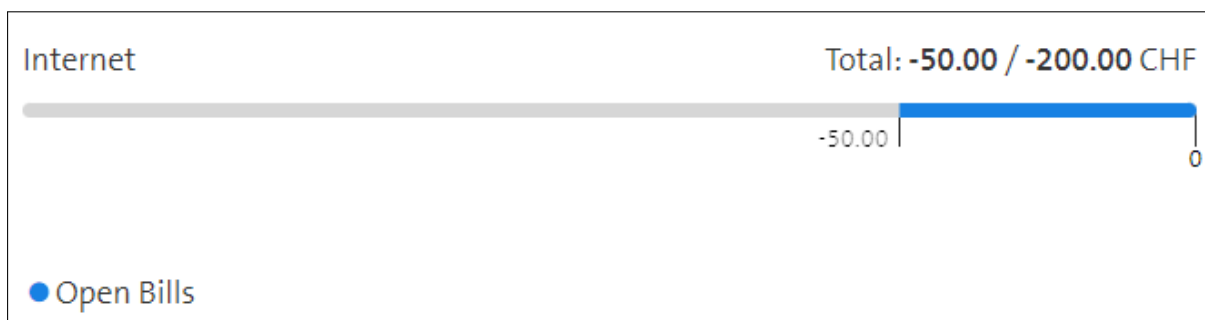
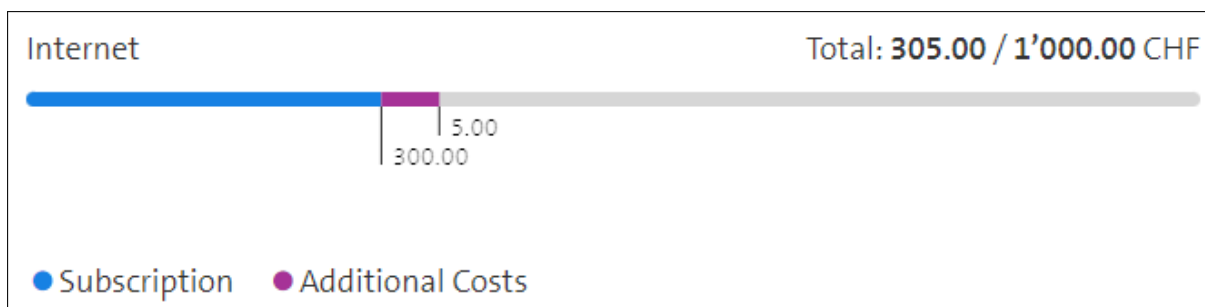
## Chart 1



## Chart 2







# Licenses

## G.1 Animate.css License

Animate.css Copyright 2021 Daniel Eden ("Licensor")

Hippocratic License Version Number: 2.1.

**Purpose.** The purpose of this License is for the Licensor named above to permit the Licensee (as defined below) broad permission, if consistent with Human Rights Laws and Human Rights Principles (as each is defined below), to use and work with the Software (as defined below) within the full scope of Licensor's copyright and patent rights, if any, in the Software, while ensuring attribution and protecting the Licensor from liability.

**Permission and Conditions.** The Licensor grants permission by this license ("License"), free of charge, to the extent of Licensor's rights under applicable copyright and patent law, to any person or entity (the "Licensee") obtaining a copy of this software and associated documentation files (the "Software"), to do everything with the Software that would otherwise infringe (i) the Licensor's copyright in the Software or (ii) any patent claims to the Software that the Licensor can license or becomes able to license, subject to all of the following terms and conditions:

\* **Acceptance.** This License is automatically offered to every person and entity subject to its terms and conditions. Licensee accepts this License and agrees to its terms and conditions by taking any action with the Software that, absent this License, would infringe any intellectual property right held by Licensor.

\* **Notice.** Licensee must ensure that everyone who gets a copy of any part of this Software from Licensee, with or without changes, also receives the License and the above copyright notice (and if included by the Licensor, patent, trademark and attribution notice). Licensee must cause any modified versions of the Software to carry prominent notices stating that Licensee changed the Software. For clarity, although Licensee is free to create modifications of the Software and distribute only the modified portion created by Licensee with additional or different terms, the portion of the Software not modified must be distributed pursuant to this License. If anyone notifies Licensee in writing that Licensee has not complied with this Notice section, Licensee can keep this License by taking all practical steps to comply within 30 days after the notice. If Licensee does not do so, Licensee's License (and all rights licensed hereunder) shall end immediately.

\* **Compliance with Human Rights Principles and Human Rights Laws.**

1. Human Rights Principles.

- (a) Licensee is advised to consult the articles of the United Nations Universal Declaration of Human Rights and the United Nations Global Compact that define recognized principles of international human rights (the "Human Rights Principles"). Licensee shall use the Software in a manner consistent with Human Rights Principles.

- (b) Unless the Licensor and Licensee agree otherwise, any dispute, controversy, or claim arising out of or relating to (i) Section 1(a) regarding Human Rights Principles, including the breach of Section 1(a), termination of this License for breach of the Human Rights Principles, or invalidity of Section 1(a) or (ii) a determination of

whether any Law is consistent or in conflict with Human Rights Principles pursuant to Section 2, below, shall be settled by arbitration in accordance with the Hague Rules on Business and Human Rights Arbitration (the "Rules"); provided, however, that Licensee may elect not to participate in such arbitration, in which event this License (and all rights licensed hereunder) shall end immediately. The number of arbitrators shall be one unless the Rules require otherwise.

Unless both the Licensor and Licensee agree to the contrary: (1) All documents and information concerning the arbitration shall be public and may be disclosed by any party; (2) The repository referred to under Article 43 of the Rules shall make available to the public in a timely manner all documents concerning the arbitration which are communicated to it, including all submissions of the parties, all evidence admitted into the record of the proceedings, all transcripts or other recordings of hearings and all orders, decisions and awards of the arbitral tribunal, subject only to the arbitral tribunal's powers to take such measures as may be necessary to safeguard the integrity of the arbitral process pursuant to Articles 18, 33, 41 and 42 of the Rules; and (3) Article 26(6) of the Rules shall not apply.

2. Human Rights Laws. The Software shall not be used by any person or entity for any systems, activities, or other uses that violate any Human Rights Laws. "Human Rights Laws" means any applicable laws, regulations, or rules (collectively, "Laws") that protect human, civil, labor, privacy, political, environmental, security, economic, due process, or similar rights; provided, however, that such Laws are consistent and not in conflict with Human Rights Principles (a dispute over the consistency or a conflict between Laws and Human Rights Principles shall be determined by arbitration as stated above). Where the Human Rights Laws of more than one jurisdiction are applicable or in conflict with respect to the use of the Software, the Human Rights Laws that are most protective of the individuals or groups harmed shall apply.

3. Indemnity. Licensee shall hold harmless and indemnify Licensor (and any other contributor) against all losses, damages, liabilities, deficiencies, claims, actions, judgments, settlements, interest, awards, penalties, fines, costs, or expenses of whatever kind, including Licensor's reasonable attorneys' fees, arising out of or relating to Licensee's use of the Software in violation of Human Rights Laws or Human Rights Principles.

\* Failure to Comply. Any failure of Licensee to act according to the terms and conditions of this License is both a breach of the License and an infringement of the intellectual property rights of the Licensor (subject to exceptions under Laws, e.g., fair use). In the event of a breach or infringement, the terms and conditions of this License may be enforced by Licensor under the Laws of any jurisdiction to which Licensee is subject. Licensee also agrees that the Licensor may enforce the terms and conditions of this License against Licensee through specific performance (or similar remedy under Laws) to the extent permitted by Laws. For clarity, except in the event of a breach of this License, infringement, or as otherwise stated in this License, Licensor may not terminate this License with Licensee.

\* Enforceability and Interpretation. If any term or provision of this License is determined to be invalid, illegal, or unenforceable by a court of competent jurisdiction, then such invalidity, illegality, or

unenforceability shall not affect any other term or provision of this License or invalidate or render unenforceable such term or provision in any other jurisdiction; provided, however, subject to a court modification pursuant to the immediately following sentence, if any term or provision of this License pertaining to Human Rights Laws or Human Rights Principles is deemed invalid, illegal, or unenforceable against Licensee by a court of competent jurisdiction, all rights in the Software granted to Licensee shall be deemed null and void as between Licensor and Licensee. Upon a determination that any term or provision is invalid, illegal, or unenforceable, to the extent permitted by Laws, the court may modify this License to affect the original purpose that the Software be used in compliance with Human Rights Principles and Human Rights Laws as closely as possible. The language in this License shall be interpreted as to its fair meaning and not strictly for or against any party.

\* Disclaimer. TO THE FULL EXTENT ALLOWED BY LAW, THIS SOFTWARE COMES "AS IS," WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, AND LICENSOR AND ANY OTHER CONTRIBUTOR SHALL NOT BE LIABLE TO ANYONE FOR ANY DAMAGES OR OTHER LIABILITY ARISING FROM, OUT OF, OR IN CONNECTION WITH THE SOFTWARE OR THIS LICENSE, UNDER ANY KIND OF LEGAL CLAIM.

This Hippocratic License is an Ethical Source license (<https://ethicalsource.dev>) and is offered for use by licensors and licensees at their own risk, on an "AS IS" basis, and with no warranties express or implied, to the maximum extent permitted by Laws.

## G.2 Anime.js License

The MIT License

Copyright (c) 2019 Julian Garnier

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



### G.3 Beautifier License

The MIT License (MIT)

Copyright (c) 2007-2018 Einar Lielmanis, Liam Newman, and contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## G.4 Chai License

MIT License

Copyright (c) 2017 Chai.js Assertion Library

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## G.5 ESLint License

Copyright OpenJS Foundation and other contributors, <www.openjsf.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## G.6 Jest License

MIT License

Copyright (c) Meta Platforms, Inc. and affiliates.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## G.7 JSLint License

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to [<https://unlicense.org/>](https://unlicense.org/)

## G.8 Less License

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to

communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or

documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf



of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

## G.9 Lit License

BSD 3-Clause License

Copyright (c) 2017 Google LLC. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## G.10 Mocha License

(The MIT License)

Copyright (c) 2011-2022 OpenJS Foundation and contributors,  
<https://openjsf.org>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## G.11 Polymer License

```
// Copyright (c) 2017 The Polymer Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
// * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
// * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
// * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

## G.12 Prettier License

Copyright © James Long and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## G.13 Puppeteer License

Apache License  
Version 2.0, January 2004  
<https://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally

submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory,



whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[ ]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright 2017 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<https://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

## G.14 Sass License

Copyright (c) 2016, Google Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## G.15 Sinon.JS License

(The BSD License)

Copyright (c) 2010–2017, Christian Johansen, christian@cjohansen.no  
All rights reserved.

Redistribution and use in source and binary forms, with or without  
modification,  
are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright  
notice,  
this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright  
notice,  
this list of conditions and the following disclaimer in the  
documentation  
and/or other materials provided with the distribution.
- \* Neither the name of Christian Johansen nor the names of his  
contributors  
may be used to endorse or promote products derived from this  
software  
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS  
IS" AND  
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
IMPLIED  
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE  
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE  
LIABLE  
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
CONSEQUENTIAL  
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
OR  
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)  
HOWEVER  
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT  
LIABILITY,  
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE  
USE OF  
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## G.16 Slim.js License

MIT License

Copyright (c) 2016 Eyal Avichay <eavichay@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## G.17 Stencil License

The MIT License (MIT)

Copyright (c) 2019-present Drifty Co.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

-----  
Third-party components, software, and/or libraries (collectively, "components") are included under the licenses specified by their authors. For information about these third-party licenses, refer to the separate legal notices governing such components at NOTICE file at the top level of the package.

## G.18 Stylus License

(The MIT License)

Copyright (c) Automattic <developer.wordpress.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## List of Tables

2.1	NFR maintainability 01	13
2.2	NFR maintainability 02	13
2.3	NFR maintainability 03	14
2.4	NFR efficiency 01	14
2.5	NFR portability 01	14
2.6	NFR usability 01	14
3.1	Design decisions	23
3.2	Contrasts	25
3.3	Utility analysis Stencil	26
3.4	Utility analysis slim.js	26
3.5	Utility analysis Polymer	27
3.6	Utility analysis Lit	27
3.7	Utility analysis Jest	28
3.8	Utility analysis Puppeteer	28
3.9	Utility analysis Mocha	29
3.10	Utility analysis Sinon.JS	29
3.11	Utility analysis Chai	30
3.12	Utility analysis anime.js	30
3.13	Utility analysis Animate.css	31
3.14	Utility analysis Sass	31
3.15	Utility analysis Stylus	32
3.16	Utility analysis Less	32
3.17	Utility analysis ESLint	33
3.18	Utility analysis JSLint	33
3.19	Utility analysis Prettier	34
3.20	Utility analysis js-beautify	34
3.21	Web component technology evaluation	35
3.22	Testing technology evaluation	35
3.23	Animation technology evaluation	35
3.24	Styling technology evaluation	35
3.25	Linting technology evaluation	36
3.26	Formatting technology evaluation	36
3.27	Technologies	37
3.28	Prototype problem factors	39
4.1	Line count	44
4.2	Error behavior	45
4.3	Missing attributes and zero values	45
4.4	Maximum and infinity behavior	46
4.5	Negative behavior	46
4.6	Other	46
4.7	Normal behavior	47
4.8	Usability tests scenarios	48
4.9	UX tests scenarios	49
6.1	Accessibility expansion overview	64
6.2	Data sorting expansion overview	65

---

6.3	Filter and search expansion overview . . . . .	65
6.4	End-to-End test expansion overview . . . . .	66
6.5	Internationalization expansion overview . . . . .	66
6.6	Line Chart expansion overview . . . . .	66
7.1	Milestone definitions . . . . .	69
8.1	Role assignment . . . . .	70
8.2	Meetings . . . . .	71
9.1	Documentation tools . . . . .	73
9.2	Code tools . . . . .	73
10.1	Risk identification . . . . .	74
10.2	Risk matrix . . . . .	74
10.3	Risk Mitigation and Prevention . . . . .	76
C.1	Error behavior - Results . . . . .	90
C.2	Zero Values - Results . . . . .	91
C.3	Maximum/Infinity behavior - Results . . . . .	92
C.4	Negative behavior - Results . . . . .	92
C.5	Other - Results . . . . .	93
C.6	Normal behavior - Results . . . . .	94
D.1	UX test results May, 23 2023 . . . . .	95
D.2	UX test results May 31, 2023 . . . . .	96
D.3	UX test results June 02, 2023 . . . . .	97
D.4	UX test results June 09, 2023 . . . . .	98
E.1	Usability test results May, 23 2023 . . . . .	99
E.2	Usability test results May, 31 2023 . . . . .	100
E.3	Usability test results June, 02 2023 . . . . .	101
E.4	Usability test results June, 09 2023 . . . . .	101



# List of Figures

1	Chart variations	ii
2	Approach overview	iii
3	Chart initialization comparison	iv
4	Horizontal bar chart web component	iv
1.1	Existing chart variations	2
1.2	Limited horizontal bar chart produced by Listing 1.1	4
1.3	Vertical bar chart from Listing 1.2	5
1.4	Vertical bar chart desktop vs. mobile view	6
1.5	Excerpt from horizontal bar chart designer rules	6
2.1	ISO 9126 [2]	9
2.2	Persona developer	10
2.3	Persona end-user	10
3.1	High level overview	16
3.2	Domain model	17
3.3	C4 model level 3 container	18
3.4	UX mockups	24
3.5	Color scheme	25
3.6	Prototype overview	38
3.7	Prototype SVG box	39
3.8	Prototype marker in Chrome	40
4.1	Folder structure	42
4.2	Test coverage report	44
6.1	Result legend	53
6.2	Result gray background bar	54
6.3	Result labelling across charts	55
6.4	Result color order	56
6.5	Result data structure	57
6.6	Result negative values	57
6.7	Result horizontal bar chart	58
6.8	Result unit and precision	58
6.9	Result minimum length	59
6.10	Result infinity behavior	59
6.11	Result maximum value behavior	60
6.12	Result stacked chart	61
6.13	Performance legacy horizontal bar chart component	62
6.14	Performance horizontal bar chart web component	63
6.15	Accessibility horizontal bar chart web component	63
7.1	Project plan	68
10.1	Risk Evolution Graph	75
11.1	Time tracking by labels	77
11.2	Time tracking by semester week	78

## List of Listings

1.1	Limited horizontal bar chart HTML initialization	4
1.2	Vertical bar chart HTML initialization	5
3.1	JSON data structure	20
3.2	Component decorator	21
3.3	Prop decorators	21
3.4	Watch decorator	21

## Bibliography

- [1] ISO/IEC, *Iso/iec 9126. software engineering – product quality*, 2001.
- [2] W. Contributors. “Iso 9126 quality(en) svg,” Wikipedia. (May 2, 2023), [Online]. Available: [https://en.wikipedia.org/wiki/ISO/IEC\\_9126#/media/File:ISO\\_9126\\_quality\\_\(en\).svg](https://en.wikipedia.org/wiki/ISO/IEC_9126#/media/File:ISO_9126_quality_(en).svg) (visited on 05/02/2023).
- [3] M. J. Hamm, *Wireframing Essentials*, 1st ed. Birmingham: Packt Publishing, Limited, 2014, p. 112, ISBN: 9781849698559.
- [4] HubSpot. “Make my persona - free buyer persona generator (2023),” HubSpot Inc. (May 14, 2023), [Online]. Available: <https://www.hubspot.com/make-my-persona> (visited on 05/14/2023).
- [5] O. Zimmermann. “Architectural decisions – the making of.” (Mar. 2, 2021), [Online]. Available: <https://ozimmer.ch/practices/2020/04/27/ArchitectureDecisionMaking.html> (visited on 11/04/2023).
- [6] A. Kirk, *Data Visualization : : a successful design process. eng*, 1st ed. Birmingham: Packt Publishing, Limited, 2012, ISBN: 9781849693479.
- [7] E. Swires-Hennessy, *Presenting Data : : How to Communicate Your Message Effectively. eng*, 1st ed. New York: John Wiley and Sons, Incorporated, 2014, ISBN: 9781118901861.
- [8] S. Few, *Show Me the Numbers: Designing Tables and Graphs to Enlighten*, English, Hardcover. Analytics Press, Sep. 1, 2004, p. 280, ISBN: 9780970601995.
- [9] M. contributors. “Web components - web apis | mdn,” Mozilla. (Jun. 5, 2023), [Online]. Available: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_components](https://developer.mozilla.org/en-US/docs/Web/API/Web_components) (visited on 06/05/2023).
- [10] Ionic. “Component api,” Ionic. (Apr. 13, 2023), [Online]. Available: <https://stenciljs.com/docs/api> (visited on 04/13/2023).
- [11] S. AG. “All the basics,” Swisscom AG. (Apr. 21, 2023), [Online]. Available: <https://sdx.swisscom.ch> (visited on 04/21/2023).
- [12] S. Krug, *Don't Make Me Think, Revisited - A Common Sense Approach to Web Usability. eng*. London: New Riders, 2014, ISBN: 9780321965516.
- [13] Ionic. “A compiler for web components,” Ionic. (Apr. 13, 2023), [Online]. Available: <https://stenciljs.com/docs/introduction> (visited on 04/13/2023).
- [14] E. Avichay. “Hello slim.js!” slim.js. (Apr. 14, 2023), [Online]. Available: <https://slimjs.com/#/welcome> (visited on 04/14/2023).
- [15] T. P. P. Authors. “Readme.md,” Polymer Project. (Apr. 14, 2023), [Online]. Available: <https://github.com/polymer/polymer> (visited on 04/14/2023).
- [16] G. LLC. “Lit,” Google. (Apr. 14, 2023), [Online]. Available: <https://lit.dev> (visited on 04/14/2023).
- [17] M. Platforms. “Jest delightful javascript testing,” Meta. (Apr. 14, 2023), [Online]. Available: <https://jestjs.io> (visited on 04/14/2023).
- [18] G. Inc. “Puppeteer | puppeteer,” Google. (Apr. 18, 2023), [Online]. Available: <https://pptr.dev> (visited on 04/18/2023).
- [19] O. Foundation. “Mocha - the fun, simple, flexible javascript testing framework,” OpenJS Foundation. (Apr. 14, 2023), [Online]. Available: <https://mochajs.org> (visited on 04/14/2023).

- [20] S. committers. "Mocha - the fun, simple, flexible javascript testing framework," Sinon community. (Apr. 16, 2023), [Online]. Available: <https://sinonjs.org> (visited on 04/16/2023).
- [21] C. A. Library. "Chai," Chai.js Assertion Library. (Apr. 16, 2023), [Online]. Available: <https://www.chaijs.com> (visited on 04/16/2023).
- [22] J. Garnier. "Readme.md," Julian Garnier. (Apr. 16, 2023), [Online]. Available: <https://github.com/juliangarnier/anime> (visited on 04/16/2023).
- [23] D. Eden. "Animate.css | a cross-browser library of css animations," Animate.css. (Apr. 16, 2023), [Online]. Available: <https://animate.style> (visited on 04/16/2023).
- [24] S. team. "Sass: Syntactically awesome style sheets," Sass. (Apr. 16, 2023), [Online]. Available: <https://sass-lang.com> (visited on 04/16/2023).
- [25] S. maintainers. "Stylus | an expressive, robust, feature-rich css language built for node.js," Automattic. (Apr. 16, 2023), [Online]. Available: <https://stylus-lang.com> (visited on 04/16/2023).
- [26] T. C. L. Team. "Getting started | less.js," Less. (Apr. 16, 2023), [Online]. Available: <https://lesscss.org> (visited on 04/16/2023).
- [27] O. Foundation. "Find and fix problems in your javascript code - eslint - pluggable javascript linter," OpenJS Foundation. (Apr. 17, 2023), [Online]. Available: <https://eslint.org> (visited on 04/17/2023).
- [28] jslint-org. "Jshint: The javascript code quality and coverage tool," jslint-org. (Apr. 17, 2023), [Online]. Available: <https://www.jshint.com> (visited on 04/17/2023).
- [29] J. Long and contributors. "Prettier - opinionated code formatter," Prettier. (Apr. 17, 2023), [Online]. Available: <https://prettier.io> (visited on 04/17/2023).
- [30] L. N. Einar Lielmanis and contributors. "Readme.md," js-beautifier. (Apr. 17, 2023), [Online]. Available: <https://github.com/beautify-web/js-beautify> (visited on 04/17/2023).
- [31] J. Krause, *Developing Web Components with TypeScript: Native Web Development Using Thin Libraries*, English. Apress, Mar. 12, 2021, p. 280, ISBN: 9781484268391.