Raine Kuosmanen

Building a data visualization solution

Metropolia University of Applied Sciences

Bachelor of Engineering

Media Engineering

Thesis

25 March 2018

Author Title Number of Pages Date	Raine Kuosmanen Building a data visualization solution 39 pages 25 March 2018
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Media Engineering
Instructors	Aarne Klemetti, Researching Lecturer

This thesis researches the possibilities of producing a software service that offers a wide range of data visualization to a plethora of customer needs. The requirement for this solution is the ability to produce fully configurable and responsive dashboard views that visualize for example data produced by digitalized forms. Furthermore, the solution requires to have an ability to separate and filter views for different user levels and needs.

This thesis introduces the basis for the responsive requirement and handles aspects regarding usability and user interface, mainly focusing on the characteristics of human visual perception and cognition. This thesis also introduces the taxonomy of data visualization methods and how they utilize the natural human perception.

As a result of this research, a REST API and a modular tool for dashboard creation were developed. Usability of the modular tool was kept as one of the main aspects of the development task. As a proof-of-concept, two data visualization dashboards for a simple digitalized form product were created by using these tools.

Additionally, a data integration to an external Business Intelligence software utilizing a similar REST API was researched.

Keywords	Data visualization, usability, user experience, software development, dashboard tool, data integration

Tekijä Otsikko Sivumäärä Aika	Raine Kuosmanen Datan visualisointityökalun rakentaminen 39 sivua 25.3.2018
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Mediatekniikka
Ohjaaja	Tutkijaopettaja Aarne Klemetti

Insinöörityössä tutkittiin mahdollisuutta kehittää ohjelmistoratkaisu, joka tarjoaa monipuolisen visualisointien kirjon laajaan asiakastarpeeseen. Ohjelmistoratkaisun vaatimuksena oli kyky tuottaa räätälöitäviä responsiivisia kojelautoja, jotka esittävät esimerkiksi sähköiseen muotoon tuotujen lomakkeiden tuottamaa dataa. Lisäksi ohjelmistoratkaisun oli tarjottava mahdollisuus rajata näkymä käyttäjätason ja -tarpeen mukaan.

Insinöörityössä perehdyttiin responsiivisen suunnittelun tarpeeseen sekä käytettävyyteen ja käyttöliittymäsuunnitteluun ihmisen visuaalisen hahmotuskyvyn näkökulmasta. Tämän ohella insinöörityössä perehdyttiin eri visualisointimenetelmiin ja niiden hyödynnettäviin ominaisuuksiin liittyen ihmisen hahmotuskykyyn. Tuloksena löydettiin muun muassa luonnollinen taipumus muodostaa yksilöistä ryhmiä niiden ominaisuuksien, kuten muodon, värin tai sijainnin, perusteella ja tilastotietoa erityyppisten internetiin yhdistettyjen laitteiden käyttömääristä.

Insinöörityössä tutkittujen asioiden pohjalta työssä kehitettiin REST API -ohjelmointirajapinta ja modulaarinen työkalu kojelautanäkymien luomiseen. Työkalun käytettävyys oli yksi kehitystyön päänäkökannoista. Kehitetyllä työkalulla luotiin kaksi eri kojelautanäkymää sen toiminnan esittämiseksi. Työkalua voidaan pitää onnistuneena, sillä kumpikin näkymä kyettiin luomaan määritellyn visualisointitarpeen mukaan.

Lisäksi työssä tutkittiin mahdollisuutta hyödyntää kehitettyä REST API -ohjelmointirajapintaa datan integroinnissa ulkoiseen järjestelmään.

Insinöörityön tulosta voidaan pitää onnistuneena, sillä kehitetyn työkalun ja REST API -ohjelmointirajapinnalla yhdistelmällä voidaan täyttää datan visualisoinnin asiakastarpeet tehokkaasti ja monipuolisesti. Työn tulos otetaan käyttöön asteittain osana digitalisoitujen palvelujen tuotteistusta. Lisäksi insinöörityö on avannut työn asiakasyrityksessä keskustelun kehitettyjen työkalujen laajemmasta hyödyntämisestä myös kojelautanäkymien ulkopuolella.

Avainsanat	datan visualisointi, käytettävyys, käyttäjäkokemus, ohjelmistokehitys, kojelautatyökalu, dataintegraatio

Metropolia

Contents

List of Abbreviations

API	Application Programming Interface. An interface used between different systems or software to exchange data or commands.
AJAX	Asynchronous JavaScript and XML. A group of techniques to include interaction in web applications. Used to fetch data with simple requests from a remote service. Enables updating only needed parts of the web application.
HTML	Hypertext Markup Language. A programming language to define elements to be rendered on a screen by a web browser.
HTTP	Hypertext Transfer Protocol. A transfer protocol used by web applications and browsers to request and receive data.
JSON	JavaScript Object Notation. An open standard file format. Can be used as a response to AJAX calls to provide updated data.
REST	Representational State Transfer. An architectural model based on HTTP protocol. Usually used in APIs to provide data according HTTP requests.

1 Introduction

1.1 Customer introduction

The customer of this research and development work is a Finnish multinational company Reslink Solutions Ltd. Reslink Solutions Ltd offers Software as a Service (SaaS) as their product. The service consists of two key components: An administrative panel portal and mobile applications. Both components offer a selection of configurations to deliver a tailor-made solution for customers in a need of a digitalization, mobility and process optimizing.

1.2 Current situation

Reslink Solutions Ltd has a third-party software that is currently used for data visualization and to offer dashboard reports regarding for example store audits based on score, location, and time. The current third-party software enables producing visualizations relatively fast and allows customizing of the dashboards based on customers' requirements. Reslink Solutions also has data visualizations in a legacy version of the administrative panel executed with Fusion Charts framework.

1.3 Research problem

The problem with the current third-party provider is that the created dashboards can be shared with customers only by embedding an iframe to the Reslink Solutions' web service environment. Iframes function as pictures, which means that the content is not responsive. This limits the device range that can show the dashboard correctly. Additionally, the current provider does not enable setting different kinds of views for different user levels without creating separate dashboards for each group. The current provider being physically located in the United States of America makes it non-compliant of the European Union's General Data Protection Regulations, which makes the solution completely inadequate in the future.

The purpose of this thesis was to search for solutions for transactional data visualization to be able to fulfil a plethora of customer needs. The research problem can be presented in two questions:

- How to build and deploy responsive data visualizations or dashboards quickly enough for different needs?
- How to enable a data integration to an external system to order more complex KPI calculation.

2 Responsive design and user experience

2.1 Responsive design

Responsive design is usually associated with a design process of a website or application that supports every display size. The need for responsive design has increased significantly during the last decade because of the grown number of handheld devices such as tablets and smartphones.

The need for responsive design has been emphasized during recent years because small touch-screen devices such as smartphones and tablets have been steadily catching up desktop devices in terms of web use. Smart devices surpassed desktop devices measured in website visits globally for the first time in (see image 1) 2016 [1].

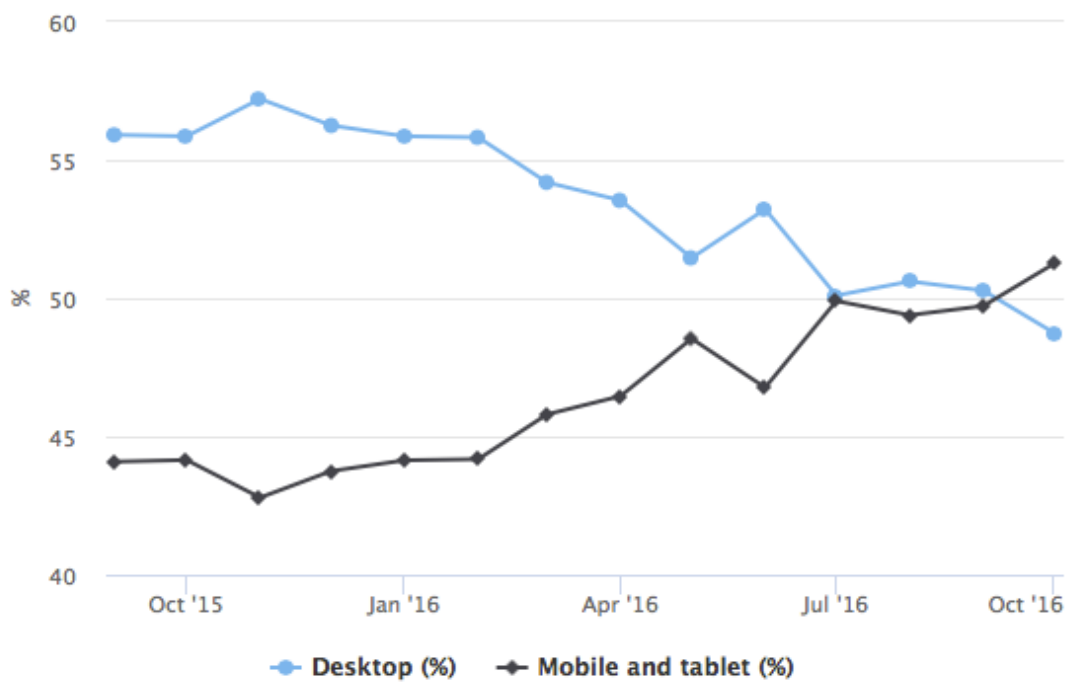

Image 1. Amount of website visits globally with desktop and mobile devices [1].

The total amount of devices connected to the internet has increased over tenfold during the last decade. However, during that same time span the amount of traditional desktop computers has remained practically the same, as the share of mobile devices such as tablets and phones has increased. The most dramatic change has been the emergence of an entirely new genre of devices such as smart televisions, household appliances and wearables, which has developed a need to reconsider whole term of user interface (see image 2). [2.]

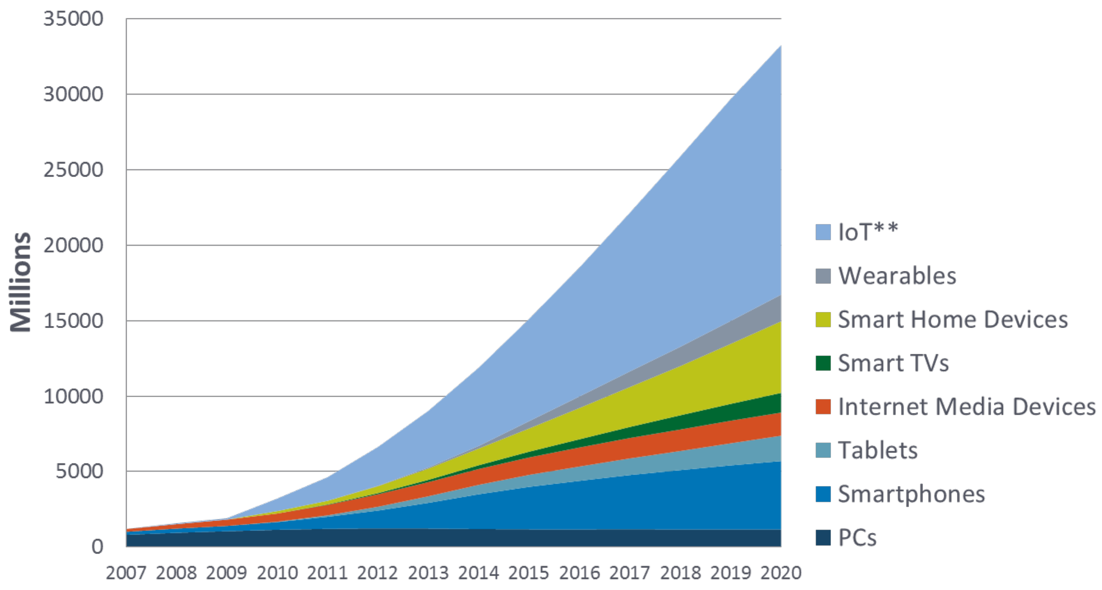

Image 2. Amount and projection of devices connected to the internet, segmented in different groups [2].

2.2 User experience

According to a dictionary definition, User Experience is ”The overall experience of a person using a product such as a website or computer application, especially in terms of how easy or pleasing it is to use” [3]. More accurately user experience incorporates features related to both pragmatic and hedonistic qualities, forming a complete experience which determines whether the user preferred the service or not. Key components are effectiveness, efficiency, and learnability, and more recently added standards joy, attractiveness, and aesthetics. [4.]

When designing a good user experience, it is advisable to acknowledge the limitations of human cognitive capacity. According to cognitive load theory a human mind can process only a few elements of information at the same time. This is because the resources are limited and directed selectively to achieve a specific goal. The theory supporting this is that memory-related cognitive processes might interfere information processing. Information processing may be hindered if the cognitive load gets too heavy for the individual user. [5.]

Generally, websites require high cognitive effort. According to cognitive load theory, to improve a websites usability it is advisable to limit stimuli directed to the user at a given time. This can be achieved by managing the amount of information concurrently available. Additionally, the presentation of the information can reduce or add to the cognitive load of the website. [5.] As presented in image 3, an average user rarely reads the page like a book: line by line from top left corner towards right bottom corner. In most cases users browse through a website quickly looking for a point of interest. [6.]

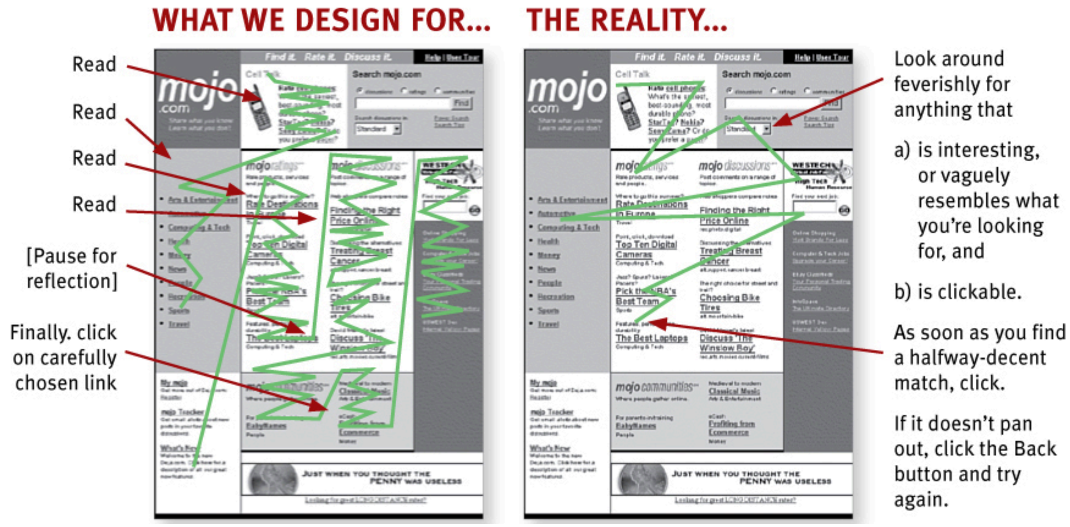

Image 3. An example of how an average user reads a website [6].

In addition to the layout of the website, also the used wording is crucial in terms of usability. More creative wordings can be used as part of a brand or a specific marketing campaign. However, unless the term is a well-known brand or product, for most users the best usability is when the used terminology leaves no room for interpretation (see image 4). [6.]

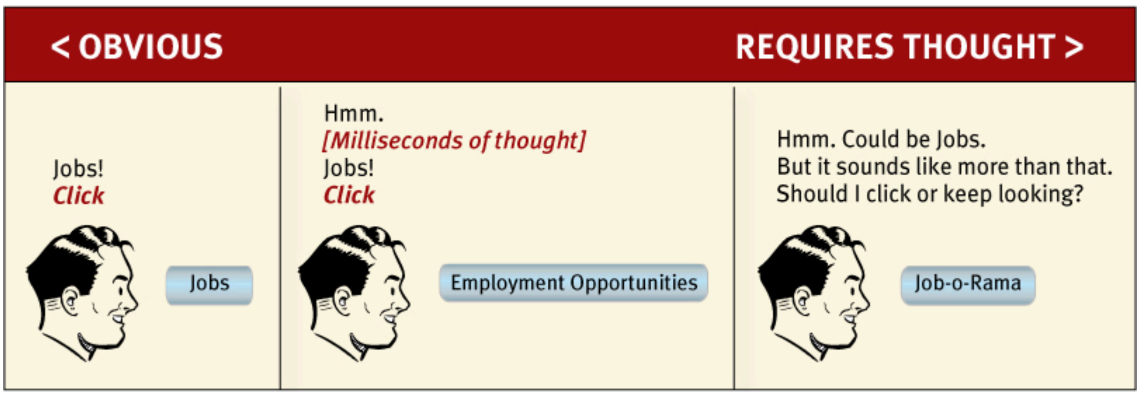

Image 4. An example of usage of terminology [6].

User preference and skill level are relevant when designing user experience. For example, with public web sites, every user group needs to be considered to ensure a pleasant user experience for everyone. Dyslexic and elderly users usually benefit from larger font size and well-chosen text and background colours. In mainstream web services user experience is imperative because it can solidify customer relationship, or the user can end up to a competing service. [4.]

Usability of non-public software is usually measured differently compared to a public website. For example, when testing a web-based professional software or a specialists' tool, learnability and optimal use are risen as main attributes. An intuitive user interface design is an instrumental part of learnability, but its role is diminished after the user gains experience and begins to memorize the working sequence. User experience or usability consists of the user interface itself, for example throughout efficiency, operability, and accessibility. Training also has a part in usability, and it can compensate possible shortcomings of the user interface caused by for example a wide variety of accessible functionalities. [7.]

Beneath the user interface functionality also affects the experience. Efficiency, mentioned also as a part of user interface, contributes also to the functionality, especially in professional software. An example of this is a page with a wide variety of selections. The sheer amount of content, chosen wordings or field placements can make user interface less intuitive. However, placing large amount of selections or content to the same view can improve the efficiency of the software with the assumption that the user has sufficient skills, training, and experience with the said functionality. [7., 8.]

3 Data visualization

3.1 Data visualization principles

The main goal of data visualization is to offer information in an easily processable form. Often raw data is displayed in a two-dimensional table, or as key-value-pairs. This raw data format offers the most specific information for each key and its value. However, it is challenging and time consuming for a human mind to interpret relative differences or trends between values. An example of this is an Anscombe's quartet, introduced by Francis Anscombe in 1970s, which includes four pairs of columns with different values (see image 5). With enough time a human mind can decipher how these pairs differ from each other as sets. When these pairs are visualized in simple xy-grids the difference between these datasets becomes instantly evident (see image 6). [9. p. 29-33]

x1	y1	x2	y2	x3	y3	x4	y4
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

Image 5. Anscombe's quartet [9 p. 29-33].

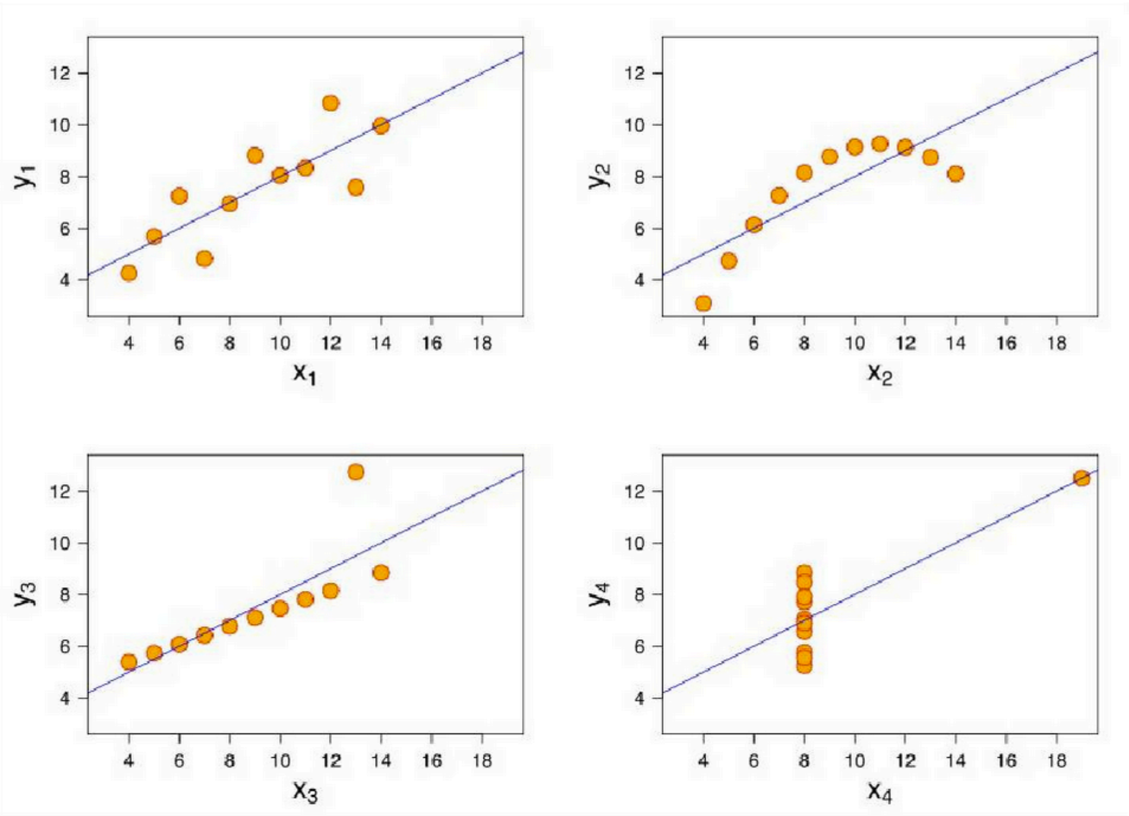

Image 6. Anscombe's quartet, visualized values of image 5 [9 p. 29-33].

Data visualization aims to utilize the natural tendencies of human perception to ensure the most efficient way to present data. Similarities between individual markers or units, and differences between larger groups of those units, are likely to be perceived naturally. This division utilizes different traits such as colour, size, shape or positioning and it happens on a subliminal level of human consciousness. [9. p. 29-33]

3.2 Principles of visualization

Minimizing cognitive effort and maximizing the data on display is the goal of data visualization. To achieve this, it is advisable to utilize known traits of human cognitive tendencies known as Gestalt principles of perception, including laws of proximity, similarity, enclosure, closure, continuity, and connection. In addition to these principles the rule of consistency, and the sheer number of visual stimuli as details affects the cognitive load. [10.]

Law of proximity is based on physical locations of the presented data. Human mind naturally understands data markers positioned close to each other as members of the same group [9 p. 29-33, 10]. An example of this is presented in image 7 that has three groups of values. Each group is similar in terms of count, size and colouring of the markers, positioning being as only distinguishable difference.

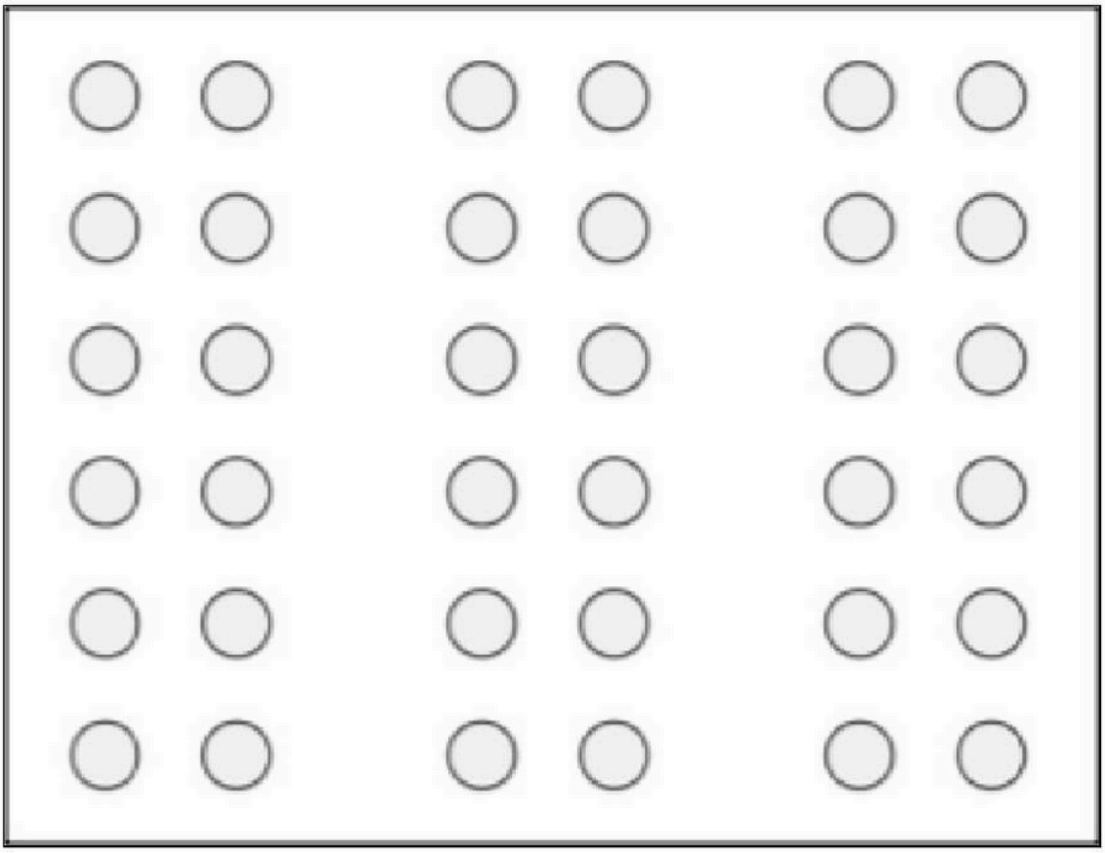

Image 7. Law of proximity presented with three similar groups of similar markers [9 p. 29-33].

Law of similarity is an instinct to find similarities and differences, and to form groups according to those traits. Attributes to indicate similarities or differences can be such as colour, shape, and size. An example of this is presented in image 8 that displays football penalty kicks and their results. In this example, the rule of similarity is used in the colouring of the football markers. [9. p. 29-33, 10.]

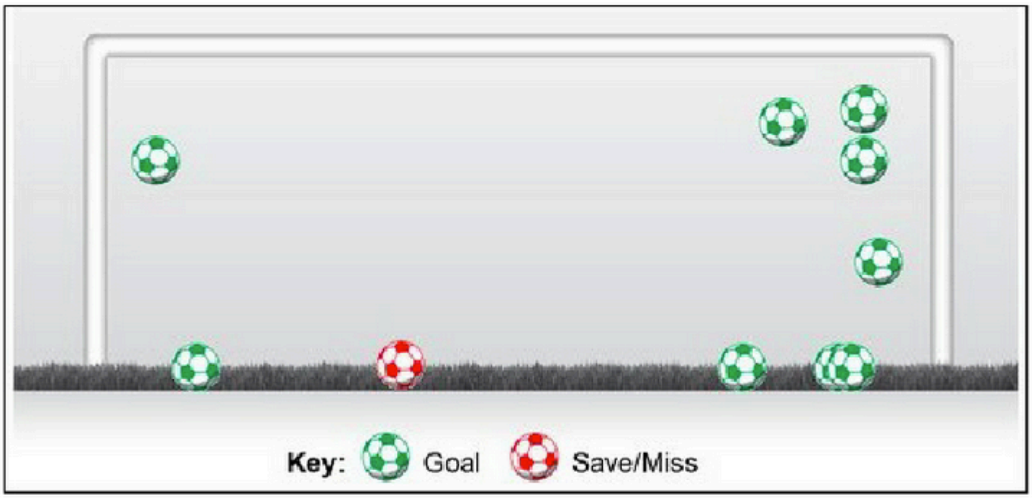

Image 8. Visualization of penalty kick results with a simple background [9, p. 29-33].

Law of enclosure leans on the natural tendency to form groups between objects according to a boundary or a background colour. Objects within the same boundary are naturally viewed as a group. For example, in image 9 there are similar objects scattered around two backgrounds. In both backgrounds there is an enclosure formed with a boundary around objects, or as a background colour beneath some objects. These enclosed objects are usually interpreted as a group. [10.]

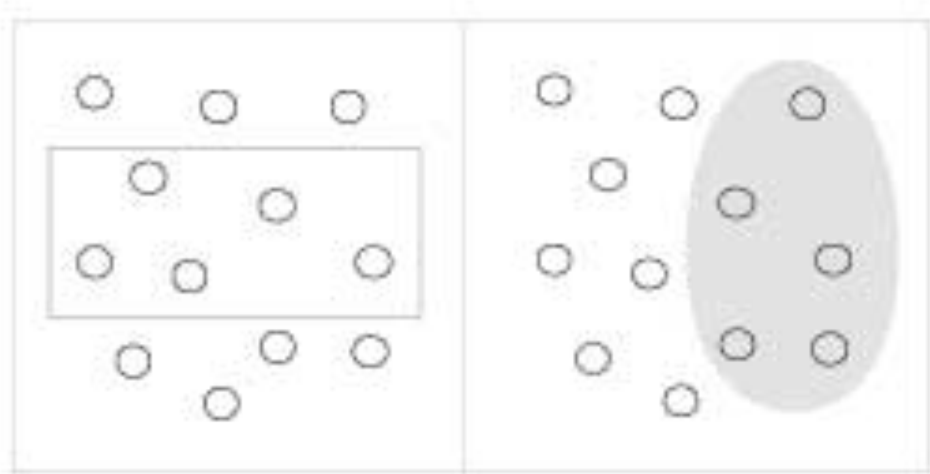

Image 9. Law of enclosure. The objects inside the boundaries in both images are usually considered as a group [10].

Law of closure is an instinct to perceive open structures as closed or completed, if it is a reasonable assumption [10]. Image 10 presents an example of this by showing multiple incomplete shapes. These shapes are usually interpreted as complete due to a fact that they are close enough of being complete and it is more reasonable to perceive them as closed than open.

Image 10. Law of closure. Usually both presented shapes are considered complete [10].

Law of continuity means that objects that appear to be a continuation of one another or that are aligned together, are perceived as a group. For example, in image 11 there is a wave presented partially crossing a boundary from behind. The wave form is not visible in parts where it crosses the boundary, but it is perceived as a continuous line. [10.]

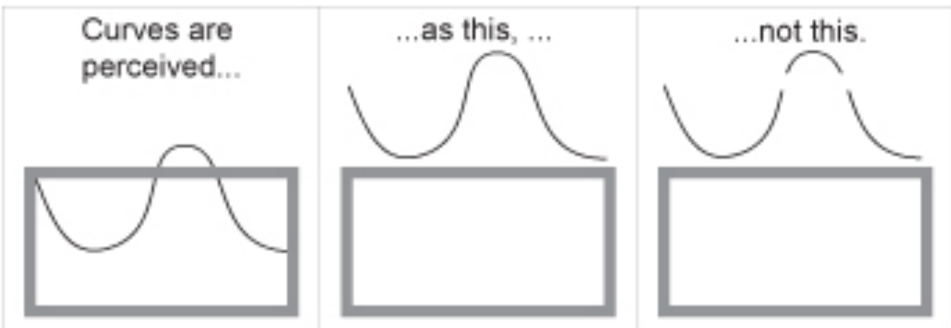

Image 11. Law of continuity. The image demonstrates how a partially visible wave form is perceived as continuous when it is reasonable to make such an assumption [10].

Law of connection is based on a natural tendency to perceive multiple separate objects that are connected by for example a line, as a group. For example, in image 12 there are four objects presented as two pairs. Each object connected to its pair with a line. [10.]

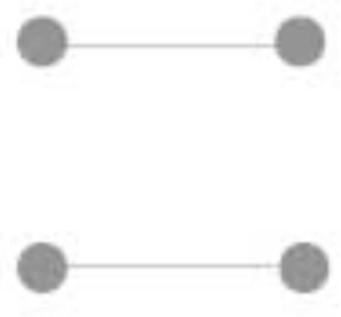

Image 12. Law of connection. Four similar objects are presented as two pairs as each object is connected to its pair with a line [10].

Rule of consistency is related to a group of visualizations meaning that visualizations showing comparable data should have consistency with for example axis values and colouring. This helps to distinguish differences or trends, that are meaningful to context. Inconsistent visualizations with for example axis values requires the viewer to read, understand and compare the values with each other to comprehend possible differences and trends. This increases cognitive load significantly and therefore makes visualization more ineffective. Rule of consistency also includes possible transformations with the visualization. This means that for example changing the timeframe should affect every visualization that shares this same value. [11.]

Every visualization practice aims to utilize the natural tendencies of human visual perception to minimize the cognitive load to ensure efficiency in understanding the visualized data. In addition to laws of proximity and similarity, and rule of consistency, it is imperative to find other ways to limit the cognitive load created by visual stimuli directed to the user. With data visualization it is advisable to present details only according demand, and to clearly separate the data layer of the visualization from the background. Image 13 presents similar data and visualization compared to image 8, and it utilizes law of similarity as colouring of the football markers. Compared to image 8 it has a detailed and colourful background layer. The result is that processing the information demands more cognitive effort compared to image 13, which makes the actual understanding of the visualization considerably slower. [9. p. 29-33]

Image 13. Visualization of penalty kick results, with a detailed background [9].

3.3 Data handling and querying

The first step in data handling is to define how it is collected and what data exactly it is. In many webservices, sites or applications, the visualization process starts with defining the collection method and choosing actual tools to collect for example data of usage. However, in some cases the data has already been formed, or it is formed continuously and the person visualizing it does not define its form, storage, or scope. For example, visualizing big data often follows this because the data is formed from multiple sources and during a longer period. [12.]

Before visualizing the data, it is imperative to have a clear understanding of the data itself, and what is the point of interest or main information in it. This defines how widely the data is queried, and can be perceived as a spectrum, presented in image 14 below. One absolute of the data querying spectrum, an unstructured overview of the data can give the wanted insight, but rarely gives specific answers. The other end of the querying spectrum is an extremely defined inquiry that may offer an accurate answer to a specific question but does not always offer context or possibility to perform any comparations. In most cases the best serving practice to find the right querying accuracy is found somewhere in the central portions of the spectrum. [12., 13.]

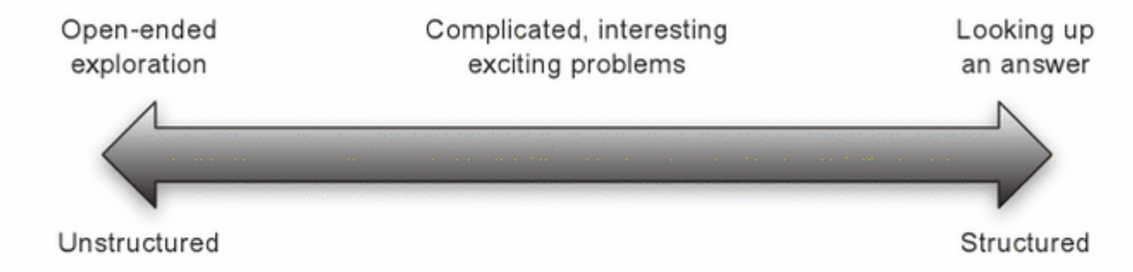

Image 14. Data querying spectrum [12].

Understanding the data and what it consists of enables the use of aggregation methods that are used to facilitate the data to a form that enables comparing values between different groups, such as gender or age. Typical aggregation methods are for example calculating averages, counting amounts of grouped values, and calculating proportions of different data groups. A common characteristic with every aggregation method is that each method relies on the data being divided in groups that can be defined by value, such as age, salary or points scored depending on the data set. This grouping is also called binned aggregation or binning (see image 15).

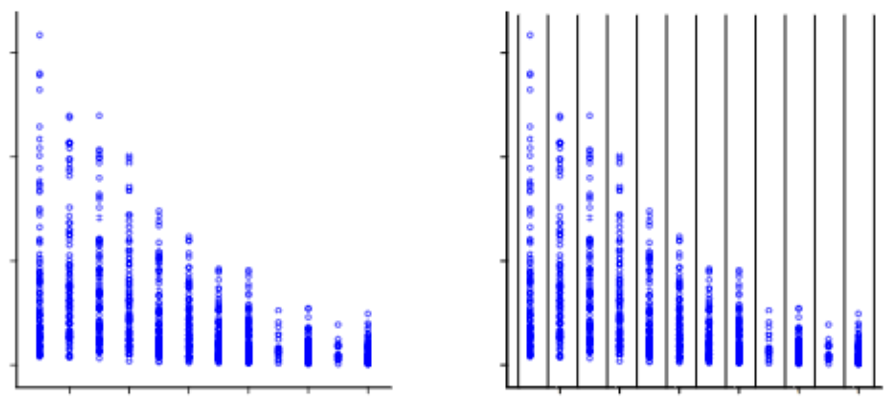

Image 15. An example of raw data (left) and data aggregated by binning method (right) [14].

Binning can be performed on various scales and criteria. It is used to even up statistical anomalies, likely caused by a measurement error, or for example to emphasize a certain quality or value of the analysed dataset (see image 16). Binning can also diminish relevant patterns or statistical peaks that would be considered valuable. These qualities make choosing the correct binning method an essential and sometimes challenging task when analysing data. [15.]

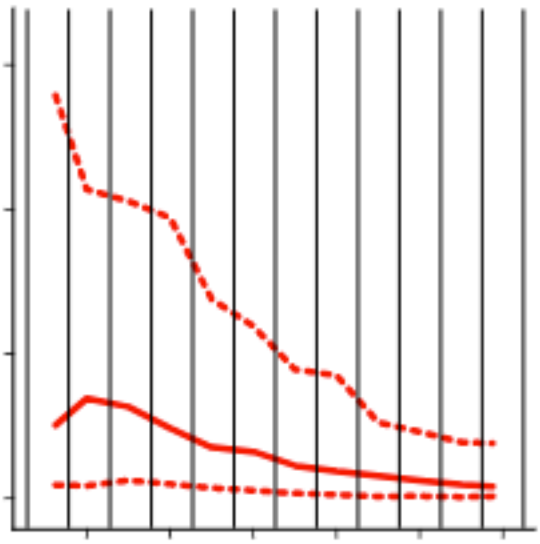

Image 16. Calculated medians and percentiles of each binned group of the image 15 [14].

The data in its type, amount and structure affects also to coordinate projection methods. In many cases data that does not follow the Gaussian plotting should be handled with Mutual Information or Joint Mutual Information techniques. With these techniques the data is handled as two-dimensional, for example value and time being the dimensions. A well-chosen technique, for example Mutual Information, can give meaningful projections when combined with right plotting. [16.]

3.4 Visualization methods

Data visualization methods can be divided into organizations. These organizations aim to present different kinds of proportions, relationships or amounts of units from the data context. Each organization has a designated purpose based on natural tendencies of human perception to observe relative differences in presented data. These different kinds of organizations and their purposes are presented below. [9.]

Showing changes over time is often a trait that is needed from the visualization. An example of this need is any kind of dashboard that presents the development of different KPI values. Any visualization able to present time as an x-axis value is suitable for this need, such as line chart or column chart (see image 17). [9.]

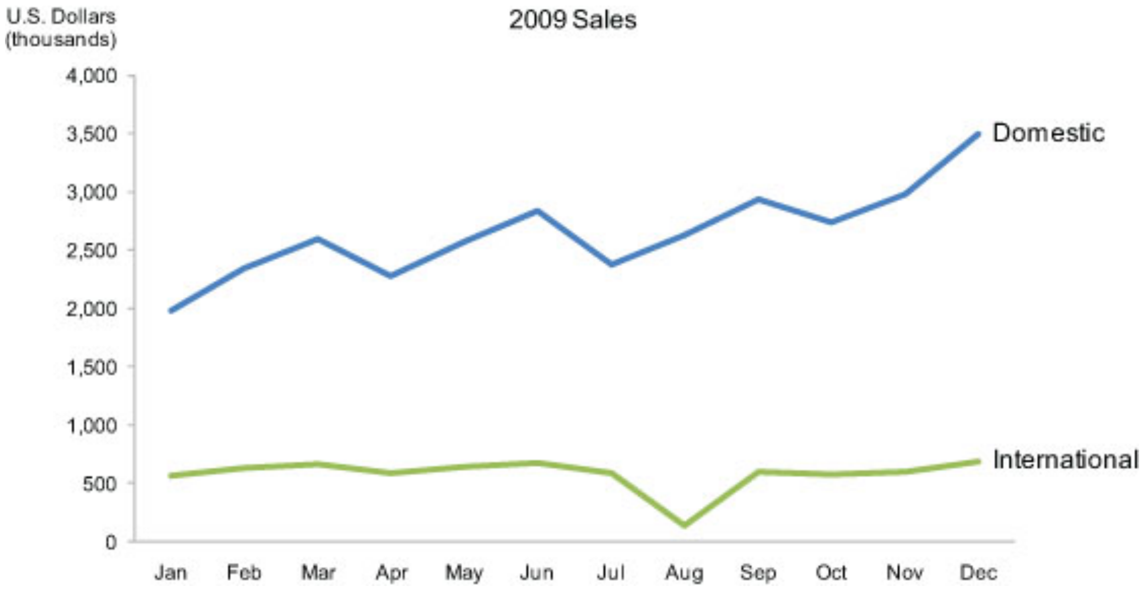

Image 17. Line chart that displays sales data divided in domestic and international sales [10].

Comparing categories is a group of methods that enables observation and comparation of relative and absolute sizes of multiple different value groups. In this method the visualization values are grouped and placed in a way that their proportions compared to each other are easily detectable. An example of this is a bar chart (see image 18). [9. p. 145-147]

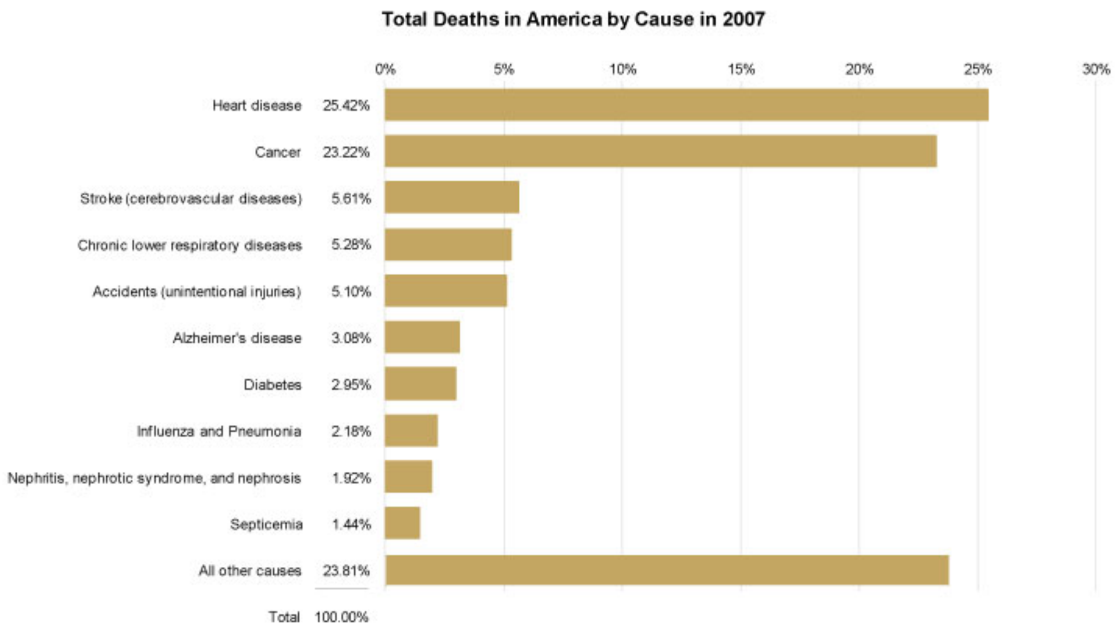

Image 18. Bar chart that displays percental proportions of different causes of death [10].

Hierarchical relationships offer perceivable information of groups and their relative size or volume differences as parts of a hierarchically higher element. This group of methods is usually presented in a circular form, which can make it difficult to perceive absolute differences if proportions are close to each other in size, or if there are too many groups presented. A common example of this is a pie chart. A more complex example is a sunburst partition (see image 19), which presents multiple hierarchical levels and their proportions. [9. p. 145-147]

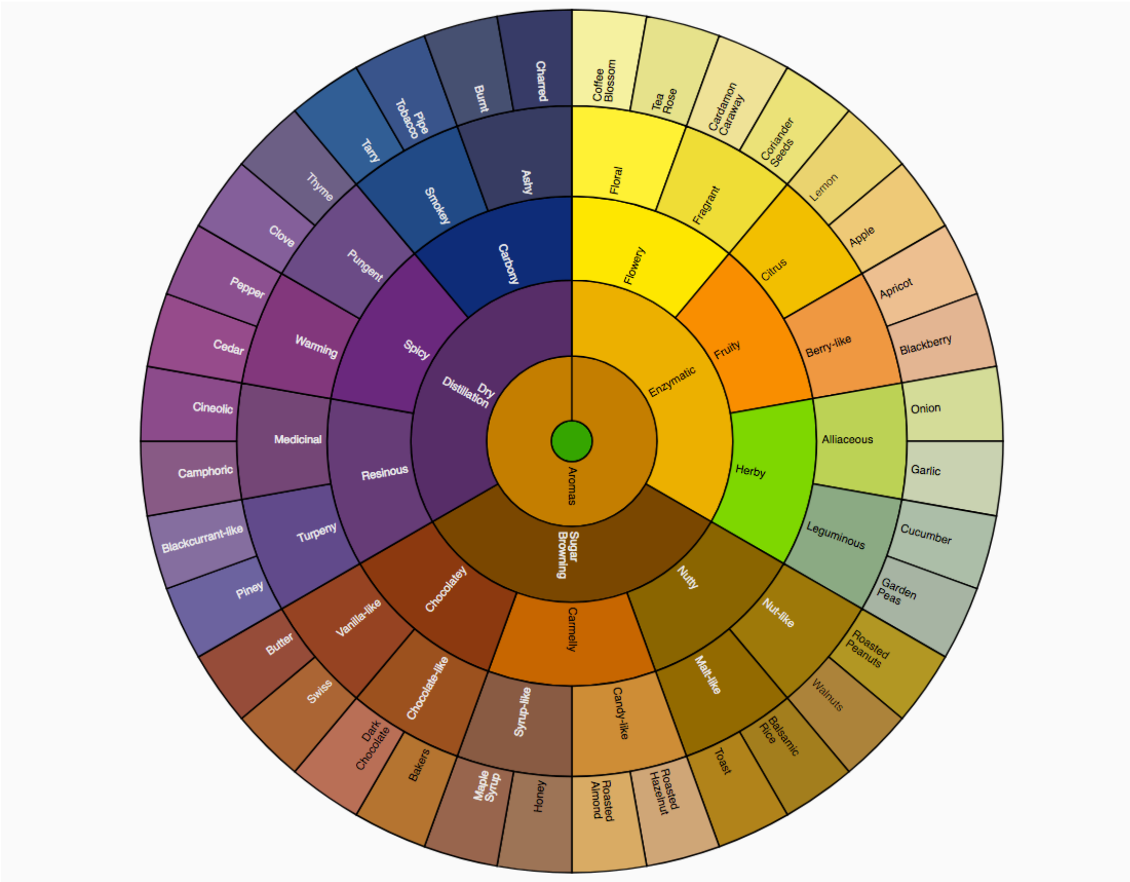

Image 19. Sunburst partition of hierarchical relationships [17].

Plotting connections and relationships is a method that presents patterns, distributions, and relationships between multivariate datasets. This group of methods presents some of the most complex visualization solutions, because it can display multiple outcomes of the same dataset. It also usually focuses on creating exploratory analysis. An example of this is a scatter plot (see image 20). [9. p. 145-147]

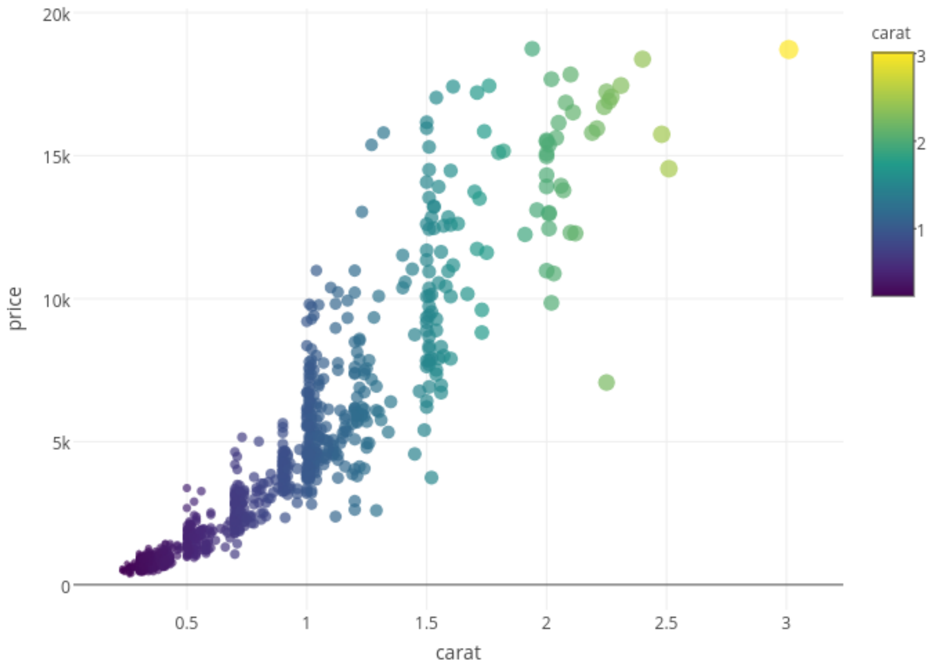

Image 20. A scatter plot visualization on how gold purity (carat) affects the price [18].

Mapping geospatial data is a method to display datasets that contain geospatial properties. This is utilized to show for example how the data proportions are divided in different geological locations. An example of this is the choropleth map (see image 21). [9. p. 145-147]

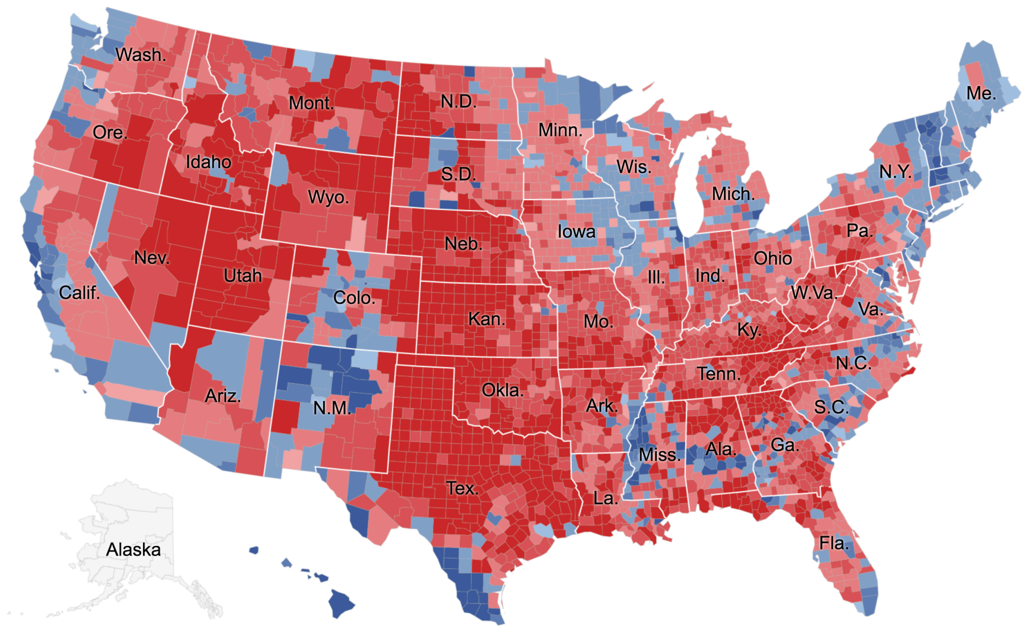

Image 21. A choropleth map showing presidential election results by county. Red indicating republican and blue indicting democrat majority [19].

The selection of an appropriate visualization method is influenced by the definition work of the visualization needs [9 p. 145-147]. Visualization methods are to be tools to achieve a desired result specified by the data itself and the customer need.

4 Dashboard design

4.1 Dashboards and their benefits

The benefits of data visualization and more precisely dashboard reports are a well-studied subject. The theory is based on the limitations and characteristics of human cognitive processes. Human cognition tends to naturally divide objects to groups according to their characteristics, such as colour, shape, positioning or surroundings. Data visualization is based on presenting a large amount of data in an easily understandable form by utilizing the traits mentioned earlier. [9. p. 145-147]

The first purpose of a dashboard is to provide a visualization of the status of a company or its specific unit. This status can indicate for example how well goals are reached, how a product is performing in comparison with a competitor's one, or how well system is working. These same views are also often used to find deviations or anomalies that may even form patterns in a longer term. Data visualization is used among other things to predict product demand, for example because of public holidays or possible flu seasons that may cause an increase in sick leaves. [20.]

Because a dashboard can present more data in a comprehensive way than for example a traditional table report, it enables people using it to be better informed. A limitation with a traditional table reports such as an Excel sheet is that the data is not well structured, thus making it difficult to make comparations with more than a few values. This problem is magnified if the user is trying to find any patterns or deviations in the data set because usually the values are relatively homogenous. Data visualization or a dashboard aggregates the data set in a way that values of same category are easier to compare to each other. This enables the whole decision-making process to be based on more data in quantity and quality. [9. p. 145-147, 21.]

Dashboard reports are only not usable when comparing high level sales figures. They are also a relevant tool for asset or employee management. Visualizing employee data can among other things be used to detect the need for training. For example, if a single person is continuously exceeding the average time with a specific type of task, it might indicate a need for further training. [21.] With assets, for example coffee automats, a dashboard can show for example if an individual coffee automat unit has a need for maintenance considerably more often than similar models. This information could indicate that either there is a major problem with the machine itself and the maintenances are only a symptom, or that the machine's production capacity is for example considerably undersized.

4.2 Context and content

The first and most important step of designing a dashboard is to understand the customer needs for the visualization [20]. The business case determines what is important to follow or highlight from the data set. For example, a dashboard for following fulfilment and results of an audit or guarding contract consist of considerably different element than a dashboard that measures sales. In the first mentioned example management is more interested in exceptions or deviations in routine. After the deviations are pointed out, it is interesting to find out if there is a pattern to them. In the latter example trends and patterns are usually considered more interesting because the context is usually higher-level decision making. Before starting the analyzation and visualization work it is imperative to determine the problem in need of solving, or the type of information that is wanted to be gathered [12].

Context of the dashboard not only includes the business case but also the person viewing the visualization. The most important feature of this person is their role in the organization, which determines how and what information should be displayed. A chief executive officer is likely to be more interested in higher level figures than an area manager. [20.]

According to studies there are three main use cases for a dashboard [22]:

- To monitor the viewer's own performance. For example, a view for an employee to observe the progress and results of work.
- To monitor the performance of others. For example, a view for an employee to compare own performance to an average result, or a view for a lower level manager to see how employees are performing.
- Administrative observation. An example of this is a dashboard of high level sales, or for system performance.

Usually a dashboard consists of multiple different visualizations that show different data based on visualization needed. Understanding the nature of the data and recognizing the valuable pieces of it is also imperative when choosing the visualization methods. The presented data affects strongly on chosen methods because each method classification is designed to show certain traits of the data sets, for example a line chart being ideal to show trends, patterns and changes happening over time. Another example are circular visualizations such as pie or donut charts that are comprehensible when comparing hierarchical relationships. However, this visualization type does not provide accurate information about quantities or their comparation due to human perception which has not evolved to support accurate decoding of areas, angles, or distances along a curve. [10.] The methodologies are based on these natural characteristics and it is imperative for the visualization choices and the dashboard to follow these classifications. [9. p. 145-147]

The readability and comprehensibility are significantly affected by the overall dashboard design including visualization placement, colouring choices, and chart ranges. It is advisable to maintain consistency to avoid misinterpretation. For example, having multiple charts visualizing similar comparable data but with different scales may cause the viewer getting a wrong impression. The objective is to display as much data as possible to maximize the effectiveness of the dashboard. On the other hand, the readability needs to be preserved. The viewer's cognitive load is increased with the amount of data or visual stimuli, which in turn increases the effort needed to comprehend the visualization. [5., 11.]

To transform the presented dataset into a more comprehensive format, it is advisable to use slice-and-dice method that means dividing the dataset into smaller and more easily understandable groups. Usually groups are divided according multiple parameters to achieve more specific results. [24.] This is the basis of the name slice-and-dice which refers to cutting the dataset into smaller pieces in multiple dimensions. For example, grouping a maintenance dataset according task type and handling organization renders it as more comparable. Adding drill down functionality to the visualizations is a method to increase the amount of data available on a same dashboard, while keeping the cognitive load smaller at any given time. The method is based on keeping the accuracy and level of the presented data in line with the user’s need, while allowing user to penetrate a deeper and more precise layer of data by clicking a chosen point of interest. Hence the name drill-down functionality. Drilling down on deviations might provide useful information and help to understand their possible causes. For example, if a specific maintenance task with a specific employee is consistently longer than average, it is likely solvable by offering more training. [20.] An opposite functionality for drill-down is called roll-up. It increases aggregation level offering a higher, less precise but more complete view, of the data set. [25.] Roll-up functionality is used among other things when comparing higher data groups. For example, instead of comparing individual persons between each other it might be more feasible to compare the different suborganizations they represent to determine if there are performance differences caused by a higher level, such as a process.

4.3 Visualization frameworks

The market offers an abundance of JavaScript-based visualization frameworks that are supported in most modern browsers. There are both commercial and open-source solutions for visualization. This section introduces D3.js framework that has become a defacto visualization framework in the market, and thus is used as a benchmark when evaluating compatible solution.

D3.js is an open source framework that approaches data visualization by binding the data to the elements. This way it is different when compared to most other frameworks that are based on libraries containing a certain amount of visualization methods linked to suitable data. This difference renders the learning curve of D3.js significantly steeper because each visualization enables but also requires a significant number of configurations. However, this cultural difference also makes D3.js a more flexible solution since it is not bound to certain supported methods and functionalities. [26.]

When choosing the visualization framework, it is imperative to understand the needed scope of the visualizations. There is ready desktop software, such as Tableau, on the market containing sophisticated data analyzation tools and support for multiple data sources. For example, when visualizing big data that has multiple data sources it was discovered that using Tableau provides sufficient results in a short time whereas D3.js was not able to entirely visualize the whole data set within given time. However, D3.js provided a highly interactive visualization of a portion of the data set, which might be the desired outcome in some use cases. [27.]

5 Visualization solution development

5.1 Building the user interface

The visualization solution was built to the Reslink Solutions' platform that already contained tools and frameworks that were used to build the user interface and functionality. To ensure the best user experience, using these tools was considered the best option [11]. The overall visual appearance was following a modified Vendroid theme throughout the platform, and it was decided to follow this theme in the visualization containers as well.

The responsive design has been implemented with Bootstrap grid framework. This enables designing visualization widgets that are rearranged and resized according to the size of the screen or window. This framework ensures the fulfilment of the responsiveness requirement. Moreover, this allows more customizability in the visualization configuration phase, since Bootstrap utilizes predefined classes for the grid size definition.

Different options were researched and compared before implementing the actual framework for data aggregation and visualization. The options consisted of two frameworks already partially used in the platform, Telerik Kendo UI and Morris.js. Two other frameworks were also taken into the consideration, D3.js and Chart.js. The criteria for the framework is as follows:

- support responsiveness
- drill-down functionality
- ability to bind data from a remote source
- include sufficient visualization and aggregation methods

D3.js is a well-known open source framework, and it was considered the most versatile framework of this comparation in terms of chart types. It has become the de-facto JavaScript visualization framework due to its vast selection of visualization types and functionalities. It was being used as a baseline for the visualization framework evaluation.

Morris.js is an open source framework that is used in Vendroid Admin Panel theme chart examples and promoted through the theme library already installed on the platform. The scarce selection of visualizations discovered in early stages of the research showed that this specific framework will not fulfil the needs for a generic visualization solution.

Chart.js is an open source framework like D3.js. This framework offers drill-down functionality by updating the data source and binding remote data with JSON. The concern with this framework was the visualization library that would probably serve most of needs but was still considerably narrower than those of D3.js or Kendo UI.

Telerik Kendo UI is a commercial framework already in use as a table and a scheduler component in the Reslink Solutions' platform. This framework enables roll-up, drill-down and slice-and-dice techniques with a seriesClick functionality by updating the visualization data. The framework supports remote data and offers a considerable amount of visualization and aggregation types.

All of the researched frameworks enabled responsiveness, which made it an insignificant term for search. The selected visualization framework was Telerik Kendo UI. It was not considered as a better option compared to an open source D3.js in terms of functionality or available visualizations. However, risk related to compatibility using a commercial product was considered smaller. Kendo UI was also considered a more viable option in terms of user interface consistency because parts of the framework were already in use in the platform.

The HTML of the platform is compiled by Smarty framework that utilizes HTML templates and enables assigning values from server-side. This framework was used with the visualization solution to keep the platform's code base uniform, and to enable using the same HTML template file to any kind of dashboard.

5.2 Modular structure, configurability, and usability aspect

The Reslink Solutions' platform that the visualization solution was built on, is based on a modular structure that determines which database tables and which functions are used, and how the platform's user interface is rendered. The modules are sectioned to higher level categories, such as" defines" including for example user credential of customer site definition, and "reports" including transaction data in a table format. To allow fast deployability and configurability, and thus ensure global nature of the solution, the visualization solution was built to utilize this structure.

The dashboard creation and configuration process were split into two new modules: Widgets and Dashboards. These two modules are used for definition and maintaining purposes. Dashboard module is used for high level default values for each widget. Widget module is used for controlling widget shape, placement, and size, and to enable overriding the default values. There needs to be a third module linked to the specific dashboard record to have access to view the created dashboard. In the platform all modules are controlled from a master module. To improve the user experience of the dashboard creation process, a necessary step of creating the module for the dashboard view was automated. To decrease the repetitiveness of the dashboard creation, sequence also creates a view module for the dashboard (see image 22). [4., 7.]

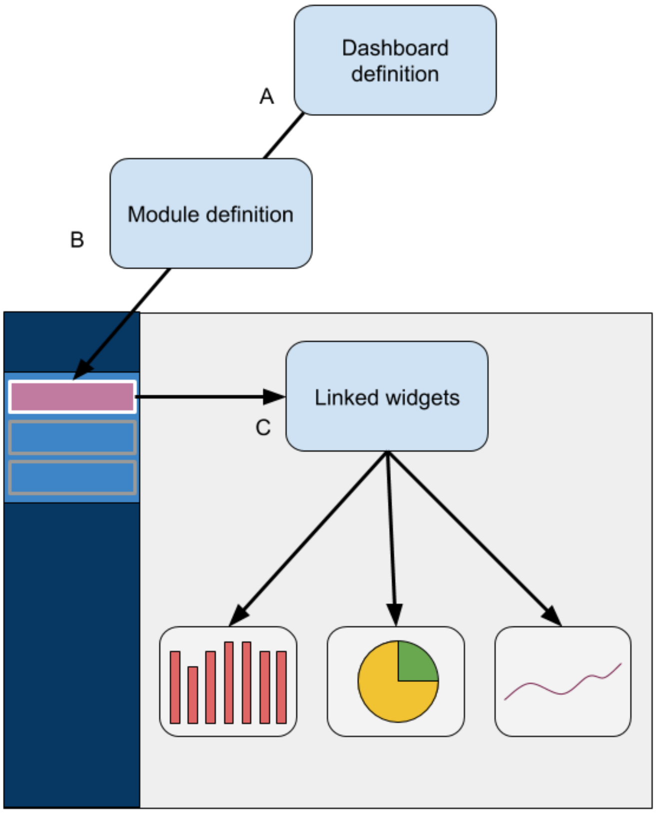

Image 22. Module usage and their logic in configuration: Dashboard configuration (A) is used to define Module configuration (B), which defines who can view the content and where in the portal it is placed. The dashboard linked to the module configuration is used to get Widget configuration (C) from the database to render the wanted widgets.

The dashboard module allows a high-level configuration for all widgets that are connected to it. The dashboard definition module was implemented to improve usability in terms of efficiency [4, 7]. An assumption was made that most of the visualization elements on the same dashboard are going to present data from the same data set. Enabling the user to define the default values for all widgets makes the dashboard configuration process faster and less repetitive. A decision to support a customized JavaScript file was made because it was deemed to be a faster task for a software developer than to have all the configurations available in for example widgets module. The available configurations are as follows:

- **Dashboard id** is the attribute that is used to tie the dashboards data and linked widgets to the part of the Reslink Solutions' platform. This definition allows a string entry allowing use of more comprehensive ids.
- **Dashboard name** is used as the displayable name of the shown dashboard.

- **Dashboard js file** gives an option to set a customized JavaScript file to fulfil for example drill-down needs. If left empty the dashboard will use the default JavaScript file.
- **Dashboard user level** determines the user level of the dashboard.
- **Dashboard form** determines a specific digitalized form to be used as a dashboard dataset.
- **Dashboard key field** sets the default column to be used as a key value of the dashboard widgets.
- **Dashboard value field** sets the default column to be used as a value of the dashboard widgets.
- **Dashboard module** determines which transaction database table the widgets linked to the dashboard use by default. This can be redefined in the widget definition.
- **Dashboard SQL** enables an accurate definition of database query result filtering. The query is set to contain location and user information joined to the selected transaction table data. With this field the default query filtering of linked widgets is set.

In turn the widget definition module offers a wider variety of configurations to be made in order to keep the built solution as generic as possible. The balance between flexibility, configurability and control, and efficiency and learnability were the basis of the widget definition process. It was deemed necessary to provide an ability to configure as many features in the widgets as possible. Therefore, similar kind of properties, such as data binning, aggregation and grouping are all separately configurable. To ensure better usability, some configurations have default values that were deemed to be most frequent choices, such as pie charts showing proportionally binned data, or a column chart showing unbinned data [7.] The available configurations are as follows:

- **Widget id and widget order** are used to determine the order in which the widgets are rendered on the dashboard. Widget order dictates the vertical and id the horizontal alignment of the widgets.
- **Widget name** is used in the header panel of the widget placeholder of the dashboard. This can be used as a topic for the widget.
- **Widget user level** allows to limit the user rights of a single widget. This is also used to display reports through the mobile app.
- **Linked dashboard** determines to which dashboard id the widget is linked to.
- **Widget type** is a classification of the widget. It is used in Kendo UI to determine what kind of chart is rendered.
- **Widget group** determines if the widget is a chart or something else, such as a map or a welcome banner. This information is used on rendering dashboard tiles, to keep widget look uniform.

- **Widget class** determines the class attributes of the HTML element that is rendered according to widget definition. This can utilize for example Bootstrap grid system by defining the widget size and behaviour on various screen sizes.
- **Widget datatype** allows to define a query level aggregation for the specific widget configurations. The implemented binning methods are a row count, average and percentage calculation.
- **Widget aggregation** allows configuring which aggregation function is used in Kendo UI chart tool. There is no default value to allow unaggregated visualizations.
- **Widget data grouping** determines which data field is used as a grouping parameter in Kendo UI charts. A default value is set as timestamp but can be set to be for example according location or employee name.
- **Widget module** defines which database table the widget module utilizes. If this is left empty the widget will use by default the setting from dashboard linked dashboard definition.
- **Widget form** determines a specific digitalized form to be used as a widget dataset.
- **Widget key field** is used to define the database column used as a key value of a key-value-pair. For example, it would be the question of a question-answer-pair. If left empty, the widget will use the default value set in dashboard definition.
- **Widget value field** defines the value field of a key-value-pair. In a question-answer-pair it would be the answer. If left empty, the widget will use the default value set in dashboard definition.
- **Widget SQL** gives an option to set a specific search criterion in the database query. For example, if it is wanted to visualize only certain values, the database query can be refined here. This field also enables having a custom data query if used in co-operation with widget binning type ’custom’. When left empty, the dashboard definition is used.

5.3 Data integration

The dashboard rendering process is divided into two separate sequences, one for a call for mobile dashboard and one for a call for an administrative dashboard. The difference between these two sequences is called the endpoint script. When making a dashboard call in the Reslink administrative platform, the executed module configuration and user session parameters are used by a dedicated generic PHP file that first makes a call for a dashboard definition by using the language key set in the module definition. This language key is compared to dashboard ids in data base and the definition of the matching dashboard is returned. The right widgets are queried from data base with a combination of the dashboard identity and user level information. All widgets are compiled to an array passed to the Smarty engine that in turn renders the HTML (see image 23).

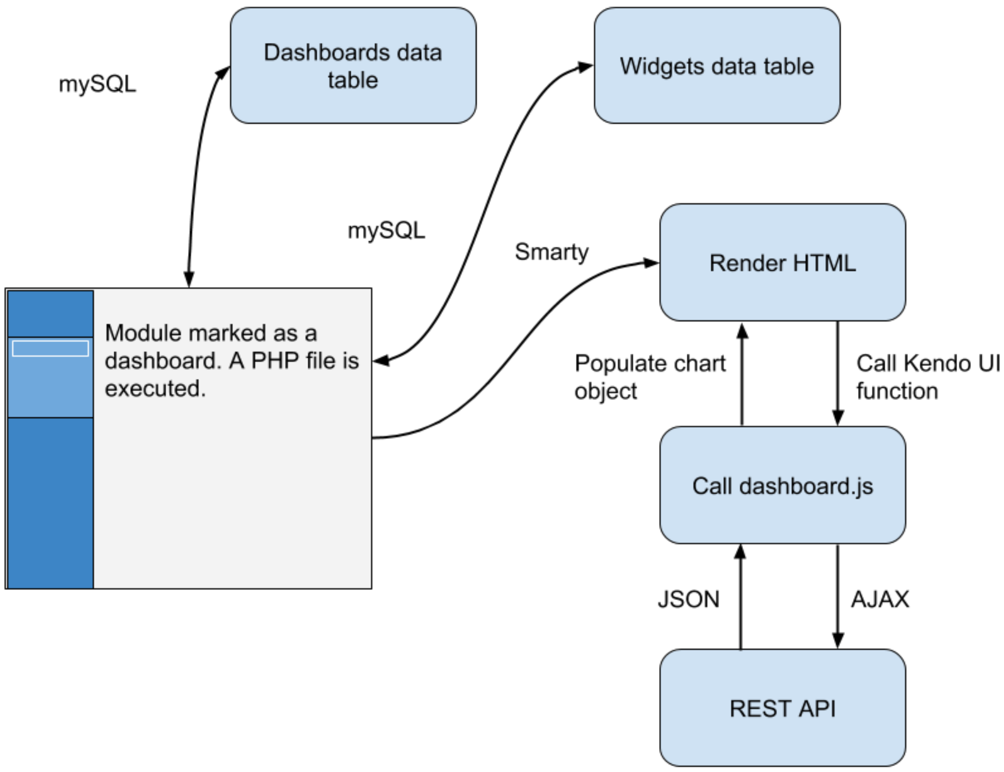

Image 23. Dashboard data connection and population

The mobile dashboard call is done by an HTTP request that has the users credentials, a dashboard handler parameter, and a dashboard name as a payload. First the endpoint handler script makes a check for the validity of the user credentials used to fetch organization and user level information. Like in the administrative dashboard call the dashboard identity, combined with user level, is used to prepare a MySQL query to fetch widgets for the dashboard. Widgets are rendered according to dashboard configuration. The difference to the administrative portal dashboard is that by default only chart widgets are included (see image 24).

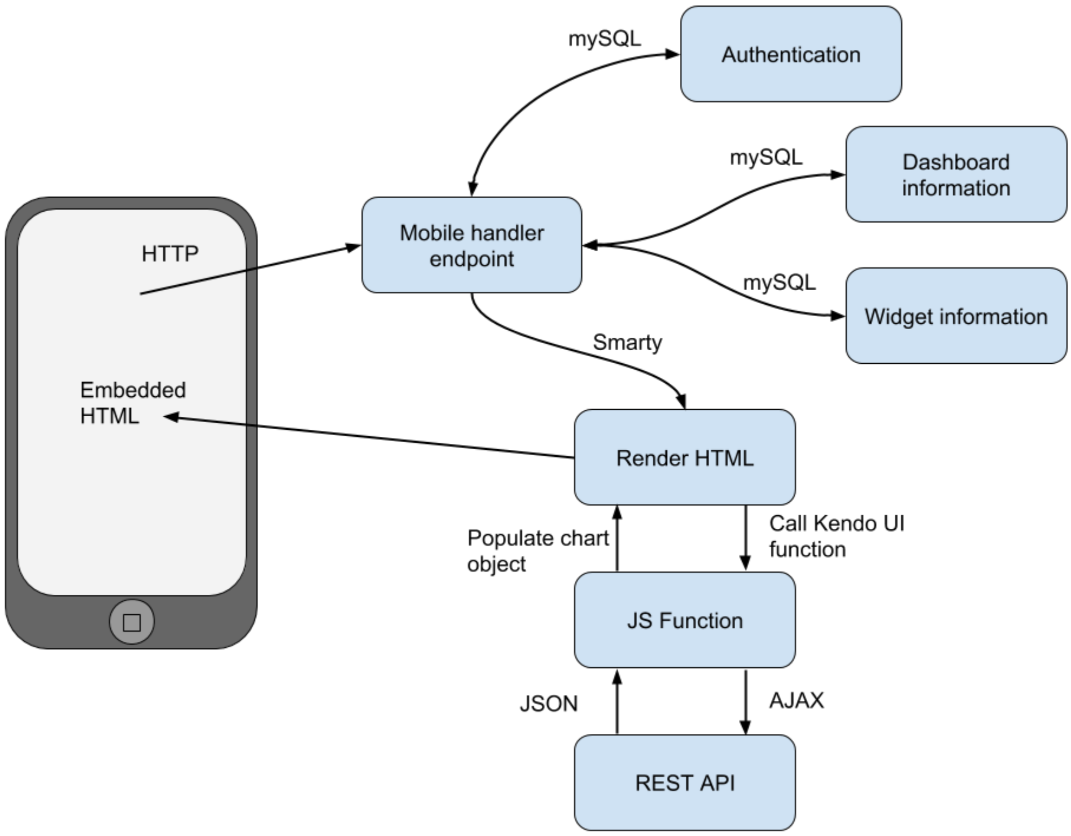

Image 24. Mobile dashboard request and data connection and population

Smarty is a framework used widely in the Reslink Solutions' platform. It allows a conditional HTML rendering according variables passed from PHP. This enables the use of a single Smarty template for every dashboard and makes same solution applicable for a variety of needs. For example, chart objects are rendered as HTML whenever the array object has a type 'chart' (listing 1).

```
{elseif $object.group==’chart’}
<div class=”panel-heading vd_bg_green”>
     <h3 class=”panel-title”>{$object.title}</h3>
     <div class=”vd_panel_menu”>
          <div data-action=”minimize” data-original-title=”Minimize”>
          <div data-action=”refresh” data-original-title=”Refresh”>
          <div data-action=”close” data-original-title=”Close”>
     </div>
</div>
<div class=”panel-body-list pd-15 style=”display: block;”>
     <div id=”{$object.id}”></div>
</div>
<script>
{$object.type}Chart(‘{$object.id}’, ‘{$object.aggr}’,‘{$object.grouping}’);
</script>
```

Listing 1. Smarty framework using conditional structure to render HTML and creating a JavaScript function definition using objects from an array of variables passed to it.

The Smarty template for dashboards makes a call for a JavaScript file, either a default or a customized one set in the dashboard definition, containing a function for each chart type. They are called dynamically for each chart object rendered according to the code example above. For example, a column chart calls a columnChart function passing chart id, aggregation method and data grouping settings to be used when rendering the chart.

Each chart function connects the created chart to the HTML element that has the chart id as an id attribute. Since the element id is a unique value this configuration enables the ability to have multiple charts of same type at the same time. The created chart object calls a data caller JavaScript function passing chart id and grouping parameter. The called function makes an AJAX call to the REST API including the chart id as a payload to the call (listing 2).

```
var dataSource = new.kendo.data.DataSource({
      transport: {
            read: {
                  url: apiUrl,
                  dataType: "json",
            },
      },
```

Listing 2. Creation of a Kendo UI DataSource object

If the data caller JavaScript function gets a response from the REST API in JSON format it will call a function that converts timestamp objects to date objects. This is because Kendo UI framework contains functionality that scales charts automatically to an adequate timeline format. For example, if visualization is from a period of a few days, the chart is rendered to show data on daily basis. If the data set contains data over months the framework will use a more compressed timeline format, for example moths or weeks. After the date conversion the data set is grouped according to widget configuration (listing 3). The response from REST API is returned as a Kendo UI data source object to the function making data call.

```
schema: {
      parse: function(data) {
            $.each(data, function(i, val){
                  val.timestamp = kendo.parseDate(val.timestamp)
            });
            return data;
      },
},
```

Listing 3. An example of timestamp objects being converted to date objects.

5.4 REST API

The data for dashboards is fetched from a REST API developed for the purpose. The data is queried by making an AJAX call and having the needed parameters in the query URL. The REST API developed for providing data for dashboard utilizes an already existing parent class that takes a handler type, dashboard in this case, as a parameter and redirects the call to the right end.

The end for dashboard makes a check for two values in the query URL: chart type and widget ID. The widget ID is used to fetch the configurations defined for the specific widget. If module, value field, key field, or SQL filtering is not set in widget configuration it is fetched from the dashboard definition. The values in widget and dashboard configuration are used to populate a generic MySQL query string which is used in most binning options. Because the generic query joins data from multiple data base tables the module setting determines prefixes to the key and value columns to ensure there are no ambiguity in the query (listing 4).

```
$sql = "select ({$pre}.{$keyField}) as 'key', ({$pre}.{$valueField} as
'value'"…
if(!empty($filter)) $sql.="$filter"
```

Listing 4. Dynamic MySQL statement that is populated by widget definition

The logic of the REST API is based on a switch sequence and data binning types are set as cases. If binning type is left empty, the API uses widget type to determine which case to use. If a case is not found and the widget type is not included in cases, the API will fall back to the default case, which uses only the generic MySQL statement populated with configuration data. The other binning options are as follows:

- Average
- Score
- Count
- Pie (widget type specific)
- Gauge (widget type specific)

If the defined MySQL statement including field, filtering and binning configurations gives a result, the API will reformat the data. Each binning case, barring gauge and pie, includes timestamps of result rows in two different formats: One with only date, and one with date and time information. Also, each binning case, with the same barring, includes the linked employee and location information to each row. Key and value columns are reported as ”key” and ”value” to each widget to keep the data references uniform and thus, as generic as possible.

5.5 Proof-of-concept dashboard

The context of the proof-of-concept dashboard built with the developed visualization builder solution is a mobile application that includes for example a simple digitalized audit form. The form produces transactional data that is calculated and presented for dashboard. The form consists of three questions with answer options red, amber and green. These options are scored as red being 1, amber being 2 and green being 3. The option also enables users to give a comment or take a picture related to the answer. The application also includes possibilities to contact Reslink Solutions' digitalization services, to leave general feedback about the application and to see visualizations straight from the mobile application. All data sent from the application is parsed into a database table as key-value-pairs maintaining also form identity, timestamp, and transaction reference information. These transaction references are also utilized to recognize which user has submitted the data.

The application is designed to be a simple give-away product and it enables a maximum organization level of three people. Because of the nature of the product, all three people are given both a manager level access to the administrative panel and an employee level access to the mobile application. These two user levels were treated as baselines in order to provide sufficient and informative dashboards, one for employee level and one for manager level users. In this case it was decided to offer the employee level visualizations only through the mobile application. This is based on previous customer cases that have indicated that other than administrative personnel seldom require credentials to the administrative panel portal. The manager level visualization dashboard is placed into the administrative panel as a landing page to show the usage of the application throughout the whole company.

The difference between these two dashboards was determined by two main factors. The first parameter is the device used to view the visualization, to design the wanted widgets to serve the purpose and offer a better user experience [28.]. The second parameter, to enhance the value and user experience of the dashboard, is determining which data is interesting from the user viewpoint, and which data can be shown and to whom.

The administrative level dashboard was designed to be more than a data visualization dashboard. It was decided to be more of a portal landing page giving information about filled forms but also welcoming the users and being used as an opportunity to present the possibilities of the platform due to the nature of the product. The visualization methods were chosen according to the give-away digitalized form context. Previous customer cases regarding audit forms have indicated that the information of an average result, the amount of auditions done over a span of time, the division of answers, and a daily average are considered as needed information. To improve the intuitiveness and thus usability, the colour theme of both dashboards, mobile and administrative, was set to follow the theme of red, amber and green [6].

A map was included to indicate the physical locations of the generated transactions [9 p. 145-147].

A radial gauge chart was implemented to indicate the all-time average score of the audition forms. The visualization method was chosen because in its simplicity it offers the information in a form that is familiar to many users. [9. p. 145-147]

A column chart was implemented to display the amount of completed audit forms. The x-axis unit is date and time to visualize the auditing intervals. [9. p. 145-147] In order to provide more information to the user while keeping the number of visual stimuli reasonable, drill-down functionality was implemented to this chart. By clicking a data series, the selected date's data is regrouped with employee information. [5., 9. p. 145-147]

An area chart was added to visualize the average score of the audit form over time. A line chart was initially considered due to its natural ability to indicate trends over time. [9. p. 145-147] The type was revised, and area chart selected to provide better readability of a larger object compared to a thin line. Also, in this case, to keep the visualization easily comprehensible, it was deemed viable to provide more accurate data with a drill-down functionality. By the first click the chart shows the division of the red, amber, and green answer from that date. By the second click it shows the proportional division between questions in the selected result. [5., 9. p. 145-147]

A pie chart was chosen to visualize the division between red, amber, and green answers. With only three groups a pie chart is a classic tool to visualize the divisional proportions of the parent group. [9. p. 145-147]

The mobile dashboard offered for employee level users was simplified to keep the user experience intact on a smaller screen. The first division was to remove all non-chart visualizations to improve the usability by removing unnecessary objects from the view. It was also deemed that the employee level user gains little added value from the radial gauge chart presenting company-wide all-time average score. To provide the most interesting information to the employee level user, all other chart type visualizations were included but showing only the user's own performance data. [22.]

5.6 Data integration to an external BI-software

To enable a broad range of visualization services, a data integration was researched. The purpose of the integration is offering an always-available stream of data to be used in an already existing Business Intelligence software. Microsoft Power BI was used as a research target because it is a widespread Business Intelligence tool, and by having a data integration to it, would cover a relatively sizeable portion of the need.

According to both example Microsoft Power BI documentation, it can handle data in JSON format [29]. Since data for the JavaScript-based dashboard solution is delivered as JSON, it was reasonable to research if the same or a similar REST API would provide the sufficient data connection. This would also enable the customer to have the data on demand.

The use of a REST API was confirmed with an experiment by Microsoft Power BI. It was able to fetch the data using only a link leading to an ending point providing a JSON response. This JSON response was successfully pivoted from key-value-pairs to a column-based data table, and further rendering visualizations of the data.

The needed REST API ending point for data integration could be for example structured in a very similar fashion than the mobile dashboard. In both cases an authentication is needed. However, whereas the mobile application is always linked to a certain customer environment and organization, the data integration API could enable data security issues with a simple mistyping or malicious intents. Most likely this API call type would need to include an API key, defined to a specific portion of the Reslink Solutions platform. The data rendering would also be different since, unlike in the mobile dashboard use case, in this case the JSON is delivered as such without a need to render HTML elements out of it (see image 25).

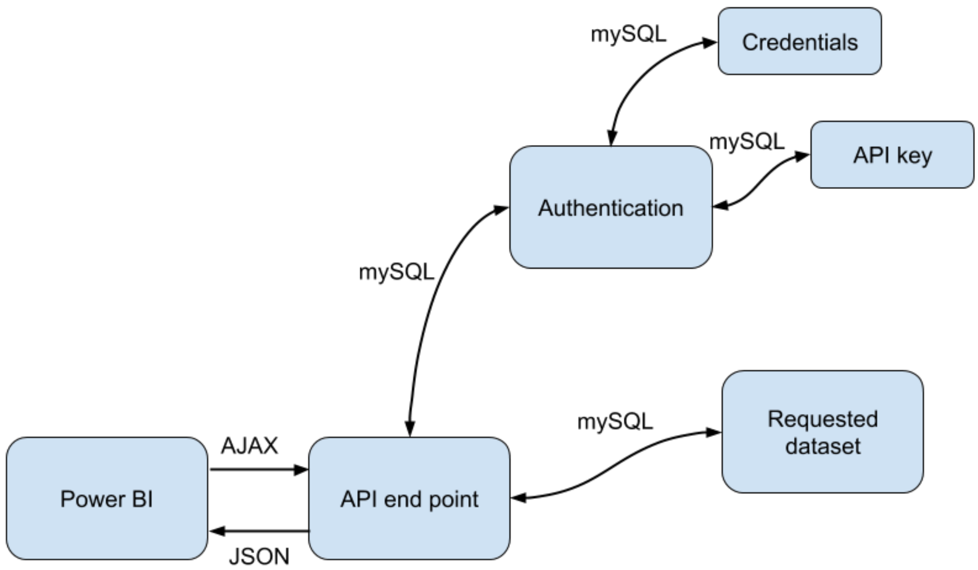

Image 25. An example structure of a data integration API call.

Microsoft Power BI supports a wide variety of data formats to be integrated into it. For example importable files, such as CSV and Excel spreadsheets, and database formats, such as MySQL and PostgreSQL, are supported. Reslink Solutions' platform already enables data sending automatically. For example, an already built-in Mobile Service API enables transaction forwarding to a web service provided by the customer. This can be utilized if the customer's web service is configured to store the data sets into a supported database, and which in turn is imported to Power BI. However, the problem with this approach is that the customer needs to have a web service configured and maintained in order to be able to have the data connection. Another example is automatic reporting that enables sending scheduled Excel or CSV reports via email. From the customer's perspective this has significantly less steps to take into consideration, but on the other hand the updated data is not available on-demand, and the automated reports only include reports from a single suborganization of the customer's administrative environment. [29.]

6 Conclusions

6.1 Research problem solution

The primary research problem of determining and developing a solution for a general tool to build responsive dashboards was approached by utilizing Reslink Solutions' modular platform structure. It was deemed vital for the success of the process to utilize the properties already found in the platform.

The first prototype required the user to define all three module layers, modules, dashboards, and widgets manually. It was determined that all these modules are needed in the process: Widgets module defining the single widgets, Dashboards module defining the high-level default values, and the Modules level defining the position and accessibility of the dashboard. The process was streamlined according to user feedback that the module for dashboard accessibility and position should be created automatically by enabling user level settings in the dashboard module. This change was made during the project and as a result, the increased efficiency improved the dashboard definition process.

The current implementation of the visualization building tool allows dashboard definition supporting multiple different data querying and aggregation settings. According to user feedback there is still room for improvement, such as adding a date range picker to the proof-of-concept dashboard. However, the current implementation documented in during this thesis can produce multiple different dashboards.

The most feasible outcome of this research and development process is the REST API, which also provides a solution to the secondary research problem of enabling data integration to an external BI software. The REST API is the centre piece that enables all needed visualizing solutions: A built dashboard, a mobile dashboard and in the future, the data integration. As such the REST API is configured to support data queries on transactions, which are sent from the mobile applications. In the future it might be needed to broaden it to support any database table and entirely customized queries instead of filtering.

6.2 Future development

The configurability and modular structure of both widgets and dashboards enables additional use cases of the developed solution in the future. The first suggested expansion is to have dashboards both as reports and as administrative. This functionality is mainly supported already due to the configurability of both querying and framework aggregation. Other, still a hypothetical, expansion is utilizing the widgets in other definition modules, such as user credentials to visualize for example the division between different user groups.

The whole variety of Kendo UI visualization types has not been utilized, but the implementation was able to provide the proof-of-concept dashboard. However, the development process raised questions about the framework's overall capabilities. Some aspects, such as hierarchically presented groups seemed to be somewhat lacking for example in donut chart type. In the future it might be feasible to reconsider another visualization frameworks should the need arise. The designed structure allows a customized JavaScript file, which enables further experimenting with other frameworks.

6.3 Overall success

This research and development process can be viewed as a success. During this process the ground work for a general visualization solution has been laid. The solution is already able to fulfil the requirements for fast deployability, configurability and responsive design, and with future iteration the scope of this solution can be broadened significantly.

The results of this work have opened possibilities for overall improvements for both usability and functionality in multiple sections of the platform, which were not involved in the process.

References

1 Titcomb, J. Mobile web usage overtakes desktop for first time. The Telegraph [document on the Internet]. The Telegraph; 2016 November 1 [cited 2017 November 3]. Available from: http://www.telegraph.co.uk/technology/2016/11/01/mobile-web-usage-overtakes-desktop-for-first-time

2 33 Billion Internet Devices By 2020: Four Connected Devices for Every Person in the World, Says Strategy Analytics [document on the Internet]. Boston: Cision PR Newswire; 2014 October 14 [cited 2017 November 5]. Available from: https://www.prnewswire.com/news-releases/33-billion-internet-devices-by-2020-four-connected-devices-for-every-person-in-the-world-says-strategy-analytics-279166141.html

3 User Experience [document on the Internet]. English Oxford Living Dictionaries [cited 2017 November 4]. Available from: https://en.oxforddictionaries.com/definition/user_experience

4 R. K. Paredes and A. A. Hernandez. Measuring the quality of user experience on web services: A case of university in the Philippines. In: 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM); 2017 Dec 1-3; Manila, Philippines: IEEE; 2017 [cited 2018 Feb 25]. pp. 1-6. Available from: IEEE Xplore

5 C. L. Caldiroli, R. Garbo, F. Pallavicini, A. Antonietti, A. Mangiatordi and F. Mantovani. How web design influences user experience: A multi-modal method for real-time assessment during web browsing. In: 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC); 2017 Jan 20; Las Vegas, USA: IEEE; 2017 [cited 2018 Feb 25]. pp. 1063-1066. Available from: IEEE Xplore

6 Krug, Steve. Don't make me think, revisited: a common-sense approach to web usability [e-book]. 3rd ed. New Riders; 2014 [cited 2018 Jan 10]. Available from: http://proquestcombo.safaribooksonline.com.ezproxy.metropolia.fi/book/web-design-and-development/9780133597271

7 Kabir MA, Rehman MU and Majumdar SI. An analytical and comparative study of software usability quality factors. In: 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS); 2016 August 26-28; Beijing, China: IEEE; 2016 [cited 2018 Feb 26]. pp. 800-803. Available from: IEEE Xplore

8 Faulkner, L and Wick, D. Cross-user analysis: Benefits of skill level comparison in usability testing. Science Dirext. [document on the Internet]; 2016 [cited 2017 Mar 11]. Available from: https://pdfs.semanticscholar.org/784c/bece69fea170e28d220be940c411208e8a46.pdf

9 Kirk, Andy. Data Visualization: a successful design process [e-book]. 1st ed. Packt Publishing; 2012 [cited 2018 Feb 28] Available from: https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=1108349

10 Few, S. The Encyclopaedia of Human-Computer Interaction, 2nd Ed., chapter 35. Interaction Design Foundation [document on the Internet]; 2013. Available from: https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/data-visualization-for-human-perception

11 Qu Z. and Hullman J. Keeping Multiple Views Consistent: Constraints, Validations, and Exceptions in Visualization Authoring. IEEE Transactions on Visualization and Computer Graphics. 2018 [cited 2018 Feb 28]; 24(1):468-477. doi: 10.1109/TVCG.2017.2744198

12 Beasley, M. Practical Web Analytics for User Experience: How Analytics Can Help You Understand Your Users [e-book]. 1st ed. San Francisco, CA: Elsevier Science & Technology; 2013 [cited 2018 Feb 20]. Available from: https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=1249219

13 Ward, M., Grinstein, G. and Keim, D. Interactive Data Visualization: Foundations, Techniques, and Applications [e-book]. Boca Raton, FL: Taylor & Francis Group; 2010 [cited 2018 Feb 27]. Available from: https://books.google.fi/books?hl=en&lr=&id=Kk7NBQAAQBAJ&oi=fnd&pg=PP1&dq=data+visualization+techniques&ots=ejSn2BN15q&sig=C4wT3WFQsWfDc0a_EHd13CLK-Nc&redir_esc=y#v=onepage&q&f=false

14 Sonehag C, Olofsson N and Simander R. Automatic binning in visual predictive checks. Department of Scientific Computing, Uppsala University [document on the Internet]; 2012 February 2 [cited 2018 Mar 2]. Available from: https://www.it.uu.se/edu/course/homepage/projektTDB/ht11/project4/Report_ht11_04.pdf

15 Ehsan H, Sharaf MA and Chrysanthis PK. Efficient Recommendation of Aggregate Data Visualizations. IEEE Transactions on Knowledge and Data Engineering. 2018 [cited 2018 Feb 27]; 30(2):263-277. doi: 10.1109/TKDE.2017.2765634

16 Gama S and Gonçalves D. 2014. Multi-level visualization of interrelated data entities. In Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14); 2014 May 27-29, Como, Italy. New York: ACM; 2014 [cited 2018 Feb 26] pp. 365-366. Available from ACM

17 Davies J. Sunburst Chart [document on the Internet]. [cited 2018 Feb 26] Available from: https://www.jasondavies.com/coffee-wheel

18 Scatter and Line Plots in R [document on the Internet]. Plotly; [cited 2018 Mar 10]. Available from: https://plot.ly/r/line-and-scatter

19 Parlapiano A. There Are Many Ways to Map Election Results - We've Tried Most of Them [document on the Internet]. New York Times; 2016 November 1 [cited 2018 Feb 20]. Available from: https://www.nytimes.com/interactive/2016/11/01/upshot/many-ways-to-map-election-results.html

20 Jain J, Ari I and Li J. Designing dashboards for managing model lifecycle. In Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology (CHiMiT '08) [serial on the Internet]. 2018 [cited 2018 Feb 25]. doi: 10.1145/1477973.1477989

21 Campos A, Vale F, and Alturas B. 2008. Human resources metrics dashboard. In Proceedings of the 26th annual ACM international conference on Design of communication (SIGDOC '08); 2008 Sep 22-24, Lisbon, Portugal. New York: ACM; 2008 [cited 2018 Mar 3], pp. 257-262. Available from. ACM

22 Schwendimann BA, Rodríguez-Triana MJ, Vozniuk A, Prieto LP, Boroujeni MS, Holzer A, Gillet D, and Dillenbourg P. Understanding learning at a glance: an overview of learning dashboard studies. In Proceedings of the Sixth International Conference on Learning Analytics & Knowledge; 2016 Apr 25-29, Edinburgh, United Kingdom. New York: ACM; 2016 [cited 2018 Mar 3]. pp. 532-533. Available from: ACM

23 Kerracher N, Kennedy J and Chalmers K. A Task Taxonomy for Temporal Graph Visualisation. in IEEE Transactions on Visualization and Computer Graphics. 2015 [cited 2018 Feb 26];21(10):1160-1172. doi: 10.1109/TVCG.2015.2424889

24 Andrews K, Osmić M, and Schagerl G. Aggregated parallel coordinates: integrating hierarchical dimensions into parallel coordinates visualisations. In Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business. 2015 Oct 21-22, Graz, Austria. New York: ACM; 2015 [cited 2018 Feb 28]. Available from: ACM

25 Amr, Tare & Stamboliyska, Rayna (2016). Practical D3.js [e-book]. Apress; 2016 [cited 2018 Mar 17] p. 53-55 Available from: https://link-springer-com.ezproxy.metropolia.fi/book/10.1007%2F978-1-4842-1928-7

26 Nair L, Shetty S & Shetty S. Interactive visual analytics on Big Data: Tableau vs D3.js. Journal of e-Learning and Knowledge Society [document on the Internet]. 2016 [cited 2018 Mar 16] Available from: https://www.learntechlib.org/p/173675/

27 Responsive Web Design [document on the Internet]. Google; 2017 September 27 [cited 2017 Nov 10]. Available from: https://developers.google.com/search/mobile-sites/mobile-seo/responsive-design

28 Data Sources in Power BI Desktop [document on the Internet]. Microsoft; 2018 March 6 [cited 2018 Mar 18]. Available from: https://docs.microsoft.com/en-us/power-bi/desktop-data-source