GIMA
Geographical Information Management and Applications

Master's Thesis

Cartographic implications of Vector Tile technology

Ingmar de Beukelaar

Universiteit Utrecht
TU Delft
University of Twente
Wageningen UR

September 2017 - March 2018
Final report - February 28th, 2018

Author: Ingmar de Beukelaar
Supervisor: Drs. Barend Köbben

Colophon

This master's thesis is commissioned by the Utrecht of University, University of Wageningen, University of Twente, and the Delft University of Technology as part of the Geographical Information Management and Applications (GIMA) - Master of Science.

Responsible Professor:
Prof.dr. Menno-Jan Kraak

Supervisor:
Drs. Barend Köbben

Contact information:
Ingmar de Beukelaar
ingmar.debeukelaar@gmail.com
UU student number: 3940330
ITC student number: s6030858

How to cite this master's thesis:
De Beukelaar, I.T.Y. (2018). *Cartographic implications of Vector Tile technology* (Unpublished master's thesis). Geographical Information Management and Applications (GIMA MSc).

Abstract

Cartographic implications of Vector Tile technology
by INGMAR DE BEUKELAAR - MARCH 2018

The Web has changed the way maps and geographical information are designed, produced and delivered as web maps by cartographers (Cartwright, Gartner, Meng, & Peterson, 2010). Most maps available on the Web today are based on well-established raster transmission methods. However, rapid technological innovation in web mapping is driving the need to utilize fast rendering tiles. Vector Tiles seem to be an emerging solution (Antoniou, Morley, & Haklay, 2009). Therefore, the goal of this thesis was to investigate the new opportunities and challenges that Vector Tiles offer for Web Cartographers. Several existing Vector Tile tools and technological solutions/workflows on how to implement Vector Tiles in Web Mapping were inventoried. Afterwards, two different workflows were assessed in terms of cartographic strengths & weaknesses. The aim was to investigate the cartographic potential of Vector Tile technology and solutions. The challenges of the thesis were to give an overview of the current state of Vector tile technology and to fill the knowledge gap between computer sciences and cartography by combining practical research regarding the emerging Vector Tile technology with cartographic theory.

Summary

Cartographic implications of Vector Tile technology
by INGMAR DE BEUKELAAR - MARCH 2018

The Web has changed the way maps and geographical information are designed, produced and delivered as web maps by cartographers (Cartwright et al., 2010). Most maps available on the Web today are based on well-established raster transmission methods. However, rapid technological innovation in web mapping is driving the need to utilize fast rendering tiles. Vector Tiles seem to be an emerging solution (Antoniou et al., 2009). Therefore, the goal of this thesis was to investigate the new opportunities and challenges that Vector Tiles offer for Web Cartographers. Several existing Vector Tile tools and technological solutions/workflows on how to implement Vector Tiles in Web Mapping were inventoried. Afterwards, two different workflows were assessed in terms of cartographic strengths & weaknesses. The aim was to investigate the cartographic potential of Vector Tile technology and solutions. The challenges of the thesis were to give an overview of the current state of Vector tile technology and to fill the knowledge gap between computer sciences and cartography by combining practical research regarding the emerging Vector Tile technology with cartographic theory.

The results show that while Vector Tile tools make it very easy for Web Cartographers to implement Vector Tile technology, that they do not always take into account how to solve certain cartographic challenges. It was identified that the main cartographic weaknesses in Vector Tile technology and workflows are the lack of control over the simplification process and the lack of support for custom projections with Vector Tiles. The cartographic strength identified was the support for styling and the ability of the Cartographer to customize these styles on the client side. The cartographic potential lies in developing the weaknesses and deploying the strengths.

It was concluded that Vector Tile technology positively contributes to the field of Cartography because it stimulates the effective communication of spatial information by means of technological improvements (i.e. the compact size and efficient caching of Vector Tiles enabling fast loading and high-resolution maps) and benefits regarding map design and interactivity (i.e. the flexible styling on the client side allowing for many map styles for the same geo-data). However, it was also identified that there could be more research on cartographic aspects regarding the too technical and performance driven Vector Tile technology which is due to an observed gap between one hand a lack of cartographic focus in computer sciences and on the other hand a technical knowledge gap with traditional cartographers. Therefore, there is a new potential and role for the cartographer to be more involved in the creation of Vector Tile tools to deploy cartographic knowledge. Developers with a background in computer sciences should take into account more the cartographic aspects in Vector Tile technology, whereas cartographers can nowadays not ignore the technical and programming part anymore. Bridging the gap between computer sciences and cartography was the aim of this thesis and at the same time the identified potential that can be deployed.

Acknowledgements

I would like to thank all the people that involved in the thesis for their support and contributions. First, I would like to thank my supervisor:

Drs. Barend Köbben, *University of Twente – Faculty of Geo–Information Science and Earth Observation (ITC)*, for his strong support with regular meetings and feedback, contacts with other people involved with Vector Tiles and not to forget the time and effort in checking this thesis.

Secondly, I would like to thank all the interviewees which made this thesis possible:

Edward Mac Gillavry, *Webmapper*, for his expertise in Web Mapping and Vector Tiles, his hospitality to invite me several times at his office and giving me the opportunity to ask questions and learn more about the practical side of Web Cartography and Vector Tiles.

Niene Boeijen, *Webmapper*, for her expertise in Web Mapping and Vector Tiles and her support to explain to me how to work with Tippecanoe and how the Vector Tile stack works.

Jeroen Hogeboom, *PDOK/Kadaster*, for his hospitality to invite me at Kadaster to show me their Vector Tiles project and discuss the user needs and opportunities of Vector Tiles.

Jan Willem Vierbergen, *ESRI Nederland*, for his interesting thoughts about the opportunities and challenges with Vector Tiles and especially is knowledge about how ESRI implements Vector Tiles for visualization and Vector basemaps.

Robert Nordan, *Norkart AS*, for his interesting thoughts about Vector Tiles and how they are suited for GIS applications, consumers and for visualization.

Prof. Stefan Keller, *The Geometa Lab, Hochschule für Technik Rapperswil (HSR)*, for his interesting thoughts and expertise about Vector Tile technology and especially his critical view on the potential and challenges of Vector Tiles for Web Cartographers.

Lukas Martinelli, *Mapbox*, for his interesting thoughts about Vector Tiles and especially his technical expertise as a computer scientist on how the technology works at Mapbox.

Lastly, I would like to thank all the other people involved that I did not mention and that have supported me while writing this thesis.

I hope you will all enjoy reading this thesis.

List of Abbreviations

AJAX	**A**synchronous **J**avaScript **A**nd **X**ML
BAG	**B**asisregistraties **A**dressen en **G**ebouwen
BGT	**B**asisregistraties **G**rootschalige **T**opgrafie
BRT	**B**asisregistratie **T**opografie
CSV	**C**omma-**S**eperated **V**alues
CSW	**C**atalog **S**ervice **W**eb
ERP	**E**xtended **R**esearch **P**roposal
GIS	**G**eographical **I**nformation **S**ystem
GDAL	**G**eospatial **D**ata **A**bstraction and **L**ibrary
GIMA	**G**eographical **I**nformation **M**anagement and **A**pplications
GML	**G**eographic **M**arkup **L**anguage
GUI	**G**raphical **U**ser **I**nterface
HSR	**H**ochschule für Technik **R**apperswil
HTML	**H**yper**T**ext **M**arkup **L**anguage
ISO	**I**nternational **S**tandards **O**rganisation
ICT	**I**nformation and **C**ommunication **T**echnology
ITC	Faculty of Geo-Information Science and Earth Observation (University of Twente)
JPEG	**J**oint**P**hotographic **E**xpert **G**roup
JSON	**J**ava**S**cript **O**bject **N**otation
KML	**K**eyhole **M**arkup **L**anguage
MVT	**M**apbox **V**ector **T**iles
OGC	**O**pen **G**eospatial **C**onsortium
OSM	**O**pen**S**treet**M**ap
PDOK	**P**ublieke **D**ienst **O**p de **K**aart
PBF	**P**rotocolbuffer **B**inary **F**ormat
PNG	**P**ortable **G**raphics
RID	**R**esearch **Id**entification
SDI	**S**patial **D**ata **I**nfrastructure
SLD	**S**tyled **L**ayer **D**escriptor
SQL	**S**tructured **Q**uery **L**anguage
SVG	**S**calable **V**ector **G**raphics
TMS	**T**ile **M**ap **S**ervice
UU	**U**niversity **U**trecht
UT	**U**niversity of **T**wente
VT	**V**ector **T**iles
WFS	**W**eb **F**eature **S**ervice
WMS	**W**eb **M**ap **S**ervice
WMTS	**W**eb **M**ap **T**iling **S**ervice
WPS	**W**eb **P**rocessing **S**ervice
WWW	**W**orld **Wide Web**
XML	e**X**tensible **M**arkup **L**anguage

Glossary

Cartographic implications of Vector Tile technology Refers to what is likely to happen for Cartographers as a result of Vector Tile technology, namely the implications the technology has on the cartographic possibilities and challenges.

Cartographic potential Refers to the cartographic possibilities or opportunities (for the future). It has an emphasis on the positive advantages that could possibly be realized (e.g. what can be done with Vector Tiles from a cartographic point of view what has not been done yet).

Cartographic strengths and weaknesses The possibilities and challenges from a cartographic point of view. Unlike cartographic potential, it refers to both opportunities and problems and not only the positive advantages that could possibly be realized.

Cartography The discipline dealing with making and using maps. In this thesis, Cartography is seen as more than just the visualization or map design phase of creating maps. It is seen as a science, craft, and technology that includes the communication of spatial information as well as all the stages from data acquisition to presentation and use Basaraner (2016).

Computer Sciences The science of exploring the digital world of information. Computer and Information Science is used as a synonym in this thesis. It is also often referred to the study of coding and engineering related to Vector Tiles.

GIS GIS can refer either to Geographic Information Systems, a system designed to capture, store, manipulate, analyze, manage, and present spatial or geographic data. It may also refer to the Geographic Information Science, the academic discipline that studies geographic information systems (Wikipedia, 2018b).

PDOK Publieke Dienst of de Kaart, a central facility in the Netherlands for unlocking geodatasets of national importance. PDOK makes digital geospatial data available as data services and files. According to PDOK, the centralization of geodata must always lead up to up-to-date, reliable and free to charge available data. Kadaster as an organization is responsible for delivering and providing PDOK services (PDOK, 2018a).

Vector Tile implementations Tools or technologies using Vector Tiles. These can, for instance, be parsers/generators, applications/plugins, command line utilities, servers or clients.

Vector Tile solution A way to implement Vector Tiles by means of one or more Vector Tile tools. It is considered to have the same meaning as a Vector Tile workflow. However, with a workflow there is an emphasis on the set-up or process steps, whereas a solution could also be one single Vector Tile tool (e.g. GeoServer for both generating and serving Vector Tiles).

Vector Tile technology The application of scientific knowledge and the collection of techniques, skills, methods, and processes regarding Vector Tiles.

Vector Tile tool An implement such as a parser/generator, application/plugin, command line utility, server or client.

Vector Tile workflow A way to implement Vector Tiles by means of one or more Vector Tile tools. It is considered to have the same meaning as a Vector Tile solution. However, a Vector Tile workflow has an emphasis on the stack. With a Vector Tile solution there is less emphasis on the set-up or process steps since a Vector Tile solution could also be one single Vector Tile tool.

Vector Tiles Packets of geographic data, packaged into pre-defined roughly-square shaped "tiles" for transfer over the web (Martinelli & Roth, 2015).

Web Cartography Web cartography is defined as *"being concerned with theoretical and practical issues that the web offers for the design, production, distribution, and use of maps and geographic information"* (Basaraner, 2016; Kraak & Brown, 2001; Peterson, 2008). Web Cartography is no different than traditional cartography because it is about the design, production and use of maps. The only difference is that is restricted to the WWW as a medium (Kraak & Brown, 2001).

Web Mapping Web Mapping is "the process of designing, implementing, generating, and delivering maps on the World Wide Web" Neumann (2008, p. 567). It is often used a synonym for WebGIS. While Web Mapping primarily deals with technological issues, Web cartography additionally studies theoretic aspects such as "the use of web maps, the evaluation, and optimization of techniques and workflows, the usability of web maps, social aspects, and more" Neumann (2008, p. 567).

WebGIS WebGIS is here used as a synonym for Web Mapping because web maps are often the presentation medium for WebGIS (Neumann, 2008). WebGIS *"denotes a type of GIS, whose client is implemented in a Web browser"* (Yang, Purves, & Weibel, 2007, p. 319).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

The Web has changed the way maps and geographical information are designed, produced and delivered (Cartwright et al., 2010). Most maps available on the Web today are created using well-established technological solutions based on raster graphics. However, rapid technological innovation in Web Mapping is driving the need to utilize fast rendering tiles. Vector Tiles seem to be an emerging solution (Antoniou et al., 2009). Therefore, the goal of this thesis is to investigate the new opportunities and challenges that Vector Tiles offer for web cartographers. The aim is also to fill the gap between Computer Sciences and Cartography by combining practical research regarding the emerging Vector Tile technology with cartographic theory. The next section will go into further detail about the motivation and problem statement of this research in order to understand the overall setting in which this research is situated.

1.2 Motivation and problem statement

Maps on the internet are nowadays a major form of spatial information delivery (Peterson, 2008). The majority of the maps available on the Web today are raster based. This is according to Antoniou et al. (2009, p. 56) *"because transmission methods for raster data over the Web are well established and easily implemented"*. While this is the case for raster techniques, more challenging issues arise for vector data as reducing the detail can cause for instance complexity or topological challenges (S. Li et al., 2011). The complexity of simplifying vector map data, is also one of the reasons why there is relatively few research on it (Yang et al., 2007). Antoniou et al. (2009) mention that vector data suffered from problems that prevented wide implementation such as on-the-fly generalization and efficient transmission over the Web. It is therefore important to perform more research on vector transmission techniques, because there are some cases where raster images are inadequate (Antoniou et al., 2009) or where vector graphic formats offer advantages for interactive mapping over raster-based maps (Carpendale, 2003). The main limitation of some raster-based mapping applications is interactivity or direct object manipulation (Antoniou et al., 2009; Bertolotto & Egenhofer, 2001), which vector techniques can offer to fulfill the wishes of certain user needs.
The tiled vector technique is one of the vector data transmission approaches that enables data reduction of vector datasets. With Vector Tiles, a dataset is split into smaller pieces and transmitted to the client where these smaller pieces are reassembled based on predicted user behavior using caching methods (S. Li et al., 2011; Quinn & Gahegan, 2010). Performing research on this topic can be of added value, because Vector Tiles have several technological advantages such as faster rendering, less bandwidth needed due to a smaller size of tile packages and more tiles that can be generated in less time (Mapdata Services, 2017). Furthermore, Vector Tiles

can come with map design, symbolization and interactivity benefits. It can offer more design flexibility by for instance styling capabilities that can be applied in the browser on the fly when requested by the client side (Cartwright, Peterson, & Gartner, 2007; Mapbox, 2017b; Mapdata Services, 2017; Quinn & Gahegan, 2010).

In order to understand the development of Vector Tile technology, and partly also why there is limited research on it, it is first is needed to understand the way maps are nowadays seen and how the cartographic discipline has had different paradigms and trends that have to be acknowledged. While on one hand cartography tends to be seen only as a visualization phase of spatial data handling, or only as the art or craft of making maps, cartography can, on the other hand, be regarded as a science that covers the entire phases of spatial data handling (Basaraner, 2016; Ramirez, 1993). The environment in which maps have been produced and used has changed considerably. The changes and trends seem to happen on two sides of cartography: in the scientific domain and in the public domain. Communication about maps and cartography is no longer done by professionals only, but also by a group of non-experts or neogeographers. Where scientists use maps with cartographic knowledge to communicate a message or to solve their problems, neogeographers, non-experts or coders want to grab the tools that are available to create their own maps (Kraak, 2011). Map users have gone from a passive role, as map readers, to a more active role, as contributors in the map mapping process (Ory, 2016). Besides, maps are being used and created by more people than ever before (Kraak, 2011). The new mapmakers have different demands and objectives, and force a change from a traditional supply-driven map production to demand driven map production. Expertise is still required, however, the role of maps has changed and expanded due to the influence of technology (Kraak, 2009).

The influence of changing technology on cartography has always been there, because the tools used in cartography are also changing, from analog tools towards computer tools (Bostock & Davies, 2013). The question a cartographer could ask is whether a cartographer can actually make a good map from code? Should cartographer be coders? Or should coders become cartographers? Cartography and coding increasingly are intertwined (Roth, Hart, Mead, & Quinn, 2014). A distinction could be made between modern map makers that are coders, and traditional map makers that are expert-cartographers. There seems to be a new role for the cartographer to provide tools that implement cartographic intent. This means that a cartographer could have a role in the toolmaking process or in the knowledge process (Köbben, 2014).

With mapping technologies facing increasing interoperability and flexibility, new opportunities are created for cartographers, such as more interactivity or interconnectivity (Kraak, 2009; L. Li, Hu, Zhu, Li, & Zhang, 2017; S. Li et al., 2011; Roth, Donohue, Sack, Wallace, & Buckingham, 2014). However, as technology evolves, there are new challenges due to the fact that it is becoming ever more difficult to maintain an overview within the current pace of technological innovation in Web Mapping (Roth, Donohue, et al., 2014). Making maps, seen as an art or science, continues to change rapidly and new mapping techniques are difficult to keep up with (Cartwright et al., 2010; Muehlenhaus, 2014). Therefore, the recent technological developments in computer- and information science ,which have strongly influenced the cartographic discipline, should be more extensively researched.

Furthermore, considering that geospatial technologies take advantage of the constant evolution of information and communication technologies, Spatial Data Infrastructure (SDI) appeared as a new paradigm in geospatial data handling (Bocher &

Ertz, 2017). Therefore, the role or status cartography is to be questioned due to fundamental changes in information and communication technologies. Cartographic knowledge within SDI can add value to spatial data by offering user-oriented design solutions and offering solutions for cartographic interoperability (Hopfstock & Grünreich, 2009).
It is important that map makers don't repeat the same mistakes while (re)using spatial data or maps. The main issue observed is that many maps on the Web or coming from SDI, often present a serious lack of cartographic knowledge (Bocher & Ertz, 2017; Muehlenhaus, 2014). This is because new available techniques for creating and distributing Web maps can be overwhelming for map authors, while tools for Web maps meeting the demands of cartographic principles and high-quality are hard to find (Cartwright et al., 2010). In line with this, Hopfstock and Grünreich (2009) underline that poor map design results are the consequence of a "too technology- and/or data-driven approach" (Bocher & Ertz, 2017). This observation highlights the need for WebGIS or computer sciences to take into account the knowledge of cartography on how to make proper maps that are not only usable but also reproducible or inter-operable. That's why bridging the gap between new technologies, such as Vector Tile technology, and cartographic knowledge is considered as being an important objective of this research.

The main assumption in this research is that the emerging Vector Tile technology positively contributes to the field of cartography because it stimulates the process of effectively communicating spatial information by means of technological improvements and benefits regarding map design and interactivity, especially on the client side. The assumption is based on the often repeated argument in the cartographic literature that the aim of cartography and maps is to effectively communicate a message or geospatial information (Basaraner, 2016; Muehlenhaus, 2014; Ormeling & Kraak, 2010; Ory, 2016). According to Kraak and Brown (2001) there is a need for the cartographer to concentrate on the effectiveness of maps and mapping technologies. Furthermore, the argument can be made that Vector Tiles serve maps more rapidly and more effective because they are rendered only when requested by a client (Mapbox, 2017b). This strengthens the cartographic effectiveness, which a key aspect to consider for the future development of Web Cartography (Kraak & Brown, 2001; Muehlenhaus, 2014). Research is needed in order to defend this assumption. The role or position of Vector Tile technology should be analyzed in the new demand-driven mapping environment and within the cartographic discipline. Such research is vital in order to fill the gap between computer sciences and cartography by combining practical research regarding Vector Tile technology with cartographic theory. In the next section, section 1.3, the relevance of this research will be further discussed.

1.3 Research relevance

The relevance of this research is, as discussed in section 1.2, to fill the research gap between technological improvements regarding Vector Tiles and cartographic theory from the cartographic scientific discipline. Defining the status of cartography as a scientific discipline and the position of Vector Tile technology within the cartographic discipline will be the first step to achieve this. A literature review on the latest developments regarding Vector Tile technology and how to implement Vector

Tiles solutions will contribute to the limited scientific research available on this subject. Next to that, by creating an inventory of the tools and technologies available on the web and assessing the strengths and weaknesses in terms of cartography of different Vector Tile workflows, a step forward in developing standardizing Vector Tile technology can be made. This inventory of current tools and technologies available regarding Vector Tiles solutions is also relevant because it gives an overview of existing and available tools in the fast pace in which nowadays new technologies and tools are developed in WebGIS. Especially for the new type of map users, as discussed earlier in the context of 'cartography from code' and 'neocartography', this can be of particular interest. It contributes to the understanding and accessibility (findability) of available tools and technologies regarding Vector Tiles.
Furthermore, identifying the cartographic potential of Vector Tile tools and technology is important to observe where more research on Vector Tile technology could be needed and what could be realized to improve Vector Tile technology and its implementation by Web Cartographers.
By analyzing where the cartographic strengths and weaknesses are in Vector Tile technology, cartographic knowledge can be made more explicit. Exploring to what extent the Vector Tile technology contributes to the cartographic need for effective communication of spatial information, is an innovation to aim at.

1.4 Research objectives

The main objective of this thesis is to **analyze the cartographic implications of Vector Tile technology**. Below, several sub-objectives are formulated that will contribute to the main objective. The sub-objectives overlap to some extend but cover the problem statement in its entirety and furthermore divides this research into logical parts.

- To understand the influence of technological developments in computer- and information science on cartography and its changing role (RO1).
- To gain more insight on the latest developments regarding Vector Tile technology by means of a literature study (RO2).
- To inventory Vector Tile tools and technologies currently available on the Web and identify how they are implemented as a solution or in a workflow (RO3).
- To relate Vector Tile technology with web-cartography theory by evaluating current Vector Tile solutions or workflows on their Cartographic strengths and weaknesses (RO4).

1.5 Research questions

To defend the statement that Vector Tile technology contributes to the field of Cartography, it is vital to research the cartographic implications caused by Vector Tile technology. Therefore the main or central research question is the following:

What are the cartographic implications of Vector Tile technology?

In order to answer the main research question and all the research objectives presented in section 1.4, the following sub-questions have been formulated:

1. What is the influence of recent technological developments in computer- and information science on cartography? (RQ1)

   The influence of changing technology on cartography and the changing role of cartographers is going to be researched (traditional cartographers vs. neo-cartographers/coders). Performing a literature review on recent technological developments with regard to the field of cartography is also needed to understand the position of web-cartography within the cartographic discipline

2. What are the latest developments regarding Vector Tile technology and what research has already been done on this? (RQ2)

   A literature review will be done in order to investigate the latest developments and research performed regarding Vector Tile technology. This sub-question relates Vector Tiles to web-cartography and contributes to the understanding of the position of Vector Tile technology within the cartographic discipline

3. What Vector Tile tools and technologies are currently available on the web and how are they implemented in a workflow? (RQ3)

   An overview will be created of which tools and technologies are currently available on the web. Several possible workflows/solutions will be identified.

4. How to assess Vector Tile tools and solutions on their cartographic strengths and weaknesses? (RQ4)

   This sub-question makes a link between research on Vector Tile technology and cartographic theory. It analyzes to what extent Vector Tile technology contributes to the effective communication of spatial information by linking it on one hand to cartographic theory and on the other hand to practical technical knowledge. The aim here is to combine cartographic theory and the findings of the experts-interviews. Furthermore, the goal is to describe how the set-up of a Vector Tile solution/workflow influences the possibilities and challenges from a cartographic point of view. The question was asked whether a certain Cartographic potential can be realized with a specific Vector Tile tool or solution/workflow.

1.6 Research scope

In section 1.3 it was described what the relevance of this thesis is and in 1.4 the objectives were presented. In this section, it is important to describe all the elements that this thesis does not cover. This is essential in order to maintain a realistic scope of this research and to be able to focus on the thesis objective to analyze the Cartographic implications of Vector Tile technology.

This thesis is not about:

- coming up with (open) standards for Vector Tiles.
- developing a standardized or unified cartographic (styling) language.
- creating a tool that implements Vector Tiles.
- coming up with a new technology or methodology regarding Vector Tiles
- creating/providing Vector Tiles itself.

1.7 Reading guide

In the next chapter, Chapter 2, the research steps and methods that apply to this thesis are first discussed.
Afterwards, Chapter 3 and Chapter 4 cover the theoretical framework of this research. The third chapter addresses an extensive literature review on the influence of changing technology on cartography and the changing role of cartographers, whereas the fourth chapter provides a literature review on the latest developments and research performed regarding Vector Tile technology.
A result of the theoretical part of this thesis, Chapter 5 presents the most important findings relevant to the theoretical framework and research. A conceptual model displays the relationships between the different concepts that are discussed.
Subsequently, Chapter 6 provides an inventory of the Vector Tile tools and technologies that can be implemented by Web Cartographers. By giving an overview, the findings describe the current state of the art of the Vector Tile technology. It is furthermore identified how the different tools and technologies can be implemented as a solution or in a Vector Tile workflow.
Chapter 7 extends the practical research part by relating Vector Tile technology with Cartographic theory. The position and role of Vector Tiles for Web Cartography is discussed and the findings from the expert-interviews are covered. Moreover, the cartographic strengths and weaknesses of two example Vector Tile workflows and assessed and it is investigated where the cartographic potential lies in the Vector Tile workflows and in Vector tile technology.
In Chapter 8, the results of the empirical research are placed in a broader context and discussed. Giving meaning to the current state and potential of Vector Tile technology for Cartography and Web Cartographers is the focus of the discussion. Lastly, Chapter 9 presents the conclusion of this thesis by answering the research questions, discussing the research limitations and proposing some recommendations for further research or for future Vector Tile projects.

Chapter 2

Methodology

This chapter discusses the research steps and methods that apply to this thesis. This research applies qualitative research methods. The research approach of the thesis is based on an investigation into the relationships between practical and scientific findings regarding Vector Tile technology and cartographic theory. Literature research as well, as practical research, are performed in order to investigate the new opportunities and challenges that Vector Tiles offer for Web Cartographers.

2.1 Research approach

The research approach of this thesis is based on a theoretical part and a practical part. The theoretical part is followed by the empirical analysis which consisted of linking the theoretical part with the practical part to assess the cartographic potential and implications of Vector Tile technology.

2.1.1 Theoretical research

The theoretical part consists of a literature review on the influence of changing technology on cartography and the changing role of cartographers. Performing a literature review on recent technological developments with regard to the field of cartography is needed to understand the current position of web-cartography within the cartographic discipline. After, the literature review discusses the latest developments and research performed regarding Vector Tile technology. Both aspects of the literature review are essential for relating Vector Tiles to web-cartography and understanding the position of Vector Tile technology within the cartographic discipline.

2.1.2 Practical research

The practical part of this research is composed of on the one hand a classification and inventory of Vector Tiles solutions and on the other hand on expert-interviews. The purpose of the interviews is to find out who is involved with Vector Tiles (the users) and what their requirements are. It should be noted that the users are not end-users in this context. After asking about their involvement with Vector Tiles, the interviewees are asked to identify the (potential) users of Vector Tile technologies. Who are the users? What are their requirements and expectations? What do they want to achieve with Vector Tiles? What is the reason to be involved with Vector Tiles? Are they aware of the benefits and drawbacks of Vector Tiles? How are current Vector Tile solutions experienced? Furthermore, the role of Vector Tiles for cartography was discussed. This includes cartographic advantages and disadvantages of Vector Tiles. Several cartographic topics or parameters was discussed and linked with Vector Tile technology. Some examples: What type of map best fits the Vector Tile technology? Do Vector Tiles face cartographic generalization problems? What about labeling or projections and Vector Tiles? Are Vector Tiles solutions well documented? How to involve users and end users more in styling maps? etc. Linking cartographic knowledge and the outcomes of the expert-interviews with the more technology-driven Vector Tiles developments and solutions, contributes, on one hand, the understanding of the emerging Vector Tile technology and on the other hand helps to fill the gap between computer sciences/web-technology and Web Cartography.

2.2 Research methods

In this thesis, for both the data collection and the data analysis qualitative research methods were chosen. The choice for qualitative research can in the first place be motivated by the fact that this is an exploratory study. Qualitative research lends itself very well to exploratory research (Boeije, 't Hart, & Hox, 2009). With qualitative research, there is a search for meaning. Qualitative research is inductive which means that it is about the understandings from patterns in the data rather than collecting data to assess hypotheses or theories. It is often referred to the term "grounded theory" which involves the inductive theorizing process for building theory. The purpose of grounded theory is to develop theory about phenomena based on a question or a collection of qualitative data (Glaser & Strauss, 1967; S. J. Taylor, Bogdan, & DeVault, 2015). Concerning the data collection, qualitative in-depth interviews were chosen. Through these interviews, the goal was to gain more insight into the role of Vector Tiles for cartography. The data analysis was done on the basis of a thematic analysis based on different cartographic topics identified in the theory which were found important for Web Cartography.
The below table summarizes the used methods and results per research question and objective (Table 2.1):

Research Objective	**Research Question**	**Research Method**	**Result**
To understand the influence of technological developments in computer- and information science on cartography and its changing role (RO1)	What is the influence of recent technological developments in computer- and information science on cartography? (RQ1)	Literature review	Chapter 3
To gain more insight on the latest developments regarding Vector Tile technology (RO2)	What are the latest developments regarding Vector Tile technology and what research has already been done on this? (RQ2)	Literature review	Chapter 4
To inventory the Vector Tile tools and technologies currently available on the Web and to give an overview of different Vector Tile solutions/workflows (RO3)	What Vector Tile tools and technologies are currently available on the web and how are they implemented in a workflow? (RQ3)	Practical research: inventory	Chapter 6
To relate Vector Tile technology with Web Cartography theory by evaluating the available Vector Tile tools and solutions/workflows on their cartographic strengths and weaknesses (RO4)	How to assess the available Vector Tiles tools and solutions/workflows on their cartographic strengths and weaknesses? (RQ4)	Practical research: expert-interviews	Chapter 7

TABLE 2.1: An overview of research methods per research question and objective

2.2.1 Data collection

For the data collection part of this thesis, in-depth interviews or semi-structured interviews were conducted. Typical of an in-depth interview is that an interview scheme is used which contains the questions and/or themes that must be addressed during the interview. Semi-structured interviews offer a good basis to structure the interviews, while also remain sufficiently open by giving space for comments or new spontaneous questions based on the answers to the main questions (Boeije et al., 2009). This means that it was possible to deviate from the predefined general questions. A topic list was used for giving structure to the interviews. The topic list consists of main questions, interspersed with themes/points and probing questions. The topics were prepared in accordance with the findings from the literature study in Chapter 3 and Chapter 4. All topics encompass themes that are considered to be important for assessing web mapping and Vector Tile solutions (Table: 2.3). Unlike a fully structured interview, the interviewer is free to follow the course of the conversation. The order/sequence of the questions on the list does not have to be kept (Boeije et al., 2009). Because the questions or sequence of the questions were not always the same for different interviews, this means that they become less valid. However, the advantage of semi-structured interviews is that there is a certain freedom to elaborate more on certain questions, without the need to follow the same sequence or structure for each interview within the research. The possibility of probing is an advantage of semi-structured interviews. According to S. J. Taylor et al. (2015, p.123) *"One of the keys to successful interviewing is knowing when and how to probe"*. Moreover, the researcher tries to maintain a kind of dialogue with the respondent. It is the task of the interviewer to obtain information of the interviewee who is as detailed as possible. In addition, it is necessary to make recordings of the in-depth interviews recordings. This way it is possible make transcription better afterwards and to stay focused during the conversation and still obtain all information. This results in more and more detailed information, what the aim is of qualitative research is (Boeije et al., 2009; S. J. Taylor et al., 2015).

2.2.2 Data analysis

Qualitative data analysis is a process that involves different stages. At the start of the data analysis it is essential becoming familiar with the data. An important step in this is making transcripts, this means that the audio recordings are written into text. Re-reading the transcripts helps becoming familiar with the data. Making a transcript is very labor intensive process and often underestimated (S. J. Taylor et al., 2015).
Another important stage of the data analysis is the search for topics or themes in the data. In order to find meaning in the data, researcher look for categories as a way of developing and refining interpretations of the data (Boeije et al., 2009; S. J. Taylor et al., 2015). This process is also known as coding and involves bringing together all the data based on major themes, ideas or concepts. Categories can also be based on concepts that have been read by the researcher in the literature. The latter is the case in this research. Existing research is used to sharpen the researcher's view and improve search strategies (Boeije et al., 2009). Coding the data provides a way of storing the data so that they can be easily retrieved for analysis and presentation (S. J. Taylor et al., 2015). With the most important codes, it is possible to build up theory in order to establish relationships and connections between the data (Boeije et al., 2009; S. J. Taylor et al., 2015). The transcriptions of the interviews are to be

Topic	What	Sources
Users and user requirements	Identifying users and user needs/requirements	Kraak and Brown (2001); Muehlenhaus (2014); Roth, Donohue, et al. (2014)
Knowledge & learning curve	To assess the knowledge of the interviewee regarding Vector Tiles and their willingness to learn more about the technology & whether it is easy or hard to learn about Vector Tile technology	Ballatore, Tahir, McArdle, and Bertolotto (2011)
Stability & Performance	Refers to the stability, reliability and performance of Vector Tile technology according to the interviewee's experience	Ballatore et al. (2011)
Scalability & Extendibility	The ability of Vector Tile technology to become more important or to continue to function well when there is a growing amount of work to handle according to the interviewee's experience	Ballatore et al. (2011)
Interoperability	How the interviewee considers the integration of different Vector Tile tools or solutions	Ballatore et al. (2011)
Formats, standards & schemes	The interviewee's opinion regarding the support of Vector Tile data formats, standards and different tiling schemes	Ballatore et al. (2011); Open Geospatial Consortium (2017a, 2017b)
Documentation and accessibility	The completeness, readability, quality and accessibility of documentation about Vector Tile technology and solutions according to the interviewee	Ballatore et al. (2011)
Community support	The support of Vector Tiles in the field of GIS and Cartography according to the interviewee	Ballatore et al. (2011)
Frequency of updates	The improvements and updates of Vector Tile solutions on the Web according to the Interviewee	Ballatore et al. (2011); Roth, Hart, et al. (2014)
Cartographic projections	The choice of projections and handling of different cartographic projects in combination with Vector Tiles	Open Geospatial Consortium (2017a, 2017b); Ormeling and Kraak (2010)
Map design and styling	Design and styling possibilities according to the interviewee	Lambert and Zanin (2016); Muehlenhaus (2014); Open Geospatial Consortium (2017b); Ormeling and Kraak (2010); Peterson (2003)
Geometry	Interviewee's opinion about the support of different types of geometries with Vector Tiling	Open Geospatial Consortium (2017a, 2017b); Ormeling and Kraak (2010)
Preservation of Topology	The challenges or opportunities of preserving topology according to the interviewee	Open Geospatial Consortium (2017a, 2017b); Ormeling and Kraak (2010)
Simplification and generalization	Handling of generalization (such as filtering options) according to the interviewee	Open Geospatial Consortium (2017a, 2017b); Ormeling and Kraak (2010)
Future possibilities and expectations	What the interviewee believes that can be done with Vector Tiles that has not been done yet	Open Geospatial Consortium (2017b)

TABLE 2.3: An overview of the topics for the expert-interviews

found in the GIMA archives. Before writing the results part of the thesis and the findings from the expert-interviews, the interviewees were contacted and given a chance to check the interview transcription. This was important to verify whether the interpretations of their sayings were correct and matched with the experiences they wanted to convey. This strengthened the internal validity of the research.

2.3 Research steps

In order to carry out the steps needed to realize the objectives of this research, the thesis was divided into five phases. Each phase can consist out of one or more parts. In total there are four parts that cover the research methodology. The steps are also depicted in a methodology flowchart in figure 2.1.

<u>Phase 1</u>
The first phase refers to a preliminary study of the research. This consists of identifying the problem statement, the research questions and the theory available on the research topic. One part of this phase is writing the Extended Research Proposal.

<u>Phase 2</u>
The second phase forms to the theoretical research of this thesis and refers to the literature studies conducted.

Part 1
The first part presents the changing role of cartography and cartographers by researching the influence of technological developments in computer- and information sciences on cartography. This part is needed in order to understand the gap between computer sciences and cartography. Only afterward, it is possible to understand the position of Vector Tiles within web-cartography.

Part 2
The second part is an investigation of the existing literature and current state regarding Vector Tile technology. In other words, the latest developments regarding Vector Tile technology are researched in more detail.

<u>Phase 3</u>
The third phase is where the practical research and results of this thesis begins. It consists of performing part 3 of this research.

Part 3
The third part refers to making an inventory of existing Vector Tiles tools and technologies found on the Web based on a proposed classification. This is needed because it is becoming ever more difficult to maintain an overview within the current pace of technological innovation in web mapping (Roth, Donohue, et al., 2014). In other words, an overview was created of which Vector Tile tools and technologies are currently available on the web and how they are implemented in a workflow. An overview of different Vector Tile solutions or workflows was given.

<u>Phase 4</u>
The fourth phase consists of linking Vector Tile technology and Cartographic theory by combining the findings from both the second phase and the third phase of the research together with the outcomes from the expert-interviews. The expert interviews are part of the practical research of this thesis.

Part 4

The fourth part of this thesis consisted of assessing the Vector Tile tools and solutions (= workflows) on their cartographic strengths and weaknesses. It was described how the set-up of a certain solution/workflow influenced the possibilities and challenges from a cartographic point of view. Linking cartographic theory from the literature review with the outcomes from the expert-interviews was central in this part. It was questioned to what extent Vector Tile technology contributes to the cartographic need for effective communication of spatial data and on the other hand what the position and role is of Vector Tile technology for cartography. Different cartographic topics, that are considered to be important and relevant for Vector Tiles, were discussed with experts in the field of Web Cartography and Vector Tiles. These results were presented and taken into account for the assessment of cartographic strengths and weaknesses two example Vector Tile workflows.

Phase 5

The last phase covers the discussion and conclusion of the research. An answer was given to the research questions and problem statement.

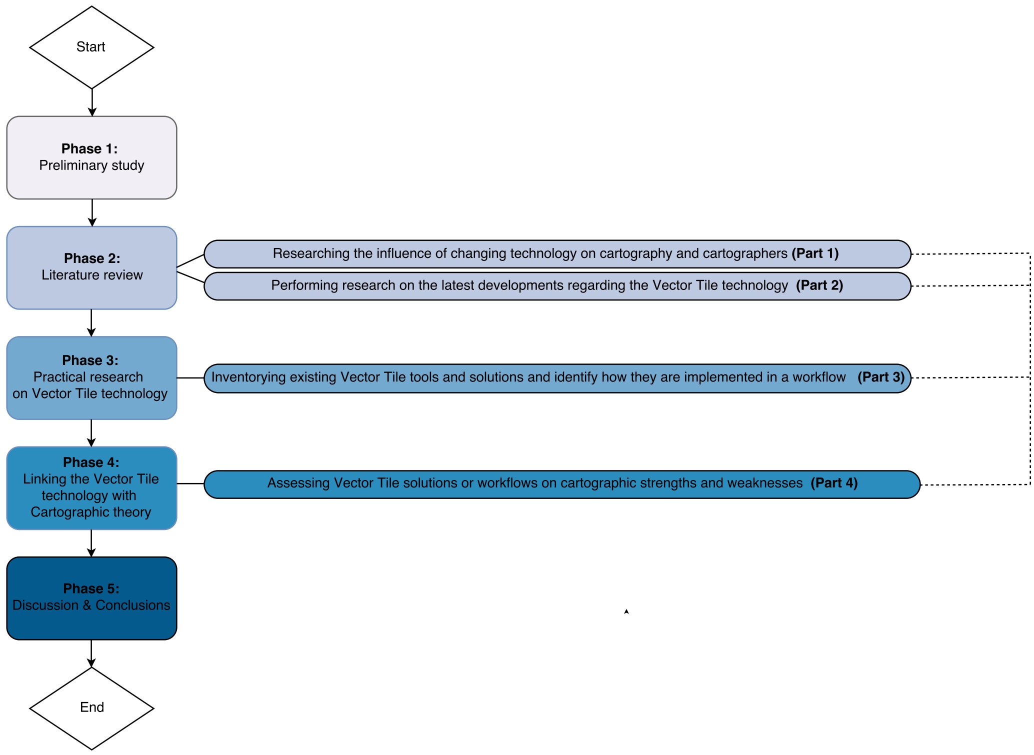

FIGURE 2.1: Flowchart of all necessary methodology steps

2.4 Time planning

This research was divided into four periods, each having its own time frame. These are (1) Research Identification; (2) Extended Research Proposal (ERP); (3) Mid-term and (4) Final Thesis. The total time to conduct this research was six months, from September 2017 until March 2018. A timeline planning was created in which the most important deliverables and phases of the project are presented (Figure 2.2).
Next to the overall timeline, a detailed table (Figure 2.3) was created as a guideline for this thesis time schedule. The table gives an indication of appointments or deadlines that were met.
The midterm results were presented in week 50 (December 14th, 2017) in Utrecht. The thesis defense is in week 13 in Wageningen (March 29th, 2018).

- Phase 1 was planned to be finalized in week 45 (November 16th, 2017). The time schedule was kept.
- Phase 2 was aimed to be ready in week 50 (December 7th, 2017). The time schedule was kept.
- Phase 3 was planned for January 2018 and was estimated to be ready in week 3 (January 19th, 2018). The time schedule was kept.
- Phase 4 was first estimated to be finished in February 2018 in week 7 (February 16th, 2018), but was finally finished in week 8 (February 25th, 2018).
- Phase 5 was first estimated to be completed in week 8 on February 23th, 2018), but was finally finished in week 9 (February 28th, 2018).

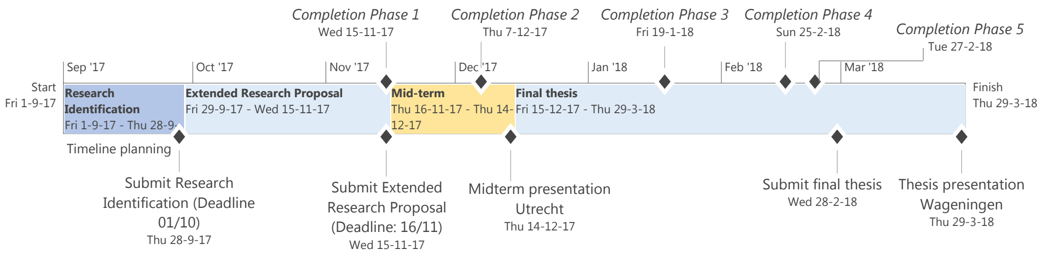

FIGURE 2.2: Timeline of the thesis planning

2.5 Risks and contingencies

Several potential risks were identified beforehand in Table 2.4. The risks were constantly monitored, reassessed and controlled. One of the most important identified risks, was the one of time complexity. This means that time allocated could be a mismatch with the time needed for a specific phase. A phase could end up to be more complex, resulting in time delay. For this reason, time evaluation was scheduled at the end of each phase, to re-adjust the estimated time where needed.

Another risk identified was the case of finding limited relevant literature on Vector Tile technology, resulting in poor quality of the theoretical framework. A measure was to investigate the available literature in more detail and to ask help from the supervisor where needed.

Furthermore, a risk was that the practical research on Vector Tile technology turned out to be more complex. Due to a lack of technical knowledge or external causes. This could result in limited results or a shortcoming in time. A measure was to start with a basic search and perform exercises and examples or to use forums for help. The risk was present during the research, however, with the help of expertise it was reduced.

Another identified risk was when linking Vector Tile technology with Cartographic theory turned out to be not possible. This could result in a failure of filling the gap in the literature or bridging the two disciplines. However, this could also lead to an investigation of why this was not possible. A measure to avoid the identified risk was performing the literature review on time and asking help from the supervisor where needed.

External risks could also lead to a delay of the thesis project or influence the way the project is done. An example of an external risk is the illness of the author or technical obstacles (e.g. computer crash).

2.6 Resources required

There were no specific data or data-sets required for this research. Open source tools and software were used where possible. Using open source tools or software has three main advantages. Firstly, problems regarding accessibility are avoided. Secondly, the Web provides a lot of documentation, tutorials and information when it comes to open source software. Third, when using open source tools, this research can be reproducible. Other researchers can use the same software packages and follow the same methodologies as used in this research.

All the required resources are described below:

Task Name	Duration	Start	Finish
Research Identification	20 days	Fri 1-9-17	Thu 28-9-17
Research thesis topic	12 days	Fri 1-9-17	Mon 18-9-17
First meeting with supervisor	1 day	Fri 1-9-17	Fri 1-9-17
Exploratory literature study	4 days	Wed 20-9-17	Mon 25-9-17
Write down essentials of research (100 words)	3 days	Tue 26-9-17	Thu 28-9-17
Submit Research Identification (Deadline 01/10)	0 days	Thu 28-9-17	Thu 28-9-17
Extended Research Proposal	34 days	Fri 29-9-17	Wed 15-11-17
Formulate research context	2 days	Mon 2-10-17	Tue 3-10-17
Formulate problem statement	7 days	Mon 23-10-17	Tue 31-10-17
Formulate research relevance	3 days	Mon 23-10-17	Wed 25-10-17
Formulate research objectives	5 days	Mon 23-10-17	Fri 27-10-17
Formulate research questions	5 days	Mon 23-10-17	Fri 27-10-17
Formulate scope and limitations	8 days	Mon 23-10-17	Wed 1-11-17
Identify tasks, create time schedule	4 days	Mon 30-10-17	Thu 2-11-17
Describe methodology (sequential list of steps)	4 days	Thu 26-10-17	Tue 31-10-17
Round off extended research proposal	3 days	Wed 1-11-17	Fri 3-11-17
Submit draft of extended research proposal to supervisor	0 days	Fri 3-11-17	Fri 3-11-17
Implement feedback supervisor to ERP	7 days	Tue 7-11-17	Wed 15-11-17
Submit Extended Research Proposal (Deadline: 16/11)	0 days	Wed 15-11-17	Wed 15-11-17
Completion Phase 1: Preliminary study		*Wed 15-11-17*	*Wed 15-11-17*
Mid-term	21 days	Thu 16-11-17	Thu 14-12-17
Implement ERP feedback	16 days	Thu 16-11-17	Thu 7-12-17
Literature review (Phase 2)	16 days	Thu 16-11-17	Thu 7-12-17
Round off theoretical chapter	9 days	Mon 27-11-17	Thu 7-12-17
Submit finished works prior to mid-term	0 days	Thu 7-12-17	Thu 7-12-17
Midterm presentation Utrecht	0 days	Thu 14-12-17	Thu 14-12-17
Completion Phase 2: Literature review		*Thu 7-12-17*	*Thu 7-12-17*
Final thesis	75 days	Fri 15-12-17	Thu 29-3-18
Carry out phase 3: empirical analysis	22 days	Sat 16-12-17	Sun 14-1-18
Write down results phase 3	5 days	Mon 15-1-18	Fri 19-1-18
Completion Phase 3: Practical research on Vector Tile technology		*Fri 19-1-18*	*Fri 19-1-18*
Carry out phase 4: empirical analysis	15 days	Mon 22-1-18	Fri 9-2-18
Write down results phase 4	15 days	Fri 9-2-18	Sun 25-2-18
Completion Phase 4: Linking the Vector Tile technology with Cartographic theory		*Sun 25-2-18*	*Sun 25-2-18*
Write discussion	3 days	Sun 25-2-18	Tue 27-2-18
Write conclusion/recommendations future research	3 days	Sun 25-2-18	Tue 27-2-18
Completion Phase 5: Discussion & Conclusions		*Sun 27-2-18*	*Sun 27-2-18*
Round off all chapters, combine chapters	2 days	Tue 27-2-18	Wed 28-2-18
Finish thesis lay-out/final structure	2days	Tue 27-2-18	Wed 28-2-18
Proof-reading, implementing resulting feedback	2 days	Tue 27-2-18	Wed 28-2-18
Submit final thesis		*Wed 28-2-18*	*Wed 28-2-18*
Thesis presentation Wageningen		*Thu 29-3-18*	*Thu 29-3-18*

FIGURE 2.3: Thesis planning

- **Software:** Different software was used to write this thesis or to conduct the empirical research. For literature and reference purposes, Mendeley and LaTeX reference library was used. For editing and writing purposes, ShareLatex was used. The research furthermore used tools that implement Vector Tiles as a part of the inventory. Most tools were open-source and findable on Github,

Risk	Result	Measure
Time complexity	Mismatch between time allocated and phase complexity	Time evaluation at the end of each phase to re-adjust
Limited relevant literature	The framework is of poor scientific quality, and good references are lacking.	Investigating literature in more detail or adjusting it to what is available, ask help from supervisor where needed
Research complexity	Results of practical research on Vector Tiles are limited	Start with basic search and exercises, perform examples. Reuse code or use forums for help. Use practical expertise from experts.
Linking Vector Tile technology with Cartographic theory does not work out	This could result in a failure of bridging both disciplines or combining practice with theory	Performing literature review on time, ask help from supervisor where needed, or investigate why this was not possible
External risks	This could result in a delay of the thesis project or influence the research	Finding other solutions or alternatives or accepting the delay

TABLE 2.4: Risks and contingencies

however some tools were proprietary such as ESRI's ArcGIS solutions. For some cases for testing Vector Tile solutions, a Linux Ubuntu virtual machine was used with the software of Oracle VM VirtualBox. It should be noted that for the end of the thesis project limited resources and software were accessible due to a laptop crash. The facilities of the University of Utrecht were used, with no option to install software or tools due to security reasons.

- **Dataware:** There was no specific data or data-set needed for this thesis. However, a lot of information was needed, such as scientific papers, books, articles, websites, exercises etc. Access to documentation was vital for testing the different Vector Tile tools. As well as open sample data for use with the Vector Tile tools. Information from the interviewee's was also considered as new data.

- **Hardware:** A proper functioning workstation with access to the internet was a requirement. For this thesis a laptop with a 13.3-inch screen, 1.7 GHz Intel Core i5, enough free space and a suitable graphic card sufficed. For the end of the thesis hardware at the University of Utrecht was used (Windows desktop computer with limited software).

- **Finances:** There was no need for financial resources.

- **People:** Fellow students, a supervisor, a professor, thesis coordinators were essential for a good realization of this thesis. Contact with experts on the field of Vector Tiles were also essential for the expert-interviews and empirical research.

Chapter 3

From Cartography to Web Cartography

This chapter takes a look at the relationships between Cartography, GIS and Web Cartography by means of a literature review. The focus is on developments and challenges in the field of Cartography and the influence of changing technology on cartography and map users. This chapter, therefore, contributes to the understanding of the position of Web Cartography within the cartographic discipline.

3.1 Cartography

Cartography is classically defined by the International Cartographic Association (ICA) as *"the discipline dealing with the art, science, and technology of making and using maps"* (Basaraner, 2016, p. 82). This definition generally reflects more the era before the use of computers for creating digital maps and geographic databases (Basaraner, 2016). In fact, defining Cartography is complex because the definition of the term cartography has gone through considerable changes during the period that the term has been in use. Before cartography started to be defined as *"the visualization of spatial information"*, it encompassed the production of maps essentially. Only after, the definition changed to "the production and use of maps", and other functions of cartography, such as analysis, storage, or communication became more important. This led to a new objective of cartography, which can be described as "passing on spatial information to support decision making" (Fernández & Buchroithner, 2013; Ormeling, 2009). Wood (2003, p. 271) also shares the opinion that cartography is vital for spatial problem solving and decision making as: *"map creation should be seen more correctly as part of the spatial problem-solving process which also involves the manipulation and use of maps. With appropriate content and design, maps can improve the comprehension and support the analysis of environmental problems and, when appropriate, help communicate this information to others"* (Basaraner, 2016). In the citation of Wood (2003) the importance of communicating spatial information to others is mentioned. Hardy and Field (2011) and MacEachren (1995) also suggest that communication is one of the main goals of cartography. As MacEachren (1995, p. 5) states: *"Communication came to be viewed as the primary function of cartography and the map was considered the vehicle for that communication"* (Ory, 2016). William J. Thomas Mitchell also shares this point of view by stating that: *"a map is a value-laden image which is used for communication"* (Fernández & Buchroithner, 2013, p. 83). With communication being an important function of cartography the question remains how effective cartography or maps are. In line with this, Hardy and Field (2011, p. 324) highlight that cartography results in an effective map: *"cartography is a language that allows the mapmaker to create graphical components on a map, so that they will be understood by the map reader, resulting in an effective map"*. According to Kraak and Brown (2001), there is a need for the cartographer to concentrate on the effectiveness of maps and mapping technologies.

While on one hand cartography tends to be seen only as a visualization phase of spatial data handling, or only as the art or craft of making maps (Basaraner, 2016; Hardy & Field, 2011), cartography can, on the other hand, be regarded as a science that covers the entire phases of spatial data handling (Basaraner, 2016; Ramirez, 1993). This includes the communication of spatial information as well as all the stages from data acquisition to presentation and use (Basaraner, 2016). If taking into account all the objectives of cartography, the main aims of cartography can be considered to be representation/visualization, exploration/analysis, and communication (Fernández & Buchroithner, 2013; Kraak & Brown, 2001; MacEachren, 1995; Ormeling, 2009; Ormeling & Kraak, 2010). D. Taylor (1994) points out that only seeing cartography as a visualization technique is very limited considering the abilities of modern cartographers. According to him, maps are not only for display but should also be for knowledge, action and development (Basaraner, 2016). In the next section, the influence of modern technology on Cartography will be further discussed. This will also contribute to the understanding of the changing role of the cartographer due to changes in technology.

3.2 The influence of changing technology on Cartography

The Internet or the Web has introduced a rapid, discontinuous change in cartography and can, therefore, be considered as a paradigm shift (Fernández & Buchroithner, 2013). Maps are being used by and created by more people than ever before (Kraak, 2011) and the demand for relevant information displays has never been greater (Bostock & Davies, 2013). The arrival of the computer and digital geographic files led to a revolution in map production because map images could be flexibly adapted for various purposes (Ormeling, 2009) and be distributed to more individuals (Fernández & Buchroithner, 2013; Kraak & Brown, 2001; Peterson, 2003).
The changes in which maps have been produced and used seem to happen on two sides of cartography: in the scientific domain and in the public domain (Kraak & Brown, 2001; Peterson, 2003). The role of cartographers and map users is no longer the same. Communication about maps and cartography is no longer done by professionals only, but also by a group of non-experts or neocartographers. Non-experts are now increasingly engaged in map making and more maps are made by people without any cartographic training or knowledge (Griffin, Robinson, & Roth, 2017). The term neocartography is, according to the ICA, being used to describe map makers that are frequently using open data and open source mapping tools. These map makers may not have come from traditional mapping backgrounds and tend to make maps for themselves. The availability of data and tools allows neocartographers in the new Web 2.0 to make their own maps (Fernández & Buchroithner, 2013; Haklay, M. Singleton & Parker, 2008). This is causing the boundaries between map producers and maps consumers to be less distinct or clear (Fernández & Buchroithner, 2013). According to Griffin et al. (2017), the distinction between map users and map designers is changing or perhaps even disappearing. Besides, one could notice that scientists use maps with cartographic knowledge to communicate a message or to solve their problems, whereas neo-cartographers, non-experts or coders want to grab the tools that are available to create their own maps (Kraak, 2011). Maps users on their turn have gone from a passive role , as map readers, to a more active role, as contributors in the map making process (Ory, 2016). The new mapmakers have different demands and objectives, and force a change from a traditional supply-driven

map production to demand driven map production. Expertise is still required, however, the role of maps has changed and expanded due to the influence of technology (Kraak, 2009).
The influence of technology on cartography also meant that the tools used in cartography changed, from analog tools towards computer tools (Bostock & Davies, 2013). It is now common practice that maps are distributed not as static images, but as source code that renders in a web browser (Bostock & Davies, 2013). The expansion of the computer in cartography is reflected by the trend that computers and web standards have become more capable (Bostock & Davies, 2013). The question a cartographer could ask is whether a cartographer can actually make a good map from code? Should cartographer be coders? Or should coders become cartographers? Cartography and coding increasingly are intertwined (Roth, Hart, et al., 2014). A distinction could be made between modern mapmakers that are coders, and traditional map makers that are expert-cartographers. Van den Berg (2017) states the following: *"The creation of especially web maps is shifting towards people who know how to code rather than people that know how to correctly design a map"*. There seems to be an increasing number of people involved in the creating of web maps that have significant programming skills instead of people who have sufficient cartographic knowledge (Roth, Hart, et al., 2014; Van den Berg, 2017). Therefore, according to Köbben (2014) there seems be a new role for the cartographer to provide tools that implement cartographic intent. According to him there is a need for *"code that thinks like a cartographer*". This means that a cartographer could have a role in the toolmaking process or in the knowledge process (Köbben, 2014). In line with this, Lambert and Zanin (2016) highlight that cartography, driven by scientific and technical progress, has for a long time been marked by the aim of locating various places with precision and fine detail. However, nowadays, the real challenge of cartography is giving intelligence to geographic data. Without questioning the necessity and usefulness of information storage which is becoming more and more accurate, the digitalization of the world does not make the reality more intelligible (Lambert & Zanin, 2016).
The use of maps on the Web can be regarded as a major advancement in cartography and opens many new opportunities (Neumann, 2008). The wide variety and rapidly expanding Web Mapping technologies available on the Web occur almost on a daily basis and have often new releases and updates (Roth, Donohue, et al., 2014). On one hand advances in mapping technologies and techniques offer major advantage for cartographers because there are now more possibilities and opportunities than ever before, on the other hand, the increasing possibilities and different technologies are difficult to keep up with (Cartwright et al., 2010; Muehlenhaus, 2014; Roth, Donohue, et al., 2014). Because new available techniques for creating and distributing digital maps can be overwhelming for map authors and map users, tools for digital maps meeting the demands of cartographic principles and high-quality are harder to find (Cartwright et al., 2010). Neumann (2008, p. 570) mentions that the Web and new mapping opportunities allows almost anyone to produce maps which *"puts geodata in the hands of untrained people who potentially violate cartographic and geographic principles and introduce flaws during the preparation, analysis, and presentation of geographic and cartographic data"*.
The influence of technology on Cartography has always been there and cartography has continually evolved to use new techniques and tools. Manual techniques that were once the basis of cartographic map design and production have now been replaced by digital workflows (Bostock & Davies, 2013; Hardy & Field, 2011). This means that the use of electronics and computers in mapping eventually led to the emergence of digital cartography and recent techniques such as GIS (Basaraner,

2016).

3.3 Relationship between Cartography and GIS

Cartography was traditionally understood as the whole mapmaking process, from map production to map analysis and interpretation. However, with the technique of GIS, cartography is now used more to cover the visualization and reproduction aspects, taking into account processed data and excluding the earlier stages of creating databases and carrying out data analysis (Basaraner, 2016; Hardy & Field, 2011). The relationship between GIS and cartography has not always been that good. With the new technology of GIS, the geospatial professionals had little cartographic knowledge or training and were too much focused on GIS analysis. This led to poor map design from GIS environments and often low-quality map work. However, in recent years cartography and map design and production have become relevant and important once more. GIS analysts have responded to the demand for high-quality maps (Hardy & Field, 2011) and the tools are better adapted for fulfilling this goal. However, the lack of cartographic knowledge still remains a major challenge for proper map production and reproduction (Bocher & Ertz, 2017; Muehlenhaus, 2014). According to Fernández and Buchroithner (2013, p. 53) the relationship between cartography and GIS results in geo-visualization which is *"a further development in cartography that takes advantage of the ability of modern computers to render changes to a map in real time, allowing users to adjust the mapped data at the same time"*. This development in cartography also led to the emergence of Web GIS/Web Mapping and Web Cartography.

3.4 Web GIS and Web Cartography

This section brings attention to the developments in WebGIS and Web Cartography.

3.4.1 Defining WebGIS and Web Cartography

The boundary between web maps and WebGIS is blurry. Therefore, WebGIS is here used as a synonym for Web Mapping because web maps are often the presentation medium for WebGIS (Neumann, 2008). WebGIS *"denotes a type of GIS, whose client is implemented in a Web browser"* (Yang et al., 2007, p. 319). This means that Web Mapping is based on the invention of the World Wide Wide (WWW) and is according to Neumann (2008, p. 567) *"the process of designing, implementing, generating, and delivering maps on the World Wide Web". While Web Mapping primarily deals with technological issues, Web cartography additionally studies theoretical aspects such as "the use of web maps, the evaluation, and optimization of techniques and workflows, the usability of web maps, social aspects, and more"*. Therefore, web cartography is no different than traditional cartography, because it is about the design, production, and use of maps. The only difference is that is restricted to the WWW as a medium (Kraak & Brown, 2001). Web cartography is defined as *"being concerned with theoretical and practical issues that the web offers for the design, production, distribution, and use of maps and geographic information"* (Basaraner, 2016; Kraak & Brown, 2001; Peterson, 2008). Even though interactive and web technologies have changed many things, they have not changed the fact that most maps are still designed to communicate a message to its audience. The medium does not matter (Muehlenhaus, 2014).

3.4.2 Web Mapping Technologies

Web Mapping technologies are described as *"the compilation of APIs, frameworks, libraries, services, etc., that all together enable the creation and dissemination of web maps"* (Kraak & Brown, 2001; Peterson, 2003; Roth, Donohue, et al., 2014). These Web Mapping technologies are continually changing and the Web is redefining how maps are made and used. The first generation of Web maps was primarily static and had no additional values compared to paper maps (Kraak & Brown, 2001; Ormeling & Kraak, 2010). Static maps were raster based, at a fixed resolution and would not allow interactivity. The ability of the Web at this time was also to sent a map image to the client or to the user's Web browser by requesting a page in which an image file was embedded and running on the Web server using data from a GIS database (Haklay, M. Singleton & Parker, 2008). The newer generation of Web Maps is more characterized by two keywords: Interactivity and Dynamics (Kraak, 2009). Martinelli and Roth (2015) describe the evolution of web maps as going from untiled static maps through raster tiles to Vector Tiles. The assumption is that the new technology of Vector Tiles also has benefits regarding interactivity. Besides, the argument can be made that Vector Tiles serve maps more rapidly and more effectively because they are rendered only when requested by a client (Mapbox, 2017b).
The new generation of Web maps is using advanced Web 2.0 technologies such as dynamic HTML (DHTML), asynchronous JavaScript and XML (AJAX), XML, JavaScript, and Scalable Vector Graphics (SVG) (Kresse & Danko, 2012). As Goodchild (2007, p. 27) noted, *"the early Web was primarily one-directional, allowing a large number of users to view the contents of a comparatively small number of sites, the new Web 2.0 is a bi-directional collaboration in which users are able to interact with and provide information to central sites, and to see that information collated and made available to others"*. Nowadays the available web application programming interfaces (APIs) are for instance relatively easy to use and have made application development more accessible, and therefore enabling a broarder community of individuals who could create, share and mash-up (geographic) information (Haklay, M. Singleton & Parker, 2008). Roth, Donohue, et al. (2014) sum up the new functions and technologies in Web Mapping that differentiates them from the first generation of web maps. Web maps are now commonly interactive, adaptive, mobile, dynamic, multiscale, and/or updated in real time (Roth, Donohue, et al., 2014).
Several studies have tried to create an overview of the available Web Mapping technologies on the Web (Ballatore et al., 2011; Roth, Donohue, et al., 2014). According to (Roth, Donohue, et al., 2014), contemporary Web Mapping technologies can mainly be organized into three broad categories: *"(1) server-side technologies used to index and query geographic information from a centralized source or, increasingly, distributed sources (e.g., the cloud), (2) client-side technologies used to render and manipulate web maps of the geographic information in the user's browser, and (3) web services or similar intermediary scripts used to relay information requests between the client and server"*. Furthermore, in a comparison of open source technologies for Web Mapping, Ballatore et al. (2011) have conducted surveys to obtain responses from the relevant online communities about a given set of characteristics. The aim was to reduce the knowledge gap regarding experiences in practice of software development in Web Mapping. The results of the surveys highlight the strengths and weaknesses of the open source technologies for Web Mapping. The characteristics used in the questionnaire are the following:

1. *Learning curve* – whether it is easy or hard to learn how to use a software.

2. *Stability* – refers to the stability and reliability of the software for use.
3. *Performance* – the performance of the software according to the user's experience.
4. *Scalability* – the ability of the software to continue to function well when there is a growing amount of work to handle.
5. *Interoperability* – the integration of the software with other technologies.
6. *Extendibility* – whether it is easy or hard to extend the software functionalities with external plugins/add-ons.
7. *Standards* – the software support for widely-adopted standards.
8. *Documentation* – the documentation of the software (e.g., completeness, readability, quality, useful examples).
9. *Community support* – e.g., the technical support offered on the project forums/mailing lists.
10. *Frequency of updates* – the new releases containing new features, improvements and bug fixes.

From the overall scores of the questionnaire, it was possible to deduce a higher satisfaction for meeting software stability and open standards than for the learning curve and project documentation (Ballatore et al., 2011).
With mapping technologies facing increasing interoperability and flexibility, new opportunities are created for cartographers, such as more interactivity or interconnectivity (Kraak, 2009; L. Li et al., 2017; S. Li et al., 2011; Roth, Donohue, et al., 2014). However, as mapping technologies evolve, there are new challenges due to the fact that it is becoming ever more challenging to maintain an overview within the current pace of technological innovation in Web Mapping (Roth, Donohue, et al., 2014).

3.4.3 Web Mapping Architectures

With the number of mapping technologies to implement Web Mapping projects becoming overwhelming, both server- and client-side technologies are to be used to implement Web Mapping projects (Neumann, 2008). This section will not list all of the existing technologies, however, will focus on the core components of a WebGIS architecture. Next to that, the most common geo-webservices and standards are presented and the role Web Cartography in Spatial Data Infrastructures is analyzed.

3.4.4 Architecture of WebGIS

WebGIS is developed utilizing the functionality of GIS and web-based computing platforms. Both platforms play significant roles in the WebGIS architecture (S. Li et al., 2011). The architecture of WebGIS is typically broken down into server-side and client-side with a middleware handling the spatial request and connecting the client-side with the server-side (Alesheikh, Helali, & Behroz, 2002). Figure 3.1 shows the minimum system architecture of WebGIS. On one side there is a Client, which is an application that can communicate with the other side, the Server, through a standard web protocol, for example HTTP. The client application can either be in the form of a web browser or standalone utility. The server-side consists of a Web Server and a GIS component (Held et al., 2004).

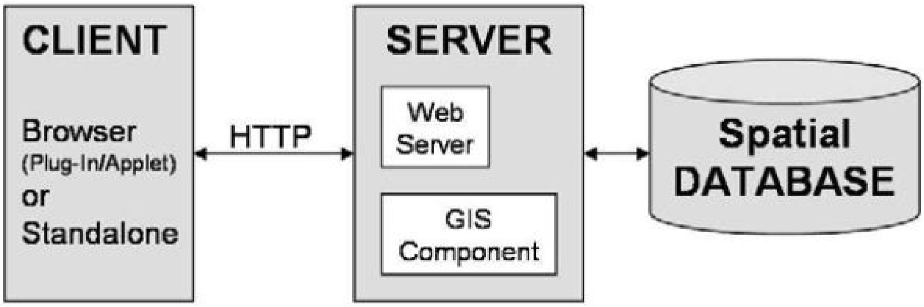

FIGURE 3.1: A typical client-server WebGIS architecture (Held et al., 2004)

Figure 3.2 depicts the main components of a WebGIS in more detail. Clients, a Web Server, a Map Server and a Data Server are the main components. The web server is responsible for handling the HTTP/Spatial request done by the client, whereas the map server provides the map itself (Nasirzadeh Dizaji & Nurhan Çelìk, 2015). Currently, the most widely used Map Servers are UMN MapServer, GeoServer and QGIS Mapserver (Ballatore et al., 2011; Lienert, Jenny, Schnabel, & Hurni, 2012).

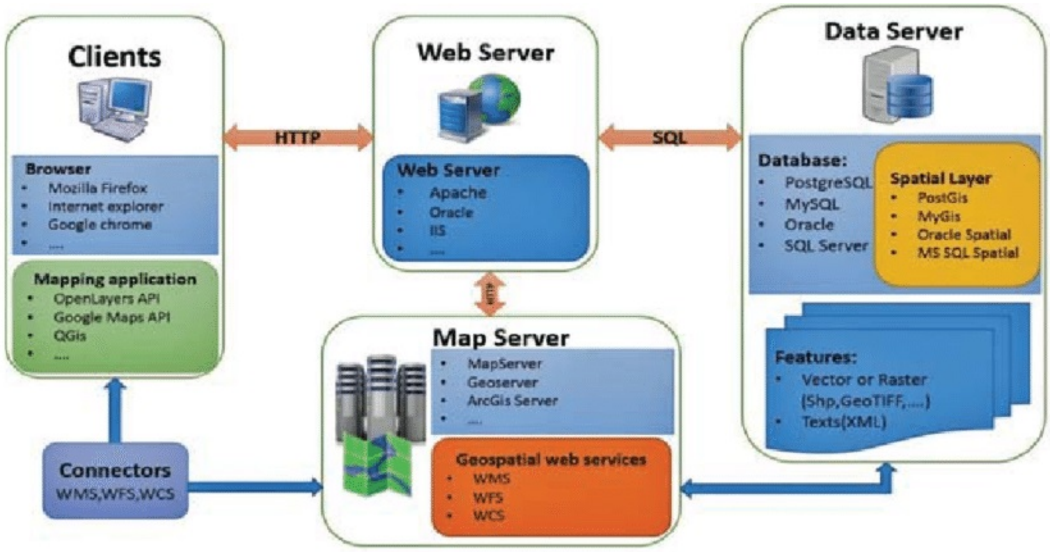

FIGURE 3.2: Structure of Web Mapping (Web-GIS) (Nasirzadeh Dizaji & Nurhan Çelìk, 2015)

3.4.5 Mapping in a Service Oriented Architecture environment

Beginning with static web map publishing, Web Mapping has evolved through interactive Web Mapping towards distributed Web Mapping services. Distributed web maps are created from distributed data sources (Neumann, 2008). In line with these developments, there seems to be a current transition to client-side mapping and tiled web maps. The development of tiled web maps is discussed in Chapter 4. While the client-server technology has long dominated the development of Web Mapping, and will still continue to have a significant role, new types of Web Mapping and GIS developments are emerging due to recent web technologies. In addition to more interactive Web Mapping services, more service-oriented web services are upcoming (S. Li et al., 2011). Figure 3.3 depicts the general principle of dissemination of maps in a Service Oriented Architecture. This set–up is being commonly used in in many Web Mapping efforts today. According to Köbben (2010), what the system has to achieve is 'automatic' and 'direct' production of maps. By 'direct' it is meant that the maps are generated on-the-fly from the data. 'Automatic' means that the maps will be generated from the data by the system working by itself with little or no direct human control. However, in most current systems this automation does not include the cartographic decisions of what map type to use for different data–types and data–instances. Therefore, there is a need for a human cartographer to set up the appropriate service configuration and to make a link between data- and visualization-type (Köbben, 2010).

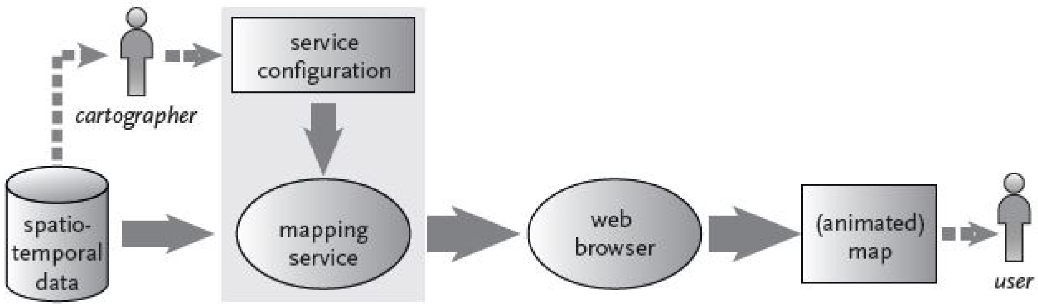

FIGURE 3.3: The general principle of dissemination of maps in a Service Oriented Architecture (Köbben, 2010)

According to Neumann (2008), the advances in client-side Web Mapping ask for putting the interoperability of existing and upcoming Web Mapping solutions in a high priority. Therefore, there is a need for developing web services based on standards. L. Li et al. (2017) note that "*the development of open standards and services has enabled the interoperability and access of geospatial data to users and applications in the form of web services, mashups and spatial data infrastructures*". The next paragraph will present the most common geo-webservices and standards, before discussing Spatial Data Infrastructures in section 3.4.7.

3.4.6 Geo-webservices and standards

Because mapping technologies can be overwhelming and geographic information can be an expensive resource, standardization is needed to promote the availability and reuse of geographic information (García et al., 2012). Interoperable web

standards have been developed by the OGC in support of Web Mapping, including WMS, WFS, web coverage service (WCS), styled layer descriptor (SLD) and geography markup language (GML) (Ballatore et al., 2011). The most common geowebservices and standards are discussed below.
The Open Geospatial Consortium (OGC) has come with a set of well–defined defacto Open Standards for geo–webservices, based on the SOA and Web Services. There are Open Web Services (OWS) specifications for most parts of the spatial data storage, analysis, and delivery process, as presented by Köbben (2010):

- Geographic data encoding: Geographic Markup Language (GML) and Keyhole Markup Language (KML);
- Spatial data delivery: the Web Coverage Service (WCS) and Web Feature Service (WFS), for querying and retrieving raster and vector data respectively. With WFS, a user makes a request for certain information, the service sends data back in GML;
- Processing of spatial data: the Web Processing Service (WPS);
- Data visualization: Web Map Service (WMS), which is by far the most mature and widest adopted OWS specification. WMS is the protocol for serving georeferenced map images over the Internet. A user makes a request for a certain location on a map, the service sends an image back. Related to WMS are the Styled Layer Descriptor (SLD) specification, for map styling, and the Web Map Context Documents (WMCD) specification, for map setup and layout;
- Describing and finding spatial data: Catalog Service Web (CSW) with a set of metadata specifications. With CSW, a user does a request for certain metadata. The service specifies the bindings and frameworks and sends the metadata back.

With OGC standards continuously being extended, many of them will or have become de-jure ISO standards (Kresse & Danko, 2012). The OGC has a Class A Liaison relationship with TC 211 (Reed, 2011). For instance, some examples of OGC standards that are also ISO standards are:

- Web Map Service (WMS)
- Geography Markup Language (GML)
- Web Feature Service (WFS)

The possibilities offered by the well-known ISO and OGC standard are useful for Web Mapping. For example, the most widely adopted Web Map Service has allowed publishing geographic information stored on multiple servers on different locations in a specific format used by multiple software. This makes it possible to rapidly produce a map through directly available information (Haklay, M. Singleton & Parker, 2008). The specified standards have made geographic information and mapping applications more interoperable. This interoperability is defined by the International Standards Organisation (ISO) as *"The capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units"* (Köbben, 2010). Having interoperability, as a priority for existing and upcoming Web Mapping solutions, has also lead to growing awareness of the importance of Spatial Data Infrastructures.

3.4.7 Web Cartography as a part of Spatial Data Infrastructures

Web cartography forms a part of the bigger concept of Spatial Data Infrastructures (SDIs). In recent years, there has been a growing awareness of the need for SDIs (Kraak & Brown, 2001). The concept of SDIs was first introduced by the U.S. National Research Council in 1993, who described SDI's as: *"a framework of technologies, policies, and institutional arrangements that together facilitate the creation, exchange, and use of geospatial data and related information resources across an information-sharing community."* (Dutta & Jayasinghe, 2015, p. 2). Within recent scientific literature, it is commonly referred to as: *"a distributed system that allows for the acquiring, processing, distributing, using, maintaining, and preserving of spatial data"* (Ostlaender, Smith, De Longueville, & Smits, 2010, p. 302). Users, data, network architecture and technical standards are considered key actors within SDIs. Service-oriented cartography can be seen as a most modern framework in the realm of SDIs, because modern cartographers are offered a significant number of distributed map productions based on services (Basaraner, 2016). Cartographic knowledge within SDI can add value to spatial data by offering user-oriented design solutions and offering solutions for cartographic interoperability (Hopfstock & Grünreich, 2009). However, the role of cartographer is declining in web maps, resulting in maps with less cartographic knowledge. The main issue observed with maps on the Web or coming from SDI, is that they often present a serious lack of cartographic knowledge (Bocher & Ertz, 2017; Muehlenhaus, 2014). This is problematic because it is important that map makers don't repeat the same mistakes while (re)using spatial data or maps. Hopfstock and Grünreich (2009) underline that poor map design results are the consequence of a *"too technology- and/or data-driven approach"* (Bocher & Ertz, 2017). This observation highlights the need to take into account the knowledge of cartography on how to design proper maps that are not only usable but also reproducible or interoperable. The next paragraph will study the topic of web map design.

3.4.8 Web Map Design

Maps on the Web require a different design and production approach to paper maps and screen maps. According to Jenny, Jenny, and Räber (2008), *"the design of a web map must be coarser and simpler than the design of a paper map so that it conveys the desired information under the less than ideal conditions of low screen resolution, increased viewing distance and shorter reading time"*. Web cartographers also have to take into account display capabilities of different screen sizes (Muehlenhaus, 2014). What constitutes the best design 'format' for Web maps is still a matter for debate and research (Peterson, 2003), which also explains why research and practice on responsive cartographic design remain in its infancy (Roth, Donohue, et al., 2014). The most important difference between designing maps for print versus the Web is according to Muehlenhaus (2014) that *"we no longer design for map readers but map users. People interact and manipulate Web maps. We no longer need to attempt to design a one-size-fits-all, optimal form of data communication. These days, it is imperative that we design our maps to be interactive and responsive to a map user's needs to facilitate communication that is more effective"*.

With web map design being important in effectively visualizing a message that map makers want to communicate, web map designers must take into account the purpose of the map and the needs and characteristics of its users, as cartographers always did (Kraak & Brown, 2001). Besides, visual hierarchy remains important when communicating the map message or information (Muehlenhaus, 2014). Muehlenhaus (2014) identifies three different new visual hierarchies for web maps (Figure 3.4). The first visual hierarchy is for reference and general-purpose maps, the second is for thematic maps on the Web, and the third is for animated maps on the Web.

VISUAL HIERARCHY LEVELS FOR WEB MAP DESIGN

	General Interest Web Maps	Thematic Web Maps	Animated Web Maps
Level 1	Title/Splash Screen Map Symbology Key Reference Data Info Windows (opened)	Title/Splash Screen Thematic Visualization Legend	Title/Splash Screen Animation Symbology Map Symbology Temporal Legend/Interface
Level 2	Base Map Base Map Labels Navigation/Directions Tools	Base Map (generalized) Info Windows (opened) Chart Graphics	Base Map Legend Info Windows (opened) Locator Map
Level 3	Map Interactivity Pan/Zoom/Rotation Tools Print/Share Map Features	Base Map Labels Map Interactivity Pan/Zoom/Rotation Tools Menus with Additional Tools	Base Map Labels Map Interactivity Pan/Zoom/Rotation Tools Menus with Additional Tools
Level 4	Locator Maps Chart Graphics Multimedia Supplements	Locator Maps Multimedia Supplements	Multimedia Supplements Chart Graphics
Level 5	Supplemental Information Attribution and Copyright Neatlines/Grids/Graticules Tool Tips	Supplemental Information Attribution and Copyright Neatlines/Grids/Graticules Tool Tips	Supplemental Information Attribution and Copyright Neatlines/Grids/Graticules Tool Tips

FIGURE 3.4: Three visual hierarchies for Web maps (Muehlenhaus, 2014)

According to Muehlenhaus (2014), clear and effective map design depends on the knowledgeable use of visual variables for the representation of map data. *"Visual variables are graphic manipulations that symbolize data in a meaningful manner"* (Muehlenhaus, 2014, p. 125). Jacques Bertin was one of the first cartographers to specifically address visual variable for use in mapmaking. The six core variables Bertin identified were shape, hue, value, orientation, texture, and size (Lambert & Zanin, 2016; Muehlenhaus, 2014; Ormeling & Kraak, 2010). Knowing when to use particular visual variables is important for designing effective maps (Muehlenhaus, 2014). Moreover, map layout, the form, and function of maps have to be considered while designing web maps (Muehlenhaus, 2014; Peterson, 2003). The map should be clear, simple, legible and balanced (Jenny et al., 2008; Peterson, 2003). Well-designed interactive maps are characterized by their relative 'emptiness', meaning that every part or element of the image visualized on the screen should be legible (Ormeling & Kraak, 2010). This means that a 'less-is-more approach' to map design is recommendable. The interactive map can be extended with additional 'hidden' information which can be accessed by the map user through interaction. This way, the user is offered the opportunity to make the less complex (Dillemuth, 2005; Peterson, 2003), with only information relevant to the user appears on screen (Brown, 1993; Dillemuth, 2005; Jenny et al., 2008). With Vector Tiles the latter is also the case because the web browser can spend its resources loading only those tiles that need to be shown to the user. For this reason, Vector Tiles can be very effective for guiding map user interpretation (Muehlenhaus, 2014).

With a shift in who creates maps and the declining role of cartographers in web maps, maps on the internet sometimes fall short of effectively conveying information. According to Lienert et al. (2012) there are three main reasons for this shortcoming: *"(a) the design of these maps sometimes does not take into account the specific*

limitations of digital displays; (b) the maps are often restricted in using standard functionality provided by the authoring software; and (c) they do not take full advantage of interactive features available in modern Web-browsers". These limitations should be taken into account in the design of web maps.
Furthermore, Roth (2012) note a fundamental duality within Cartography and map design between representation and interaction. According to Roth (2017) there is a *"need to consider interaction as a fundamental complement to representation in cartography and visualization because many maps today are highly interactive and delivered online or through mobile devices"*. The term is defined by Roth (2013) as "the dialogue between a human (a) and a map (c) mediated through a computing device (b) to emphasize digital interactions" (Figure: 3.5).

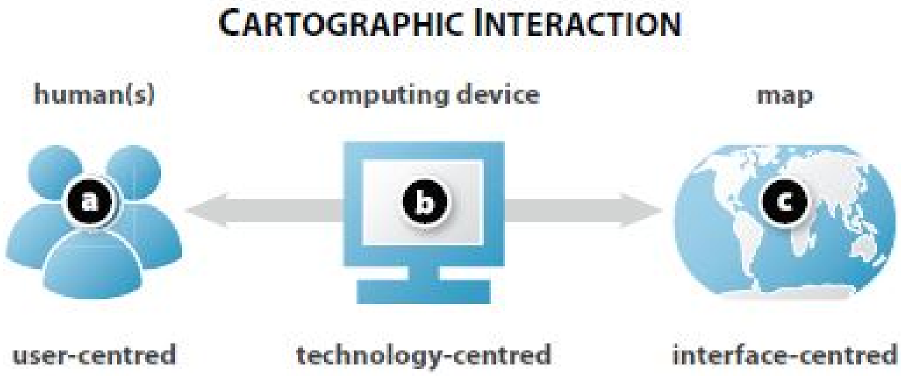

FIGURE 3.5: Cartographic interaction (Roth, 2013)

Roth (2013) highlights that *"a science of cartographic interaction must begin with a consideration of the overall experience of cartographic interaction and not immediately focus upon the implementation and use of cartographic interfaces supporting these interactions"*. Therefore, it can be vital to consider, in the context to Web Cartography, cartographic interaction as a way to test the cartographic effectiveness or user experience of cartographic maps, tools or technologies.
Besides, it is important to notice the changes in the way web maps are designed. The creation of especially web maps is shifting towards people who have the knowledge to code rather than people that know how to design a map correctly. Cartographers are often trying to improve their coding skills. Cartography and coding are increasingly intertwined (Roth, Donohue, et al., 2014). According to Roth, Ross, and MacEachren (2015), cartographic design rules are often incorrectly used by coders due to a lack of cartographic knowledge causing maps to increasingly fail. This can lead to frustrating and ineffective user experiences. This means that according to Roth, Donohue, et al. (2014) cartographers should have a special role of user experience (UX) designers to possibly resolve problems of cartographic incorrect or ineffective web maps. With cartographers having the experience and the required knowledge to create correct (web) maps, they should be involved in the user-centered design and evaluation of prototypes to streamline the development and promote a positive user experience with web maps.

3.4.9 Web Map Performance

In WebGIS, web map design is also linked to web map performance. Muehlenhaus (2014) note that "performance emerges as a critical challenge when large numbers of users, complex processing, and large volumes of datasets are involved". Also for Vector Tiles, performance plays an important role, because with rapid technological innovation in Web Mapping there is a need to utilize fast rendering tiles (Antoniou et al., 2009). Vector Tiles have several technological advantages such as faster rendering, less bandwidth needed due to a smaller size of tile packages and more tiles that can be generated in less time (Mapdata Services, 2017). With advantages such as more interactivity and more direct object manipulation, performance is a crucial factor for Vector Tiles and web maps in general. Therefore, several performance indicators can be considered. According to S. Li et al. (2011), each of the components and their connections in the WebGIS architecture have to be considered when testing the performance of a WebGIS. The authors propose the performance indicators: time, memory, reliability, and interoperability. Time complexity refers to the time duration between the request by an end user and the response at the client side. Space complexity refers to the memory needed to conduct processing requests and generating responses. Reliability refers to the availability, accessibility, and accuracy of a WebGIS service at all times for users at different places. The last performance indicator is the one of interoperability which means that a WebGIS can be reused or shared with other WebGISs (Figure: 3.6).
Relating these performance indicators to Vector Tiles, one could say that technology considerations for its performance are bandwidth and processing, because file- and download sizes determine the speed at which interaction occurs (Bostock & Davies, 2013; Hardy & Field, 2011; Roth, 2013) and response or loading times, because users want fast rendered tiles (Antoniou et al., 2009; Roth, 2013). Loading times of interactive maps or Vector Tiles are important since users might dislike waiting for relatively long loading times. Several authors have proved this while evaluating the impact of page load times on user satisfaction (Butkiewicz, Madhyastha, & Sekar, 2011; Ceaparu, Lazar, Bessiere, Robinson, & Shneiderman, 2004; Gardner, 2011; Ramsay, Barbesi, & Preece, 1998; Skadberg & Kimmel, 2004). Where slow pages result in increased user frustration (Ramsay et al., 1998), fast websites are perceived as more interesting and appealing to users (Ramsay et al., 1998; Skadberg & Kimmel, 2004). The same applies to web maps and Vector Tiles when considering loading times. Therefore, image or caching compression strategies are often used to reduce the size of images or tiles (Gardner, 2011; Kang, Kim, & Kim, 2001; S. Li et al., 2011; Loechel & Schmid, 2013; Peterson, 2003; Quinn & Gahegan, 2010; Yang et al., 2007). Tiles can allow rapid response to user requests by creating in advance small fragments of maps that are requested at various pre-defined scales and assembling them into a cache (Quinn & Gahegan, 2010). Caching time is therefore important to take into account when testing the performance of tiles. Next to caching time, the size of the tiles, disk space and the cartographic scale of the map are to be considered when testing the performance of Web Mapping (Quinn & Gahegan, 2010). The next chapter will cover the topic of Vector Tile technology which is emerging in the field of Web Cartography.

Different perspectives of WebGIS performance.

	Host and maintainer	Designer and developer	End user
Time	How many resources are needed to reduce time and complexity? Restricted by available budget, network and computing infrastructure.	Strategies, architectures, algorithms and implementations can reduce time complexity. Restricted by available technology, tools, and designer's and developer's knowledge and experience.	How much time is needed to wait for a response? 8-second rule applies. Restricted by the server and network performance and web client computer if client side computing is involved.
Memory	How much memory is needed for a well-performing WebGIS? Restricted by available budget and technology.	What database design, data structure and organization methods can make full use of available memory? Restricted by database and designer's and developer's knowledge and experience.	How much memory will be used by a WebGIS client? Depending on client side function and computing design.
Reliability	How to ensure reliability? Redundancy system and better system configuration.	How to deploy a stable and reliable system? Hardware and software redundancy, exception detect and process.	Fast and smooth access to a WebGIS is desirable.
Interoperability	Transparency? Scalable? Easy to maintain and upgrade?	How many days does it take to develop the system? Can any technologies be reused?	Compatibility with a client hardware and software? Familiar user interface and free and/or open-source clients.

FIGURE 3.6: Different perspectives on WebGIS performance (S. Li et al., 2011)

Chapter 4

Vector Tile technology

The previous chapter discussed Web Mapping and the different technological changes it has undergone in recent years. This chapter brings attention to the emerging technology of Vector Tiles. The topic of Web Mapping with Vector data is first studied in this chapter. After, the technology of Vector Tiles is presented, as well as their advantages compared to raster tiles. Furthermore, vector data transmission approaches over the Web are covered and different Vector Tile technologies as found in the scientific literature are exposed.

4.1 Toward Web Mapping with Vector Data

There have been some drastic changes in the way maps work since the appearance of the first web maps. According to Gaffuri (2012), it is necessary to change the approach of how web maps are made in order enable more interactivity and improve the user experience of web maps. As the majority of maps available on the web are raster-based, because their solutions are well established and easily implemented, there are specific cases where vector maps offer more opportunities and are more adequate than raster maps (Antoniou et al., 2009; Gaffuri, 2012). Therefore, the solution to go further in Web Mapping interactivity is to fully open Web Mapping to vector data (Gaffuri, 2012).

4.1.1 Benefits and Challenges of Vector Web Mapping

In Web Mapping, direct interaction is still a problematic task since the majority of Web maps are raster-based due to mature raster images delivery methods (Shang, 2015). Vector maps can respond to the limitations of raster maps since it comes with main benefits regarding interactivity or direct object manipulation. Vector maps enable a direct interaction of the user with the map objects which web maps based on raster data cannot offer (Antoniou et al., 2009; Gaffuri, 2012). Several authors, Gaffuri (2012); Lienert et al. (2012); Schnabel and Hurni (2009); Shang (2015), among others, have summarized the advantages that vector graphics can offer for interactive mapping. These advantages are listed below:

- They are scalable without loss of information;
- Vector map applications allow users to obtain the semantic information of a map object;
- The symbolization, such as the line width, fill color or transparency, is adjustable on-the-fly.
- The symbolization and geometry can be animated;
- Map features can be shown and hidden without regenerating and reloading the entire map. Vector maps are also independent of resolution, the map image can be redrawn using the source data every time users zoom or pan the map;

- Attributes can be attached to each individual map feature. Users can have access to the attributes of map objects, as well as, to the external data linked to these objects. This is improving map content personalization;
- Map features can be generated on-the-fly (e.g. diagrams);
- The geometry can be changed, allowing for projection to other coordinate systems without loss of information;
- With vector data, simple geo-processes can be performed on the client side, for instance, computing the area of a parcel or the length of a road or.

The above-mentioned benefits of vector graphics highlight the opportunities that vector maps van offer above raster maps, however, shifting from raster to vector Web Mapping also come with some challenges. As mentioned by Gaffuri (2012), *"especially improving the use of vector data in Web Mapping is often shown as the next challenge of Web Mapping"*. The author notes that the main obstacle to the development of vector Web Mapping is performance, because web maps are expected to be fast maps. Gaffuri (2012) notes hereby that: *"existing web maps based on vector data usually do not meet the minimal requirements in terms of display speed. For this reason, the raster maps have been preferred until now"*. Moreover, vector maps suffered from setbacks that prevented wide implementation. The large volume of vector data, the variety of formats with the disadvantages of vector encoding and the lack of standards have caused problems for the implementations of vector maps (Antoniou et al., 2009). However, taking into consideration that vector solutions offer opportunities on the client side and that client device memory, processing, and connection capacities are always improving, Web Mapping with vector data is becoming an accepted approach. Furthermore, with the methods of vector data becoming more mature, there is a promising opportunity for wider implementation of Vector Tiles (Gaffuri, 2012).

4.1.2 Vector-based technologies for Web Mapping

Vector-based Internet technologies are continuously developing and changing, Lienert et al. (2012) have presented an overview of different vector-based Internet mapping technologies, for different user groups and use cases. The most important ones mentioned by the authors are the following:

- *Flash/Flex*: Flash was designed by Adobe and originally developed by Macromedia for animated Web-based vector graphics.
- *XAML/Silverlight*: Microsoft's Silverlight consists of an XML-based vector graphics description language, known as XAML.
- *SVG*: SVG stands for Scalable Vector Graphics and is an XML format for vector graphics. It is a recommended standard of the W3C consortium. All modern Web-browsers can draw SVG without the use of a plug-in. The direct support in the browser and the large variety of visual effects and vector elements are the main advantages.
- *JavaFX*: JavaFX, being developed by Oracle, is a framework for the development of Rich Internet Applications (RIA). JavaFX enables developing large business applications in which maps come into play. However, it is not as widely used as Flash or Silverlight.

- *Canvas*: Canvas is a part of HTML5. It is an element which uses JavaScript commands for rendering graphic primitives.
- *WebGL*: WebGL is a javascript API for rending 2D and 3D graphics for Web applications. For instance, the popular Mapbox GL JS javascript library is based on WebGL to render interactive maps from Vector Tiles and Mapbox styles (Mapbox, 2017a).

Technologies such as Adobe Flash and Flex, or Microsoft Silverlight may better meet the needs of design-oriented cartographers, whereas JavaFX, Canvas or WebGL reach a smaller number of cartographers and need more programming skill (Lienert et al., 2012). For the client, vector graphics need to be rendered into vector maps for final display. Currently, the vector graphic APIs that are supported in HTML5 are Canvas, SVG and WebGL (Shang, 2015). While vector based technologies for Web Mapping are contiunually evolving, vector data in Web Mapping is still not widely implemented in contrast to raster data due to the lack of standards and well-established and integrated approaches to support efficient vector Web Mapping (Antoniou et al., 2009; Gaffuri, 2012). However, some approaches exist for vector Web Mapping, which are discussed in the next paragraph.

4.1.3 Approaches for Vector Web Mapping

Vector spatial datasets represent collections of spatial entities such as points, lines and polygons that are connected through spatial relations (Bertolotto & McArdle, 2011). Different approaches to transmitting these large sized vector map data over the Web are still in the early stages and present many challenges (Antoniou et al., 2009; Bertolotto & Egenhofer, 2001; Bertolotto & McArdle, 2011; Gaffuri, 2012; Shang, 2015). One approach is to use vector data in Web Mapping by displaying vector data on top of the raster base map. The major drawback of this approach is that the vector data size has to be small in order to be transmitted at once from the server to the client (Shang, 2015). Too detailed vector data or too large vector data size cause long loading times and slow transferring, and rendering of the vector data. This performance problem may also cause the final map not to be legible (Gaffuri, 2012). Other traditional methods for delivering vector datasets over the Web focused on progressive transmission (Antoniou et al., 2009; Bertolotto & McArdle, 2011). As Gaffuri (2012) explains, the principle of vector data progressive transmission methods is to *"load progressively the points composing the object geometries, and display the loaded data continuously, before the full transmission is complete"*. This means that the data are progressively displayed from a simplified view toward a more detailed view (Gaffuri, 2012). Vector Tiles are typically organized and created on different level of details or zoom levels. This means that the data needs to be simplified and generalized. This can be done using generalization algorithms (Ingensand et al., 2016). The preprocessing of Vector Tiles, such as the filtering,geometry clipping and cartographic generalization operation are performed in advance (GIS Wiki HSR, 2018). The progressive transmission methods are based on applying vector data generalization algorithms (Antoniou et al., 2009; Bertolotto & McArdle, 2011), such as the Douglas-Peucker (Douglas & Peucker, 2011) algorithm and the Visvalingam algorithm (Visvalingam & Whyatt, 1992). Progressive transmission does not contribute to solving the performance problem of vector data (Gaffuri, 2012), but the generalization process applied, helps to decrease the level of detail of a map. However, when decreasing the level of detail of a map, problems occur regarding the

preservation of topology (Bertolotto & McArdle, 2011) and the preservation of consistency (Bertolotto & Egenhofer, 2001). Not only do the progressive transmission methods deal with the complexity of simplifying vector map data while maintaining consistent topology, but the generalization algorithms also involve complex and time-consuming calculations and therefore cannot generally be applied for real-time Web Mapping. Progressive vector transmission remains for this reason challenging, as well as, due to the fact that effective compression techniques are currently lacking (Bertolotto & Egenhofer, 2001).
Another approach to improve the performance of web vector maps is according to Gaffuri (2012) to use specific data vector data formats that are small and compressed formats for improving transfer duration. Most vector formats are based on Extensible Markup Language (XML), the best known are Keyhole Markup Language (KML), Geography Markup Language GML, Styled Layer Descriptor (SLD) and Scalable Vector Graphics (SVG). The above formats are efficient for spatial data exchange, however, are less suited for fast transmission of vector map data. The GeoJSON format ,however, is another format that has been developed especially for fast transmission (Gaffuri, 2012).
Vector Tiling is another relatively new vector data transmission method that aims at cutting vector data into smaller pieces. It is inspired by the raster tiling concept in raster maps. Compared with progressive methods, it is relatively new and easy to implement in Web Mapping practice (Ingensand et al., 2016; Shang, 2015). The next section covers the approach of Vector Tiles and the latest trends regarding this technology.

4.2 Vector Tiles

Vector Tiles are small rectangular entities of vector data. Lopez et al. (2017) define a Vector Tile as *"a vector representation of geographic data covering a spatially contiguous and rectangular extent"*. The tiled vector techniques transmit small pieces of vector data to the client where they are reassembled based on predicted user behavior using a tiled caching mechanism (Antoniou et al., 2009; Bertolotto & McArdle, 2011). This means that Vector Tiling ensures that only the data which is suitable depending on its view and zoom level are requested and loaded by the client (Gaffuri, 2012). In this section, it will be explained how map tiling works and what the advantages or disadvantages are of Vector Tiles compared to raster tiles. Next to that the latest developments regarding Vector Tiles, as found in the scientific literature, will be discussed.

4.2.1 How Map Tiling woks

In order to understand how map tiling works, it is needed to understand the concepts of tile-based Web Mapping systems (Sample & Ioup, 2010; Shang, 2015). Sample and Ioup (2010) have defined the core properties of tile-based mapping systems:

1. Tile-based mapping systems have multiple discrete zoom levels, each corresponding to a fixed map scale. These zoom levels are adopted to present different level of details (LODs) of map data.

2. Each zoom level consists of multiple tile images that together form the map view.

3. Tiled images are sent from the server to the client with minimal processing.

4. Image tiles are accessible using a discrete addressing scheme and are primarily distributed utilizing a client/server system architecture.

Considering the above core properties, the main concepts of tile-based Web Mapping systems are zoom levels and tiling. Maps in lower zoom levels show less detail and are smaller in size, whereas higher zoom levels increase detail and increase the physical size of the displayed map. By enlarging or reducing the visualized area by zooming in or out, the level of detail updates accordingly (García et al., 2012), as depicted in Figure 4.1. In the pyramid system, cells in a high-resolution map area are split and sub-divided into a set of lower-resolution quadrants for display on a smaller scale. It is a generalization process occurring at different scale levels, creating increasingly smaller grids, going from the highest level of detail (large scale) toward the lowest level of detail (small scale) (Antoniou et al., 2009; Quinn & Gahegan, 2010). The opposite can also occur as the contents of one tile at a low zoom level (level 0) can be split into many tiles at a higher zoom level (level 1). In this case, the process starts with one single tile of the entire world, where after this tile is divided up into four equal parts. The process is repeated until the desired zoom level is reached (Nordan, 2012).

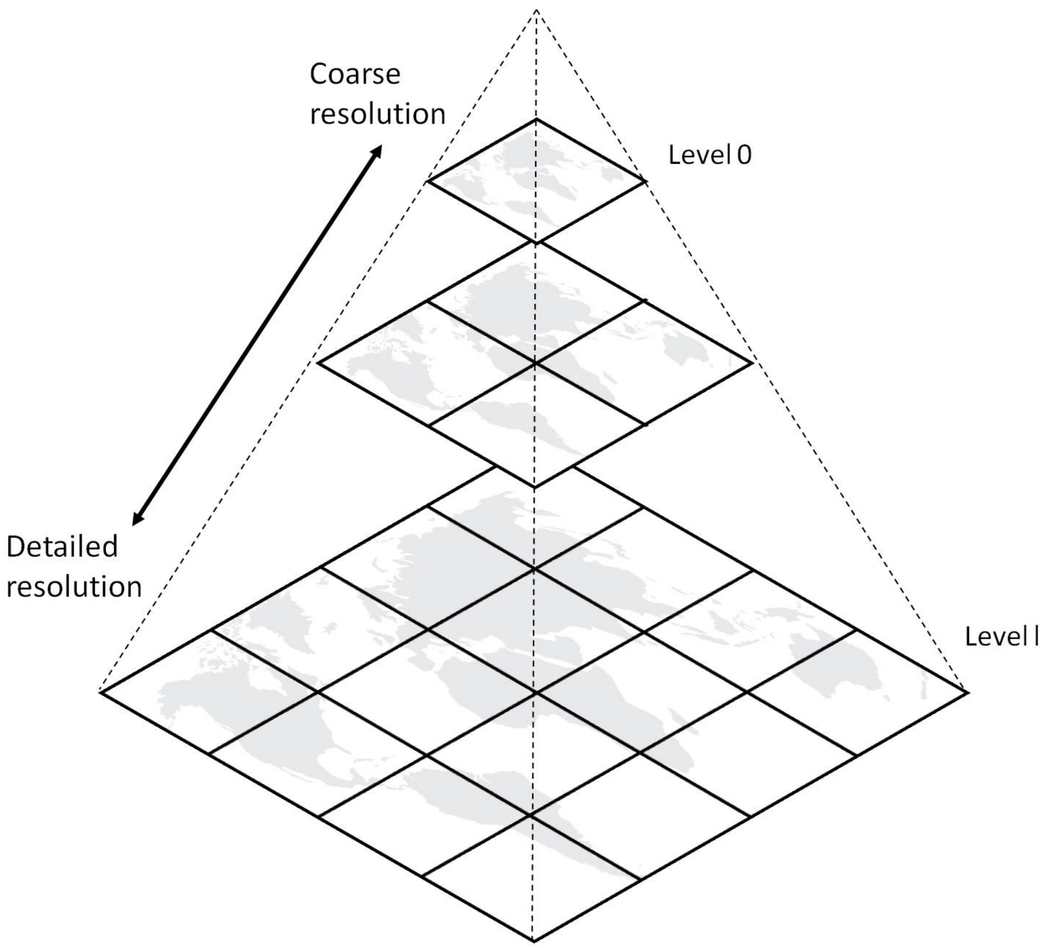

FIGURE 4.1: Tile pyramid representation(García et al., 2012)

Tiles and zoom levels give the user a better user experience (Shang, 2015). Moreover, the smaller the size of map data, the shorter the loading time is between the server and the client. Tiling can reduce the initial loading time and is, therefore, offering a more friendly experience to the client (Kang et al., 2001). Another benefit for the

client's experience is that tiles allow the client to retrieve the pieces of a map as needed, which means that only the tiles requested at a given scales are displayed and no further data retrieval is needed for data at other scales (Quinn & Gahegan, 2010).
One essential technical element behind tiled maps is Asynchronous JavaScript and XML (AJAX). AJAX enables dynamically sending map requests to a server or browser by caching memory in the background without the need to reload the entire HTML page (Shang, 2015). According to Kang et al. (2001), while tiling can minimize the initial user's response time, it is not able to minimize the total response time. Therefore the authors have proposed an efficient tile-prefetching algorithm to minimize the total response time. Tiles that are likely to be accessed should be saved in the cache for future reusing. This enables less communication delay to get the required tiles from the server (Kang et al., 2001). Quinn and Gahegan (2010) also present a model for determining high-priority areas for tile caching. The model takes into account variables that are of interest to Web map users according to previous research.

4.2.2 Vector Tiling approaches

Tile-based Web maps are often raster-based maps. However, major players have shifted to using Vector Tiles (Martinelli & Roth, 2015). Vector Tiles are similar to raster tiles, but instead of pre-rendered raster images, the spatial data used is a vector representation which is inside the tile (Antoniou et al., 2009; Shang, 2015). Figure 4.2 depicts the principle of Vector Tiling, a vector dataset is split into equally smaller sized rectangle pieces at a fixed scale.

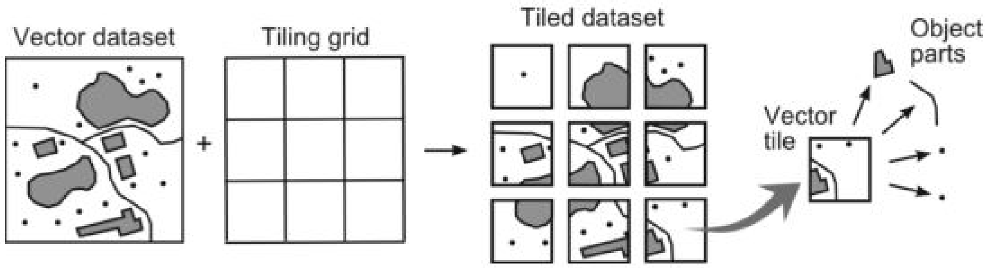

FIGURE 4.2: Principle of Vector Tiling (Gaffuri, 2012)

Multiple solutions exist to address Vector Tiling, as noted by the Open Geospatial Consortium (2017a) Testbed-12. For Vector Tiling, there are two major approaches identified for turning vector features into tiles: (1) Render-based tiling which is focused on visualization of the vector features, and (2) Feature-based tiling which is focused on maintaining the integrity of vector features for storage and analytics. Google Map, Apple Maps and Mapbox, all use approaches that can be classified as render-based tiling. With the render-based tiling approach clients can read the tiles without the need to do any feature geometry assembly, reprojections or other intensive operations (Open Geospatial Consortium, 2017a). Antoniou et al. (2009) discussed that some early efforts on Vector Tiling had been introduced based on the off-line preparation of Vector Tiles. The main disadvantage of these methods according to the authors is that they *"do not provide a merging mechanism of the tiles at the client side"*. The focus is mainly on the simple visualization of the map. This results in a critical issue that the map has segmented entities at the edges or borders of the tiles leading to inconsistencies and visual/graphic discontinuities (Antoniou et al., 2009; L. Li et al., 2017). Therefore, Antoniou et al. (2009) suggest that a *"mechanism providing a solution for on-the-fly tiling of the data on the server and the merging of the tiles on the client will be more appropriate"*. With this mechanism, the client-server interaction takes place asynchronously using AJAX requests and the caching takes place on the client side. Moreover, offloading the rendering process of Vector Tiles to the client side can offer several benefits, such as faster maps, more interactivity, map design benefits and therefore better user experiences (Martinelli & Roth, 2015). As suggested, Vector Tiles offer several opportunities, especially for the development of Web Mapping systems and on the client side. The next section compares Vector Tiles with raster tiles by discussing the opportunities of Vector Tiles while highlighting the limitations of raster tiles and challenges of Vector Tiles.

4.2.3 Vector Tiles compared to Raster Tiles

Vector Tiling is relatively new compared to raster tiling in Web Mapping and not well established yet due to a general lack of standards regarding Vector Tile-based Web Mapping systems (Gaffuri, 2012; Shang, 2015). However, raster tiles have their limitations as well. Back in the nineties, Tomlin (1990) stated: *"Yes raster is faster, but raster is vaster, and vector just seems more corrector"*. This statement was used to explain that raster rendering is very fast and that there was nothing faster than raster tiles because they had pre-rendered content. However, the statement is not entirely accurate anymore, because when going to high-resolution devices, the raster tiles can become very big in size, making vector data actually better suitable for faster data transmission. Nowadays vector data can be rendered more efficiently with Graphics Processing Units (GPUs) (e.g. WebGL via the Browser) (Williams, Craig and Punt, Edie, 2016). Furthermore, according to Peterson (2003), the statement by Tomlin did not consider the ability of vector graphics to associate attribute data and interactivity, something that raster graphics are limited in. Moreover, with raster tiles, the limitation is that they have fixed zoom levels in contrast to Vector Tiles that are scalable and can be manipulated easily. The design, building time and maintenance of raster tiles can be very time-consuming because for each zoom level a tile needs to be created and each map style must be created in a separate raster tileset. Raster tiles are just images, labeling is pre-set and cannot be changed and no geo-processing calculations can be performed on spatial features (e.g. calculating the distance between two points). Furthermore, raster times lack flexibility in terms of style and content (Schmidt & Weiser, 2012). Instead of pre-styled raster images, Vector Tiles can be manipulated on the client-side. This means that when requesting a tile, map users can design their own maps by applying dynamic labeling or customizing different styles on the fly in the browser. This is offering map design, symbolization and interactivity benefits for map users. Vector Tiles enable access to attributes because they contain source data such as geometries, road names, area types, etc. Furthermore, Vector Tiles are smaller than raster tiles, enabling higher resolution maps, efficient caching, faster rendering and better user experience (Cartwright et al., 2007; GIS Wiki HSR, 2018; Lopez et al., 2017; Mapbox, 2017b; Mapdata Services, 2017; Martinelli & Roth, 2015; Quinn & Gahegan, 2010). Where Vector Tiles have advantages over raster tiles or share similarities with raster tiling problems and solutions, Vector Tiles come with some unique challenges as identified by the Open Geospatial Consortium (2017a) Testbed-12:

- Data coherence challenges: data coherence refers to the ability to assemble and access. A feature when reading Vector Tiles a tile should contain all necessary information to assemble a feature that crosses multiple tiles back into its original form.

- Defining multiple levels of detail: for vector data this is issue is more complicated, however two techniques can be identified being feature filtering and feature generalization.

- Tile sectioning: for vector data, there is no way to associate a feature with a tile. With features crossing tile boundaries, one needs to define an approach to associate features with tiles.

- Unique feature identification: the need to identify the source feature.

4.2.4 Vector Tiles providers

Several existing providers already create Vector Tiles. It should be noted that often providers still own the data because they wish to promote their products or services. A few Vector Tiles providers have been listed below:

- *Mapbox*: The MapBoxGL framework, released to render into vector maps on the browser (Lopez et al., 2017): MapBoxGl offloads the rendering work to the client. With MapBox Studio the client can have complete design control by creating custom maps according to their needs (Martinelli & Roth, 2015). Maputnik is a free and open visual editor for the Mapbox GL styles (Lopez et al., 2017).

- *Mapzen*: Mapzen provides API access to their public Vector Tiles. Mapzen states that access and the platform should remain free and Open Source. Mapzen however, does not give access to the entire raw data and one is bound to the limitations of the service (Martinelli & Roth, 2015). Note: Mapzen shut down its services end of January 2018 (Mapzen, 2018).

- *Kartotherian*: The Kartotherian has the goal to provide a free Map service free for use for everyone. It is a Maps Tile service for Wikipedia. The data cannot be downloaded, because it is only a service (Martinelli & Roth, 2015).

- *Apple and Google*: Google and Apple are using Vector Tiles, but their tiles are not accessible for the general public and use a proprietary format. Only their services can be used, such as Google Maps (Martinelli & Roth, 2015).

- *Thunderforest*: Thunderforest is a tiles provider for use in Web Mapping libraries or applications. Mapnik and PostGIS at its core. Vector Tilesets are available in the MVT format, which means that they are compatible with a wide range of rendering libraries.

- *OpenMapTiles.org*: The OpenMapTiles project is a successor of OSM2VectorTiles, a project which started as a student bachelor thesis at HSR Rapperswil in cooperation with Klokan Technologies GmbH. The project turns the publicly available OpenStreetMap data into ready-to-use packages containing Vector Tiles for the whole planet, individual countries and major cities. Downloaded map tiles can be displayed on websites with JavaScript viewers (GIS Wiki HSR, 2018; Martinelli & Roth, 2015; OpenMapTiles, 2018a)

- *Mapcat.com* Mapcat provides Vector Tiles conform to the Mapbox Vector Tile Specification. It is possible to use Mapcat services and vector-based map tiles with OpenLayers or with Mapbox GL JS. The ol-mapbox-style JavaScript library enables to create a OpenLayers map from Mapbox compatible stylesheet. Mapcat Vector Tile data schema is based on and extends the well-known Open Map Tiles Schema (Mapcat, 2018). There are plans to integrate Mapcat Vector Tiles as another provider in the QGIS Vector Tile Reader plugin (Keller, 2018).

Providing and creating Vector Tiles requires a sound knowledge and understanding of Web Mapping technologies. This is one of the reasons why according to Martinelli and Roth (2015) Vector Tiles are not always adopted by the mainstream yet. Several tiling solutions and tiling formats furthermore exist, and are discussed in the next sections.

4.2.5 Vector Tiling solutions identified by OGC Testbed-13

The Open Geospatial Consortium (OGC) have come up with a draft of the Vector Tiles Engineering Report which captures the requirements, solutions, and implementation experiences of the Vector Tiling work package in OGC Testbed-13. In the report several Vector Tiling solutions have been analyzed:

- Mapbox Vector Tiles
- Cesium 3D Tiles
- Esri I3S
- Ecere Gnosis
- GeoServer Vector Tiles Extension

The different solutions that currently provide Vector Tiling support were assessed based on the following parameters:

- Support for different projection systems
- Support for styling
- The tiling scheme and how tiles are addressed
- Support for different types of geometries
- Support for 3D data
- Handling of generalization
- The role of different response formats such as GeoJSON, TopoJSON, etc
- How attributes are handled
- How the solution may or may not align with the OGC standards baseline
- Sustainability
- The ability of the client (if a client exists) to reassemble features
- Support for moving features
- The possibility of combining several layers in one tile
- Which operations are possible with vector features (e.g. write support, etc.)

Some of the above parameters are interesting to implement for assessing the cartographic advantages and disadvantages of the Vector Tile solutions that are inventoried in this thesis. Besides, the Open Geospatial Consortium (2017a) Testbed-12 also comes with several use cases or topics that can be seen as important for discussing the role of Vector Tiles for cartography. Styling possibilities (e.g. labelling), preservation of topology, simplification and generalization (e.g. filtering) as well as the use and choice of tiling schemes or projections are all elements to take into account.

4.2.6 Vector Formats and Vector Tile Encoding

Vector Tiles can be stored in file systems or in database structures. There are several formats that are considered to be interesting for Vector Tiling (Table: 4.1). Two common formats for transferring vector data on the Web are XML-based such as GML and JSON-based formats such as TopoJSON and GeoJSON (Ingensand et al., 2015). Both GeoJSON and TopoJSON are human readable formats. However, there are more geospatial applications that support tiles in this GeoJSON compared to TopoJSON which has limited Vector Tiling applications using it (Open Geospatial Consortium, 2017b). Where GeoJSON and TopoJSON can be used, Mapbox Vector Tiles (MVT) is the recommendable format since it is widely supported in many vector data applications (Open Geospatial Consortium, 2017b). The Mapbox Vector Tile Specification is the current de facto standard (Lopez et al., 2017). The GeoPackage Encoding Standard is the OGC counterpart to the Mapbox Vector Tiles Specification. Google and Apple are also using Vector Tiles. However, these providers have proprietary formats which are not openly accessible to the public (Martinelli & Roth, 2015). The Vector Tile encoding remains a challenging issue because the way that the Vector Tile is encoded affect the transmission efficiency of the Vector Tiles over the Web (Shang, 2015). XML and JSON encoded tiles are easy to implement and widely accepted. GeoJSON is, for instance, human-readable and offering easy debugging. GeoJSON can, compared XML-based encoding formats, be more easily and quickly parsed by a computer (L. Li et al., 2017). GeoJSON, therefore, has improved transmission efficiency, however, other binary formats such as Google Protocol Buffers can also offer more simplicity and performance because it was designed to be smaller and faster than XML or GeoJSON. The problem is that XML or GeoJSON encoding formats may offer low transmision efficiency with large vector datasets (Ingensand et al., 2015; L. Li et al., 2017; Open Geospatial Consortium, 2017b; Shang, 2015).

4.2.7 Tiled Vector Web Map Services

From the point of view of the web service, Vector Tiles and raster tiles can be served through the same interface, however, are encoded in a different format, as discussed in the previous section (Lopez et al., 2017). Vector Tiles, furthermore, belong to view services and not to download services. They are a representation of vector data intended for visualization (Lopez et al., 2017). Today no open and widely adopted standard exists for the implementation of web services involving Vector Tiles (Ingensand et al., 2016). However, the Open Geospatial Consortium (OGC) released the Web Map Tile Service (WMTS) as the standard protocol for tile-based Web Mapping applications (Sample & Ioup, 2010; Shang, 2015). This map service standard has been developed based on WMS and the “pyramid technique”, which have been adopted by most large-scale Web Mapping systems (L. Li et al., 2017). Open Geospatial Consortium (2017b) has recently started testing different approaches for the generation of Vector Tiles from source data, the creation of Vector Tiling services and the

GML	The Geography Markup Language is an XML-based format which is often used with Web Feature Services (WFS). It is an OGC standard which means that it is widely implemented. The advantage of GML is that it supports all types of geometries. The disadvantage is the weight of the data due to the fact that GML is based on XML
GeoJSON	GeoJSON is a human readable JSON-based format. It is not an OGC standard. The advantage is that it is widely used in the context of web applications since this kind of data is easier to parse and to integrate within a JavaScript application.
TopoJSON	TopoJSON is also JSON-based and human readable. The advantage of TopoJSON is also that it is easy to parse with JavaScript applications and that the data is even more compact than GeoJSON. However, the disadvantage is that there is no widespread support for this format.
PBF & MVT	The Google's Protocol Buffer Format (PBF) is a binary format that has been used for the Mapbox Vector Tiles (MVT) format. The advantage of PBF and MVT is that the data is very compact. The disadvantage is that the binary data needs to be converted. MBtiles is a file format (SQLite binary file) for storing tilesets incl. raster and Vector Tiles with metadata

TABLE 4.1: An overview of Vector formats (Open Geospatial Consortium, 2017b, p. 20)

implementation of Vector Tiles clients with the aim of delivering faster, lighter and more robust vector data via the web. The presented work of the Vector Tiling Engineering report demonstrated three different approaches to Vector Tiling geospatial web services:

- Approach 1 - Web Feature Service (WFS) with Vector Tiles extensions
- Approach 2 - Web Feature Service (WMTS) with Vector Tiles extensions
- Approach 3 - Unified Map Service, unifying WMS, WMTS, WFS & WCS capabilities with shared semantics

A detailed description of these approaches is out of scope in this thesis. The focus lies more on Vector Tiling solutions for the generation of Vector Tiles from source data, serving Vector Tiles and the implementation of Vector Tiles clients. The next section discusses the topic of serving Vector Tiles.

4.2.8 What we use Vector Tiles for

Norman, P. (2018) has identified several cases for which Vector Tiles could be used for. Vector Tiles could be used for (1): Analysis; (2) for converting them to raster tiles on the server and (3) for sending them to the client and render them on the client-side. The last use is considered here to be the main use. Currently, on the Web, the main use of Vector Tiles can be seen with Vector Tile basemaps that are served to the client. Vector Tiles can, therefore, be considered useful for visualization and for quickly rendering multiple similar map styles and changing colors (Norman, P., 2018).

4.2.9 Generating and serving Vector Tiles

There are different uses of Vector Tiles. For instance, one could only consume them, or one could generate Vector Tiles themselves and serve them to the client. Chapter 6 gives an overview of different uses and uses cases of Vector Tiles and how they are positioned in a workflow.

Regarding research on serving Vector Tiles, Norman (2016) has tested different options to serve Vector Tiles, as depicted in Figure 4.3. He has compared the server options by testing them based on characteristics such as full planet, updates, data sources, and output formats.

Server	Full planet	Diff updates	Non-OSM data	GeoJSON	TopoJSON	Mapbox Vector Tiles
node-mapnik	Yes	Yes	Yes	Some	No	Yes
Tilezen tileserver	Yes	Yes	Yes	Yes	Yes	Yes
Tegola	Yes	Yes	Yes	No	No	Yes
t-rex	Yes	Yes	Yes	No	No	Yes
TileStache	Yes	Yes	Yes	Yes	No	Yes
Tilemaker	No	No	Yes	No	No	Yes
VectorTileCreator	Unknown	No	No	No	No	No

FIGURE 4.3: Different Vector Tile server options (Norman, 2016)

Shang (2015) also has summarized different Vector Tile servers according to their supported Vector Tile encoding formats and whether they are open source or commercial, the result is depicted in Figure 4.4.

Vector Tile Server	Tile Format Supported	Open Source	Commercial
TileStache	GeoJSON,TopoJSON,ArcJSON, MVT (Mapnik Vector Tiles)	Yes	No
Vector-tile-server	Google Protocol Buffers	Yes	No
Node-tile-server	GeoJSON	Yes	No
MapBox Tile Server	Google Protocol Buffers	No	Yes

FIGURE 4.4: Summary of Vector Tile Servers (Shang, 2015)

Due to the current pace in which technological developments occur, it is hard to keep up with all the Vector Tiles solutions. Even though, TileStache is often mentioned as a Vector Tile server that supports most vector encoding formats and supporting most popular spatial databases or files (Martinelli & Roth, 2015; Nordan, 2012; Norman, 2016; Shang, 2015), other solutions might also be useful for serving Vector Tiles. Therefore, in this thesis an inventory will be made of the current options to generate and/or serve Vector Tiles, to give more overview of the different options. The findings will be part of a chapter describing the current state of the Vector Tile technology, by focusing on the available tools on the web that implementing Vector Tiles.

Chapter 5

Conceptual Model

In the previous chapters, relevant concepts in Cartography and Web Cartography and regarding the Vector Tile technology were introduced. In this chapter, the most important findings relevant to this research are described. The main concepts are displayed in a conceptual model (Figure 5.1) in which the relationships between the different concepts are made clear.

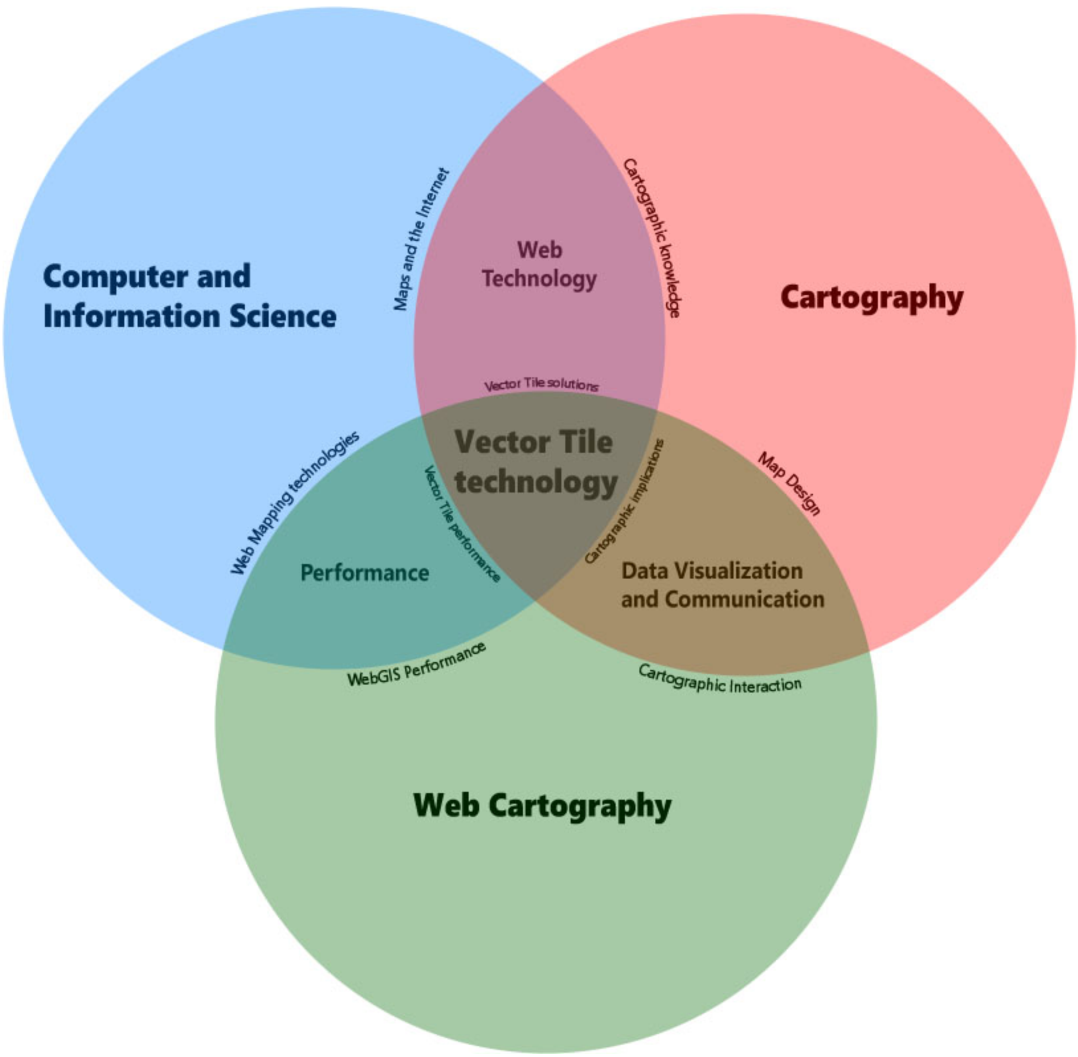

FIGURE 5.1: Conceptual model

This research is placed in several fields of research: Computer and Information Science, Cartography and Web Cartography. The concepts surrounding these fields are interrelated and come together in the middle to Vector Tile Technology, which is the focus of this research.
Research in Chapter 3 demonstrated that the influence of technology on Cartography was very significant, making internet maps the major medium of spatial information delivery over the web. With technology becoming more performing and internet maps becoming more interactive, new opportunities are presented for Cartography.
The technology of Vector Tiling was presented in chapter 4. Vector Tiles are an emerging and promising solution of fast transmission of vector data over the web. Scientific research regarding Vector Tile technology is still at early stages and there is still a lot of areas of vector tiling that can be researched in greater detail. While a lot of research focuses on Vector Tile performance, such as transmission methods and data reducing or improving techniques, less research has been done on the implementation of Vector Tiles. Therefore, a more user-oriented evaluation is needed.
For discovering the cartographic implications of Vector Tiles, research has to be done on the tools and technologies available on the Web. This will result in an inventory of the Vector Tile tools solutions available on the web and an overview of different Vector Tile solutions/workflows. Afterwards, these Vector Tiles solutions are assessed by considering findings from cartographic theory and expert-interviews. The question should be asked whether Vector Tiles solutions stimulate an effective approach for communicating a message, as cartographers always have done with maps.
The knowledge gap between computer and information science on one side and web cartography on the other side has to be filled. The aim is to bridge the different fields and their relationships together. Even though all fields in the conceptual model have their share in the Vector Tile technology, some relationships are less clear than others. The gap between cartographic knowledge and technology is one good example where there is a knowledge gap that needs to be filled. Other relationships are better connected, such Map Design and Data Visualization. Of particular interest in this thesis, is how Vector Tiles can bridge the divide between technological aspects of computer sciences and the theoretical framework and knowledge of Cartography and Web cartography. The next chapter proposes an inventory of current Vector Tile solutions for Web Cartographers.

Chapter 6

Inventory of Vector Tile tools & solutions for Web Cartographers

Earlier in this thesis, it was mentioned that due to the current pace in which technological developments occur, it is hard to keep up with all the web mapping technologies (Cartwright et al., 2010; Muehlenhaus, 2014; Roth, Donohue, et al., 2014). Some authors have already attempted to give an overview of web mapping technologies (Ballatore et al., 2011; Roth, Donohue, et al., 2014) or of Vector Tile server options (Norman, 2016). In this chapter, an inventory will be provided of the current options that can be used to generate, serve and render/visualize Vector Tiles. Different Vector Tile tools will be shortly described in a table that together forms an overview of the available tools and technologies on the web for implementing Vector Tiles. Not all the Vector Tile implementations are listed in this inventory. However, it was aimed at presenting the most important ones that were found on the Web during this thesis project. By giving an overview, the findings describe the current state of the Vector Tile technology.

6.1 Classifying Vector Tile tools

Google Maps started to offer Vector Tiles for their desktop client already back in 2013. While using the Google Maps API is it is even possible to make your own styles with a styling wizard called "Google Maps API styling wizard" (Google Maps, 2018). However, Google Maps is not the only one using Vector Tiles for their maps. There is an important variety of tools that implement Vector Tiles on the Web. The current pace in which Vector Tile tools and technologies change and develop is hard to keep up with. Therefore, there is a need to create an overview of the current Vector Tile tools and solutions. One way to classify Vector Tile tools is to differentiate them by their function or use. As earlier discussed in the theoretical framework, in Chapter 3, contemporary web mapping technologies can be organized into three broad categories, namely (1) server-side technologies, (2) client-side technologies and (3) web services or intermediary scripts. For Vector Tile technologies especially the server-side and client-side are relevant. Therefore the focus in the inventory is on Vector Tiles tools that generate Vector Tiles, convert Vector Tile formats, serve Vector Tiles or render Vector Tiles. Typically, on the web the following type of Vector Tiles tools can be found: (1) Parsers & Generators, (2) Clients, (3) Applications/Plugins, (4) Command-Line Utilities and (5) Servers (Github Mapbox, 2018). Furthermore, according to Mac Gillavry (2017), when it comes to generating, serving and rendering Vector Tiles, there seems to be the following divide:

1. Mapnik-rendering in the OpenStreetMap ecosystem
2. WebGIS extended with Vector Tiles
3. Database-rendering

The Vector Tile tools and technologies found on the web are classified based on the above divide. The three different categories in this divide give a way to create an overview of different Vector Tile solutions, however it should be noted that the categories can overlap with each other since some Vector Tile tools could fit in one or more categories. This means that the boundaries between the three different categories are arbitrary and do not have hard or closed boundaries. For each Vector Tile tool it is stated what type of tool it is, a short description is given, links are provided on where to find documentation and it is mentioned whether the solution was tested successfully during this thesis project or not. It was decided to add information in the inventory about testing tools in order to inform the reader about the current state of the Vector Tile tool. In other words, the aim was to raise awareness to the reader whether at the moment of writing the thesis, the tool was implementable or not. Different answer options were set up for the column "Tested?", namely: (1) "Not tested", (2) "Tested successfully", and (3) "Tested, but unsuccesful". "Not tested" means that there was no time or no priority to test the tool. A successfully tested tool means that the documentation was still accessible and up to date while writing this thesis, as well as that there was enough expertise or knowledge to make the solution work. "Tested, but unsuccesful" means that either the tool was outdated, did not work, or that there was a lack of expertise or knowledge to make the tool work. It was added to the comments why the tool was tested unsuccesfully. It should be noted that testing each tool in detail was not the aim of this thesis. Testing the Vector Tiles tools or solutions on their performance was also out of the scope of this thesis. The focus lied on giving an overview of available Vector Tile tools and solutions. Afterwards, in a further chapter, it was possible to investigate where the cartographic strengths and weaknesses can be found in the Vector Tile solutions. Two Vector Tiles workflows and their relevant Vector Tile tools were finally assessed on their cartographic strengths and weaknesses and cartographic potential in Chapter 7.

6.2 Inventory of current Vector Tile tools & technologies

6.2.1 Mapnik-rendering in the OpenStreetMap ecosystem

Mapnik is an open source mapping toolkit for map rendering. It supports a variety of geospatial data formats and provides flexible styling options for designing many different kinds of maps (Wiki OpenStreetMap, 2018). It can read PostGIS, ESRI shapefiles, .osm files, TIFF rasters, any GDAL or OGR supported formats, CSV files, and many more. Important users of Mapnik are the OpenStreetMap project (OSM) and Mapbox. Mapnik can then be used to render the OpenStreetMap data into maps with the appearance the user wants. The Mapbox Vector Tile Specification is widely implemented for the Vector Tile tools in this category. Table 6.2.1 presents an overview of Vector Tile tools that implement the Mapbox Vector Tile Specification in the Mapnik/OSM ecosystem.

ID	Name	Type of tool	Description	Links (click on text to open in browser)	Comments	Tested?
1	Mapnik-vector-tile	Parsers & Generators	A Mapnik implemention of Mapbox Vector Tile specification. C++ Vector Tile read/write.	Github [https://github.com/mapbox/mapnik-vector-tile]	Provides C++ headers that support rendering geodata into Vector Tiles and rendering Vector Tiles into images.	Not tested
2	Mapbox-vector-tile by Tilezen	Parsers & Generators	Python package for encoding & decoding Mapbox Vector Tiles	Github [https://github.com/tilezen/mapbox-vector-tile]		Not tested
3	Kosmtik	Application	Make maps with OpenStreetMap and Mapnik.	Github [https://github.com/kosmtik/kosmtik]	Exports to common formats (Mapnik XML, PNG...). Plugin: kosmtik-mbtiles-export: export your project in MBTiles. Only Carto based projects are supported.	Tested successfully
4	MVT styler	Application	MVT Styler is an editor of vector styles for interactive maps. Implementation of Mapbox Vector Tile specification	Sputnikmaps [http://sputnik-maps.github.io/mvt-styler/]; Github [https://github.com/sputnik-maps/mvt-styler]		Not tested
5	Maputnik	Application	Maputnik is a free and open source visual editor for the Mapbox GL style specification	Maputnik-Editor [https://maputnik.github.io/editor]; OpenMapTiles-Editor [http://editor. openmaptiles.org/]	Once you are done editing the style you can download the modified style in JSON format	Tested successfully
6	Mapbox studio	Application	Desktop design studio for both creating Vector Tiles from raw geodata and for rendering them on-the-fly into image tiles.	MapboxStudio [https://www.mapbox.com/mapbox-studio/]; Github [https://github.com/mapbox/mapbox-studio]	Design on top of Mapbox template styles. A point-and-click interface built for designers and cartographers. Import or create custom data layers.	Tested successfully

7	Mapbox Cartogram	Application	An application to create custom maps by uploading a picture and selecting colors you want to use. It works with the Mapbox style API and it is possible to open styles in Mapbox Studio style editor.	MapboxCartogram [https://www.mapbox.com/cartogram/]		Tested successfully
8	Mapbox GL Inspect	Plugin	A plugin for Mapbox GL to add an inspect control to view all features of the vector sources and allows hovering over features to see their properties.	Github[https://github.com/lukasmartinelli/ mapbox-gl-inspect]	Not tested due to time constraints	Not tested
9	TileStache	Server	Python-based server application that can serve up map tiles based on rendered geographic data	TileStache [http://tilestache.org/]; DigitalGeography [http://www.digital-geography.com/set-tileserver-using-tilestache-gunicorn-nginx/]; GithubForum [https://cmhh.github.io/post/tiles/]	Setting up a Mapnik source for use with TileStache requires a Mapnik XML file, that can be created with TileMill. Documentation of TileStache outdated which gave some testing problems	Test, but not successfully
10	Tileserver by Tilezen	Server	A lightweight tileserver to share code paths with tilequeue for tile generation based on Mapzen Vector Tile Service	Github [https://github.com/tilezen/tileserver]; Github [https://github.com/tilezen/vector-datasource]	Mapzen's Vector Tile service was shut down end of january 2018 [https://mapzen.com/blog/shutdown/]; [https://www.wired.com/story/ mapzen-shuts-down/]	Not tested

11	OpenMapTiles & Tileserver GL	Server	OSM Vector Tiles provider & Server. OpenMapTiles is a project aiming to create world maps from open data. It consists of a set of tools allowing everyone to create his own vector map tiles from OpenStreetMap data for hosting, self-hosting or offline use.	OpenMapTiles [https://openmaptiles.org/]; Github [https://github.com/openmaptiles/openmaptiles]; Pedro Sousa Blog [http://build-failed .blogspot.nl/2017/02 /playing-with-mapbox- vector-tiles-part-1.html]	From Klokan Technologies GmbH. Former projectname: OSM2VectorTiles.org. It uses pre-generated OSM tiles that can be used to set up your own tileserver with TileServer GL using Docker.	Tested successfully
12	Kartotherian	Server	Wikipedia Maps Tile Server	Github [https://github.com/kartotherian/ kartotherian]		Not tested
13	MapboxGL	Client	JavaScript library that uses WebGL to render interactive maps from Vector Tiles and Mapbox styles	MapboxGL [https://www.mapbox.com/help/define-mapbox-gl/]; Github [https://github.com/mapbox/mapbox-gl-js]	It's a part of the MapboxGL ecosystem. Not open-source.	Tested successfully
14	OpenLayers	Client	JavaScript vector & raster library.	OpenLayers [https://openlayers.org/en/latest/examples/osm-vector-tiles.html]; Github [https://github.com/openlayers/ openlayers/pull/4219]; OpenLayers [https://openlayers.org/en/master/examples/ mapbox-vector-tiles.html]	The OSM Vector Tiles example offered by OpenLayers uses the Mapzen Vector Tile service which was shut down end of January 2018. This solution was tested before the shut down. The Mapbox Vector Tiles example worked without issues with a Mapbox API key	Tested successfully

15	Leaflet	Client	Lightweight Javascript open-source library.	LeafletJS [http://leafletjs.com/]; Github-MapboxGLleaflet [https://github.com/mapbox/mapbox-gl-leaflet]; Github-LeafletVectorGrid [https://github.com/Leaflet/Leaflet.VectorGrid]; OpenMapTiles [https://openmaptiles.org/docs/website/leaflet/]	Leaflet doesn't support Vector Tiles by default. But it can load and render the Vector Tiles directly by the help of the mapbox-gl-leaflet plugin and VectorGrid plugin. The plugins are experimental and it are not developed or actively supported by Mapbox.	Tested successfully
16	Tangram from Mapzen	Client	Tangram is another open source WebGL based rendering client which supports the Mapbox Vector Tiles specification. Tangram is designed to use vector data sources such as Mapzen's Vector Tile Service, which is a tiled, hosted version of the OpenStreetMap database.	Mapzen [https://mapzen.com/products/tangram/]; Github [https://github.com/tangrams/tangram]	The solution was tested before Mapzen shut down its services end of January 2018 [https://mapzen.com/blog/shutdown/]; [https://www.wired.com/story/ mapzen-shuts-down/] [https://mapzen.com/blog/migration/]	Tested successfully
17	Nextzen	Client	Long-term support for Mapzen maps, vector & terrain tiles	Mapzen [https://mapzen.com/blog/long-term-support-mapzen-maps/]	Not tested because the project was still setting up while writing this thesis chapter	Not tested

18	Tilemaker	Command-line utility	Make OpenStreetMap Vector Tiles without the stack	Tilemaker [https://github.com/systemed/tilemaker]	Tilemaker creates Vector Tiles (in Mapbox Vector Tile format) from an .osm.pbf planet extract. I tested it and created a mbtile file from a osm.pbf file downloaded from download.geofabrik.de.	Tested successfully
19	OSM VectorTileCreation	Command-line utility	VectorTileCreator is part of KDE Marble and takes the unique approach of creating tiles of raw OSM data	Techbase [https://techbase.kde.org/Marble/ OSMVectorTileCreation]; PaulNorman [http://www.paulnorman.ca//2016/11/serving-vector-tiles/]		Not tested
20	Osm2pgsql	Command-line utility	Tool to import OpenStreetMap data into own database	LearnOSM [http://learnosm.org/en/osm-data/osm2pgsql/]	Alternatives for linux: Osmosis and Imposm. Tested in Windows: used a .pbl file from openstreetmap, downloaded from geofabrik.	Tested successfully
21	Tiler by Geovation	Command line utility	command line orientied pipeline for taking vector data in formats such as Shapefiles, and transforming them into raw Vector Tiles and MBTiles files	Github [https://github.com/Geovation/tiler]	Project was made possibly thanks to Tippecanoe from Mapbox, ogr2ogr from OSGeo and PostGIS Docker Container from Tim Suttom. The tool was not tested due to a lack of time	Not tested

22	MVT Driver & MBTiles driver by GDAL/OGR	Command line utility	MVT driver can read and write Mapbox Vector Tile files, as standalone files, uncompressed or gzip-compressed (typical extensions are .pbf, .mvt, .mvt.gz), or a tileset at a given zoom level of such files.	GDAL MVT Driver [http://gdal.org/drv_mvt.html]; GDAL MBtiles Driver [http://gdal.org/frmt_mbtiles.html]	Mapbox Vector Tiles stored within a SQLite container conforming to the MBTiles format are handled by the MBTiles driver. Solution not tested due to a lack of time and expertise	Not tested

TABLE 6.1: Overview of Vector Tile tools based on Mapnik-rendering in the OpenStreetMap ecosystem

6.2.2 WebGIS extended with Vector Tiles

WebGIS is here considered to be the process of using maps delivered by geographic information systems and is, therefore, more than just web cartography. WebGIS is a type of distributed information system, comprising at least a server and a client, where the server is a GIS server and the client is a web browser, desktop application, or mobile application. The previous category of 'Mapnik-rendering in the OpenStreetMap ecosystem' is also considered to be Web Mapping, however, here in the category of 'WebGIS extended with Vector Tiles' the focus lies on Desktop-GIS and Web-GIS that implement Vector Tiles without a direct link with OpenStreetMap or the Mapbox Vector Tile Specification. While mapping tools such as OpenStreetMap, OpenLayers and Mapbox have already been mentioned as leaders in Web GIS, ArcGIS by ESRI, QGIS and GeoServer are other important players in WebGIS that have Vector Tile extensions for implementing Vector Tiles. ArcGIS has proprietary tools to generate or to serve Vector Tiles based on the Mapbox Vector Tile Specification, whereas QGIS and GeoServer are open-source and also have their own plugins or extensions for implementing Vector Tiles. Table 6.2.2 presents the different Desktop-GIS or WebGIS options that can implement Vector Tiles.

ID	Name	Type of tool	Description	Links (click on text to open in browser)	Comments	Tested?
23	ESRI ArcGIS Pro	Application/ Client	Generate Vector Tiles from maps authored in ArcGIS Pro or imported from ArcMap	ESRI [http://www.esri.com/en/arcgis/products/arcgis-pro/overview]; ESRI Training [https://www.esri.com/training/catalog/5851736fd33f8b0b47b78e26/creating-vector-tiles-in-arcgis-pro/]	Used ESRI vector basemaps that are delivered as Vector Tiles (PBF format and rendered on the client-side based on a style file that is delivered with the Vector Tiles. The "Creating Vector Tiles in ArcGIS Pro" Training needs an ESRI organizational account	Tested successfully
24	ArcGIS Online & Portal for ArcGIS	Server	Supports serving Vector Tiles and rendering in the mapping application powered by the ArcGIS API for JavaScript. A Vector Tile service is an ArcGIS Server web service originating from a Vector Tile package in ArcGIS Pro. Vector Tile services (also known as Vector Tile layers) enable to share and consume Vector Tiles in Portal for ArcGIS and in custom applications.	ArcGIS online [http://www.esri.com/software/arcgis/arcgisonline]; ArcGIS Enterprise [http://www.esri.com/en/arcgis/products/arcgis-enterprise/overview]; ArcGIS Vector Tile Service [http://enterprise.arcgis.com/en/server/latest/publish-services/linux/vector-tile-services.htm]	All Vector Tile services begin inside ArcGIS Pro, where you author a Vector Tile package. The Vector Tile services have been tested successfully in ArcGIS online	Tested successfully

25	ESRI Vector Basemap Style Editor	Application	A styling app that allows you to quickly and easily turn on and off feature layers, find features by colors, apply bulk color changes for all features with that specific color, and more	ESRI-VectorBasemapStyleEditor [https://maps.esri.com/jg/VectorBasemapStyleEditor/index.html]; Github [https://github.com/jgrayson-apl/VectorBasemapStyleEditor]; ESRI-Blog [https://blogs.esri.com/esri/arcgis/2016/07/11/65959/]	The tool is a beta and was not officialy realeased as a product by ESRI	Tested successfully
26	ESRI Vector Style JSON Editor	Application	Styling app that exposes the json code in one half of the window, and the map in the other half.	Github [http://esri.github.io/arcgis-vectortile-style-editor/]; ESRI-Blog [https://blogs.esri.com /esri/arcgis/2016/07/11/65959/]	The tool is still a beta	Tested successfully
27	GeoServer with Vector Tile extensions	Server	GeoServer is a Java based open source server software that allows users to edit and share geospatial data and uses open standards to publish GIS data. The latest release of GeoServer adds support for creating Vector Tiles in GeoJSON, TopoJSON, and MapBox Vector Tiles format through its WMS service.	GeoServer [http://docs.geoserver.org/latest/en/user/ extensions/vectortiles/tutorial.html]; YouTube [https://www.youtube.com/watch?v=xdc67aZVO7E]		Tested successfully
28	QGIS Vector Tile reader plugin	Client	QGIS Python plugin which reads Mapbox Vector Tiles from local MBTiles file or remote format.	GitHub [https://github.com/geometalab/ Vector-Tiles-Reader-QGIS-Plugin]; Geometa Lab HSR [https://giswiki.hsr.ch/ Vector-Tiles-Reader-QGIS-Plugin]		Tested successfully

TABLE 6.2: Overview of Vector Tile tools that are based on WebGIS extended with Vector Tiles

6.2.3 Database-rendering

The last category concerns Vector Tile tools that are based on database-rendering. This means that spatial databases can store an amount of geographic data or contain spatial features that are afterwards used to be rendered in a client. Different options exist to serve spatial features from a spatial database. Table 6.2.3 gives an overview of different Vector Tile tools when it comes to database-rendering.

ID	Name	Type of tool	Description	Links (click on text to open in browser)	Comments	Tested?
29	PostGIS as a Vector Tile generator	Parsers & Generators	PostGIS can be used to generate Vector Tiles	ZimmiPosts [https://www.zimmi.cz/posts/2017/ postgis-as-a-mapbox-vector-tiles-generator/]; Medium [https://medium.com/tantotanto/vector-tiles-postgis-and-openlayers-258a3b0ce4b6]; OpenMapTiles [https://openmaptiles.org/docs/generate/ custom-vector-from-postgis/]; Carto blog [https://carto.com/blog/inside/ MVT-mapnik-vs-postgis/]	Not tested because it was not a priorirty and due to a personal lack of knowledge about PostGIS	Not tested
30	Tegola	Server	Vector Tile server written in Go (MVT)	Tegola [http://tegola.io/]	Followed these steps: [http://tegola.io/getting-started/]	Tested successfully
31	Tippecanoe (with ogr2ogr if needed)	Command-line utility	Builds Vector Tilesets from large (or small) collections of GeoJSON, Geobuf, or CSV features. Ogr2ogr converts simple features data between file formats (e.g. Shapefile to GeoJSON).	Github [https://github.com/mapbox/tippecanoe]; Github Maptime [https://github.com/maptime-ams/vector-tiles-workshop] Mapbox [https://www.mapbox.com/help/large-data-tippecanoe/]; Pedro Sousa Blog [http://build-failed .blogspot.nl/2017/03/playing-with-mapbox- vector-tiles-part-2.html]		Tested successfully

32	T-rex	Server	Vector Tile server specialized on publishing MVT tiles from your own data	t-rex [http://t-rex.tileserver.ch/]; Github [https://github.com/t-rex-tileserver/t-rex]; Devhub [https://devhub.io/repos/pka-t-rexpy]	Works very straightforward on windows. Good documentation compared to others. Mbutil (python module to convert to mbtiles)	Tested successfully

TABLE 6.3: Overview of Vector Tile tools that can be place under the Database-rendering environment

6.3 Overview of Vector Tile solutions or workflows

Before giving an overview of different Vector Tile solutions or workflows, it is important to make a distinction between a Vector Tile tool and a Vector Tile solution. A Vector Tile tool is an implement such as a parser/generator, application/plugin, command line utility, server or client. These tools have been listed in the inventory above. A Vector Tile solution is considered to be a way to implement these Vector Tile tools. A Vector Tile workflow is considered the same as a Vector Tile solution, however, there is an emphasis on the set-up or different process steps needed to realize a certain goal with Vector Tiles. It all depends on what you want to do with Vector Tiles. It the only goal is to style Vector Tiles (e.g. by using Mapbox Studio), then a single Vector Tile tool could suffice as a solution. However, if you want to do the whole stack of generating, serving and rendering Vector Tiles than you would need several steps to realize this goal and it would be better to talk about a Vector Tile workflow. For this reason, it is also vital to define different uses and use cases with Vector Tiles, which is discussed below.

6.3.1 Defining different uses and use cases of Vector Tiles

There are several uses or use cases possible when implementing Vector Tiles. Uses and use cases should be seen here as possibly steps or components in the Vector Tile workflow and not as examples for which Vector Tiles are used. In other words, it tells something about the list of actions or event steps that can be the case when achieving a goal with Vector Tiles. The identified different use cases with Vector Tiles are briefly listed below:

1. *Downloading:*
   (a) Downloading pre-generated Vector Tiles
   (b) Downloading geo-data to use to generate Vector Tiles
2. *Generating:* Creating own geo-data or creating Vector Tiles from own geo-data or from downloaded geo-data
3. *Serving:* Publishing or serving Vector Tiles (with a Tile Server or without by means of a Web Server)
4. *Rendering* Visualizing Vector Tiles with a client
5. *Styling:* Customizing visualizations with Vector Tiles

The Styling part could also be before the Rendering part. However, this sequence was chosen by the author in order to highlight the advantage of Vector Tiles to change map styles once the rendering on the client side has been done. This is different than with raster tiles where the styles have to be made before serving them to the client. Another argument is that a cartographer would first want to see how the data looks like, once they are in Vector Tiles, before applying different styles to them.

Figure 6.1 shows different uses or use cases possible with Vector Tiles. Some examples of solutions are given for each type of use. As can be observed, it is possible to have a variety of scenarios or use cases, for instance you could (1) only download pre-generated Vector Tiles, (2) download pre-generated Vector Tiles and render/visualize them directly without styling, (3) download Vector Tiles to serve them and style them or (4) you could bake your own Vector Tiles, serve them, visualize them and eventually even style them, etc. Summing up all the possible use cases is a difficult and unclear task and not the aim here. The aim is to give a clear overview of some obvious solutions or workflows. The next subsection will propose some obvious and relevant solutions or workflows based on possible uses and use cases. It should be noted that some solutions are open source whereas others commercial. The intention is not to promote any but to give an overview of current working options.

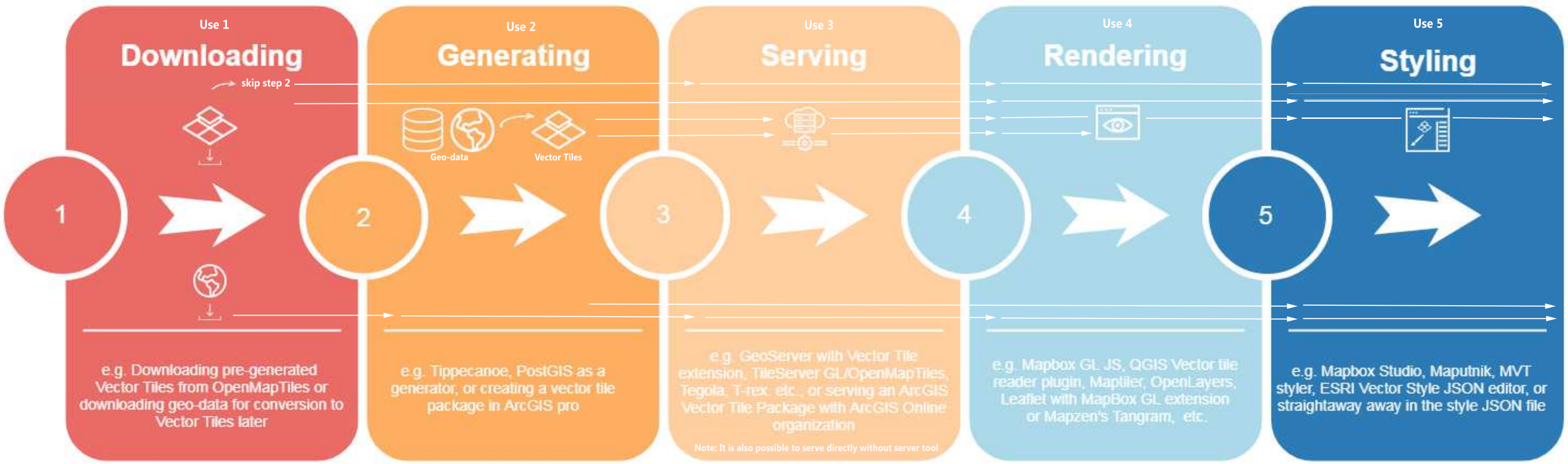

FIGURE 6.1: Different possible uses and use cases with Vector Tiles

6.3.2 Identifying different Vector Tile solutions or workflows

Different possible Vector Tile solutions or workflows can be identified in practice. It is not possible to list them all. Therefore, a selection has been made by the author. The listed workflows in this subsection are taking into account the different possible uses with Vector Tiles (Figure 6.2). Examples of workflows are given based on what was found most recognizable or straightforward by the author. Even though the selection is to some extent arbitrary, which is a point discussed in the research limitations of this thesis (Chapter 9.3), one could also say that this helps the Web Cartographer who aims at migrating to Vector Tiles to see which are the most appropriate solutions currently possible. Since it is not always easy to keep up with all the different Vector Tile tools or solutions (Boeijen, 2018b) or Web Mapping technologies (Cartwright et al., 2010; Muehlenhaus, 2014; Roth, Donohue, et al., 2014), it helps to have some proposed Vector Tile solutions or workflows:

	Use	W1	W2	W3	W4	W5	W6	W7	W8
1	Downloading	✓	✓	✓		✓	✓	✓	✓
2	Generating			✓	✓	✓	✓	✓	✓
3	Serving			✓	✓	✓	✓	✓	✓
4	Rendering	✓	✓	✓	✓	✓	✓	✓	✓
5	Styling	✓	✓	✓	✓	✓	✓	✓	✓

FIGURE 6.2: Uses cases of Vector Tiles for each workflow

- **Workflow 1: OpenMapTiles → QGIS Vector Tile Reader plugin (*Downloading pre-generated Vector Tiles, rendering them directly and eventually styling them*)**

  A first possible workflow is to download pre-generated Vector Tiles (e.g. at OpenMapTiles), render them directly (e.g. In QGIS with Vector Tile Reader plugin) and eventually style them (e.g. in QGIS or by using a JSON style file) (Figure: 6.3). This workflow is probably only an appropriate solution for a Web Cartographer if the use case is to discover Vector Tiles and how they work.

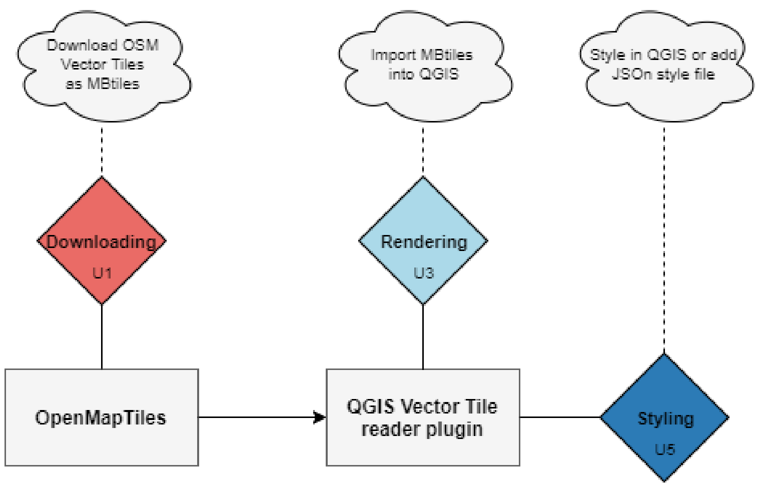

FIGURE 6.3: Workflow 1: OpenMapTiles → QGIS Vector Tile Reader plugin

OpenMapTiles
OpenMapTiles could in this workflow be used to download pre-generated Vector Tiles from OpenStreetMap data. The pre-generated Vector Tiles can be downloaded as MBTiles (Figure: 6.4). It should be noted that this solution is only free if the map tiles are for an open source project or open data project, non-commercial personal object, evaluation or education purpose. For commercial or company web it is a paid solution (OpenMapTiles, 2018c).

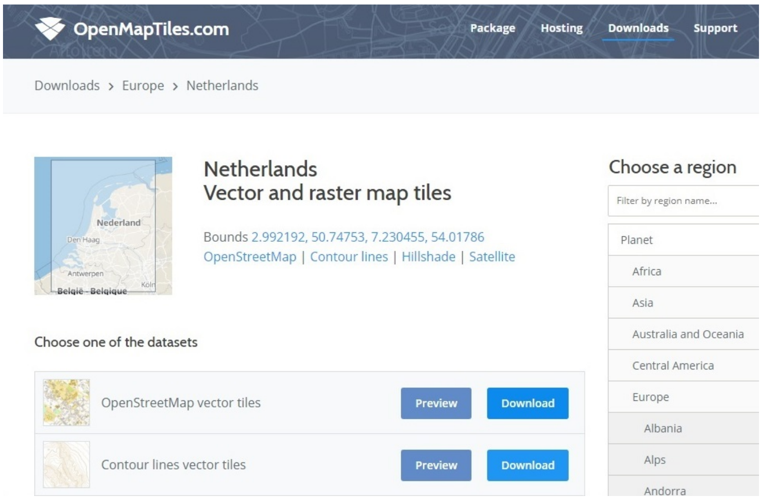

FIGURE 6.4: Example of downloading OSM The Netherlands data in MBTiles from OpenMapTiles (OpenMapTiles, 2018c)

QGIS Vector Tile Reader plugin
The QGIS Vector Tile Reader plugin could in this workflow be used to import the downloaded MBTiles into the QGIS client (Figure: 6.5). Once rendered it is possible to change the visualization or to change styles by adding a JSON style file.

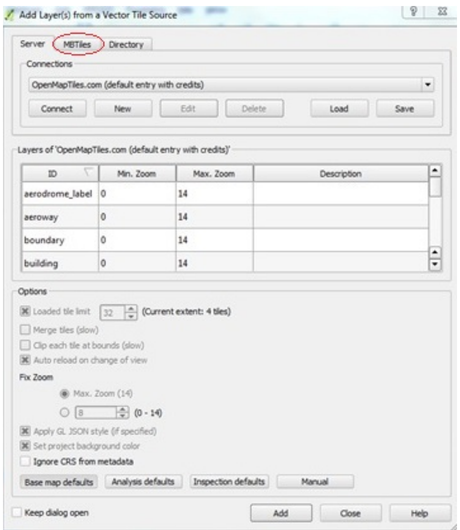

FIGURE 6.5: Adding a Vector Tile layer (.mbtiles) into QGIS with the Vector Tile reader plugin

Handy links (Date accessed: 23-02-2018):

- https://openmaptiles.com/downloads/planet/
- https://github.com/geometalab/Vector-Tiles-Reader-QGIS-Plugin
- https://giswiki.hsr.ch/Vector_Tiles_Reader_QGIS_Plugin
- https://github.com/mapbox/mbtiles-spec

- **Workflow 2: OpenMapTiles → Mapbox Studio (*Downloading pre-generated Vector Tiles, rendering and styling them directly without serving them*)**

Similar to workflow 1, one could download pre-generated Vector Tiles (e.g. MBTiles from OpenMaptiles). Afterwards, another rendering and styling solution can be implemented in the workflow. In this second workflow the example of Mapbox Studio was taken for rendering and styling Vector Tiles. Even though it is possible to host the styled Vector Tiles afterwards with Mapbox solutions and use Mapbox GL JS to publish and render a map in the Web, it was not taken into account in this workflow (Figure: 6.6).

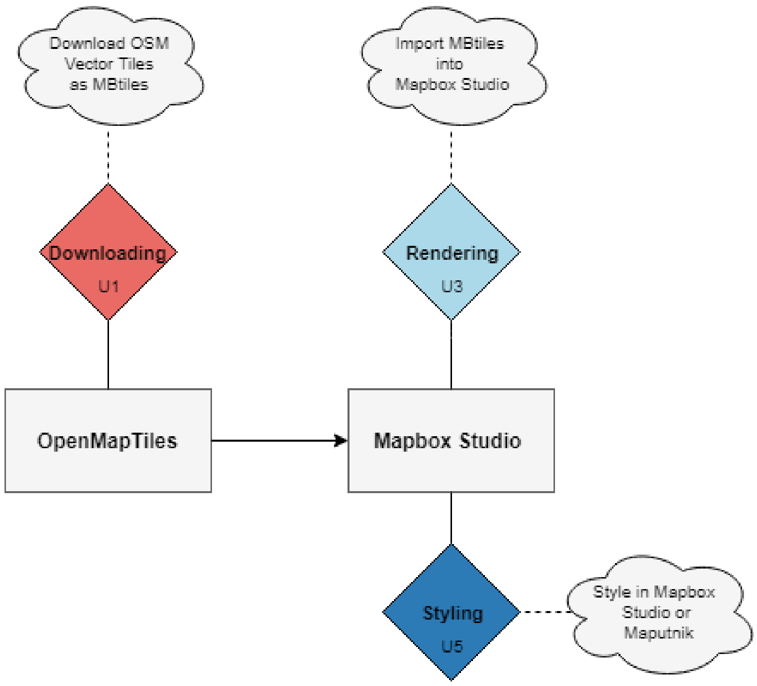

FIGURE 6.6: Workflow 2: OpenMapTiles → Mapbox Studio

OpenMapTiles
OpenMapTiles could in this workflow be used to download pre-generated Vector Tiles from OpenStreetMap data. The pre-generated Vector Tiles can be downloaded as MBtiles (Figure: 6.4). It should be noted that this solution is only free if the map tiles are for an open source project or open data project, non-commercial personal object, evaluation or education purpose. For commercial or company web it is a paid solution (OpenMapTiles, 2018c).

Mapbox Studio
One obvious choice for rendering and styling the Vector Tiles is using Mapbox Studio, an online service where you can load data, style and publish maps (Figure: 6.7). Mapbox provides various price ranges, with a non-commercial free plan that allows 50.000 map views per month, 50GB tileset storage and 5GB dataset storage, which is a good solution for new Web Cartographers who want to test out Vector Tiles (Mapbox, 2018b; Sousa, Pedro, 2017). Maputnik is an open source alternative to Mapbox Studio and could be used instead of Mapbox Studio.

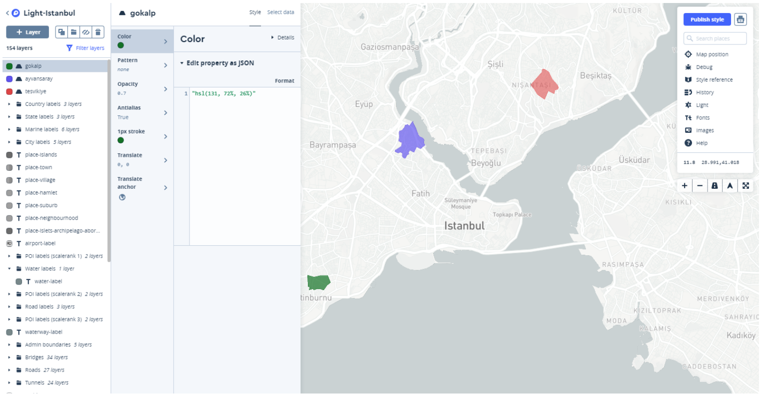

FIGURE 6.7: Example of styling MBTiles in Mapbox Studio

Handy links (Date accessed: 24-02-2018)::

- https://openmaptiles.com/downloads/planet/
- https://www.mapbox.com/mapbox-studio/
- http://build-failed.blogspot.nl/2017/02/playing-with-mapbox-vector-tiles-part-1.html
- https://maputnik.github.io/editor/

- **Workflow 3: Geofabrik → Tilemaker → TileServer GL → Mapbox GL JS (*Downloading geo-data, converting them to Vector Tiles, serving them, rendering them and eventually styling them*)**

The third workflow assumes that the user does not want to use his own data but prefers to download geographic data for conversion to Vector Tiles (Figure: 6.8). This should not be confused with downloading pre-generated Vector Tiles. In this example, the user downloads an .osm.pbf planet extract, as typically downloaded from providers like Geofabrik. This OpenStreetMap data can be converted with the Tilemaker tool to an output MBtiles file. The MBTiles can be served with TileServer GL from Klokan Technologies (or without tile server) and rendered directly with a client such as Mapbox GL JS. A JSON style file can be used to customize the map visualization if needed. This can be done before the rendering on the client, or after.

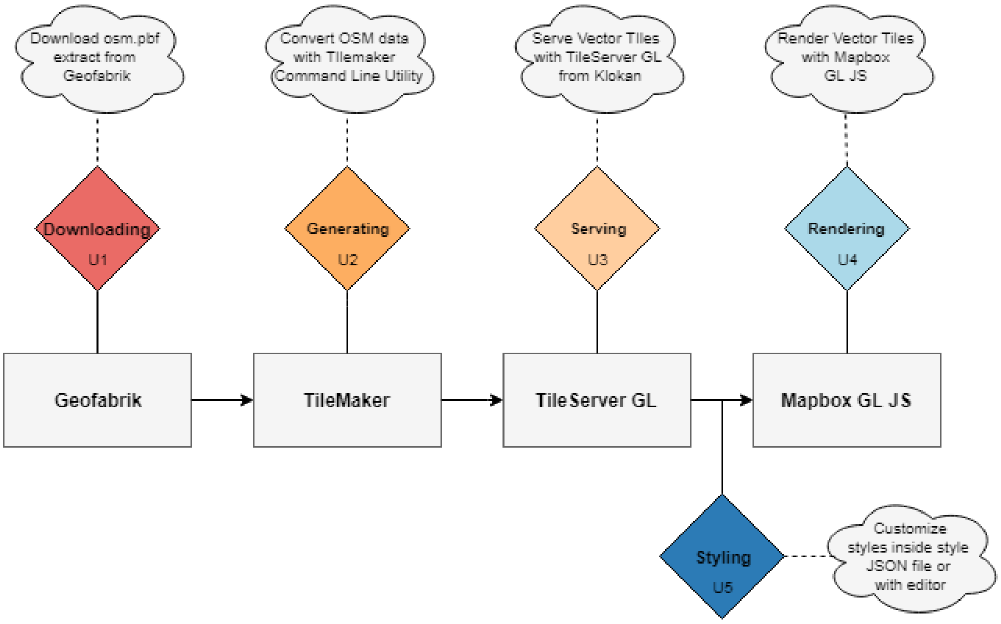

FIGURE 6.8: Workflow 3: Geofabrik → Tilemaker → TileServer GL → Mapbox GL JS

Geofabrik
In this workflow, Geobabrik's free download server is used to download data extracts (.osm.pbf) from the OpenStreetMap project (Figure 6.9).

FIGURE 6.9: Example of downloading OSM extract data from Geofabrik download server (Geofabrik, 2018)

TileMaker
Tilemaker is a solution to create Vector Tiles (in MVT format) from an .osm.pbf planet extract. It aims to be 'stack-free': no need for a database and there is only one executable to install. It is possible to have an output as individual files, or to a MBTiles. (Github Tilemaker, 2018).

TileServer GL
Once the Vector Tiles are generated, a tile-server could be used to serve the vector tiles via HTTP to the client. This is not necessary though because you could also serve Vector Tiles directly without tile-server, but in this workflow the example of TileServer GL has been taken. TileServer GL can be installed

via Docker or via NPM. There is also a TileServer-gl-light version. When configuring TileServer GL this can be done in a regular JSON configuration file. It defines the behavior of the application and the paths to the data and styles. The styles can be customized in a JSON style file which has to be added as a new style to the TileServer configuration file. TileServer GL enables to have tile URL specifying the location of the vector tiles data for later use in a client such as MapboxGL or OpenLayers.

Mapbox GL JS
Mapbox GL JS is a JavaScript library that uses WebGL to render interactive maps from vector tiles and Mapbox styles (Mapbox, 2017a). In this workflow, the generated and served Vector Tiles can be viewed with TileServer GL directly with MapboxGL. It is also possible to create an index.html file in which you define the tile URL and path to the style according to the Mapbox Vector Tile Specification. The integration of Mapbox GL JS with TileServer GL is very well and therefore this solution is most straightforward.

Handy links (Date accessed: 23-02-2018):

- https://www.geofabrik.de/data/download.html
- https://github.com/systemed/tilemaker/blob/master/README.md
- https://github.com/klokantech/tileserver-gl
- http://build-failed.blogspot.nl/2017/02/playing-with-mapbox-vector-tiles-part-1.html
- https://openmaptiles.org/docs/host/tileserver-gl/
- https://www.youtube.com/watch?v=rOg4VnSAnI4
- https://tileserver.readthedocs.io/en/latest/usage.html
- https://www.mapbox.com/mapbox-gl-js/style-spec
- https://www.mapbox.com/mapbox-gl-js/api

- **Workflow 4: GeoJSON.io → Tippecanoe → TileServer GL → OpenLayers (*Creating own geo-data, generating Vector Tiles, serving them, rendering and styling them*)**

  Workflow 4 shows an example of creating own geo-data for use with Vector Tiles. It is possible to create your own data in a Shapefile with QGIS and export is as GeoJSON. In this workflow and example was given of using geojson.io as a tool to create geo-data as GeoJSON directly, without the need to convert geodata using ogr2ogr for instance. The output GeoJSON file can be converted with the Tippecanoe command line utility to an output MBTiles file. The MBTiles can be served with TileServer GL from Klokan Technologies (or without tile server) and rendered directly by means of a client such as OpenLayers. A JSON style file can be used to customize the map visualization if needed. This can be done before the rendering, or after (Figure: 6.10).

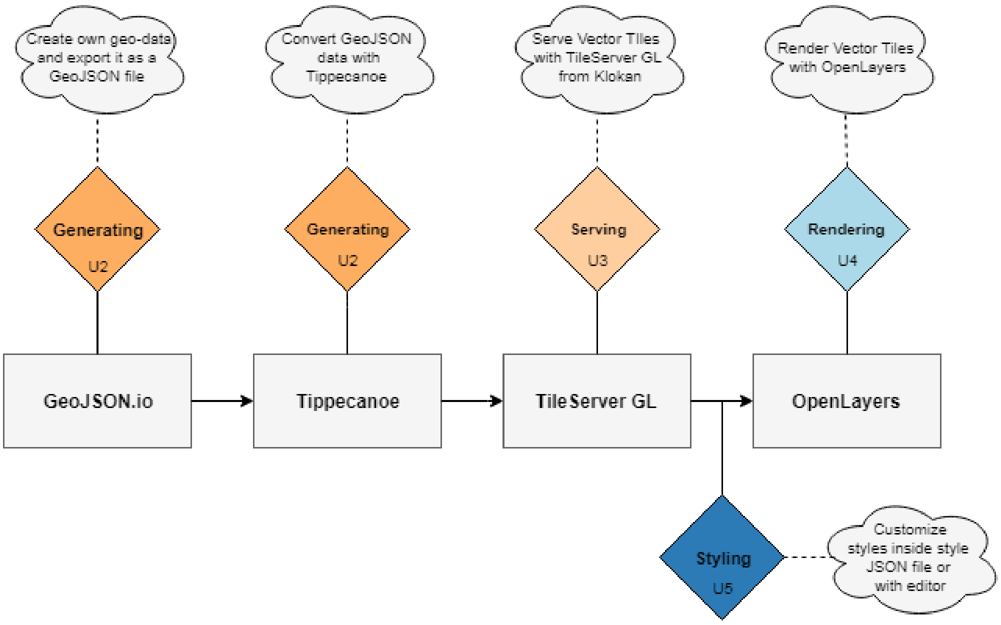

FIGURE 6.10: Workflow 4: GeoJSON.io → Tippecanoe → TileServer GL → OpenLayers

GeoJSON.io
GeoJSON.io is an open source project and website to create, change, and publish maps (Figure: 6.11). With GeoJSON.io shapes can be drawn in the browser and exported as a GeoJSON file. It is a very handy tool to easily and quickly create your own geo-data. In this workflow, GeoJSON.io is used to export a GeoJSON file for later conversion to Vector Tiles.

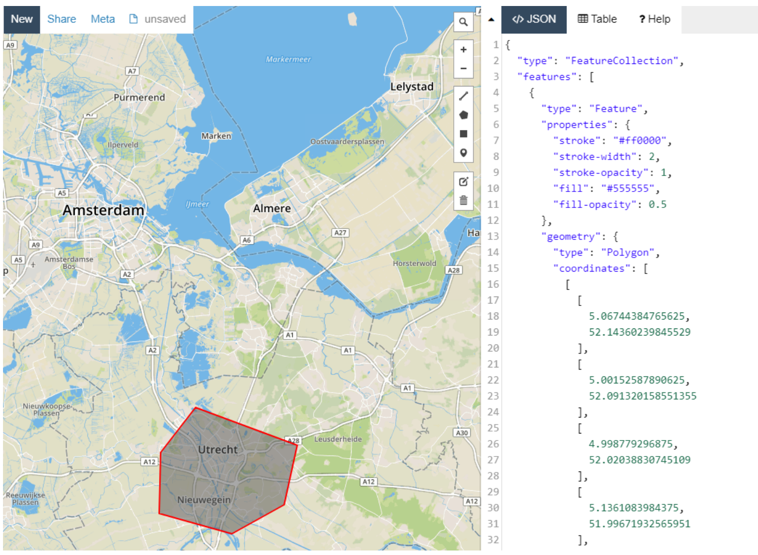

FIGURE 6.11: Example of creating own geo-data with GeoJSON.io tool

Tippecanoe
Tippecanoe makes it possible to create your own tiles with an GeoJSON as an input file. It is a Command-line utility for generating Vector Tiles. It is

available in Docker. The output is an MBTiles file or it can export Vector Tiles individually in a file map structure on a server directory. In this workflow example, the Vector Tiles are exported as MBTiles.

TileServer GL
Once the Vector Tiles are generated, a-tile server could be used to serve the vector tiles via HTTP to the client. This is not necessary though because you could also serve Vector Tiles directly without tile-server, but in this workflow the example of TileServer GL has been taken. TileServer GL can be installed via Docker or via . There is also a TileServer-gl-light version. When configuring TileServer GL this can be done in a regular JSON configuration file. It defines the behavior of the application and the paths to the data and styles. The styles can be customized in a JSON style file which has to be added as a new style to the TileServer configuration file. TileServer GL enables to have tile URL specifying the location of the vector tiles data for later use in a client such as MapboxGL or OpenLayers.

OpenLayers
OpenLayers is an open source JavaScript library. Vector Tiles can be displayed in OpenLayers. The ol-mapbox-style plugin which can be imported as a module or added in the HTML file should be used to work with Mapbox Style objects in this workflow.

Handy links (Date accessed: 23-02-2018):

- http://geojson.io/
- https://github.com/mapbox/tippecanoe
- http://build-failed.blogspot.nl/2017/03/playing-with-mapbox-vector-tiles-part-2.html
- https://www.mapbox.com/help/large-data-tippecanoe/
- https://github.com/maptime-ams/vector-tiles-workshop/wiki/Optional-making-your-own-tiles
- https://github.com/boundlessgeo/ol-mapbox-style
- https://openmaptiles.org/docs/website/openlayers/
- https://boundlessgeo.com/2017/01/using-mapbox-style-objects-open-layers/
- https://openlayers.org/workshop/en/vectortile/ugly.html

- **Workflow 5: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS (*Downloading geo-data, generating Vector Tiles, serving them without tile server, styling and rendering them*)**

Workflow 5 shows the stack used by Webmapper together with PDOK (Boeijen, 2018b, 2018c; Hogeboom, 2018; Mac Gillavry, 2018b) to create Vector Tiles for the BRT and BGT Vector Tile project. Everything is done directly on the server in this workflow. The first step is to download data and import it into a PostGIS database with ogr2ogr. Once the geo-data are imported the attributes can be edited or the data can be simplified. For the Webmapper case the data was also transformed to EPSG:4326 (WGS84). After being finished with the data, it should be exported to GeoJSON with ogr2gr in order to use it with

Tippecanoe. Tippecanoe converts the GeoJSON features to individual .pbf files in the server's directory. These can be served directly with the Web Server without need of a tile server. The final step is to create a map with MapboxGL JS and to write a style file in JSON. This JSON style file can be customized afterward straight in the code or with a style editor such as Maputnik. MapboxGL JS does the rendering part as a client (Figure: 6.12).

Niene Boeijen (2018c) describes the following steps that are all done directly on the server:

1. Downloading data
2. Important data in Database [ogr2ogr]
3. Editing data [PostGIS, SQL]
4. Removing attributes
5. Transform to EPSG:4326
6. Simplify data if needed. This is most convenient to do it in the data itself in order to have more control on it.
7. Export data to GeoJSON [ogr2org]
8. Generate Vector TIles with Tippecanoe. Result: Tile cache on the server in a directory structure [Tippecanoe]
9. Create map with Mapbox GL JS and write the style JSON [HTML CSS JS JSON Mapbox GL JS].

If needed steps 3 to 6 can be re-done to optimize the data and the Vector Tiles.

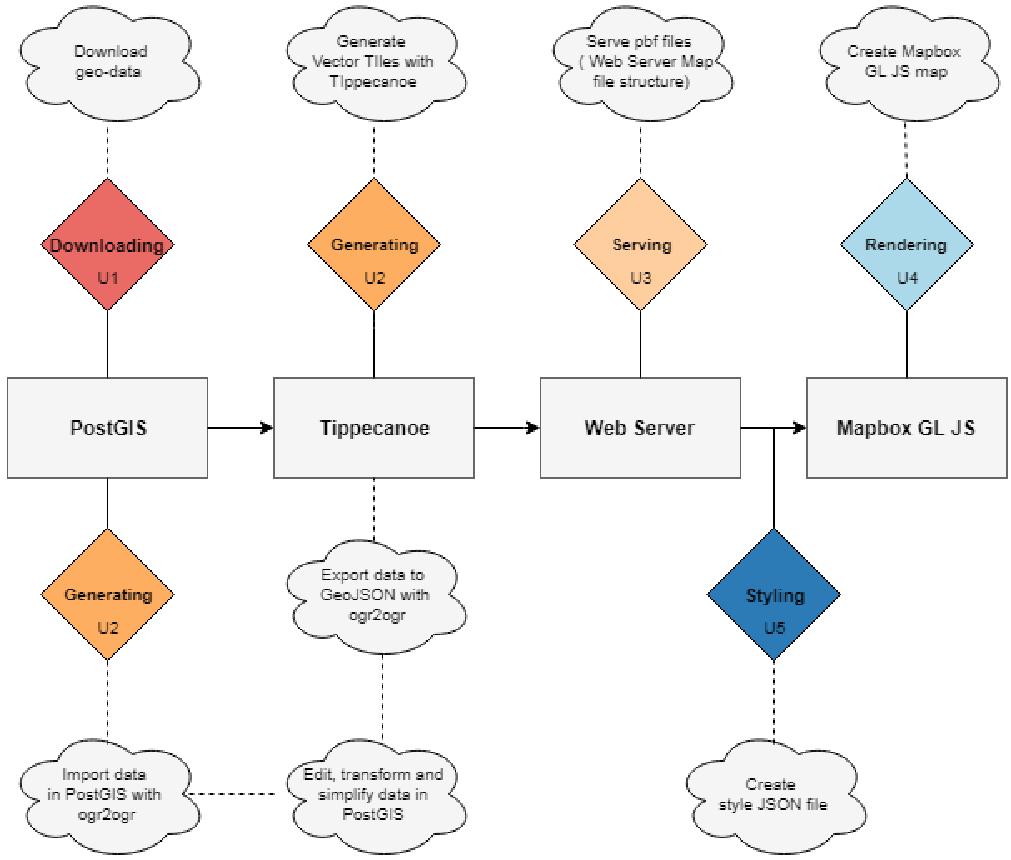

FIGURE 6.12: Workflow 5: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS (Webmapper - Tippecanoe solution)

PostGIS
PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL database. It works with SQL and follows specification from the Open Geospatial Consortium (OGC). It is possible to generate vector tiles directly from PostGIS. However, it was considered to leave this example out due to a lack of expertise by the author to work with the PostGIS Vector Tile extension. PostGIS is used in this workflow as a database where to edit data attributes and to simplify the data. Ogr2ogr was used to import and export the data from the Database.

Tippecanoe
Tippecanoe makes it possible to create your own tiles with a GeoJSON as an input file. It is a Command-line utility for generating Vector Tiles. It is available in Docker. The output is an MBTiles file, or it can export Vector Tiles individually in a file map structure on a server directory. In this workflow example, the Vector Tiles are exported as individual pbf files on the Server.

Serving the tiles with a Web Server without a tile-server
In this workflow, the generated tiles with Tippecanoe are served without the need of a tile server directly from the Web Server. This is a fast solution and

there are some examples on how to do it on the Web (see References below).

Mapbox GL JS
Mapbox GL JS is a JavaScript library that uses WebGL to render interactive maps from vector tiles and Mapbox styles (Mapbox, 2017a). In this workflow, a map is created with MapboxGL JS. It is also possible to create an index.html file in which you define the tile URL and path to the style according to the Mapbox Vector Tile Specification. The integration of Mapbox GL JS with Mapbox Vector Tiles is the most straightforward solution.

Handy links (Date accessed: 23-02-2018):

- https://github.com/PDOK/vectortiles-bgt-brt
- https://github.com/mapbox/tippecanoe
- http://build-failed.blogspot.nl/2017/02/playing-with-mapbox-vector-tiles-part-1.html
- http://build-failed.blogspot.nl/2017/03/playing-with-mapbox-vector-tiles-part-2.html
- https://www.mapbox.com/help/large-data-tippecanoe/
- https://github.com/maptime-ams/vector-tiles-workshop/wiki/Optional-making-your-own-tiles
- https://postgis.net/documentation/
- https://github.com/klokantech/mapbox-gl-js-offline-example
- https://www.mapbox.com/mapbox-gl-js/api/
- https://forum.pdok.nl/t/vector-tiles-brt-en-bgt-via-pdok/1103/14
- https://maputnik.github.io/editor/12.24/47.37181/8.54807
- https://geovation.github.io/build-your-own-static-vector-tile-pipeline

- **Workflow 6: PostGIS → GeoServer → GeoServer → OpenLayers (*Downloading geo-data, generating Vector Tiles, serving them, rendering them and eventually styling them*)**

Workflows 6 is the Vector Tile solution that uses GeoServer with Vector Tile extension as a Vector Tile generator and tile-server (Figure: 6.13). Like the previous workflow, the geo-data were first downloaded and imported into the PostGIS database. Afterwards, the data were directly imported into GeoServer tool without the need to convert. GeoServer generates the Mapbox Vector Tiles.

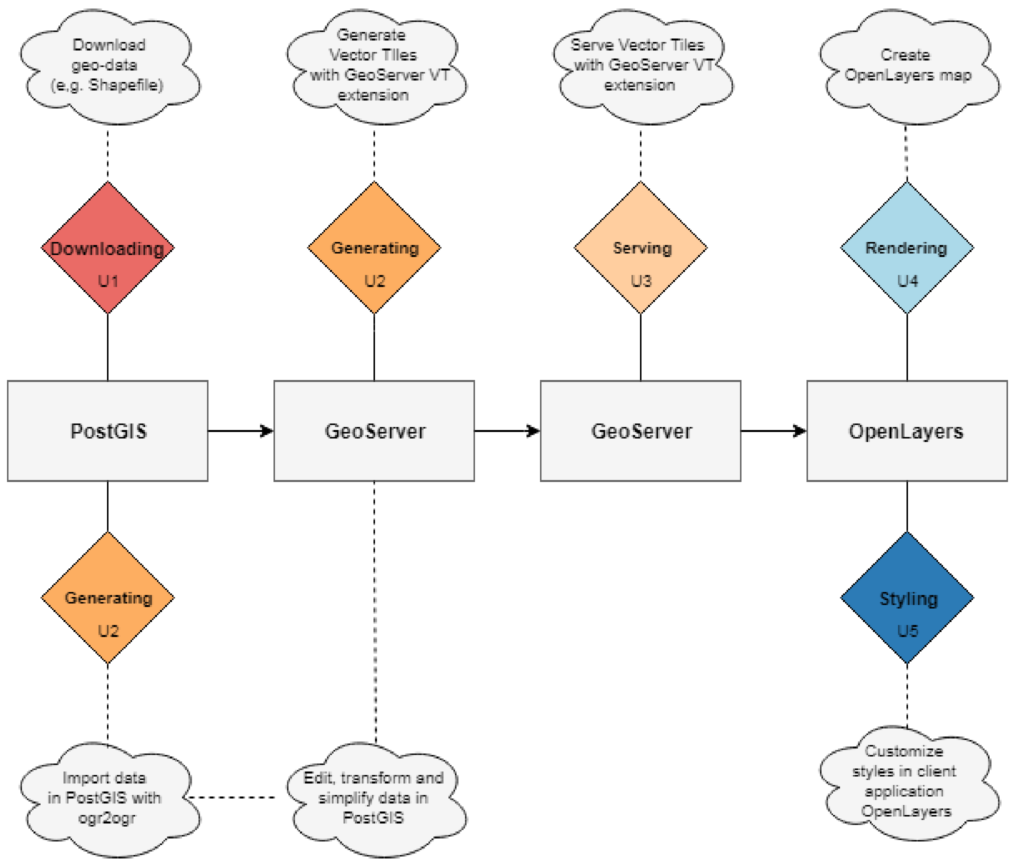

FIGURE 6.13: Workflow 6: PostGIS → GeoServer → GeoServer → OpenLayers (Geoserver solution)

PostGIS
PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL database. It works with SQL and follows specification from the Open Geospatial Consortium (OGC). It is possible to generate vector tiles directly from PostGIS, however it was considered to leave this example out due to a lack of expertise by the author to work with the PostGIS Vector Tile extension. PostGIS is used in this workflow as a database where to edit data attributes and to simplify the data. Ogr2ogr was used to import the data from the Database.

GeoServer
In this Workflow, GeoServer is both used to generate Vector Tiles as well as to serve Vector Tiles. First, the imported data (e.g. Shapefile) or data from PostGIS database has to be published through GeoWebCache. With the Tile-Caching a tile format has to be defined. It is possible to choose out of GeoJSON TopoJSON and MVT. MVT is the recommended format.

OpenLayers
After generating and serving the Vector Tiles, OpenLayers can be added inside the GeoServer Data Directory to create a map and directly apply style changes in the OpenLayers client application. OpenLayers is an open source JavaScript library. Vector Tiles can be displayed in OpenLayers and the ol-mapbox-style

plugin, which can be imported as a module or added in the HTML file, can be used to work with Mapbox Style objects (Styles written in Mapbox GL JSON).

Handy links (Date accessed: 24-02-2018):

- http://docs.geoserver.org/latest/en/user/extensions/vectortiles/tutorial.html
- https://postgis.net/documentation/
- https://openlayers.org/workshop/en/vectortile/ugly.html
- https://boundlessgeo.com/2017/01/using-mapbox-style-objects-open-layers/
- https://github.com/boundlessgeo/ol-mapbox-style
- https://www.youtube.com/watch?v=xdc67aZVO7E

- **Workflow 7: PostGIS → t-rex/ Tegola → OpenLayers/MapboxGL (*Downloading geo-data, generate Vector Tiles, serve them with server tool, render them and eventually style them*)**

This workflow is somehow similar to the previous workflow because it uses tools to both generate and serve Vector Tiles by means of one solution. One solution in this workflow could be t-rex either Tegola, which are in some way similar. They enable to generate Vector Tiles from data in the database and serve them as MVT. Afterwards, styles can be customized with the client application or written in the Mapbox GL JSON format which is used by viewers such as Mapbox GL and Maputnik (Figure: 6.14).

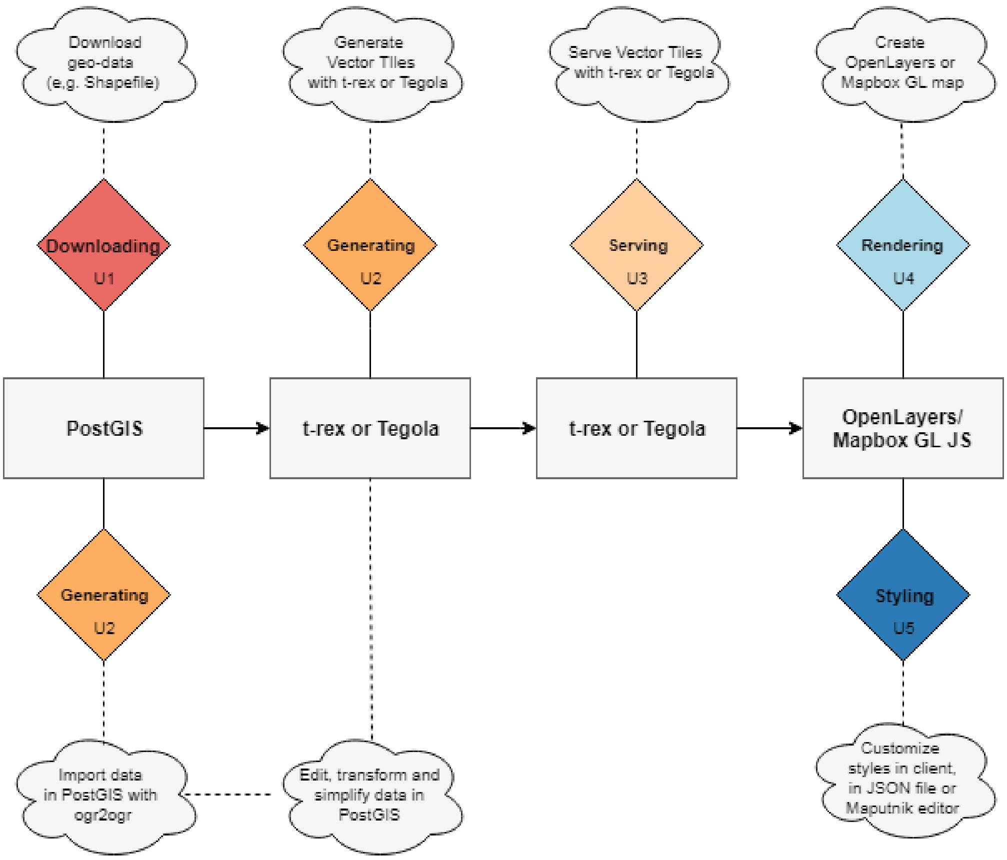

FIGURE 6.14: Workflow 7: PostGIS → t-rex/ Tegola → OpenLayers/MapboxGL

PostGIS
PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL database. It works with SQL and follows specification from the Open Geospatial Consortium (OGC). It is possible to generate vector tiles directly from PostGIS, however, it was considered to leave this example out due to a lack of expertise by the author to work with the PostGIS Vector Tile extension. PostGIS is used in this workflow as a database where to edit data attributes and to simplify the data. Ogr2ogr was used to import the data from the Database.

t-rex
In this Workflow t-rex can be used to generate Vector Tiles as well as to serve Vector Tiles (Figure: 6.15,) It is an open source tile-server written in Rust that publishes MVT tiles from own data such as data in a PostGIS database). It has a built-in web server, built-in caching, built-in viewer and has experimental support for embedded Mapbox GL styling according to the Mapbox Style Specification (TOML). Furthermore, it has support for custom tile grids, next to two built-in grids, Web Mercator and WGS84. The solution has been tested successfully by following the documentation, but has not been deployed further.

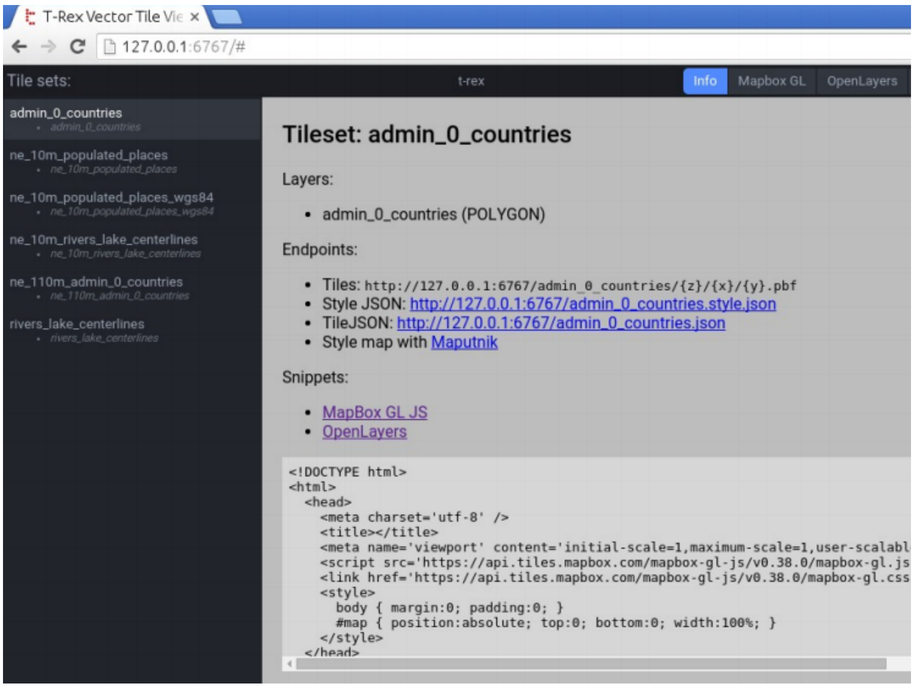

FIGURE 6.15: Example of t-rex as a generator and tile-server for Vector Tiles

Tegola
Another alternative for t-rex is Tegola, which is also an open source tile-server that publishes data from a PostGIS database to MVT tiles. It is written in GO.

It has an embedded viewer with auto-generated style for quick data visualization and inspection. Tegola has support for Web Mercator (3857) and WGS84 (4326) projections. The solution has been tested successfully by following the documentation, but has not been deployed further.

OpenLayers or Mapbox GL JS as a client
After generating and serving the Vector Tiles, OpenLayers or Mapbox GL JS can be used as a client renderer. For this workflow, styles can be written and customized in the Mapbox GL JSON format which is used by Mapbox GL viewers, Maputnik and others. It is possible to read these styles in OpenLayers with the ol-mapbox-style plugin.

Handy links (Date accessed: 24-02-2018):

- http://t-rex.tileserver.ch/t-rex_vector_tile_server.pdf
- http://tegola.io/
- https://postgis.net/documentation/
- https://openlayers.org/workshop/en/vectortile/ugly.html
- https://boundlessgeo.com/2017/01/using-mapbox-style-objects-open-layers/
- https://github.com/boundlessgeo/ol-mapbox-style
- http://t-rex.tileserver.ch/
- http://blog.sourcepole.ch/assets/2017/t-rex-foss4g2017.pdf

- **Workflow 8: ESRI ArcGIS Pro → ArcGIS Online (*Downloading geo-data, generate Vector Tiles Package, publish them on ArcGIS online organization, render them and eventually style them*)**

  The last proposed worfklow is very different from the other workflows because it uses the entire ESRI ecosystem, which is proprietary. Although ESRI's Vector Tiles are built based on the Mapbox Vector Tile specification, it is not open source. ArcGIS license and ArcGIS Online account is needed. Therefore, this workflow will only shortly be presented as an identified and possible solution. To create or to publish Vector Tiles in this workflow, the ESRI ArcGIS PRO and ArcGIS online stack was used (Figure: 6.16). ESRI's ArcGIS Pro is to be used in this workflow in order to create Vector Tiles in a Vector Tile Package that can be published to ArcGIS Online. Styles and fonts can be customized in ArcGIS pro before generating the Vector Tiles. Once published as a Vector Tile layer ESRI Vector Basemaps styles can be edited in the style JSON file or in the Vector Tile Style Editor. Vector Tile basemaps from ESRI online can be rendered in leaflet with the esri-leaflet-vector plugin.

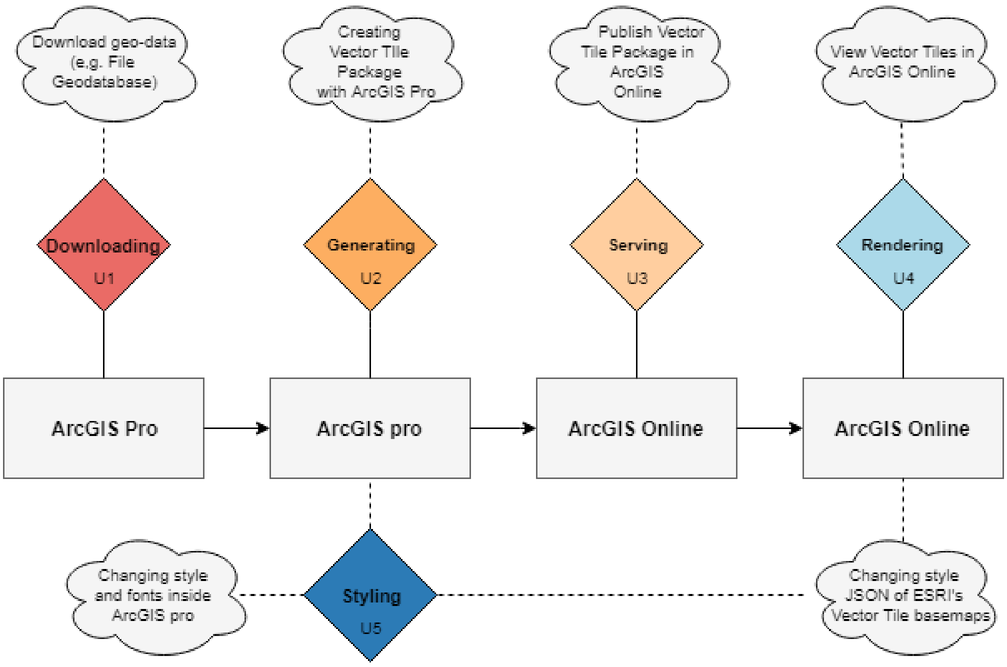

FIGURE 6.16: Workflow 8: ESRI ArcGIS Pro → ArcGIS Online

Handy links (Date accessed: 24-02-2018):

- https://www.esri.com/training/catalog/5851736fd33f8b0b47b78e26/creating-vector-tiles-in-arcgis-pro/
- https://community.esri.com/thread/173243
- http://pro.arcgis.com/en/pro-app/help/mapping/map-authoring/author-a-map-for-vector-tile-creation.htm
- https://github.com/Esri/arcgis-vectortile-style-editor
- https://github.com/jgrayson-apl/VectorBasemapStyleEditor
- https://maps.esri.com/jg/VectorBasemapStyleEditor/
- https://blogs.esri.com/esri/arcgis/2016/07/11/65959/
- https://github.com/Esri/esri-leaflet-vector

Chapter 7

Relating Vector Tile Technology to Web Cartography

This chapter presents the results of phase 4 of this research which consists of relating Vector Tile technology with Cartographic theory. The literature studies in Chapter 3 and Chapter 4 have provided the background of this research and serve as a starting point and input for the remainder of this thesis. In this chapter, the position and role of Vector Tiles for Web Cartography will be discussed by applying findings of the literature study and the expert-interviews. The inventory of different Vector Tile tools and solutions, conducted in the previous chapter, was used to assess the cartographic strengths and weaknesses of two Vector Tile workflows examples. It was investigated where the cartographic potential lies in the Vector Tile workflows and in Vector tile technology.

7.1 The position of Vector Tiles in Web Cartography

Vector Tiles are not entirely new in practice, they have already been used by the Google Maps Android client since December 2010 and on the desktop client since 2013 (Wikipedia, 2018c). Vector Tiles for rendering OpenStreetMap data with Mapnik renderer and Mapbox Vector Tiles were both launched in 2013 as well (Mapbox, 2018a; Wikipedia, 2018c). Even though the leaders in providing and implementing Vector Tiles have started already a few years ago, it is still an emerging solution for (web)cartography and not widely implemented or mainstream yet in the GIS-field. Besides, there is still relatively few theory or scientific literature regarding Vector Tile technology. The position of Vector Tiles in (web)cartography and GIS-environment can therefore still be considered as emerging. The reason for Vector Tile technology for still being in a developing stage, is that other Web Mapping technology such as raster tiles or web map services are much better developed. Professionals in the geo-field are therefore sometimes hard to convince to migrate to the emerging technology of Vector Tiles, since they already have a good working solution. This aspect was also mentioned by Wouter Visscher, software developer at the Dutch Cadastre, Land Registry and national mapping agency in the Netherlands. According to him, Vector Tiles are an effective product to distribute geographic information. However, the market is still hard to convince. Wouter Visscher (2018) states that: *"Vector Tiles are sometimes seen as the future, however, it already has been proved by major leaders such as Google Maps and Mapbox that Vector Tiles are not new and a working solution for visualizing spatial data. Working professionals stating that WFS or other Web Mapping solutions are sufficient for their needs, are people that do not possess sufficient knowledge about the vector tiling technique. The GIS community is divided and to bridge Vector Tile technology with other Web Mapping technologies is too challenging for those who believe Vector Tiles are too complicated"*. These sayings imply that there is a (technical) knowledge gap that prevents Vector Tiles from wide implementation. A link can be made with the literature where it was stated that Vector Tiles need a good understanding of Web Mapping technologies. According to Martinelli and Roth (2015, p. iv): *"Producing Vector Tiles requires a good understanding of map technologies and sufficient computing power. This is the reason why Vector Tiles aren't adopted by the mainstream yet"*. Low cartographic knowledge was also identified as a possible risk while creating a workflow to generate Vector Tiles from OpenStreetMap data (Martinelli & Roth, 2015). Moreover, vector maps suffered from other setbacks that prevented wide implementation. The large volume of vector data, the variety of formats with the disadvantages of vector encoding and the lack of standards have caused problems for the implementations of vector maps (Antoniou et al., 2009). Therefore, more research is needed. The advantages or challenges of Vector Tile technology are not always understood or known by the entire GIS or cartography community yet. It should also be mentioned that ESRI, as Leader in Global GIS Market, have only added the support for Vector Tiles in ArcGIS Server and ArcGIS Pro since February 2016 (Wikipedia, 2018c).

To understand the position of Vector Tiles in Web Cartography it is needed to understand the evolution of web maps. In Chapter 3 the influence of technology on Cartography was elaborated. It was discussed that maps have evolved from paper maps to digital maps. The first digital or web maps were static maps that can be distinguished from dynamic maps (Kraak & Brown, 2001; Mitchell, 2005). Maps using Vector Tiles solutions can be identified as dynamic web maps since Vector Tiles are characterized by their opportunities for more interactivity. With Web Mapping having gone through different technological changes in recent years, a history of web maps that can be identified. Martinelli and Roth (2015) describe the evolution of web maps as going from untiled static maps through raster tiles to Vector Tiles. They identified different phases:

- *Phase 1 - Untiled static maps*: In the beginning WMS servers generated static images for the viewport of the map.
- *Phase 2 - Raster tiles*: In 2005 Google introduced Google Maps and XYZ tiles which delivered a raster image for coordinates specified by a tile index.
- *Phase 2.5 - Raster tiles with vector overlays*: To provide a level of interactivity, tools like Leaflet support rendering vector data like SVG on top of a raster-based maps.
- *Phase 2.75 - Raster tiles from Vector Tiles*: For backwards compatibility and faster serving of raster tiles, Vector Tiles where introduced to avoid querying a database.
- *Phase 3 - Vector Tiles*: Vector Tiles are delivered directly to the browser and rendered by WebGL based clients.

Kalberer (2017) also notes that web maps have gone from static view images towards Vector Tiles. The author identifies three groups: (1) WMS which had no tiling problems (labels etc.); (2) WMTS which came with scalability and caching on the server and client side; and (3) Vector Tiles that offer scalability, caching, interactivity, flexible styling and Hi-DPI.

Furthermore, in order to understand the position of Vector Tiles in web cartography, it is helpful to identify which type of dynamic web maps, Vector Tile technology can be used for. Mac Gillavry (2018a) mentions that Vector Tiles can be used in combination with a variety of types of web maps. There are possibilities for extrusion, animations, thematic maps (eg. choropleth maps, chorochromatic maps, dot distribution maps etc.). The author proposes a classification of different types of web maps, based on three types of maps that were earlier used for identifying the three visual hierarchies for Web maps by Muehlenhaus (2014) (Figure 3.4). The three map types identified are:

- *General-Interest or Topographic Web Maps*: Vector Tiles are mainly used for general-purpose maps that can be for reference or orientation purposes or can show topographic features (points, lines and polygons) such as water, roads, railway lines, parks, boundaries etc. Therefore topographic and cadastral maps showing ownership boundaries, houses and parcel numbers can be considered as general-interest web maps because they give an understanding of location and features of an area. Numerous general purpose maps are also collaborative web maps where users collaborate to create and improve the Web Mapping experience (e.g. OpenStreetMap).
- *Thematic Web Maps*: Vector Tiles can also be used in combination with basic vector map data that visualize a particular theme connected with a specific geographic area. Thematic Web maps are in contrast to general reference maps, which show the variety of phenomena together. The thematic map focuses on a specific theme or geographical area (such as population density in The Netherlands). Thematic maps can be statistical or socio-economic maps with qualitative or quantitative data used (dot maps, dot density maps, choropleth maps, chorochromatic maps, isoline maps, diagram maps, flowline maps, cartogram maps, proportional point symbol maps (Ormeling & Kraak, 2010)). Some thematic maps are also analytical web maps that offer GIS analysis.
- *Animated Web Maps*: For some rare cases, Vector Tile technology can be used for animated maps. Animated web maps emphasize a spatial or temporal change of an occurrence (DiBiase, MacEachren, Krygier, & Reeves, 1992). Web maps with real-time animation can visualize for instance traffic congestion, weather or other monitoring. Non-temporal animated maps can be used for fly through animation, cartographic zoom animation, classification animation or generalization animation (Wikipedia, 2018a).

For the three different web map types, Vector Tile technology can be applied. However, with Mapbox being the leader in Vector Tiles and Mapbox Vector Tile Specification being the de-facto standard, most maps are based on visualizing basic vector map data such as points, lines and polygons for general purpose maps. Even though there are a lot of possibilities with Vector Tiles, it should be realized that rendering is slower with polygons or complicated vector data compared to point data. Not every type of vector data is fully optimized yet for use with Vector Tiles. Niene Boeijen, Web Cartographer at Webmapper, mentions that most maps coming from Mapbox Vector Tile Web Mapping are made of point data (Boeijen, 2018a). Point data is well optimized for use with Vector Tiles and understandable for everyone. In line with this, Rober Nordan from Norkart AS mentions that most Mapbox maps with Vector Tiles are meant for general public consumption and not specialist consumption. This also explains why Mapbox only uses WGS84 Mercator projection (Nordan, 2015). The question arises whether Vector Tiles can also easily be implemented with other cartographic projections. Furthermore, how to cope with generalization? What to take into account regarding map design with Vector Tiles? How important is the preservation of topology with Vector Tiles? These are examples of topics that should be studied while investigating the cartographic implications of Vector Tiles. In the next section, the role of Vector Tiles for (web)cartography will be discussed based on a combination of findings from the literature review and results from the expert-interviews.

7.2 The role of Vector Tiles for Web Cartographers

In order to understand the role of Vector Tiles for Web Cartographers, several experts involved with Vector Tiles in the computer sciences or cartography field have been interviewed. The expert interviews tackled several cartographic topics that were considered important in the literature (Table: 2.3) and handled the expert's opinion regarding the opportunities and challenges of Vector Tiles for Web Cartographers. The findings of the expert-interviews are presented below according to the cartographic topics in the interview topic list (Appendix: A).

7.2.1 Users and user requirements

The first topic is about users and user requirements with Vector Tile technology. The aim was to find out who the users are of Vector Tiles and how they intend Vector Tiles to be used.

Identifying users of Vector Tiles
While on one hand Vector Tiles are seen as an effective product for consumers or customers, they are on the other hand seen as an emerging solution for producers or developers. This divide was also became very obvious during the interviews. On the one hand, Robert Nordan (2018) from Norkart AS and Willem Jan Vierbergen (2018) from ESRI Nederland mention the important use of Vector Tiles for consumers and customers. Here the consumers and customers are users that wish easily zoomable and fast maps. Customers can be end-users or consumers using Vector Tiles for use by their own end-users. For instance, customers of ESRI products are seen as the consumers of Vector Tiles produced by ESRI. On the other hand, Edward Mac Gillavry (2018b) and Niene Boeijen (2018b) from Webmapper believe that especially at the developer's side Vector Tiles are implementable. They believe that the need for Vector Tiles does not come from the consumers or end-users, because they should not realize the change in technology. End-users are not aware of consuming Vector Tiles and are more concerned with how fast a map is transferred and displayed on their screen. Edward Mac Gillavry (2018a) gives an example: *"End-users are not aware of consuming Vector Tiles. An example is Google Maps, where you are not always aware of it that they provide maps in Vector Tiles. So I believe it is especially at the developer's side that Vector Tiles are implementable"*. Niene Boeijen (2018b) adds to this that the need of Vector Tiles is more at the developer's side: *"It is more the need for developers to provide a website with a fast loading map or to provide another visualization to their customer"*. In line with this, Jeroen Hogeboom (2018) from PDOK/Kadaster mentions that he also sees the user of Vector Tiles as a developer who wants to create its own visualization (or a view) of a map on an online platform. These developers are in this case often cartographers. Prof. Stefan Keller (2018) draws attention to the fact that Vector Tiles are mostly made for cartographers since all the styles for Mapbox Vector Tiles have been designed for cartographers by cartographers. This also means that a link with the theory can be made because Vector Tiles are made for map users. Earlier in the literature, a shift from map readers to map users was observed (Kraak, 2009; Muehlenhaus, 2014; Ory, 2016). This shift from passive map readers to more active map users with Web Cartography is clearly becoming more obvious with Vector

Tiles as well. There is no longer a supply-driven map production by cartographers, but a demand driven map production to design interactive and responsive maps that reply to the map user's need.
While there still seems to be a divide on who the main users are of Vector Tiles, it is not possible to say that it is one or the other. Lukas Martinelli (2018) from Mapbox highlights that at Mapbox they see both their customers as themselves as users. For the company and developers, the benefit is off-setting rendering costs and less caching with Vector Tiles, while for the end-users Vector Tiles are a step towards being resolution independent with seamless zooming and more dynamic data. In the next paragraph, it is further discussed which user requirements of Vector Tiles can be identified.

Understanding user requirements of Vector Tiles

It is important to understand what users want with Vector Tiles. Earlier two main groups could be identified, namely: (1) End-users, customers or Vector Tiles consumers and (2) Developers, producers and Vector Tiles providers. Often, a requirement of the producer is to meet the requirement of the end-user. A good example is given by Willem Jan Vierbergen (2018) who highlighted the requirement of ESRI Nederland to produce Vector Tiles in the Projected coordinate system for the Netherlands (Rijksdriehoeksstelsel) since this is what customers were expecting to use with their data. Another example is that for end-users it is important to have a choice of maps in different styles, while for producers it is important to create these styles and custom visualizations for them (Boeijen, 2018a; Hogeboom, 2018; Keller, 2018; Mac Gillavry, 2018a; Martinelli & Roth, 2015; Vierbergen, 2018). The same applies when it comes to the requirement of having fast loading maps, the end-users expect that maps are displayed fast on their screen, while the developer also wants to serve and transfer Vector Tiles these maps as efficient and fast as possible. Edward Mac Gillavry (2018b) and Niene Boeijen (2018b) mention that it is important that map interaction becomes faster.
Next to that, for both user groups it is a requirement to be resolution independent which refers to Vector Tiles offering smooth and seamless zooming between the different zoom levels. The ability to have more dynamic content in the maps is a requirement for both user groups as well. Furthermore, according to Lukas Martinelli (2018) it is important for the developer that the specification is open in order to have different encoders for use with the Mapbox Vector Tile specification.

The main use of Vector Tiles

Currently, the main use of Vector Tiles is visualization. Willem Jan Vierbergen (2018) draws attention to the fact that currently Vector Tile technology focuses itself very much on general basemaps which are used for reference or as background give users a new option for visualization. It is also possible to use Vector Tiles for a broader aim or for other kind of maps, however, this has not been done a lot yet in practice. The same applies to the use of Vector Tiles for analysis purposes. The interviewees agree on the fact that analysis is not the main focus or priority of Vector Tiles. Robert Nordan (2018) and Willem JanVierbergen (2018) believe that users that wish to do analysis are already more advanced GIS professionals, whereas Vector Tiled basemaps are for everyone to consume. Similarly, Edward Mac Gillavry (2018b) notes that it is possible to think about scenarios of end-users doing analysis on different parameters, although these end-users are rather geospatial researchers than consumers. Besides, Jeroen Hogeboom (2018) reminds that when adding more data into Vector Tiles for analysis, that they might become less attractive for clients as they might use up more performance. The latter also has consequences for the main requirement of Vector Tiles to be fast and efficient visualized. However, there are some cases that can be mentioned where Vector Tiles are not only used for visualization but also for analysis. For instance, Niene Boeijen (2018b) mentions an example where point data in Vector Tiles is used as storage medium for analysis. In that case it is possible to easily perform calculations on a specific tile instead of having to do calculations on the whole map. For certain analysis, the OSM QA tiles mentioned by Lukas Martinelli (2018) and Prof. Stefan Keller (2018) can be used. These Vector Tiles have limitations because for example, it is not possible to calculate a street length straight forward, because it's possibly cropped along a tile. This would mean that a client has to merge it before analysis. Even though there are some use cases for analysis with Vector Tiles, the main use is visualization. In the next paragraph, the topic of knowledge and learning curve will be discussed.

7.2.2 Knowledge and learning curve

When considering knowledge and learning curve as a topic, it was asked to the interviewees whether they considered it difficult to learn about Vector Tiles and how to use the technology. Furthermore, it was questioned whether they agreed or disagreed with the statement that Vector Tiles are not mainstream yet due to a lack of cartographic knowledge. Lastly, it was questioned what aspects the experts wish to learn more about regarding Vector Tiles.

The difficulty to learn about Vector Tiles and using the technology

There is no clear answer on whether it is difficult or not to learn about Vector Tiles and how to use them because it depends on the experience of the developer. Lukas Martinelli (2018) and Robert Nordan (2018) mention that in some way Vector Tiles are conceptually harder to understand compared to for instance raster tiles since it is another approach. However, Edward Mac Gillavry (2018b) and Jeroen Hogeboom (2018) also note that Vector Tiles are not more difficult than setting up a tile cache or even are somehow similar to raster tiles for persons that are already familiar in the field. Lukas Martinelli (2018) summarizes it well: *"I think that the tool on top of Vector Tiles makes it as easy as other technologies such as raster tiles, however, conceptually understanding is harder"*. Vector Tiles are harder to understand because it is different than just loading a grid of images. It is needed to understand the conceptual link of having raw data, a Vector Tile format and an rendered image. Robert Nordan (2018) notes that with Vector Tiles it is not possible to know directly what is inside the tile because you would need a tool or the whole stack to see more than just a stream of numbers. While Vector Tiles are conceptually harder to understand than just loading a grid of images, the tools for Vector Tiles make it easier to use them. Willem Jan Vierbergen (2018) points out that ArcGIS Pro as a tool makes it somehow easier as a starting point, since it is not necessarily needed to dive into the hard code. Furthermore, Edward Mac Gillavry (2018b) notes that there are a lot of different choices and tools when it comes to deciding where and how to generate Vector Tiles. This makes it easier but also challenging, as Niene Boeijen (2018b) highlights that you are very dependable on external developments regarding Vector Tile technology: *"Viewers to view or to style Vector Tiles are continuously changing and finding the best approach to do this. You are always looking for a good solution, and once you have something working, it is possible that there has been something else developed which works better"*. The fact that it can go very fast with the developments when it comes to Vector Tiles makes it easy and challenging at the same time to work with them.

Lack of cartographic knowledge? Vector Tiles mainstream?
It is not possible to say that due to a lack of cartographic knowledge Vector Tiles are not mainstream yet. First of all, one could say that Vector Tiles are in fact mainstream already, as Robert Nordan (2018), Edward Mac Gillavry (2018b) and Lukas Martinelli (2018) mention that consumers have been consuming Vector Tiles already for a very long time. It is, for instance, obvious for them how to use them on a mobile phone on Google Maps. However, the problem observed by Willem Jan Vierbergen (2018) and Edward Mac Gillavry (2018b) is that Vector Tiles are not always mainstream yet due to a lack of ignorance/awareness or because they are slowly adopted by cartographic community or organizations working with GIS. These organizations or customers sometimes still have to get used to Vector Tiles and Vector Tiles only become mainstream for them once software like ArcGIS and QGIS support Vector Tiles, which they started doing. Prof. Stefan Keller (2018) states the following about this: *"At the time that ESRI started to use Vector Tiles from Mapbox, I knew that this would be the sign that Vector Tile technology was a technology to stay for a while."*
Secondly, it is important to differentiate between cartographic knowledge and technical knowledge when looking at working with Vector Tiles. According to Prof. Stefan Keller (2018), the knowledge gap is somehow on the education side, where traditional cartographers are not acquainted with programming or technical aspects from computer sciences: *"It is really leading-edge technology for cartography on the web. My explanation is simply that a lot of cartographers have not the knowledge of the technology because it is programming oriented."*. Niene Boeijen (2018b) also believes that there are enough cartographers that find it difficult to work with Vector Tiles and web maps in general due to a lack of technical or programming knowledge/skills. Jeroen Hogeboom (2018) even suggests that it is nowadays perhaps not possible anymore to avoid the technical part for cartographers. When it comes to the cartographic knowledge gap, this seems not to be the case for consumers or end-users because they have been consuming Vector Tiles already for a long time and are not making maps or Vector Tiles themselves. However, the cartographic knowledge gap can be observed with developers with a background in computer sciences. As Prof. Stefan Keller (2018) observes: *"Every semester I realize how computer sciences underestimate the difficulties of cartography like generalization or visualization constraints. Of course, there is a knowledge gap on the computer science side, because they not always have the feeling for cartographic design"*. Furthermore, Lukas Martinelli (2018) would say that everywhere where custom projections matter, Vector Tiles are not mainstream yet. Therefore there seems a lot of work to be done yet for traditional cartographers and modern computer scientists to combine cartographic knowledge with programming/technical knowledge.
Both the cartographic knowledge gap for computers scientists as the technical knowledge gap for traditional cartographers observed by the interviewees can be linked with the distinction found in the literature between modern map makers that are coders, and traditional map makers that are expert-cartographers (Roth, Hart, et al., 2014; Van den Berg, 2017). As Van den Berg (2017) mentioned: *"The creation of especially web maps is shifting towards people who know how to code rather than people that know how to correctly design a map"*. This also seems to be the case in practice.

What experts want to learn more about regarding Vector Tiles
In general, the interviewees mention that they all know how Mapbox Vector Tiles work. However, what they want to learn more about differs. Edward Mac Gillavry

(2018b) and Niene Boeijen (2018b) note that the process of rendering stays something complicated. They remark that if you have more knowledge about how this rendering works, it gives you more knowledge about what it means for cartography and styling or even interaction. Niene Boeijen (2018b) also mentions that she would like to learn personally more about how to work with PostGIS as Vector Tile generator.
Lukas Martinelli (2018) indicates that he would like to understand better how features go across tiles and how they are clipped and reassembled on the client side.
Prof. Stefan Keller (2018) states that there is always more to learn, but that for him it is important to see what should go into the documentation of Vector Tiles. He believes that currently the specs. of metadata of tiles, TileJSON and MBTiles are way behind the current status (e.g. defining what are mandatory fields). Later in this section, the topic of documentation will be further discussed. In the next topic, stability & performance of Vector Tile technology and Vector Tiles tools will be handled first.

7.2.3 Stability & Performance

This topic refers to the stability, reliability and performance of Vector Tile technology according to the interviewee's experience. Furthermore, it was questioned whether the interviewees believe that performance is more important than cartographic rules or design.

Stability & Performance of Vector Tile technology
In general, the interviewees believe that Vector Tile technology is stable and performing well enough. For instance, Willem Jan Vierbergen (2018) believes that Vector Tile technology is stable enough and very attractive for end-users since Vector Tiles are smaller in data size for storage and therefore better suited compared to raster tiles. Lukas Martinelli (2018) remarks that the fact that the technology didn't change that much lately shows that it is stable: *"It powers so many customers right now that it is basically well developed. It is in the maturity curve, it is slowly arriving, it is not new anymore in my opinion"*. Robert Nordan (2018) also sees Mapbox Vector Tiles as stable and well developed. However, he remarks that the only possible problem lies at the client-side, because it is more resource-intensive and demanding. Prof. Stefan Keller (2018) states that there is another problem, even though the performance is good, he believes that the specs. from Mapbox are behind the current state of Vector Tiles usage which makes the de-facto standard unstable. Furthermore, Edward Mac Gillavry (2018b) and Niene Boeijen (2018b) indicate that it is not always possible to follow a certain approach because of encountered challenges. For instance, with the usual stack of PostGIS - Geoserver Vector Tile plugin or with the current approach of Tippecanoe it is difficult to adjust simplification or to have control over this.

Performance more important than cartographic rules or design?
With the question whether performance is more important than cartographic rules or design, it depends on who you are making maps and Vector Tiles for. In line with this, Jeroen Hogeboom (2018) believes that this question is more a question for the user since it depends on their needs. He mentions that higher levels of details might need more performance, but if wanted they can add more resources for higher performance. In general, traditional cartographers tend to look more at map design instead of performance, whereas developers look more at performance. This aspect clearly appears in what Robert Nordan (2018) has to say about performance and proper map design: *"As an engineer, I think that proper map design and cartographic rules*

are more important. But if you look at the users, performance is more important because they don't care if the maps don't look properly as long as it loads fast". One could say that it is about finding compromises between performance and proper map design (Mac Gillavry, 2018b).

7.2.4 Scalability & Extendibility

The topic of Scalability & Extendibility refers to the ability of Vector Tile technology to become more important or to continue to function well according to the interviewee's opinion.

Scalability & Extendibility of Vector Tile technology
Vector Tiles are seen as a good solution by the interviewees, both currently and for the future, for cartographers. For instance, Prof. Stefan Keller (2018) states the following: *"I always have been or still am convinced that Vector Tiles are the future of Web Mapping because they have been designed since the beginning as efficient as possible, they sorted out as much data as possible, it was always speed while still having as much as data needed and control over the styles"*. Lukas Martinelli (2018) mentions that it will even be adopted now by the slower adopters and therefore be more important. As well for mobile applications and devices without internet connection Vector Tiles are important according to Edward Mac Gillavry (2018b) and Niene Boeijen (2018b). They mention for instance the need of a good reference for fieldworkers. Furthermore, Robert Nordan (2018) and Willem Jan Vierbergen (2018) believe that Vector Tiles get easier for users since hardware, devices and browsers get more powerful and capable. The CPU of the client is an important aspect, that nowadays is no real obstacle anymore, because nowadays most devices are good enough for Vector Tiles. If still a drawback has to be mentioned, Prof. Stefan Keller (2018) draws attention to the lack of standardization of Mapbox Vector Tiles metadata which might hold back the eco-system and slow down further innovation.

Vector Tiles, a need for a more complex and resource demanding infrastructure?
It would be false to state that Vector Tiles need a more complex and resource demanding infrastructure, because this depends for who, either the developer/provider or the end-user/client.
When looking at a developers point of view, Vector Tiles don't need a more complex infrastructure than compared to for instance raster tiles. Edward Mac Gillavry (2018b) states: *"If you take a smart approach not at all. It is basically a map file structure with different pbf files and a server to serve them. If you don't produce Vector Tiles on the fly, which I think is often the case, you can best separate the generation of tiles from the serving of tiles"*. Willem Jan Vierbergen (2018) remarks that the most important advantage is that less data storage is needed with Vector Tiles for the operational infrastructure. Furthermore, Lukas Martinelli (2018) remarks that the tools for generating and serving Vector Tiles make it really easy for developers.
If considering the client-side, it can be more complex or resource demanding according to Jeroen Hogeboom (2018) and Robert Nordan (2018), even though it is getting better with technology becoming more performant. Jeroen's Hogeboom (2018) answer summarizes well both the client-side and provider-side: *" Vector Tiles demand more from the client-side, however, for the Vector Tile provider it can help to alleviate the server by giving more resources/responsibility to the client. WMTS technologies become costly compared to Vector Tiles which can diminish the investments costs for the data*

providers. Logistics and infrastructure is a lot of work. Using a new tool is not always easy because of migration from one technology to another".
Another aspect to mention is that problems can occur with Vector Tiles at the client side when, as Mac Gillavry (2018b) mentions, some environments do not support WebGL in the browser: *We had this experience with one of our clients where thin clients on desktops didn't support WebGL"*.

7.2.5 Interoperability

The topic of interoperability refers to the interviewee's opinion regarding the integration of different Vector Tile tools or solution. Even though interoperability is often considered to be important, the general point of view of the interviewees is that the solutions are not entirely interoperable. However, Edward Mac Gillavry (2018b) notes the following on this subject: *"ArcGIS Pro, QGIS, Openlayer, Leaflet, Mapbox...from the viewer side it is possible to use it together. Even in D3 you can load Vector Tiles. On the server side it really depends, if you want to have the whole stack with GeoServer with a plugin then it is a little extension, however, if you migrate to dedicated Vector Tiles rendering such as Tegola or Tippecanoe then you have less integration"*. Furthermore, Willem Jan Vierbergen (2018) mentions that interoperability could be technically possible, however, that it should be kept in mind that it is not always allowed to use Vector Tiles provided by ESRI with other Web Mapping tools. Customers of ESRI have to accept the terms of use, which also means that they have to use the content inside ESRI software.

7.2.6 Formats, standards & schemes

The topic of Formats, standards & schemes discusses the interviewee's opinion about the lack of standards for Vector Tiles and the ability to make or have standards for Vector Tiles.

The lack of standards for Vector Tiles
If questioning whether the variety of formats is a disadvantage, Robert Nordan (2018) answers that he doesn't see it as a disadvantage. He doesn't see competition: *"There are a lot of formats, but I don't see a competition anymore since MVT have won it completely for me*. Although Robert Nordan mentions being in favor of MVt, he also remarks that the good thing is that you don't necessarily need to use Mapbox solutions, but there are also alternative solutions or options. Prof. Stefan Keller (2018) shares this point of view, according to him, if one solution has a major share, then it doesn't need to be standardized as far as the spec. is open for anyone to implement. Edward Mac Gillavry (2018b) does not see it as a disadvantage either, he believes that the different vector formats are all about the choice that you want to make. The choice depends on the size of the data. Jeroen Hogeboom (2018) agrees that what works for one, might not work for another. In general, one could say that the MVT and Mapbox tiling scheme is most optimized and fits best the needs of the interviewees.

The ability to make or have standards for Vector Tiles
According to Prof. Stefan Keller (2018) and Lukas Martinelli (2018), there is already a standard, the one of Mapbox Vector Tiles specification. However, this is not an OGC standard, but a de-facto standard. If OGC is to make a standard for Vector Tiles, then Prof. Stefan Keller (2018) and Willem Jan Vierbergen (2018) identify some challenges

or requirements. Prof. Stefan Keller Keller (2018) believes that if OGC tries in the next two years to make something about Vector Tiles with the current requirements from the stakeholders, it will be doomed to fail. He highlights that OGC ignored one of the main requirements of Vector Tiles to be fast and efficient in the Testbed-13 Vector Tiles Engineering report. With the wish to integrate Vector Tiles with a Web Feature Service, it will have consequences for the speed of Vector Tiles. Willem Jan Vierbergen (2018) mentions that there is no real need for a standard for the ESRI content, however, in case customers want to make their own Vector Tiles, it could be useful to have a standard. He notes that if there will be an OGC standard, it is important for ESRI Nederland that it will be a standard in which there is the ability to use more projections.

7.2.7 Documentation and accessibility

This topic refers to the completeness, readability, quality and accessibility of documentation about Vector Tile technology and solutions according to the interviewees. According to the interviewees, documentation about Mapbox Vector Tiles are easy to find, which is also a reason why they have chosen to go with that solution. However, Prof. Stefan Keller (2018) draws attention to the specs of metadata of Mapbox Vector Tiles being behind the current state of Vector Tiles usage. He wishes that Mapbox comes with a new release of TileJSON, to write the de facto status quo of the software they are using. However, according to him, Mapbox perhaps fears that they will break data at their customer's side when they change the behavior of their current software components. For everything that is other than Mapbox Vector Tiles, Edward Mac Gillavry (2018b) mentions that the quality of the documentation can be very case specific. Niene Boeijen (2018b) for instance indicates that when she encountered an issue in the use of a Vector Tile tool, that it is something less well-documented because it is a difficult aspect in the making of the Vector Tile tool. However, that in most cases it is possible to sort it out by means of documentation and discussions on the developer's websites, forums or Github. Jeroen Hogeboom (2018) mentions that even though there is some good documentation, there could be more to find on Google. When it comes specifically to documentation for ESRI customers, then Willem Jan Vierbergen (2018) notes that there is enough to find on how to make good maps with Vector Tiles. Interesting to mention as well is that at ESRI Nederland they have not made documentation on how to change Vector Tile styles since the Vector Tiles are still in beta and they want to wait with it. Willem Jan Vierbergen (2018) mentions that, currently, they are not stimulating customers to change styles while using Vector Tiles, because the data scheme currently used can still be updated in the upcoming months which could mean that the styles changed by the customers could not work anymore afterwards.

7.2.8 Community support

The topic of community support refers to the opinion of the interviewees whether there is enough support for Vector Tiles in the field of GIS and Cartography.
In the ArcGIS community, Willem Jan Vierbergen (2018) observes that there are currently not a lot of people making their own Vector Tiles: "*There are perhaps a few customers that are doing this, but in general we make them at ESRI and our customers consume them*". Furthermore, he mentions that the ESRI content team of the Netherlands sometimes has meetings with other contents teams from other counties. In these meetings, Vector Tiles are often a topic which comes back. Similarly, but not only

for the ArcGIS community, Jeroen Hogeboom (2018) mentions that he would like to see more collaboration and meetings between different organizations and users in the GI-field in the Netherlands to stimulate for instance a Vector Tile community. He believes that it is important to have a platform to present findings or ask questions to learn from each other. Lukas Martinelli (2018) thinks that GIS applications slowly move to Vector Tiles and that there will be enough community support. Edward Mac Gillavry (2018a) also believes that it goes slow and mentions that the Cartography or GIS-field not always picks up Vector Tiles yet, however, he can imagine that they are welcoming the technology once that they are aware of the advantages of Vector Tiles. On the other hand, Prof. Stefan Keller (2018) also notes that there are still specialist that are very knowledgeable about all the technology, but sill stick to more traditional styles such as Mapnik or CartoCSS. There does not seem to be a lack of community support in the field of GIS and Cartography, but moreover a lack of programming knowledge or *"lecturers that have a hard time keeping up with technological advances and experience the Stockholm syndrome towards major GIS vendors"* (Mac Gillavry, 2018b). Moreover, Prof. Stefan Keller (2018) mentions that some still prefer to use raster tiles over Vector Tiles: *"Some stick to the server-side rendering, to the raster tiles and to the Mapnik styling technology, which I understand" (Keller, 2018)*.

7.2.9 Frequency of updates

This topic discusses the current pace in which Vector Tile tools and technologies develop and whether it is a challenge to keep up with all the Vector Tile solutions according to the interviewees. The expert's opinions are divided regarding this subject. On one hand, some think that it is difficult to keep up with all the different Vector Tile tools and solutions, whereas on the other hand some think it is not going too fast.

Edward Mac Gillavry (2018b) mentions that it is very case specific, but that the update of tools can go very fast because they are almost occurring on a daily basis. Niene Boeijen (2018b) believes that it is hard to keep up with all the Vector Tile solutions. For instance, she points out that she is not aware of what has been updated with the GeoServer plugin for Vector Tiles. Niene Boeijen (2018b) gives another example of OpenMapTiles that one year ago provided Vector Tiles from OSM data in a very simple base map of the world, and now is very detailed: *"I was overwhelmed that in a half a year time, the tiles offered by OpenMapTiles were more detailed and contained more data such as the BGT"*.

Other interviewees seem to be less overwhelmed by the current pace in which Vector Tile tools and technologies develop. Willem Jan Vierbergen (2018) draws attention to the fact that there are challenges, however, that for each challenge there is a solution. He furthermore mentions that the data scheme is crucial and that he can imagine that in 3 years the needs regarding data can change: *"Changes in what kind of data customers want, and other updates, may lead to the problem that all styles that customers have modified won't work anymore*. However, Willem Jan Vierbergen (2018) doesn't see Vector Tiles as something which changes the Web Mapping technology dramatically. In line with this, Prof. Stefan Keller (2018) doesn't expect too much to show up anymore. He doesn't see it as something challenging to keep up with all the different Vector Tile tools and solutions. Lukas Martinelli (2018) and agrees that it is not hard to keep up with all the Vector Tile solutions. Both Martinelli (2018) and Hogeboom (2018) find that the development of Vector Tile tools and solutions is not going too fast.

7.2.10 Cartographic projections

This topic deals with cartographic projections. It was asked during the interviews what the interviewees think about the use of different projected coordinate systems. The aim was to find out to what extent cartographic projections play a role for the experts.
Edward Mac Gillavry (2018b) mentions that ESRI & Klokan technologies have made it possible to add different projections for Vector Tiles. That is a supplement to the Mapbox Vector Tile specification. For the PDOK/Kadaster project of BRT and BGT in Vector Tiles, he doesn't see it at as a disadvantage to use the Web Mercator projection: *"I see it as an advantage, because it simplifies the learning curve for people in the Netherlands, because they expect World Mercator"*. Jeroen Hogeboom (2018) remarks that the choice and use of Web Mercator is because it is compatible with a lot of applications and the Mapbox Vector Tile specification. According to him, it must be awaited or seen later whether users want a custom projection or not, currently the Vector Tile service is still in beta.
Lukas Martinelli (2018) also notes that it is possible to create a fake projection that translates it into Web Mercator: *"You can create a new tile grid that is specific for you local projection, it does not require a lot of hacking, it is do-able."*. He furthermore states that usually for Mapbox it is not a big concern since Mapbox is global and it is hard to justify for local projections.
Prof. Stefan Keller (2018) believes that projections are very important: "It is difficult for computer scientists to think that the world is covered by geographic coordinates, and that the Mercator projection is the only one, but in fact that is not true". He moreover gives an example of Klokan Technologies applying a trick to let the Mapbox tools and libraries think that it is Mercator, but it was the Swiss coordinate system. Prof. Stefan Keller (2018) draws attention to the fact that the Vector Tile specification does not say it needs to be Mercator: ""*It is a misunderstanding that Vector Tiles are sticking to Mercator. The specification does not mention a coordinate system"*. Willem Jan Vierbergen (2018) also claims that projections are very important: *"For our use and users, the RD projection is very important. I think that more than 80% of the organization users and the majority of the ArcGIS users in the Netherlands use RD for their own data that they want to visualize on a map"*.

7.2.11 Map design and styling

The topic of Map design and styling discusses the advantages and disadvantages of styling with Vector Tiles according to the interviewee's opinion. Map design and styling are often seen as the most obvious part of Cartography. It will be first discussed what the identified advantages are, and afterwards what map design or styling challenges with Vector Tiles are present according to the interviewees.

Map design and styling advantages with Vector Tiles
The interviewees mention several major advantages that come with Vector Tiles. For instance, Willem Jan Vierbergen (2018) states that for map makers it is an advantage that styling can be changed to the simplify the map by removing elements that you don't want to see: *"The ability to decide which elements on a map are most important by applying styling to them is stimulating a better understandability and composition of the map"*. Prof. Stefan Keller (2018) also sees Vector Tiles as an advantage to communicate the visual science: *"Vector Tiles have an advantage that they have vectors to display with no rough lines, so you get the perfect outline, perfect contrast, perfect high-resolution*

display". Next to the advantage of better communicating a message based on control over the styling process, the interviewees mention the advantages of Vector Tiles that it works with vectors. Working with vectors means that the map features look sharp on each zoom level, this also leads to being resolution independent with seamless zooming and more dynamic data (Boeijen, 2018b; Martinelli, 2018). Working with Vector Tiles also means that it is not needed to make a style for each zoom level and that it is possible to control this in the JSON style file. Willem Jan Vierbergen (2018) expresses that *"the main advantage for Web Cartographers is that it is not needed to make tiles and styles for every different zoom level"*. Niene Boeijen (2018b) gives a similar but even more specific example on this subject that an advantage of vectors is that it fluently interpolates between different zoom levels (e.g. for label sizes): /textit"it is very different than older technologies where you had to specify for each zoom level and had an image as an output". The advantage is that there is less code needed to write down and that the result looks smoother or sharper between the different zoom levels. Jeroen Hogeboom (2018) and Willem Jan Vierbergen (2018) also indicate that they like the option that users can decide what kind of styling they want to see and that it can create a good visualization on each zoom level.
Not only the ability of different zoom levels that the user looks at is considered as an advantage regarding map design, it is also the way that users interact these days (Martinelli, 2018; Vierbergen, 2018). In the literature, it was observed by Muehlenhaus (2014) that interactivity advantages make the map user more in control. According to the same author: *"map designers can build redundant interactive mechanisms into their maps (e.g., multiple methods of zooming or changing data layers) to help people digest the information they are receiving in individually more palatable ways"*. However, too much interactivity can also potentially minimize the clear communication of information in the map (Muehlenhaus, 2014). Besides, end-users possibly are not aware of color rules while changing the styles of the maps which can result in less effective web maps. Therefore, the role of the cartographer is to design maps that according to Muehlenhaus (2014) *" are interactive and responsive to a maps user's need to facilitate communication that is more effective"*.
When considering in especial labeling with Vector Tiles, Robert Nordan (2018) highlights that place names, words, letters are better placed and smarter positioned: *"Dynamic labeling allows size and font types to be changed on the fly unlike raster tiles where labels are pre-set"*. To add to this, Edward Mac Gillavry (2018a) remarks some advantages about labels when rotating and having a tilt view of the map: *It is also nice that with labels you can rotate them in the viewer and still read them easily in the reading direction"*. Therefore, one could say that Vector Tiles offer advantages regarding labeling. However, on the basis of the interviews, it can be noted that labeling also still causes some challenges.

Map design and styling challenges with Vector Tiles

Willem Jan (Vierbergen, 2018) highlights that for labels there are still some challenges because it is not possible to rotate the labels based on a point or polygon, it is not possible to give direction to it. However, this problem seems to be possible to solve by using lines (Mac Gillavry, 2018b; Martinelli, 2018; PDOK, 2018b; Vierbergen, 2018). Lukas Martinelli (2018) remarks that for print maps labeling could be a disadvantage, however, not for mobile where dynamic labeling should be embraced according to him.
Another labeling problem concerns the repetition of labels with Vector Tiles (Keller, 2018; Mac Gillavry, 2018b; PDOK, 2018b; Vierbergen, 2018). Willem Jan Vierbergen (2018), for instance, states that the repetition of labels can be seen anywhere with

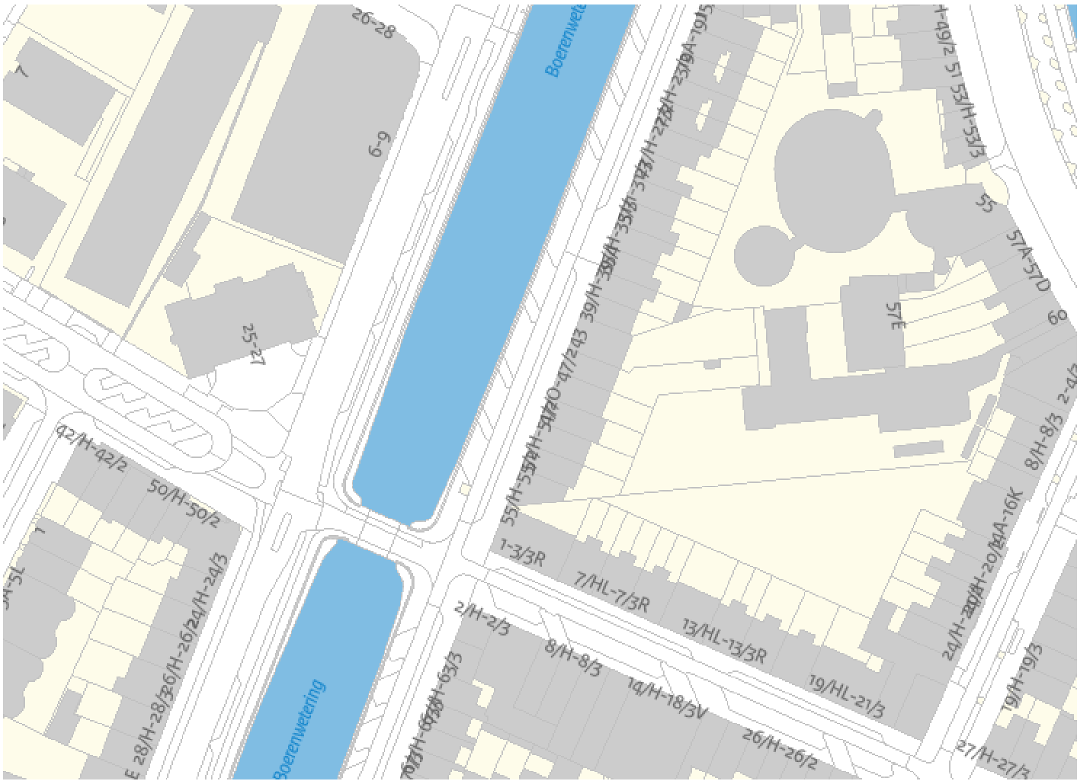

FIGURE 7.1: PDOK Vector Tiles BGT in Beta; label challenge with point data (PDOK, 2018b)

Vector Tiles (Figure: 7.2), also with tiles used by Google. Prof. Stefan Keller (2018) observes that there are many discussion on how to reconnect the lines together that could be used to solve the repetition of street name labels on every tile, however, that this is feasible.

FIGURE 7.2: ESRI Topo RD Vector Tiles in Beta; repetition of labels issue (PDOK, 2018b)

Another different challenge with Vector Tiles, other than labeling, is the challenge for hillshading (Keller, 2018; Martinelli, 2018; Vierbergen, 2018). According to Willem

Jan Vierbergen (2018) is is hard to find a good way to visualize hillshading, which has a lot of colors, with Vector Tiles. He notes that Mapbox has found some solutions with vectors and different classes of transparency (e.g. Mapbox Terrain). However, Willem Jan Vierbergen (2018) points out that even though hill shades for Vector Tiles is theoretically possible, that it is not going to be better than raster hill shading. Nevertheless, Prof. Keller (2018) remarks that he would like to see more representations for hillshading, which he believes is interesting and feasible: : *"you can transform a continuous surface to a polygon Vector Tile. You get an efficient polygon which is somehow divided into many different classes"*. However, this solution seems to be still less detailed than hill shades with raster tiles, where there more classes of darkness possible.
In terms of styling, there are other arguments of having raster tiles or server-side rendering instead of Vector Tiles according to Prof. Stefan Keller (2018). He mentions the ability to print, the ability to control the position of the labeling better. Furthermore, a disadvantage of Vector Tiles is that it is somehow quite a completely different styling language compared to CartoCSS, Mapnik or SLD's, which means yet another new styling language (GIS Wiki HSR, 2018; Keller, 2018).

7.2.12 Geometry

When it comes to the topic of Geometry related to Vector Tiles, all geometries are considered important according to Willem Jan Vierbergen (2018). Jeroen Hogeboom (2018) mentions that currently lines, points and polygons in Vector Tiles are well supported.
Furthermore, Robert Nordan (2018) highlights that theoretically Vector Tiles could do anything when it comes to geometries, however, that in practice Mapbox Vector Tiles are often only used with simple features. He mentions that for professional applications the support of different geometries could be better, but that for consumers there is no noticeable difference.
When considering the topic of geometry with Vector Tiles is moreover important to look at how rendering works in WebGL. According to Edward Mac Gillavry (2018b) polygons are perhaps easiest to render, and fonts most challenging. He points out that geometry does not always have to be very accurate for visualization.

7.2.13 Preservation of topology

The next topic discusses the preservation of topology with Vector Tiles. Willem Jan Vierbergen (2018) notes that with the BGT topology is a challenge. Not all topological rules are always kept by all source holders according to him. However, for the ESRI Vector Tiles, the file geodatabase used as an input for the Vector Tiles is already well prepared and checked on topological errors. Jeroen Hogeboom (2018) also believes that for the PDOK Vector Tile Service the topology is reasonably in sync with what there is right now.
However, according to Robert Nordan (2018) there is no real good solution yet for the preservation of topology because it is all based on the visual: *"Nobody really found a sensible way of preserving topology"*. Prof. Stefan Keller (2018) notes that it is a challenge with how polygons are displayed and eventually merged and aggregated back. Niene Boeijen (2018b) also mentions that is is good to look at topology when taking the example of polygons where there might sometimes show up slivers during the simplification process. The next topic handles the simplification and generalization topic with Vector Tiles in more detail.

7.2.14 Simplification and generalization

This topic deals with simplification and generalization opportunities or challenges with Vector Tiles according to the interviewee's opinion. The interviewees mention several aspects.
Robert Nordan (2018) highlights that Vector Tilea are all simplified and generalized and that the real issue is that it is not possible to get back to the unsimplified/ not geralized version of it.
Next to that, Prof. Stefan Keller (2018) notes that there re many open questions regarding generalization, however, not strictly related to Vector Tiles. Lukas (Mac Gillavry, 2018b) also believes that generalization of Vector Tiles is not more challenging than for the traditional raster cartography. Prof. Stefan Keller (2018) remarked that when visiting some ICA working groups of generalization topics, he wanted to explain to that cartographers are in the best position to start research on generalization based on Vector Tile encoding and technology, but that they were simply not aware of what Vector Tiles are. The most important challenge encountered by Prof Stefan Keller (2018) is the fact that it is needed to pre-generate the Vector Tiles at every level, and then to decide how much to generalize beforehand. Niene Boeijen (2018b) also points out the important role of simplification/ generalization before generating Vector Tiles: ""*the tools that are there right now offer simplification, however, it remains difficult to keep control over this process. This is the reason why I preferably already simplify my own data beforehand (e.g. Merging of roads)*". This means that in some cases it remains better to apply changes already in the database. This process requires cartographic knowledge and can be time-consuming (Boeijen, 2018b; Vierbergen, 2018). Furthermore, Willem Jan Vierbergen (2018) mentions that in the beginning it was difficult to get used to the simplification process of Vector Tiles basemaps on the lowest level of detail, because with raster tiles it is usual to start on the highest level of detail.

7.2.15 Future possibilities and expectations

The last topic handles the future possibilities and expectations of Vector Tile technology. In other words, it refers to what the interviewees believe that can be done with Vector Tiles that has not been done yet, as well as what they expect from the future of Vector Tiles.
First of all, there could be a lot more work done on projections for Vector Tiles. According to Lukas Martinelli (2018), projections could be interesting because: *"It would be cool to see how local products apply other projections with Vector Tiles themselves"*. Prof. Stefan Keller (2018) also mentions that he would like to see in the future more work with other projections. According to him, it is currently a lack of the specification: *"You could put it the other way around, it is a needed extension of the current specification"*. The fact that it was a needed extension is proved by ESRI which provides Vector Tiles in local projections for the Netherlands. As mentioned earlier, according to Willem Jan (Vierbergen, 2018), it was a requirement because of the users demand. Next to that, Robert Nordan (2018) remarks that projection problems can be a drawback for GIS professionals, that tend to not use Vector Tiles in that case.
Secondly, Prof. Stefan Keller (2018) and Willem Jan Vierbergen (2018) indicate that the hill shading technique of transforming a continuous surface to a polygon Vector Tile with new client-side rendering can be an interesting topic to do more research on, as well as 3D Vector Tiles. Prof. Stefan Keller (2018) mentions the example of roofs in 3D or tilt view that could be visualized more realistically when investing

more time in it. In line with this, Niene Boeijen (2018b) also mentions that she would like to see more possibilities with extrusions: *It is possible to lift up buildings or polygons with extrusion, however, for water, for instance, I would like to see under-extrusion*. To continue the wish list, Prof. Stefan Keller Keller (2018) believes that it is interesting to investigate how to tackle best the creation of hachures/ contour lines (e.g. Rock depiction) for steep slopes in Vector Tiles.
Moreover, Willem Jan Vierbergen (2018) mentions that there is still a technical barrier for showing pop-ups with Vector Tiles because if a lot of attributes are taken into account, the data size of the Vector Tiles gets bigger. According to Robart Nordan (2015), there is a need for a sort of WFS for Vector Tiles, the missing piece is the analysis part. However, he remarks that analysis is not the focus of Vector Tiles, and, therefore, there won't change much in the future. Furthermore, Nordan (2018) puts an emphasis on the fact that he has great respect for consumers even though he would say that they don't really make use of all the possibilities in Vector Tiles: *"They simply don't need a lot of things we cartographers like to dream about"*.
Another aspect that could be solved is making Vector Tiles better usable with client GIS applications. Robert Nordan (2018) would like to find a way for making Vector Tiles work with GIS applications. Similarly, a wish of Lukas Martinelli (2018) is that there would be multiple open source renderer clients, not just Mapbox GL. He mentions that the good thing is that there is OpenLayers and Mapzen Tangram that can render Vector Tiles.
Furthermore, Jeroen Hogeboom (2018) would like to see more accessible and findable data in Vector Tiles in the future. He believes that more data can be converted Vector Tiles than currently is done. The most important part is to make Vector Tiles more accessible, according to Jeroen Hogeboom (2018) the current released Vector Tile service, demo viewer and styles, make it more accessible: *It depends on the technical barrier for cartographers, but it gives opportunities for visualizing data based on your wishes*.
There are many possibilities, options and wishes with Vector Tiles. However, Niene Boeijen (2018b) highlights that the problem with more options is that everyone wants different options and that it is not possible to implement everything. Therefore, the next section presents some main aspects that are considered to be important for Web Cartographers when working with Vector Tiles. These aspects cover the cartographic potential and challenges of Vector Tile technology for Web Cartographers.

7.3 Advantages and disadvantages of Vector Tile technology for Web Cartographers

Based on the different cartographic topics covered while discussing Vector Tiles with the interviewees, one can come up with several advantages and disadvantages of Vector Tiles for Web Cartographers. The author summarized the following advantages of Vector Tiles:

- Suitability of Vector Tiles for mobile applications and offline use (this because of the compact size of Vector Tiles and efficiency for transfer).
- Potential to use Vector Tiles as an exchange format (also because of the compact size of Vector Tiles resulting in efficient data transfer and lower bandwidth usage and costs for hosting).

- Suitability for interactive and informative maps (because of flexibility and adaptive character, e.g. it is possible to get attributes information of the map objects or to rotate and tilt the map all while labels stay right-side up which make the experience more user-friendly).
- Ability to create maps with better resolution and communicate the visual science better (because of working with vectors and having seamless zooming advantages). This means that they are well suited for communicating and visualizing data (e.g. changing styles based on zoom level is handy, it is possible to make features only visible on a specific zoom level).
- Easier to make custom styles or visualizations to better answer the users needs (because there is no need to make a style for each tile, customizable right away in the code on the client side).
- Client-side rendering and the advantage that Vector Tiles can be quickly rendered for multiple similar map styles. Only need to tile the data once to have multiple maps. Map clients can access the vector data directly.
- The current tools to generate, serve and to style Vector Tiles make it very easy for Web Cartographers to implement Vector Tiles.

On the other hand, there are some disadvantages or challenges with Vector Tile technology that could be identified:

- More complicated to conceptually understand compared to image tiles.
- Less suited for analysis or editing of data (because Vector Tiles almost always lose some data due to filtering and generalization).
- Not suited for data with a lot of attributes. The more attributes, the bigger in size the Vector Tiles get. Vector Tiles should stay compact to keep its main use to offer fast loading maps.
- Little control over the simplification and generalization process with the Vector Tile tools
- Still some styling challenges such as for instance drawbacks like the repetition of labels belonging to different feature classes.
- No unified styling language and Mapbox Style Specification being a low level styling language (e.g. no syntax to group things). Realistic styles become hard and complex to maintain according to Keller (2018). Also making good styles in OpenLayers is a difficult task according to Martinelli (2018).
- The current Mapbox Vector Tile Specification says nothing about the use of different coordinate system projections. Even though it is possible, there is no real support for it.
- Being dependable on external developments regarding all the different options with Vector Tiles and Vector Tile tools. Difficult to keep up with all the different solutions and possibilities.
- No standardization of OGC/ISO or mature standardized tools, however, defacto standard with Mapbox Vector Specification.
- Challenges with client-side rendering with slower or older devices (because rendering on the client side takes more CPU and graphics power on the client).

7.4 Cartographic strengths and weaknesses of Vector Tile tools and solutions

After having identified the cartographic advantages and disadvantages of Vector Tile technology, it is interesting to understand the cartographic strengths and weaknesses while implementing Vector Tiles. What are the strengths of a particular tool or solution and where does this apply in the workflow? What is a cartographic aspect that is better supported by one solution compared to the other? Which tool or solution is best suited for Web Cartographers when wanting to create Vector Tiles? Which tool or solution is most adequate for serving Vector Tiles tot the client? These are questions one could ask.
Based on cartographic topics from the theory and cartographic aspects that were found important by the interviewees, some cartographic aspects can be identified to assess the cartographic strengths and weaknesses of Vector Tile solutions. The following parameters were identified and selected by the author:

1. Quality of documentation (findability, Up-to-dateness , completeness, etc.).
2. Difficulty in implementing a tool or solution (technical or cartographic knowledge required in a workflow or for implementing a tool)
3. Risk of vendor lock-in (ability to switch to alternatives if needed, integration with other solutions, open source/proprietary)
4. Support for different geometries
5. Handling of generalization (generalization/filtering techniques used and control over the simplification process
6. Support for different map projections
7. Support for different formats (e.g. input formats, storage and output formats)
8. Support for styling (e.g. difficulty of changing styles)

Even though the choice for a solution or workflow depends on more choices than only cartographic aspects, the aim here is to give an idea of the cartographic strengths and weaknesses of certain Vector Tile solutions.
Not all the Vector Tile tools and solutions/workflows presented earlier in Chapter 6 are taken into account due to time constraints and the fact that not every Vector Tile tool and solution was fully deployed by the author. It should be noted that it was not the main goal of the thesis to test out all the different Vector Tile tools and solutions. The aim of the assessment is to give an idea of where the cartographic strengths and challenges lie in a Vector Tile workflow. Therefore some example tools or solutions were considered in the assessment:

- PostGIS as a tool for storing, editing, transforming and simplifying the data before generation of the Vector Tiles
- Tippecanoe as a tool for generating Vector Tiles and a Web Server as a solution for serving the tiles
- Geoserver as a tool for both generating and serving Vector Tiles
- OpenLayers as a client tool for rendering and styling Vector Tiles

- Mapbox GL JS as another client tool for rendering and styling Vector Tiles

These mentioned tools can be found in the workflows of:

1. PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS (Workflow 5, in Chapter 6, Figure 6.12)
2. PostGIS → GeoServer → GeoServer → OpenLayers (Workflow 6, in Chapter 6, Figure 6.13)

Both workflows discussed in Chapter 6 are Vector Tile stacks for custom data. This means that the use case is to generate own Vector Tiles and serve them to the client. Figure 7.3 shows an example of how a Vector Tile stack for custom data works.

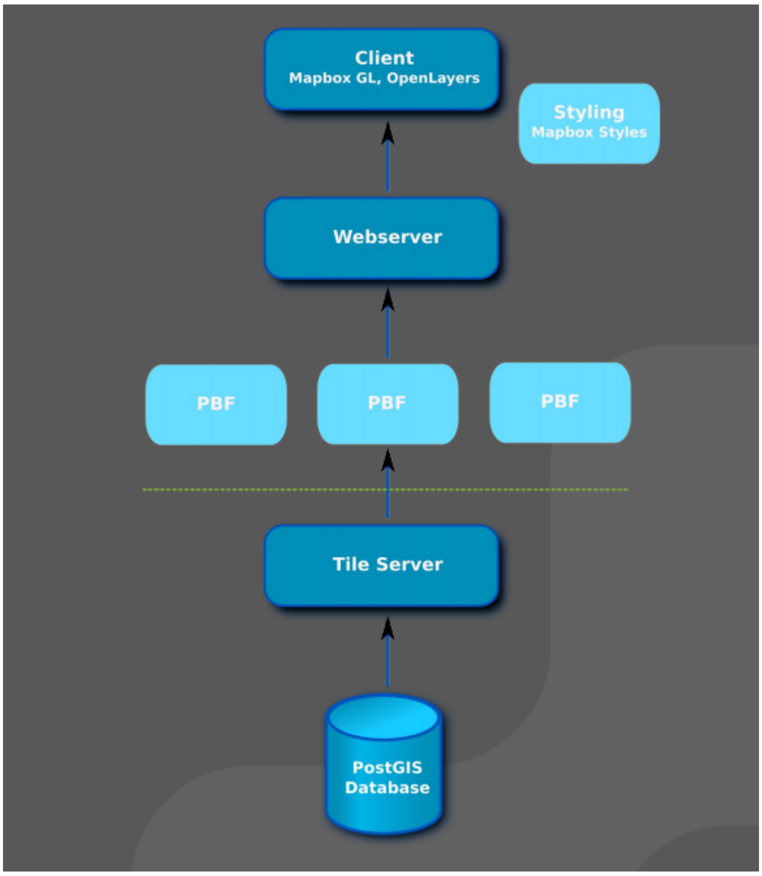

FIGURE 7.3: Vector Tile stack for custom data (Kalberer, 2017)

By assessing the cartographic strengths and weaknesses of the selected Vector Tile tools and solutions/workflows, it is possible to identify where the cartographic potential comes in, in the Vector Tile workflows. In other words, the question can be asked whether a certain cartographic potential can be realized with a specific Vector Tile tool or solution/workflow. The two selected workflows and their tools were first compared and assessed on their cartographic strengths and weaknesses according to the cartographic parameters. The assessment is structured in the following paragraphs by discussing each cartographic parameter one by one.

7.4.1 Quality of documentation

The quality of the documentation refers to the findability, completeness and up to up-to-dateness of the documentation that can be found on the web on by means of other sources regarding the use of a specific tool or technology. The findings are based on the authors experience while testing the solutions or on experiences and opinions from the interviewees. For both workflows, one could say that the quality of the documentation was good and up-to-date enough for implementing the tools and solutions successfully. The links to the online documentation for the different tools can be found in Chapter 6 under the name of "handy links".

Workflow 5: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS

Documentation on PostGIS as a spatial database extension
PostGIS as a spatial database extension is well documented by means of online documentation and workshops. There are also some good cookbooks on how to work with PostGIS or workshops by Boundless. There are no challenges encountered while using PostGIS for the storing, editing, transforming and simplifying of data for this workflow assessed.

Documentation on Tippecanoe as a Vector Tile generator
The documentation on how to use Tippecanoe as a Vector Tile generator is fair. It should be noted that there is really a large amount of websites or blogs to find that describe how to use Tippecanoe, however, the documentation on how to install Tippecanoe with Docker and some code use examples are mainly written for Linux operating systems. Therefore it could be more complete. The author experienced some challenges while using docker for Windows. Fortunately, there was some expertise at Webmapper to tackle this challenge and make the solution work on Windows.

Documentation on how to use a Web Server for serving Vector Tiles
The documentation on how to serve Vector Tiles without the need of a server tool is fair. There are some examples given on the Web on how to do this. Someone with good knowledge about servers would not need any documentation.

Documentation of Mapbox GL JS as a client or renderer of Vector Tiles
The documentation on how to use Mapbox GL JS as a client renderer is excellent. There are many examples of how to implement it. Mapbox GL JS general strength is that it was mainly written for use with Vector Tiles and the Mapbox Vector Tile Specification.

Workflow 6: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS

Documentation on PostGIS as a spatial database extension
PostGIS as a spatial database extension is well documented by means of online documentation and workshops. There are also some good cookbooks on how to work with PostGIS or workshops by Boundless. There are no challenges encountered while using PostGIS for the storing, editing, transforming and simplifying of data

for this workflow assessed.

Documentation on GeoServer as a Vector Tile generator and server
The documentation on how to use Geoserver as a Vector Tile generator and server is very good. It offers a tutorial on how to implement the solution and it even gives examples of how to change styles in OpenLayers as a client. The only part of documentation which was harder, or almost not, to find was regarding the handling of generalization and GeoServer.

Documentation of OpenLayers as a client or renderer of Vector Tiles
The documentation on how to use OpenLayers with Vector Tiles is good. However, it was not always easy to find. While the examples of how to use the API and how to use OSM or Mapbox Vector Tiles in Openlayers are easy to find, it is harder to find documentation on how to style Vector Tiles in OpenLayers. The author had to do some effort in order to find a Vector Tile workshop for OpenLayers. Fortunately Boundless also gives a complete documentation on how to use Mapbox Style Objects With OpenLayers.

7.4.2 Difficulty in implementing a tool or solution

The previous parameter discussed the findability, completeness and up-to-dateness of the documentation of the Vector Tile tools for both workflows. These aspects are also considered to be important when assessing the difficulty of implementing a certain tool or solution. Not only documentation is important, but also the amount of technical knowledge or cartographic knowledge required for implementing a tool or solution is an aspect to consider. In general, there is not a lot of technical knowledge required for implementing the Vector Tile tools in both workflows, making them rather easy to implement the solutions. It should be observed that the workflow of creating Vector Tiles with Tippecanoe and serving them based on a file directory structure by means of a Web Server (workflow 5) requires more technical knowledge compared to the solution and workflow of using Geoserver as a Vector Tile generator and server (workflow 6). The cartographic knowledge part comes in handy while using PostGIS as a tool for simplifying the geo-data used in both workflows.

Workflow 5: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS

Implementing PostGIS as a spatial database extension
PostGIS as a spatial database extension is easy to implement, however, there is some knowledge required about how to work with Postgresql and SQL. This means that a Web Cartographer aiming to use this tool should have this technical knowledge in order to implement it in the workflow. There are no challenges encountered while using PostGIS for the storing, editing, transforming and simplifying of data for this workflow assessed.

Implementing Tippecanoe as a Vector Tile generator
Tippecanoe is very easy to implement. However, it is a command line utility meaning that it requires some technical knowledge on how to use Docker or the command prompt. If this knowledge is present, then the tool is easy to implement. The author

experienced quite some challenges to make Tippecanoe work with Docker on Windows.

Implementing a Web Server to serve Vector Tiles to the client This part requires some technical and computer science or technology knowledge regarding servers and their components. A Web Cartographer with knowledge should not face any challenges while implementing a simple http to serve Vector Tiles to the client. For a traditional cartographer, this is perhaps more difficult, but this is very case specific.
Implementing Mapbox GL JS as a client or renderer of Vector Tiles
Mapbox GL JS as very straightforward to use. Knowledge about Javascript, HTML and CSS is of added value, however, the author is of the opinion that any Web Cartographer could easily implement this solution. Traditional cartographers might encounter some challenges, although it depends on the person. The library is very well documented by Mapbox which makes is easy to implement.

Workflow 6: PostGIS → GeoServer → GeoServer → OpenLayers

Implementing PostGIS as a spatial database extension
PostGIS as a spatial database extension is easy to implement, however, there is some knowledge required about how to work with Postgresql and SQL. This means that a Web Cartographer aiming to use this tool should have this technical knowledge in order to implement it in the workflow. There are no challenges encountered while using PostGIS for the storing, editing, transforming and simplifying of data for this workflow assessed.

Implementing GeoServer as a Vector Tile generator and server
When considering the implementation of GeoServer as a Vector Tile generator and server one could say that it is very easy to install and to use. Installing the tool does not need any technical knowledge and the author's opinion is that basically anyone could use this tool with the documentation provided.

Implementing OpenLayers as a client or renderer of Vector Tiles
OpenLayers as a client or renderer of Vector Tiles is easy to implement. Knowledge about Javascript, HTML and CSS is of added value, however, the author believes that any Web Cartographer could easily implement this solution. Especially Web Cartographers that already have experience in using OpenLayers as a Web Mapping library would face no major challenges. Traditional cartographers might encounter some challenges, although it depends per person.

7.4.3 Risk of vendor lock-in

In general, there is a higher risk of vendor lock-in with vector tiles. The interviewees have mentioned that they find it important to have alternatives in case a tool or solution stops working or is depreciated (e.g. in case Mapbox runs out of money). Therefore, it is important to look at whether there are alternatives while implementing a certain tool or solution. Furthermore, the interviewees have mentioned that they find it important to have good integrated open source tools. First of all, it should be noted that the workflows chosen to compare on their cartographic strengths and weaknesses are just two possible options, and there are many alternatives to generate and serve vector tiles other than these solutions. However, it can be interesting to investigate for each tool individually whether there is a high or low risk of vendor lock-in. In other words, whether there are many or few alternatives tools.

Workflow 5: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS

Risk of vendor lock-in with PostGIS
To the best of the authors knowledge, there are other options to PostGIS and Postgresql to edit, manage, transform and simplify the data. One could also choose not to have data in a database by creating a Shapefile and exporting it to GeoJSON or by creating a GeoJSON file directly. It all depends on what kind of data and how large the data is that the Web Cartographers wants to convert to Vector Tiles. A strength is that PostGIS is open source and well integrated with other tools in the workflow (e.g. ogr2ogr to convert data formats).

Risk of vendor lock-in with Tippecanoe from Mapbox
There are other alternatives for creating Vector Tiles than Tippecanoe (e.g. Tilemaker) which means that there is not a high risk of vendor lock-in. The strength of Tippecanoe is that it is open source, however, the weakness is that the user is very dependable on the developments regarding the tool its functionalities. The Web Cartographer has little say over this aspect and is limited to the available options.

Risk of vendor lock-in when serving tiles by means of a Web Server
There is no risk at all of vendor lock-in for this part since there is no tile server needed. However, you would need an http server in order to serve the Vector Tiles.

Risk of vendor lock-in with Mapbox GL JS
The risk of vendor lock-in is questionable with Mapbox GL. The weakness is that Mapbox GL JS is that you depend very much on the Mapbox ecosystem, however, the Mapbox Vector Tile specification is open source. In order to use Mapbox GL JS you need a Mapbox account and key. There are alternatives to Mapbox GL as a client or as a renderer, such as OpenLayers or leaflet, however, it is not always convenient to migrate to these alternatives once a project has already been created.

Workflow 6: PostGIS → GeoServer → GeoServer → OpenLayers

Risk of vendor lock-in with PostGIS
To the best of the author's knowledge there are other options to PostGIS and Postgresql to edit, manage, transform and simplify the data. One could also choose not to have data in a database by creating a Shapefile and exporting it to GeoJSON or by creating a GeoJSON file directly. It all depends on what kind of data and how large the data is that the Web Cartographers wants to convert to Vector Tiles. A strength is that PostGIS is open source and well integrated with other tools in the workflow (e.g. GeoServer that can import data from PostGIS).

Risk of vendor lock-in with GeoServer as a Vector Tile generator and server
Other tools offer the same solution of both generating and serving vector tiles, some examples are t-rex and Tegola. Therefore the risk of vendor lock-in is very low. It should be noted that there are many more alternatives as a way to generate vector tiles (e.g. Tilemaker or Tippecanoe) as well as to serve vector tiles (e.g. without a tile server or with another alternative such as TileServer GL).

Risk of vendor lock-in with OpenLayers
The risk of vendor lock-in is lower with OpenLayers compared to Mapbox GL as an alternative since it is open source. However, it is not always convenient to migrate to an alternative such as Mapbox GL once a project has already been created. Furthermore, with Mapbox GL you would need a Mapbox account and a key which makes the risk of vendor lock-in higher.

7.4.4 Support for different geometries

The support of different geometries is a cartographic parameter to take into account. Theoretically, Vector Tiles could do anything when it comes to geometries. However, in practice, when the use of Vector Tiles for basemaps and visualization is considered, Vector Tiles tools tend to work more with simple features. The interviewees mentioned that all geometries, such as points, lines and polygons are important. For this cartographic parameter, is was not relevant to assess the tools individually on their support for different geometries, because basically all tools in both workflows support points, lines, polygons and even multi-part features. There were no differences identified between the solution of creating Vector Tiles with Tippecanoe and serving them based on a file directory structure by means of a Web Server (Workflow 5) and the solution of using Geoserver as a Vector Tile generator and server (Workflow 6). One could interpret this as a cartographic strength of all tools compared in this assessment of the two workflows.

7.4.5 Handling of generalization

Handling of generalization here refers to the generalization/ filtering techniques used in the solutions and the amount of control the Web Cartographer has over the simplification process.

Workflow 5: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS

PostGIS to simplify data before generating Vector Tiles
The cartographic strength of PostGIS is that it enables to have strong control over the simplification and generalization process. This is vital in both workflows since in general the tools to generate Vector Tiles offer little control over the simplification process. This aspect was also mentioned by NieneBoeijen (2018b). Therefore, one could say that the cartographic potential in both workflows for simplifying data comes before the actual generation of Vector Tiles. The cartographic knowledge and know-how can be implemented before using the tools of Tippecanoe or Geoserver as Vector Tile generators.

Tippecanoe and the handling of generalization
As mentioned earlier, Tippecanoe as a tool for generating Vector Tiles gives little control to the Web Cartographer over the simplification process. Even though there are some options (e.g. Filtering feature attributes, dropping a fixed fraction of features by zoom level, dropping tightly overlapping features, multiplying the tolerance for line and polygon simplification by scale, attempts to improve shared polygon boundaries, etc.) possible with Tippecanoe, there is in general little control over the generalization process. This is considered as a cartographic weakness of the tool.

Mapbox Vector Tile Specification and Simplification
The Mapbox Vector Tile specification states that it doesn't cover simplification. Even though the conversion from geographic coordinates to Vector Tile coordinates is an important step, there are many different ways that simplification can be implemented prior to Vector Tile encoding. This is considered as a cartographic weakness while using Vector Tile tools in the Mapbox ecosystem. Mapbox states that any map data that is uploaded to Mapbox studio is converted into Vector tiles (Mapbox, 2017b). Mapbox supports filtering and generalization using grid-snapping and the Douglas-Peucker algorithm (Open Geospatial Consortium, 2017b).

Workflow 6: PostGIS → GeoServer → GeoServer → OpenLayers

PostGIS to simplify data before generating Vector Tiles
The cartographic strength of PostGIS is that it enables to have strong control over the simplification and generalization process. This is vital in both workflows since in general the tools to generate Vector Tiles offer little control over the simplification process. This aspect was also mentioned by NieneBoeijen (2018b). Therefore, one could say that the cartographic potential in both workflows for simplifying data comes before the actual generation of Vector Tiles. The cartographic knowledge and know-how can be implemented before using the tools of Tippecanoe or Geoserver as Vector Tile generators.

GeoServer and the handling of generalization
The author had a hard time finding anything about the handling of generalization and GeoServer and does not fully understand how to cope with this in GeoServer. What can be said, is that, first of all, it is mentioned in the OGC Testbed-13 Vector Tile Engineering report that Geoserver does not use any specific generalization technique Open Geospatial Consortium (2017b).
Moreover, Niene Boeijen (2018b) and Edward Mac Gillavry (2018b) mentioned that in earlier experiments with Vector Tiles and generalization they encountered generalization problems because they had no say over this. Therefore, one could conclude, that similar to the Tippecanoe tool, one has little control over the simplification process with GeoServer and Vector Tile tools in general that are used to generate Vector Tiles. A better option is to simplify the data before adding to GeoServer, in PostGIS with SQL for instance.

7.4.6 Support for different map projections

This cartographic parameter refers to the support for different map projections. Cartographic projections are important due to the fact that a variety of different systems exist for different regions and purposes. One can imagine that Mercator base maps are not always suited (e.g. not suited to do visual inspection on arctic sea ice coverage).
The interviewees mentioned that they would like to see more maps with Vector Tiles in custom or local projections because this can be a need of customers that have their data in another coordinate system. Therefore, the ability to have custom map projections is an important cartographic aspect to take into account. Here it will be assessed for each workflow how the support is for different map projections.

Workflow 5: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS

Tippecanoe, Mapbox GL & Mapbox Vector Tile Specification
The Mapbox specification of Tippecanoe states that the projection of the input data must be specified and that currently supported EPSG:4326 (WGS84, the default) and EPSG:3857 (Web Mercator) are supported. It states that in general you should use WGS84 for your input files if at all possible (Github Mapbox Tippecanoe, 2018).
There is no support for other projections then Web Mercator, however, the Mapbox Vector Tile Specification does not state that the Vector Tiles have to be in Web Mercator projection. In fact, there are some hacks possible to use custom coordinate projection systems with Vector Tiles (Keller, 2018; Mac Gillavry, 2018b; Martinelli, 2018; OpenMapTiles, 2018b; Thakker, Anand, 2017). Although vector tiles are usually created only in Web Mercator projection (EPSG:3857), it is possible to encode and display the vector tileset in any other coordinate system (OpenMapTiles, 2018b). To the author's opinion, it remains complex or unclear how to do this without the appropriate technical knowledge. Therefore this remains a cartographic weakness, but also a cartographic potential.

Workflow 6: PostGIS → GeoServer → GeoServer → OpenLayers

GeoServer and the support for different projections
In this workflow, Geoserver tends to be more flexible for support of different projections compared to the other workflow. GeoServer supports all projected coordinates systems. It has not been deployed by the author, however, it can be considered as a cartographic strength of the solution.

7.4.7 Support for different formats

This cartophic parameter covers the support for different formats which refers to what input formats, storage and output formats are supported by a specific relevant tool in the workflows assessed.

Workflow 5: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS

PostGIS and the support for different formats
Different formats can be imported into PostGIS and Postgresql by using command line tools such as ogr2ogr or shp2sql. The same applies for the export, where for instance in this workflow the data in the database was exported to a GeoJSON file.

Tippecanoe and the support for different formats
Tippecanoe supports mainly GeoJSON as an input format, however, it also support CSV input files for point geometries. It can output as MBtiles or it can write tiles to the specified directory instead of to an MBtile file. The possibility to store tiles in folders (hierarchical structure) is considered as an advantage in this workflow.

Workflow 6: PostGIS → GeoServer → GeoServer → OpenLayers

PostGIS and the support for different formats
Different formats can be imported into PostGIS and Postgresql by using command

line tools such as ogr2ogr or shp2sql. This approach was not needed in this workflow because GeoServer can import directly from PostGIS database. This is considered as a cartographic strength in this workflow.

GeoServer and the support for different formats
Most common Vector data sources of Geoserver are (WFS), PostGIS or Shapefile. The fact that the data can be imported directly from the PostGIS database is seen as a cartographic strength in this workflow. The data is stored in a folder hierarchy in GeoServer.The output formats of GeoServer for Vector Tiles are GeoJSON, TopoJSON and Mapbox Vector Tiles (MVT). Even thouhg, Mapbox Vector Tiles are the recommended output format by GeoServer it is seen as a cartographic strengths that it supports different response formats.

7.4.8 Support for styling

This cartographic parameter covers the support for styling which says something about the ability to change styles and how difficult this is to do.

Workflow 5: PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS

Mapbox GL JS and Mapbox Style Specification as a way to style
When implementing Mapbox GL JS as a client and rendering it is very straightforward on how to change styles by specifying the style file according to the Mapbox Style Specification. Mapbox Style has some major advantages because it is in JSON file and easy to read and modify. It enables to describe the whole map, not just a single layer, because it has built-in concept of sources and layers. Furthermore, it has functions to control the appearance accross a range of zoom levels or resolutions (Boundless, 2017; Mac Gillavry, 2018b). The fact that changing the style is applied on the client side is seen as a major cartographic advantage because it enables different maps styles for one Vector Tile set. Furthermore, it offers fast and high-resolution maps that better communicate the visual science. The spatial message can be more effectively communicated and the maps user experience is also improved.

Workflow 6: PostGIS → GeoServer → GeoServer → OpenLayers

OpenLayers as a client and way to style
In the workflow of GeoServer there is no need to reconfigure GeoServer because rendering is done by the client and styling can be applied directly in the OpenLayers client. This is seen as a major cartographic advantage because it enables different maps styles for one Vector Tile set. Furthermore, it offers fast and high-resolution maps that better communicate the visual science. The spatial message can be more effectively communicated and the maps user experience is also improved. It should be noted that it is hard to write good styles in OpenLayers (Martinelli, 2018), it requires requires basic JavaScript skills because it is done with functions (Boundless, 2017). However, there is also a possible to use Mapbox Style objects in OpenLayers by using the ol-mapbox-style plugin. Mapbox Style has some major advantages because it is in JSON file and easy to read and modify. It enables to describe the whole map, not just a single layer, because it has built-in concept of sources and layers. Furthermore it has functions to control the appearance across a range of zoom levels or resolutions (Boundless, 2017; Mac Gillavry, 2018b).

7.5 Cartographic potential in the Vector Tile workflows

After having described for each cartographic parameter where the cartographic strengths and weaknesses are, this section aimed at identifying the Cartographic potential in the Vector Tile workflows assessed. Figure 7.4 and Figure 7.5 first gives an overview of where the cartographic strengths and weaknesses occur in each separate workflow. Figure 7.6 compares the cartographic strengths and weaknesses of both workflows according to the cartographic parameters.

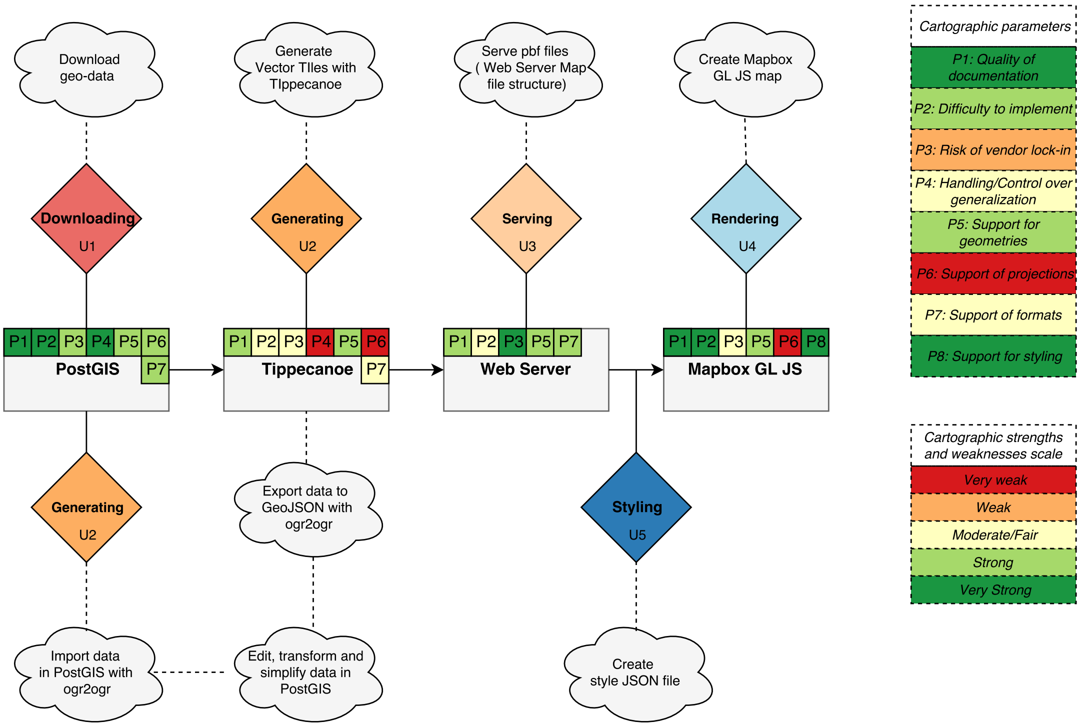

FIGURE 7.4: Overview of cartographic strengths and weaknesses in the assessed Vector Tile workflow of PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS (Workflow 5)

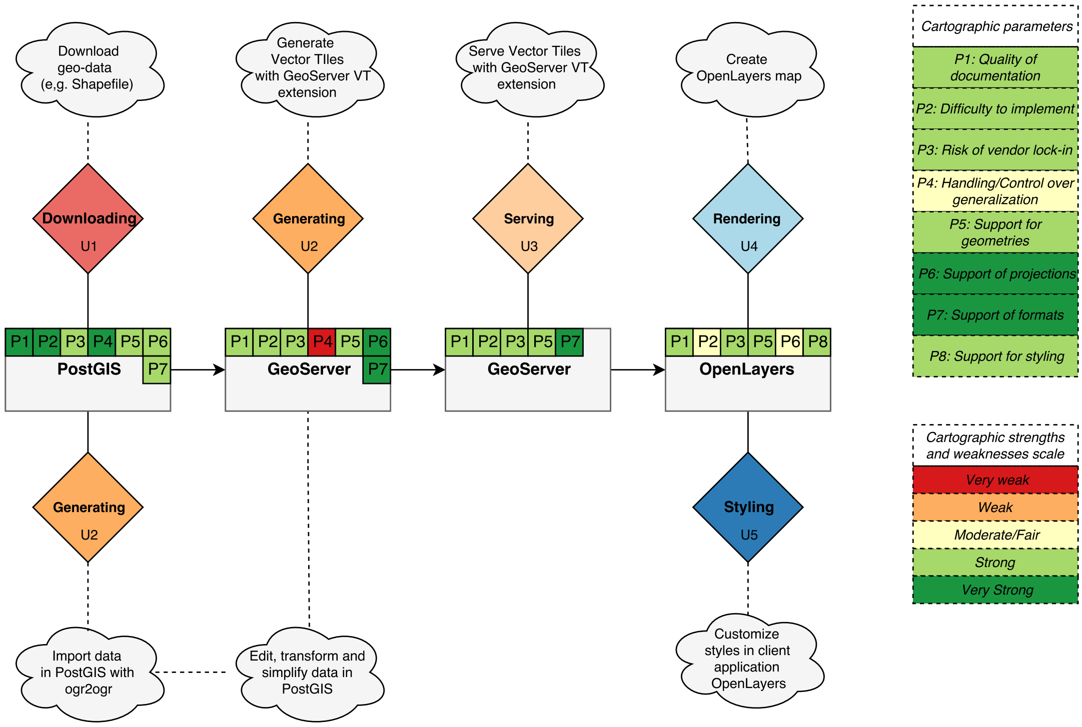

FIGURE 7.5: Overview of cartographic strengths and weaknesses in the assessed Vector Tile workflow of PostGIS → GeoServer → GeoServer → OpenLayers (Workflow 6)

Cartographic parameters		W5	W6	Workflow 5 Strengths	Workflow 5 Weaknesses	Workflow 6 Strengths	Workflow 6 Weaknesses
P1	*Quality of Documentation*			Mapbox GL documention is very good. A lot to find about Mapbox Tippecanoe. PostGIS has complete workshops	Examples on how to use Tippecanoe and Docker in Windows could be more complete.	Good tutorial for GeoServer and complete workshops for PostGIS	Little documentation to find on how GeoServer handles generalization
P2	*Difficulty to implement*			Very straightforward solution with generation and serving of tiles separated from each other, Tippecanoe and Mapbox GL JS easy to implement	Requires some technical knowledge	Easy to install GeoServer and to use it without any significant technical knowledge required	Some Javascript knowledge required for writing good styles directly in OpenLayers. Little to find about generalization.
P3	*Risk of vendor lock-in*			Enough alternatives, and no need for a tile server	Mapbox ecosystem higher risk of lock in	Enough alternatives, no risk of vendor lock-in	
P4	*Handling/control generalization*			PostGIS for the simplification of the data	Little control over simplification with Tippecanoe	PostGIS for the simplification of the data	Little control over simplification with GeoServer
P5	*Support for different geometries*			Support for points, lines, polygons and multi part features		Support for points, lines, polygons and multi part features	
P6	*Support for different projections*			It is possible, if the technical know-how is there	No support by Mapbox for custom projections	GeoServer supports all coordinate systems and projections	Could be some more examples on how to do this with GeoServer and OpenLayers
P7	*Support for different formats*			It is possible to convert different formats with ogr2ogr for import and export of PostGIS database. Pbf files on http server considered as advantage	Tippecanoe supports mainly supports GeoJSON as input file, not always suited	Different Vector Tile output formats for GeoServer. Integration with PostGIS.	
P8	*Support for styling*			Mapbox Style Specification and JSON very easy and written for Vector Tiles	Remains a low level styling language	There is a Mapbox Style plugin for OpenLayers	Harder to write good styles in OpenLayers due to Javascript knowledge required

Cartographic strengths and weaknesses assessment scale

Very weak	Weak	Moderate/Fair	Strong	Very strong

Workflow 5 (W5): PostGIS → Tippecanoe → File directory on Web Server → Mapbox GL JS
Workflow 6 (W6): PostGIS → GeoServer → GeoServer → OpenLayers

FIGURE 7.6: Overview of cartographic strengths and weaknesses in the Vector Tile workflows (W5 & W6)

The most important findings and identified cartographic potentials in the workflows are discussed below. What is the best solution and where is the cartographic potential in the workflows? An answer to these questions is difficult, it all depends on the Web Cartographer's technical expertise and what he or she wants to do with the data (how large, what format, frequently updating or not) when converting data into Vector Tiles. Therefore, there is no best solution. It depends on what your goal is and which requirements there are by the end user.

Cartographic potential of implementing custom projections
When looking at the support for projections, it is considered as a weakness of Tippecanoe, Mapbox GL and the Mapbox Vector Tile Specification (basically the Mapbox ecosystem) that there is no support for custom projections. However, it should be noted that the Mapbox spec does not say that Mercator Projection has to be used. It is technically possible to have other projections than the Web Mercator projection, however, it still requires some know-how on how to do this. One could say that, regarding the cartographic potential of implementing custom projections, GeoServer would be a better solution since it has support for all projection systems.

Cartographic potential of having more control over the generalization process
A cartographic potential that can be identified in the workflows, and what has not been realized yet, is to further research and develop ways to enable more control over the simplification process in the Vector Tile tools itself. Currently, the cartographic strength of generalization lies before the generation of Vector Tiles, which is before the implementation of a Vector Tile tool. Therefore, it could be a cartographic potential to move this strength towards inside the Vector Tile tools. One could see the implementation of cartographic knowledge regarding generalization and simplification while editing the data in PostGIS database as a cartographic potential or strength for Web Cartographers in the Vector Tile stack that already has been realized.

Cartographic potential of different formats with Vector Tiles
The choice for a specific Vector Tile solution or workflow can depend very much on the data used, how big the size is of the data and what how often the Vector Tiles need to be updated. The most straightforward and easiest solution is to use Mapbox Studio and Mapbox GL JS. The drawback of this, is that the Web Cartographer would depend on the products of Mapbox. There is a high risk of vendor lock-in. Furthermore, the data can be too large to upload in Mapbox Studio. Mapbox offers another solution with Tippecanoe to convert the data into MBTiles that are smaller files. This is a very easy and straightforward command line solution. This major advantage of Tippecanoe is that it is possible to make one entire cache which is advantageous because it takes less time to serve the tiles. In case the tiles must first be generated every time when requested by GeoServer it would take more time. However, Tippecanoe has its disadvantages as well. One disadvantage is that it needs a GeoJSON as an input file, which can be less suitable when having many features or large data sets, although it is still possible to split up layers (e.g. one GeoJSON for the water layer, one GeoJSON for the buildings, etc.) (Boeijen, 2018d).
One could say that for larger datasets it could be better suited to use GeoServer as a Vector Tile generator solution if the Web Cartographer does not want to bother with splitting up layers. Furthermore, GeoServer has a good integration with PostGIS and offers support for different input data formats and output Vector Tiles formats, which can be seen as a cartographic strength. Both workflows are not very complex or resource demanding when you take a smart approach. However, for most uses it would be best to separate the generation of tiles from the serving of tiles, like this is the case with the workflow of creating Vector Tiles with Tippecanoe and serving them based on a file directory structure by means of a Web Server (Workflow 5). The

reason is because it leads to better map user experience. Even if there is no direct link with Cartography, the fact that you just need a map file structure with different .pbf files enables the creation of a very fast loading map which leads to a positive map user experience. However, if the aim is to generate Vector Tiles on the fly, with tiles that need frequent updating, it would be better to import data into PostGIS and use a tool such as GeoServer to generate the requested tiles dynamically from the database. This means that when there are changes in the data, this is directly taken into account when serving the data as Vector Tiles. (Boeijen, 2018d; Gardner, James, 2018).

Cartographic potential of having more open source alternatives
As can be noted in Figure 7.6, Workflow 5 has a higher risk of vendor lock-in due to the fact that the solution depends very much on Mapbox solutions. Therefore, in the same line as what the interviewees have mentioned, it would be a cartographic potential to develop more open source tools that can be used to implement Vector Tiles. Currently there are good working solutions, however, every solution has its disadvantages and challenges which could be further developed or solved. This is a potential for Web Cartographers.

Cartographic potential of extending current documentation
It can be observed that the documentation currently regarding the use of the tools in both workflows is good, however, there is always space for improvements. The current specification could be further updated (e.g. standardizing MVT metadata). Since the Mapbox Vector Tile specification is open source there are opportunities for Web Cartographers to come up with suggestions for improvements. Furthermore, according to the author, there could be more examples or tutorials available on how to implement the tools.

Cartographic potential of styling with Vector Tiles
One could argue that the cartographic strength lies in the rendering and styling part of the workflows because the rendering and styling happens on the client side leading to several cartographic advantages such as sharper visualizations and better contrast or visual hierarchies to communicate the visual science. One main strength is that Vector Tiles allow different map styles for the same Vector Tile dataset.
While in both workflows the styles are changed directly in the client application or in the JSON style file according to the Mapbox Style specification, it is also possible to use a style editor. The strength is that there are some style editors available to change styles, however, they are sometimes seen as more difficult to use than applying changes right away in the code. A cartographic potential is to develop alternative open source client renderers and style editors that work as good as the Mapbox Studio Classic solution for raster tiles. Currently, Maputnik exists as an open source editor to style, however, according to the author this solution is not as user-friendly as Mapbox Studio. With the current Mapbox Studio you are dependable on Mapbox their services and keys and internet connection, which is seen as a disadvantage.
The cartographic potential of styling Vector Tiles is the most obvious one and is already deployed. The next section discusses the Cartographic potential of the overall Vector Tile technology that still can be realized.

7.6 Cartographic potential of Vector Tile Technology

When taking into account relevant cartographic topics that appeared in the theoretical framework and the cartographic aspects that were found important by the interviewees, it is possible to identify the cartographic potential of Vector Tile technology. Cartographic potential refers here what can be done with Vector Tiles from a cartographic point of view what has not been done yet. It also refers to what still can be researched in more detail in order to improve the use of Vector Tile technology for Web Cartographers. The following cartographic opportunities or potential have been identified and summarized by the author:

- The potential to do more research on the use and support for custom projections. Similar to what ESRI did by extending the Mapbox Vector Tile Specification for use with the Dutch RD projection, there could be more documentation on how to do this for open source solutions.
- The potential of making good looking open source styles for Vector Tile basemaps.
- The potential of coming with a unified styling language, that is as easy as the Mapbox Style language for Vector Tiles but at the same time is less low level by having more possibilities.
- The potential of implementing and combining cartographic knowledge in Vector Tile technology with technical knowledge on how the technology works. This is a gap that still could be filled better.
- The potential of doing more research on encoding and generalization. It remains a difficult topic for Vector Tiles. Important is that geometries should be valid.
- The potential of encoding better geometries that spread multiple tiles and solve labeling challenges such as repetition of labels.
- The potential of finding a good way to have hill shading with Vector Tiles. Currently, raster tiles are better suited.
- The potential of further developing the use of labels and challenges with label repetition with Vector Tiles.
- The potential of extending documentation of Mapbox Vector Tiles metadata (e.g. what metadata is mandatory).
- The potential of improving attribute handling and enabling pop-ups for displaying information about attributes.

Chapter 8

Discussion

This thesis importantly fills a knowledge gap by presenting the available literature and knowledge about the current state of Vector Tile technology and the expert's opinions about the opportunities of Vector Tile technology for Cartographers. Vector Tile technology is in this research considered as a technology that can be deployed by Web Cartographers. The focus lies on what opportunities and challenges there are for Web Cartographers while implementing Vector Tile technologies and workflows.

While the focus of this thesis was on the Web Cartographer as the user of Vector Tile technology, the Web Cartographer was not always considered as the main user by the interviewees. In fact, one could see the customers or end-users as the main users of Vector Tile technology, because they consume the end products, the actual fast loading Vector Tile basemaps created by Cartographers and Web Developers.
The different backgrounds of the interviewees became apparent during the empirical research. There are three groups that could be identified: (1) Producers or developers with a computer science background, (2) Producers or developers with a Cartographic background and (3) Product owners or content engineers. For the first group of interviewees, performance of the technology (e.g. loading speed of tiles, size and encoding of tiles) seemed to be matter the most, whereas for the second group, cartographic opportunities and challenges identified played an even important role (e.g. styling benefits, projections or lack of control over simplification process). For the last group, it was in particular interesting to discuss the advantages of Vector Tiles for users (e.g. Vector Tiles resulting in highly interactive maps with better resolution).

Although the different backgrounds of the interviewee's led to different opinions regarding Vector Tiles, there were also many similar points of view which could be identified. When considering the general outcome of the interview, it can be said that visualization plays an important role in Vector Tile technology because it has a focus on general basemaps which are used for reference or as topographic maps. These are maps that are more and more used by nowadays map users. What can be observed is that map users are also more and more involved in Web Mapping and the creation of maps. They don't read maps, but they use them (Muehlenhaus, 2014). With Vector Tiles, map users even can change the map visualizations since they can change styles on the fly on the client side. Therefore, one could observe a shift from product, to service, to user-driven cartography. Where first the focus was on producing maps for map readers, afterward services with raster tiles, now the user expects to consume fast interactive maps with Vector Tiles. For the end-users, performance seems to be more important than the actual design or cartography part of Web Maps. Styling and web design is not a priority for consumers because the majority won't do anything about changing styles. Customizing map visualizations remains something for the producer, for the Web Cartographer, it is a cartographic potential of Vector Tile technology for them.

Vector Tile technology is a technology that has been there already for quite some years, therefore, it can be argued that Vector Tiles are mainstream. However, one could also say that Vector Tiles are not mainstream yet because they are not welcomed or adopted by everyone yet. Currently, not everyone is acquainted with Vector Tiles and not all possibilities in Vector Tiles are being used. One reason why Vector Tile technology is still being in an emerging or developing stage is that other Web Mapping technologies such as raster tiles are much better developed. Professionals in the geographic field are therefore sometimes hard to convince to migrate to the technology of Vector Tiles since they already have a good working solution. One could observe that if a certain technology works well, the main drawback is that users are not always interested in changing anything about it.
The main argument why Vector Tiles are perhaps not widely implemented yet by all developers or cartographers is that they are harder to understand conceptually. The Vector Tile workflows may not be obvious either since the link has to be made between the different stages of the Vector Tile pipeline or stack. There is a need to understand that there is raw data encoded and transferred as tiles and then decoded back to an image. It is a conceptual link that has to be made. Moreover, Vector Tile technology requires technical or programming knowledge that traditional cartographers could see as a technical barrier. It can furthermore be challenging to keep an overview of all the different developments and possibilities, especially since Vector Tile technology is developing fast.

While Vector Tile tools developed to generate or to serve Vector Tiles make it very easy for Web Cartographers to implement the Vector Tile technology, they do not always take into account how to solve certain cartographic challenges. It was noted that the main cartographic weaknesses in the assessed workflows are the lack of control over the simplification process in the Vector Tile tools and the lack of support for custom projections. The major strength identified is the support for styles and the ability of the Cartographer to customize these on the client side. One could say that the weaknesses can be set up against the strengths since a challenge could also mean a potential to do more research on. The potential lies in developing the weaknesses and deploying the strengths. There is a new role for the Web Cartographers to respond to these challenges and to help out developers with a computer science background and lack of cartographic knowledge. The technical knowledge of the technical computer scientists is something where Cartographers, in turn, could learn from. The gap between the two fields became obvious both in the literature studies as well as in practice while discussing Vector Tiles with the experts. A potential is to bridge this gap and to focus more on the cartographic aspects that are involved in Vector Tile technology because in practice the focus seems to be too technology- or performance driven. With cartographic aspects, one should not only consider map design but also important cartographic topics such as generalization, projections or topology. This is an aspect that tends to be forgotten or underestimated by some developers that see Cartography only as the design or as a visual science. According to the author, the role of cartography in the Vector Tile workflow is the whole process from generating to rendering Vector Tiles, not only the styling. One could argue that before using the tools the real cartography part comes in. Before using tools the Cartographer has to think about defining what data to put in the map, for who the map is intended and what it will be used for. It is a vital part of cartographic knowledge and decisions that a computer or a tool cannot do. It remains a cartographic knowledge aspect that can be regarded as a potential for Cartographers.

Chapter 9

Conclusion

In this final chapter, the conclusions of this thesis are presented. First, an answer will be given to the research sub-questions, whereafter the central question will be covered. Afterwards, the research limitations and reflections are discussed. These research limitations lead to some recommendations for future research or Vector Tile projects that are in line with the cartographic potential identified in Chapter 7.

9.1 Answering the research sub-questions

The main objective of this thesis was to fill the gap between Computer Sciences and Cartography by combining practical research regarding Vector Tile technology with cartographic theory. The first part was to perform an extensive literature review on the influence of changing technology on cartography. This was needed in order to understand the way maps are nowadays seen and how changing technology influenced cartography and map users. This chapter, therefore, contributes to the understanding of the position of Web Cartography within the cartographic discipline. The first research sub-question was:

What is the influence of recent technological developments in computer- and information science on cartography? (RQ1)

The first part of the theoretical framework, in Chapter 3, revisited the definitions of Cartography. What can be said is that while on one hand cartography tends to be seen only as a visualization phase of spatial data handling, or only as the art or craft of making maps, cartography can, on the other hand, be regarded as a science that covers the entire phases of spatial data handling (Basaraner, 2016; Ramirez, 1993).
Furthermore, it is important to understand that the environment in which maps have been produced and used has changed considerably. The use of maps on the Web can be regarded as a major advancement in cartography and opens many new opportunities (Neumann, 2008). It can be concluded that communication about maps is no longer done by professionals only, but also by a group of non-experts or neogeographers. Where scientists use maps with cartographic knowledge to communicate a message or to solve their problems, neogeographers, non-experts or coders want to grab the tools that are available to create their own maps (Kraak, 2011). Map users have gone from a passive role, as map readers, to a more active role, as contributors in the map mapping process (Ory, 2016).
Besides, maps are being used and created by more people than ever before (Kraak, 2011). The new mapmakers have different demands and objectives, and force a change from a traditional supply-driven map production to demand driven map production. Expertise is still required, however, the role of maps has changed and expanded due to the influence of technology (Kraak, 2009).
The influence of changing technology on cartography has always been there, because the tools used in cartography are also changing, from analog tools towards

computer tools (Bostock & Davies, 2013). A distinction could be made between modern map makers that are coders, and traditional map makers that are expert-cartographers. There seems to be a new role for the cartographer to provide tools that implement cartographic intent. This means that a cartographer could have a role in the toolmaking process or in the knowledge process (Köbben, 2014). It can be concluded that Cartographers should be more involved in the creation of Vector Tile tools to deploy cartographic knowledge. Chapter 4 reviewed the current research available on Vector Tile technology. The corresponding research sub-question was the following:

What are the latest developments regarding Vector Tile technology and what research has already been done on this?(RQ2)

The tiled vector technique is seen as an efficient approach of vector data transmission that enables data reduction of vector datasets. Several studies have researched vector data transmission approaches over the Web, however, there is relatively few research on the Vector Tiling technique. Existing research has studied different Vector Tiling approaches and mentioned the advantages of Vector Tiles compared to other technologies such as raster tiles. It can be concluded that the offloading of the rendering process of Vector Tiles to the client side can offer several benefits, such as faster maps, more interactivity, map design benefits and therefore better user experiences. For this reason, Vector Tiles offer many opportunities for Web Cartographers. However, it was mentioned in the literature studies that with new technological developments it is becoming more challenging to keep up with all the mapping technologies and to maintain overview within the current pace of technological innovation in web mapping (Muehlenhaus, 2014; Roth, Donohue, et al., 2014). The same applies to Vector Tile technology, which is developing fast. Therefore there is is a need for providing an overview of current Vector Tiles tools and technologies available on the Web, which was an answer to the third research sub-question:

What Vector Tiles tools and technologies are currently available on the web and how are they implemented in a workflow? (RQ3)

The fast-paced character of Vector Tile technology and the speed in which Vector Tile tools and technologies develop is overwhelming. The inventory presented in 6 presented an overview of the current Vector Tile tools and technologies available on the Web. One can see that there are different type of tools: Parsers & Generators, Clients, Applications/Plugins, Command-Line Utilities and Servers. The inventory of current Vector Tile tools was presented based on the divide of (1) Mapnik-rendering in the OpenStreetMap ecosystem, (2) WebGIS extended with Vector Tiles and (3) Database-rendering. A description of each tool, some relevant links to documentation and tutorials have been provided in the inventory. The author has tested many of Vector Tools presented to give an indication the current state of the Vector Tile tool. In other words, the aim was to raise awareness of the tools that are implementable for Vector Tile projects.
What can be concluded is that even though there is some overview proposed, it is still hard to keep up with all the different tools. Furthermore, what works for one, might not work for the other. In general, one could express that it remains a challenge to be aware of the possibilities of each tool. One should be willing to put time and effort into reading all the documentation.

Furthermore, it was important to identify where and how the inventoried tools can be implemented in a workflow. Therefore an overview was given of different possible workflows with Vector Tiles while taking into account the different uses cases possible with Vector Tiles: downloading, generating, serving, rendering and styling. The next step of this research was to assess Vector Tiles workflows on their cartographic strengths and weaknesses. Therefore, two example workflows were selected to be assessed based on some cartographic parameters that were found important in the literature or as a result of the findings from the expert-interviews. The research sub-question that covered the cartographic assessment of the workflows was the following:

How to assess Vector Tile tools and solutions on their cartographic strengths and weaknesses? (RQ4)

The Vector Tile strengths and weaknesses in the Vector Tile workflows were assessed base on cartographic parameters: (1) Quality of documentation, (2) Difficulty to implement, (3) Risk of vendor lock-in, (4) Support for different geometries, (5) Handling of generalization, (6) Support for different map projections, (7) Support for different formats and (8) Support for styling. When taking into account all these parameters, it can be concluded from the results that the main cartographic weaknesses identified in the Vector Tile workflows are the lack of control over the simplification process in the Vector Tile tools and the lack of support for custom projections. The major strength identified is the support for Styles. The potential lies in developing the weaknesses and deploying the strengths.

9.2 Answering the central research question

The combination of the research sub-questions answers the main question:

What are the Cartographic implications of Vector Tile technology?

The main objective of this thesis was to investigate the opportunities and challenges of Vector Tile technology as well as the cartographic potential that can be realized in the future. It was researched that Vector Tile technology comes with many advantages for both performance and map design. However, the focus of Vector Tile technology tends to be more on performance or on the technical part in practice. It could be observed that the technological advantages of Vector Tiles, such as fast rendering and small data size, are often found most important or mentioned by the interviewees or the literature. The author believes that it should be about making compromises between both performance and map design. Therefore, it is important to review the role of Cartography for Vector Tile technology since it is not only about displaying geographic data and efficient date encoding, but also about communicating a clear message to the map user. It can be concluded that Vector Tile technology can answer to this need. In fact, the main assumption that the emerging Vector Tile technology positively contributes to the field of cartography is very likely because it stimulates the process of effectively communicating spatial information by means of technological improvements and benefits regarding map design and interactivity, especially on the client side. This means that Vector Tiles effectively communicate a message if the advantages of Vector Tiles for flexible map design are considered. The main benefit in a design point of view is that Vector Tiles enable to remove a lot of elements by means of simplifying and styling which makes the map less complex and more legible and clear for the map reader. Because Vector Tiles are already simplified, and because they contain vector data, the resulted maps have high contrast, high resolution and offer a clear visual hierarchy. Furthermore, they are well suited for Cartography because they are created for the new target group of map users who are no longer map readers, but map users that want to interact and manipulate maps. Vector Tiles respond well to the advantages of interactive maps that enable to have additional 'hidden' information which can be accessed by the map user through interaction (e.g zooming). The ability to have different information on different zoom levels and to style accordingly stimulates a "less is more approach" of effective Web Mapping. With only loading the tiles and information relevant to the user, Vector Tiles offer a good user experience.

While Vector Tile technology has many opportunities for Web Cartographers, there is still an important gap to be bridged. There are many challenges and cartographic topics that can be further researched in Vector Tile technology.
One example is that maps might become less meaningful due to a loss of control by cartographers in the simplification process. It can observed that, currently, the Vector Tile tools do not offer enough control over this process. Another main challenge lies in the support of having custom projections with Vector Tiles. These cartographic challenges form a potential for Web Cartographers to further research. The explanation why these challenges are still present in Vector Tile technology is that a gap between computer sciences and cartography can be observed. On one hand, computer scientists have a lack of cartographic knowledge or underestimate cartographic topics such as generalization and projections, while on the other hand traditional cartographers are not acquainted with programming or technical aspects from computer sciences. This gap can be linked with the distinction found in the literature between modern mapmakers that are coders, and traditional map makers that are expert-cartographers (Roth, Hart, et al., 2014; Van den Berg, 2017). As Van den Berg (2017) mentioned: *"The creation of especially web maps is shifting towards people who know how to code rather than people that know how to correctly design a map"*. This seems also to be the case in practice. Therefore, there seems a lot of work to be done yet for traditional cartographers and modern computer scientists to combine cartographic knowledge with programming/technical knowledge. This means that a cartographer should have a role in the toolmaking process or in the knowledge process (Köbben, 2014). With Vector Tiles, cartographers become more and more developers, thus should be involved in the toolmaking process. However, one could also say that it is not about the tools, but it is more about the knowledge, both cartographic and programming. The author believes that nowadays it is not possible anymore to avoid the technical part for cartographers. This is in line with what Köbben (2014) mentioned, that there is a need for *"code that thinks like a cartographer"*. The real cartographic part, which should be seen as both a challenge and potential, lies in the knowledge process before using the tools. For example, deciding what data to simplify before generating Vector Tiles. Developers with a computer sciences background should take into account should take into account more the cartographic aspects in Vector Tile technology.
It can be concluded that bridging the gap between computer sciences and cartography was the aim of this thesis and at the same time the identified potential that can be deployed.

The next section discusses the research limitations. The research limitations lead to new potential topics or aspects to further investigate.

9.3 Research limitations and reflection

Although the aim of this thesis was to bridge the gap between computer sciences and cartography, it was not possible to solve this entirely, because the gap is still there. The thesis raised awareness of the gap and proposed some aspects of cartographic potential, however, it would have been of scientific relevance writing the thesis together with a computer science student. This could have led to more practical results, instead of exploratory research and a identified potential. This is considered as a missed chance, but an opportunity for future cartography and computer scientists to do so.

The main challenge was to take into account cartographic topics or theory, which did not always match with the very technical or performance driven technology of Vector Tiling. It was difficult to come up with cartographic criteria on which to assess cartographic advantages or disadvantages of Vector Tiles. The traditional theory regarding the visual variables by Jacques Bertin was difficult to relate to Web Cartography or Vector Tile technology.
Besides, the research could have taken more cartographic parameters into account, such as the handling of attributes or preservation of topology. The choice of cartographic parameters was to some extent arbitrary, being a limitation of the assessment on the cartographic strengths and weaknesses in the Vector Tile workflows.

Furthermore, it was hard to keep the scope of this research. The topic is very broad, and even during the interviews, it was difficult to stay with the topic of cartography. This was also due to the fact that often performance or technical aspects were mentioned by the interviewees, which is not surprising since the main use and advantage of Vector Tiles is for speed because of small data storage size.
It was also hard to keep the scope on the meaning of Vector Tile technology for Web Cartographers during the interviews because it was often referred to the opportunities of Vector Tiles for consumers and end-users. Vector Tiled (base)maps were often seen as a product or service for end-users, whereas the focus in this thesis lies on the opportunities and potential of Vector Tiles for Cartographers. This was sometimes forgotten and it was also due to some confusions about who the 'user' is of Vector Tiles. The user of Vector Tiles could be a developer, a cartographer or a consumer/end user. This was not always clear and reflected by the different views and focus on that aspect by the interviewees.

Another research limitation is that the developments in Vector Tile technology change very fast, and especially the Vector Tile tools, meaning that in a few years or even months, the tools could have already been further developed or deprecated. This means that the provided inventory of Vector Tile tools and solutions could be outdated when reading this thesis. A specific tool was perhaps implementable on the moment of writing this thesis, but not anymore afterwards. An example was Mapzen that shut down its services on February 1, 2018. This was just in the middle of inventorying Vector Tile tools in Chapter 6.It should also be noted that ESRI and PDOK released Vector Tile services which are still in beta. This means that interesting outcomes or feedback was not taken into account in this thesis. Once they are out rolled, this could give more insights not being mentioned here. Moreover, due to time constraints, and a lack of expertise by the author, not all Vector Tile tools inventoried in this thesis could be tested or fully deployed.

An aspect to mention is that this thesis consisted of exploratory and qualitative research, which leads to some limitations that the findings of the interviews were often generalized where in reality it was not always possible to draw hard conclusions. Both the opinions of the interviewees as well as the assessment of Vector Tile solutions on their cartographic strengths and weaknesses were therefore to some extent very subjective or arbitrary.

When considering the personal reflection of the author it was difficult to conceptually understand how Vector Tiling works. Furthermore, there was some confusion about the use of terms or terminology. The author had trouble to define all the different terms in the available documentation and specification. Besides, it was not always clear what a Vector Tile solution was because it can be a tool or the workflow. Therefore, a glossary was added to this thesis to clarify some concepts.

9.4 Recommendations for future research or projects

The recommendations for future research or projects can be drawn from the research limitations and wishes of the expert interviewees regarding Vector Tile technology. The recommendations, therefore, go almost hand in hand with the cartographic potential of Vector Tile technology identified in Chapter 7.6.
In general, one could state that the tools have made it very easy for Web Cartographers to implement Vector Tiles, however, there are still some remaining issues regarding Vector Tile technology that need further research. Therefore, there are some recommendations for future research:

- More research on how to combine cartographic knowledge with technical knowledge regarding Vector Tile technology. The potential is to combine both fields of computer sciences and cartography. How can this best be tackled? How to stimulate Cartographers to do research on Vector Tiles? How to stimulate developers that have a background in computers science to take into account cartographic aspects of Web Mapping with Vector Tiles and not only focus on the technology, on performance or tools. A recommendation is to have more projects with both computer scientists and cartographers involved.

- More research on the current cartographic weaknesses of Vector Tile technology. How to gain more control over the simplification process in Vector Tile tools? How to implement and support custom projections with Vector Tiles? How to tackle the labeling challenges of Vector tiles (e.g. repetition of labels or geometries based on shared features in multiple tiles)? How to design good styles for Vector Tile basemaps? How to improve or standardize styling languages? etc.

- More research on the use and opportunities of Vector Tile technology for end-users, customers and consumers. This research was focused on the opportunities of Vector Tile technology for cartographers. However, it could also be interesting to do research on how end-users experience Vector Tiles.

- More meetings and projects to discuss Vector Tiles and to improve current documentation.
- More implementations of Vector Tiles and more open source clients and editors for Vector Tiles.

References

Alesheikh, A. A., Helali, H., & Behroz, H. (2002). Web GIS: technologies and its applications. In *Symposium on geospatial theory, processing and applications* (Vol. 15). Retrieved from https://www.researchgate.net/publication/228714608_Web_GIS_Technologies_and_its_applications

Antoniou, V., Morley, J., & Haklay, M. M. (2009). Tiled vectors: A method for vector transmission over the web. In *International symposium on web and wireless geographical information systems* (pp. 56–71). Springer.

Ballatore, A., Tahir, A., McArdle, G., & Bertolotto, M. (2011). A comparison of open source geospatial technologies for web mapping. *International Journal of Web Engineering and Technology*, *6*(4), 354. Retrieved from http://www.inderscience.com/link.php?id=43440

Basaraner, M. (2016). Revisiting cartography: towards identifying and developing a modern and comprehensive framework. *Geocarto International*, *31*(1), 71–91. Retrieved from www.tandfonline.com

Bertolotto, M., & Egenhofer, M. J. (2001). Progressive Transmission of Vector Map Data over the World Wide Web. *GeoInformatica*, *5*(4), 345–373. Retrieved from https://pdfs.semanticscholar.org/75dc/de425c6e62cb4c11eb8ae60847e2bad12a0b.pdf

Bertolotto, M., & McArdle, G. (2011). Data reduction techniques for web and mobile GIS. *Advances in Web-based GIS, Mapping Services and Applications*, *9*, 139.

Bocher, E., & Ertz, O. (2017). A redesign of OGC Symbology Encoding standard for sharing cartography. Retrieved from https://peerj.com/preprints/2415.pdf

Boeije, H., 't Hart, H., & Hox, J. (2009). *Onderzoeksmethoden*. Den Haag: Boom Lemma Uitgevers.

Boeijen, N. (2018a, January 19th). *Personal communication at Kadaster in Apeldoorn.*

Boeijen, N. (2018b, February 6th). *Personal communication/interview at Webmapper in Utrecht.*

Boeijen, N. (2018c, February 19th). *Personal e-mail communication.*

Boeijen, N. (2018d, February 26th). *Personal e-mail communication.*

Bostock, M., & Davies, J. (2013). Code as Cartography. *The Cartographic Journal*, *50*(2), 129–135. Retrieved from www.tandfonline.com

Boundless. (2017). *Using Mapbox Style objects OpenLayers.* Retrieved from https://boundlessgeo.com/2017/01/using-mapbox-style-objects-open-layers/ (Date accessed: 24-02-2018)

Brown, A. (1993). Map design for screen displays. *The Cartographic Journal*, *30*(2), 129–135.

Butkiewicz, M., Madhyastha, H. V., & Sekar, V. (2011). Understanding website complexity: measurements, metrics, and implications. In *Proceedings of the 2011 acm sigcomm conference on internet measurement conference* (pp. 313–328).

Carpendale. (2003). Considering Visual Variables as a basis for Information Visualisation. , 175–188. Retrieved from https://prism.ucalgary.ca/bitstream/1880/45758/2/2001-693-16.pdf

Cartwright, W., Gartner, G., Meng, L., & Peterson, M. P. (2010). *Lecture Notes in Geoinformation and Cartography.* Retrieved from www.springer.com

Cartwright, W., Peterson, M. P., & Gartner, G. (2007). *Multimedia Cartography*. Heidelberg Berlin: Springer.

Ceaparu, I., Lazar, J., Bessiere, K., Robinson, J., & Shneiderman, B. (2004). Determining causes and severity of end-user frustration. *International journal of human-computer interaction*, *17*(3), 333–356.

DiBiase, D., MacEachren, A. M., Krygier, J. B., & Reeves, C. (1992). Animation and the role of map design in scientific visualization. *Cartography and geographic information systems*, *19*(4), 201–214.

Dillemuth, J. (2005). Map design evaluation for mobile display. *Cartography and Geographic Information Science*, *32*(4), 285–301.

Douglas, D. H., & Peucker, T. K. (2011). Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Classics in Cartography: Reflections on Influential Articles from Cartographica*, 15–28.

Dutta, R., & Jayasinghe, S. (2015). Integrating Spatial Data Infrastructure in Monitoring Climate Change Impacts: A Research Oriented Approach. *Approach (TS-3: SDI Development*.

Fernández, P. I. A., & Buchroithner, M. F. (2013). *Paradigms in cartography: An epistemological review of the 20th and 21st centuries*. Springer Science & Business Media.

Gaffuri, J. (2012). Toward web mapping with vector data. *Geographic information science*, 87–101.

García, R., de Castro, J. P., Verdú, E., Verdú, M. J., & Regueras, L. M. (2012). Web map tile services for spatial data infrastructures: Management and optimization. In *Cartography-a tool for spatial analysis.* InTech.

Gardner, B. S. (2011). Responsive web design: Enriching the user experience. *Sigma Journal: Inside the Digital Ecosystem*, *11*(1), 13–19.

Gardner, James. (2018). *Build Your Own Static Vector Tile Pipeline.* Retrieved from `https://geovation.github.io/build-your-own-static-vector-tile-pipeline` (Date accessed: 25-02-2018)

Geofabrik. (2018). *Geofabrik Downloads.* Retrieved from `http://download.geofabrik.de/` (Date accessed: 23-02-2018)

GIS Wiki HSR. (2018). *Vector Tiles.* University of Applied Sciences Rapperswil. Retrieved from `https://giswiki.hsr.ch/Vector_Tiles` (Date accessed: 11-01-2018)

Github Mapbox. (2018). *Awesome-vector-tiles, implementation of The Mapbox Vector Tile specification.* Retrieved from `https://github.com/mapbox/awesome-vector-tiles` (Date accessed: 02-12-2017)

Github Mapbox Tippecanoe. (2018). *Mapbox Tippecanoe Specification.* Retrieved from `https://github.com/mapbox/tippecanoe` (Date accessed: 25-02-2017)

Github Tilemaker. (2018). *Vector Tiles.* Retrieved from `https://github.com/systemed/tilemaker` (Date accessed: 23-02-2018)

Glaser, B., & Strauss, A. (1967). Grounded theory: The discovery of grounded theory. *Sociology The Journal Of The British Sociological Association*, *12*, 27–49.

Goodchild, M. (2007). *Citizens as voluntary sensors: spatial data infrastructure in the world of Web 2.0. IJSDIR 2: 24–32.*

Google Maps. (2018). *Google Maps API Styling Wizard.* Retrieved from `https://mapstyle.withgoogle.com/` (Date accessed: 06-02-2017)

Griffin, A. L., Robinson, A. C., & Roth, R. E. (2017). Envisioning the future of cartographic research.

Haklay, M. Singleton, A., & Parker, C. (2008). Web mapping 2 . 0 – the emergence of the GeoWeb. *Geography Compass*, *6*(2008), 2011–2039.

Hardy, P., & Field, K. (2011). Portrayal and cartography. In *Springer handbook of geographic information* (pp. 179–190). Springer.

Held, G., Rahman, A., & Zlatanova, S. (2004). Web 3D GIS for Urban Environments.

Hogeboom, J. (2018, February 8th). *Personal communication at Kadaster Apeldoorn.*

Hopfstock, A., & Grünreich, D. (2009). A user-oriented map design in the sdi environment using the example of a european reference map at medium scale. In *Proceedings of the 24th international cartographic conference.* Santiago de Chile, Chile: ICC.

Ingensand, J., Nappez, M., Moullet, C., Gasser, L., Ertz, O., & Composto, S. (2015). Vector tiles for the Swiss Federal Geoportal. *Geomatics Workbooks n° 12*, 717–124. Retrieved from `http://geomatica.como.polimi.it/workbooks/n12/FOSS4G-eu15_submission_143.pdf`

Ingensand, J., Nappez, M., Moullet, C., Gasser, L., Ertz, O., & Composto, S. (2016). Implementation of Tiled Vector Services: A Case Study. In *SDW@ GIScience* (pp. 26–34).

Jenny, B., Jenny, H., & Räber, S. (2008). Map design for the Internet. In *International perspectives on maps and the internet* (pp. 31–48). Springer.

Kalberer, P. (2017). *Vector Tiles - Introduction & Usage with QGIS.* Sourcepole AG, Zürich. Retrieved from `http://t-rex.tileserver.ch/Vector-tiles-and-QGIS.pdf`

Kang, Y.-K., Kim, K.-C., & Kim, Y.-S. (2001). Probability-Based Tile Pre-fetching and Cache Replacement Algorithms for Web Geographical Information Systems. *LNCS*, *2151*, 127–140. Retrieved from `https://pdfs.semanticscholar.org/e7f5/636a075f622c761c61384ee994b79d9cccb4.pdf`

Keller, S. (2018, February 8th). *Personal online communication/interview.*

Kraak, M.-J. (2009). Internet - Web mapping. In *International encyclopedia of human geography : pt. cartography and GIS* (pp. 562–568). Amsterdam, The Netherlands: Elsevier.

Kraak, M.-J. (2011). Is there a need for neo-cartography. *Cartography and geographic information science*, *38*(2), 73–78.

Kraak, M.-J., & Brown, A. (2001). *Web cartography : developments and prospects*. London New York: Taylor & Francis.

Kresse, W., & Danko, D. M. (2012). *Springer handbook of geographic information*. Springer Science & Business Media.

Köbben, B. (2010). *A short introduction to Geo-webservices: Lecture note on the history and principles of Geo-webservices.* International Institute for Geo-information Science Earth Observation (ITC).

Köbben, B. (2014). *Cartography from code. . . ? Lecture at FOSS4G FDX Conference.* ITC University of Twente. Retrieved from `https://vimeo.com/106848055` (Date accessed: 02-10-2017)

Lambert, N., & Zanin, C. (2016). *Manuel de cartographie: principes, méthodes, applications*. Armand Colin.

Li, L., Hu, W., Zhu, H., Li, Y., & Zhang, H. (2017). Tiled vector data model for the geographical features of symbolized maps. *PloS one*, *12*(5), e0176387.

Li, S., Veenendaal, B., & Dragićević, S. (2011). *Advances in web-based GIS, mapping services and applications.* Taylor & Francis Group, London. Retrieved from `http://www.crcnetbase.com/doi/10.1201/b11080-3`

Lienert, C., Jenny, B., Schnabel, O., & Hurni, L. (2012). Current trends in vector-based Internet mapping: A technical review. In *Online maps with apis and webservices* (pp. 23–36). Springer.

Loechel, A., & Schmid, S. (2013). Caching techniques for high-performance Web Map

Services. *Press) In: International Journal of Spatial Data Infrastructures Research*.

Lopez, E., Béjar, R., Barrera, J., Lopez-Pellicer, F. J., Rodríguez, A. F., & Abad, P. (2017). Support for vector tiles in INSPIRE view services. Retrieved from https://inspire.ec.europa.eu/sites/default/files/presentations/INSPIRE2017_VectorTiles.pdf

MacEachren, A. (1995). *How Maps Work: Representation, Visualization, and Design*. The Guilford Press.

Mac Gillavry, E. (2017, December 20th). *Personal e-mail communication.*

Mac Gillavry, E. (2018a, January 19th). *Personal communication at Kadaster Apeldoorn.*

Mac Gillavry, E. (2018b, February 6th). *Personal communication/interview at Webmapper.*

Mapbox. (2017a). *Mapbox GL JS.* Retrieved from https://www.mapbox.com/mapbox-gl-js/api/ (Date accessed: 12-10-2017)

Mapbox. (2017b). *Vector Tiles.* Retrieved from https://www.mapbox.com/vector-tiles/ (Date accessed: 12-10-2017)

Mapbox. (2018a). *About Mapbox.* Retrieved from https://www.mapbox.com/about/ (Date accessed: 06-01-2018)

Mapbox. (2018b). *Pricing Mapbox.* Retrieved from https://www.mapbox.com/pricing/ (Date accessed: 23-02-2018)

Mapcat. (2018). *Mapcat documentation.* Retrieved from https://docs.mapcat.com/ (Date accessed: 14-02-2018)

Mapdata Services. (2017). *Vector tiles vs Raster tiles - the pros and cons.* Retrieved from https://mapdataservices.wordpress.com/2017/02/22/vector-tiles-vs-raster-tiles-the-pros-and-cons/ (Date accessed: 22-11-2017)

Mapzen. (2018). *Mapzen Services Alternatives.* Retrieved from https://mapzen.com/blog/migration/ (Date accessed: 03-01-2018)

Martinelli, L. (2018, February 13th). *Personal online communication/interview.*

Martinelli, L., & Roth, M. (2015). *Vector Tiles from OpenStreetMap* (Unpublished doctoral dissertation). HSR Hochschule für Technik Rapperswil.

Mitchell, T. (2005). *Web mapping illustrated*. O'Reilly Media, Inc.

Muehlenhaus, I. (2014). *Web cartography: map design for interactive and mobile devices*. CRC Press.

Nasirzadeh Dizaji, R., & Nurhan Çelìk, R. (2015, 10). Open source geo-information technology for making special purpose web-mapping application. *Coordinates*, *11*(10).

Neumann, A. (2008). Web mapping and web cartography. In *Encyclopedia of GIS* (pp. 1261–1269). Springer.

Nordan, R. P. V. (2012). *An investigation of potential methods for topology preservation in interactive vector tile map applications* (Unpublished master's thesis). NTNU, Institutt for bygg, anlegg og transport.

Nordan, R. P. V. (2015). *Stuffing your vector tiles full of data - Presentation at FOSS4G Seoul.* Retrieved from https://vimeo.com/142335630 (Date accessed: 04-01-2017)

Nordan, R. P. V. (2018, February 5th). *Personal online communication.*

Norman, P. (2016). Serving Vector Tiles. Retrieved from http://www.paulnorman.ca/blog/2016/11/serving-vector-tiles/

Norman, P. (2018). *Overview of vector tiles.* Retrieved from https://www.youtube.com/watch?v=savQWL0kq_g (Date accessed: 23-02-2018)

Open Geospatial Consortium. (2017a). *OGC Testbed-12: Vector Tiles Engineering Report.* Retrieved from http://www.opengis.net/doc/PER/t12-A008 (Date accessed: 10-01-2017)

Open Geospatial Consortium. (2017b). *OGC Testbed-13: Vector Tiles Engineering Report (Draft).* Retrieved from `http://www.opengis.net/doc/PER/t13-ID` (Date accessed: 09-01-2017)

OpenMapTiles. (2018a). *About OpenMapTiles.org project.* Retrieved from `https://openmaptiles.org/about/` (Date accessed: 16-01-2017)

OpenMapTiles. (2018b). *OpenMapTiles Coordinate Systems.* Retrieved from `https://openmaptiles.com/coordinate-systems/` (Date accessed: 25-02-2018)

OpenMapTiles. (2018c). *OpenMapTiles Downloads.* Retrieved from `https://openmaptiles.com/downloads/planet/` (Date accessed: 23-02-2018)

Ormeling, F. (2009). From Ortelius to OpenStreetMap–Transformation of the Map into a Multifunctional Signpost1. In *Cartography in Central and Eastern Europe* (pp. 1–16). Springer.

Ormeling, F., & Kraak, M.-J. (2010). *Cartography - Visualization of Spatial Data*. Harlow New York: Prentice Hall.

Ory, J. (2016). *Connaissances pour la conception et la perception de styles topographiques.* Université Paris-Est.

Ostlaender, N., Smith, R. S., De Longueville, B., & Smits, P. (2010). What Volunteered Geographic Information is (good for)-designing a methodology for comparative analysis of existing applications to classify VGI and its uses. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International* (pp. 1422–1425).

PDOK. (2018a). *About PDOK.* Retrieved from `https://www.pdok.nl/en/about-pdok` (Date accessed: 19-02-2018)

PDOK. (2018b). *PDOK Forum: Vector Tiles BRT en BGT via PDOK, discussion with Frank Steggink about labels.* Retrieved from `https://forum.pdok.nl/t/vector-tiles-brt-en-bgt-via-pdok/1103/10` (Date accessed: 20-02-2018)

Peterson, M. P. (2003). *Maps and the Internet*. Elsevier.

Peterson, M. P. (2008). *International perspectives on maps and the Internet*. Heidelberg Berlin: Springer.

Quinn, S., & Gahegan, M. (2010). A predictive model for frequently viewed tiles in a web map. *Transactions in GIS*, *14*(2), 193–216.

Ramirez, J. R. (1993). Development of a cartographic language. In *European Conference on Spatial Information Theory* (pp. 92–112). Springer. Retrieved from `https://link.springer.com/content/pdf/10.1007/3-540-57207-4{_}8.pdf`

Ramsay, J., Barbesi, A., & Preece, J. (1998). A psychological investigation of long retrieval times on the World Wide Web. *Interacting with computers*, *10*(1), 77–86.

Reed, C. (2011). OGC standards: Enabling the geospatial web. In *Advances in web-based GIS, mapping services and applications.* Taylor & Francis Group, London.

Roth, R. (2012). Cartographic interaction primitives: Framework and synthesis. *The Cartographic Journal*, *49*(4), 376–395.

Roth, R. (2013). Interactive maps: What we know and what we need to know. *Journal of Spatial Information Science*, *2013*(6), 59–115.

Roth, R. (2017). User Interface and User Experience (UI/UX) Design. *The Geographic Information Science & Technology Body of Knowledge (2nd Quarter 2017 Edition).*

Roth, R., Donohue, R., Sack, C., Wallace, T., & Buckingham, T. (2014). A process for keeping pace with evolving web mapping technologies. *Cartographic Perspectives*, *2014*(78), 25–52.

Roth, R., Hart, D., Mead, R., & Quinn, C. (2014). Design before you code: Using wireframes in support of interactive and web-based mapping.

Roth, R., Ross, K. S., & MacEachren, A. M. (2015). User-centered design for interactive maps: A case study in crime analysis. *ISPRS International Journal of Geo-Information*, *4*(1), 262–301.

Sample, J. T., & Ioup, E. (2010). *Tile-based geospatial information systems: principles and practices*. Springer Science & Business Media.

Schmidt, M., & Weiser, P. (2012). Online Maps with APIs and WebServices. *Springer-Verlag Berlin Heidelberg*, 10. Retrieved from `http://www.springerlink.com/index/10.1007/978-3-642-27485-5`

Schnabel, O., & Hurni, L. (2009). Cartographic web applications–developments and trends. In *Proceedings of the 24th international cartography conference, Santiago.*

Shang, X. (2015). *A Study on Efficient Vector Mapping With Vector Tiles Based on Cloud Server Architecture* (Unpublished doctoral dissertation). University of Calgary.

Skadberg, Y. X., & Kimmel, J. R. (2004). Visitors' flow experience while browsing a Web site: its measurement, contributing factors and consequences. *Computers in human behavior*, *20*(3), 403–422.

Sousa, Pedro. (2017). *Playing with Mapbox Vector Tiles, Part 1: End-to-end experiment.* Retrieved from `http://build-failed.blogspot.nl/2017/02/playing-with-mapbox-vector-tiles-part-1.html` (Date accessed: 23-02-2018)

Taylor, D. (1994). Cartography for knowledge, action and development: retrospective and prospective. *The Cartographic Journal*, *31*(1), 52–55.

Taylor, S. J., Bogdan, R., & DeVault, M. (2015). *Introduction to qualitative research methods: A guidebook and resource*. John Wiley & Sons.

Thakker, Anand. (2017). *Dirty Reprojectors: How to trick your web maps into non-mercator projections.* Retrieved from `https://developmentseed.org/blog/2016/12/15/dirty-reprojectors/` (Date accessed: 25-02-2018)

Tomlin, C. D. (1990). *Geographic information systems and cartographic modelling* (No. 910.011 T659g). New Jersey, US: Prentice-Hall.

Van den Berg, J. (2017). *Towards a Dynamic Isochrone Map: Adding Spatiotemporal Traffic and Population data* (Unpublished master's thesis). Geographical Management and Applications, GIMA Msc, The Netherlands.

Vierbergen, W. J. (2018, February 7th). *Personal online communication/interview at ESRI Nederland.*

Visscher, W. (2018, January 19th). *Personal communication at Kadaster Apeldoorn.*

Visvalingam, M., & Whyatt, J. (1992). Line generalisation by repeated elimination of the smallest area. *Cartographic Journal*, *30*(1), 46–51.

Wiki OpenStreetMap. (2018). *Mapnik.* Retrieved from `https://wiki.openstreetmap.org/wiki/Mapnik` (Date accessed: 03-01-2018)

Wikipedia. (2018a). *Animated mapping.* Retrieved from `https://en.wikipedia.org/wiki/Animated_mapping` (Date accessed: 19-11-2017)

Wikipedia. (2018b). *Geographic information system.* Retrieved from `https://en.wikipedia.org/wiki/Geographic_information_system` (Date accessed: 25-02-2018)

Wikipedia. (2018c). *Vector tiles.* Retrieved from `https://en.wikipedia.org/wiki/Vector_tiles` (Date accessed: 03-01-2018)

Williams, Craig and Punt, Edie. (2016). *Video recording of ESRI Event: Desktop Mapping Creating Vector Tiles.* Retrieved from `https://www.youtube.com/watch?v=L2ds2rb-d5c` (Date accessed: 02-10-2017)

Wood, M. (2003). Cartography in the age of geographic information science. *Geograficky casopis slovenskej akademie vied*, *55*, 267–278.

Yang, B., Purves, R., & Weibel, R. (2007). Efficient transmission of vector data over the Internet. Retrieved from `www.tandfonline.com`

Appendix A

Appendix: Interview topic list for Expert-interviews

1. Introduction

- Personal introduction
- Background information thesis research
- The role of the interviewee (Explain why this person was asked for the interview)
- Interview procedure (Ask permission for audio recording and other formalities)

2. Vector Tile technology and Cartography

1. ***Users and use***:
    - What is your experience with Vector Tile technology? Are you aware of the current state of the technology or its opportunities?
    - Are you currently using a specific vector tile solution? How are you involved with the vector tile technology? Do you consider yourself as a producer or consumer?
    - Who do you see as the users of vector tiles are in general? Can you identify some groups?
    - What are according to you the needs/requirements of producers regarding vector tiles?
    - What are according to you he needs/requirements of end-users regarding vector tiles?
    - What do you aim at or want to do with vector tiles? What are your requirements? How do you intend vector tiles to be used?
    - To what extent to do you believe that vector tiles are primary used for visualization and not for analysis?
2. ***Knowledge and learning curve***:
    - How do you consider your knowledge about vector tile technology? Are you aware of the theoretical advantages and challenges of vector tiles? What are they according to you?
    - Do you consider it difficult to learn how vector tiles work and to gain skills on how to use the technology?
    - Is there something you wish to learn more about regarding vector tiles?

- According to you, how difficult is it for the producer to produce vector tiles?
- According to you, how difficult is it for the consumer to consumer vector tiles?
- Do you agree with the following statement "There is a lack of cartographic knowledge, that's why vector tiles are not mainstream yet"?

3. ***Stability***:

   - Do you consider vector tile technology to be stable? Well developed?
   - How stable and reliable are vector tiles solutions on the market according to you? Do they match your needs/requirements?

4. ***Performance***:

   - Is vector tile technology performing good enough for your needs?
   - In your opinion, do vector tiles score better in performance than other web mapping technologies or solutions?
   - Do you consider performance to be more important than cartographic rules or proper map design (especially if you look at what your goal is with vector tiles) ?

5. ***Scalability & Extendibility***:

   - Do you believe the vector tile technology is able to grow or to become more important?
   - Can it handle specific use cases that can not be done with other web mapping technologies?
   - Do you agree with the following statement: "Vector tiles implies a complex and resource demanding infrastructure?"
   - Do you think that the vector tile technology can be extended with other new emerging technologies?

6. ***Interoperability***:

   - How do you consider the integration of different vector tile tools or solutions?

7. ***Formats, standards and schemes***:

   - What do you think of the different vector tile formats? Is the variety of formats a disadvantage or advantage? Is there a specific format that fits best your needs?
   - Is the fact that there are no current standards seen as a major problem for the use of vector tiles for cartographic purposes?
   - In terms of different tiling schemes, what is for you the best approach in addressing vector tiles?
   - What do you think of the ability to make or have standards for vector tiles?

8. ***Documentation and accessibility***:

- Is there according to you enough documentation and knowledge to find about vector tile technology?
- What about the vector tile solutions and their documentation? Are they hard to find?
- Is documentation accessible? Can anyone be involved with vector tiling web mapping? What about costs?

9. ***Community support***:
   - Is there enough support for vector tiles in the field of GIS and Cartography?
   - Is the technical support offered for vector tile solutions meeting your needs, or good enough?

10. ***Frequency of updates***:
    - Are vector tile solutions and its documentation well updated?
    - What do you think about the pace in which vector tile tools and technologies develop? Does it form a challenge to keep up with all the vector tile solutions for you?

11. ***Cartographic projections***:
    - For your use of vector tiles, to what extend to cartographic projections play a role?
    - How important is the use of different or a specific projection for your use of web maps?

12. ***Map design and styling***:
    - How do you see the advantages of vector tiles for styling?
    - Can you think of any styling challenges or disadvantages with vector tiles?
    - Do maps using vector tiles offer a clear visual hierarchy? Does it help to draw attention to certain elements of the map and push those of less importance further down? Can map elements be well differentiated?
    - Do maps using vector tiles stimulate do send a clear message to the map reader/consumer? Are they legible, understandable? Recognizable? Do they offer good composition?
    - Do you agree with the following statement: "Vector tiles rendered in a browser means yet another new styling language", is everyone capable or willing to get along with this new styling language?
    - Do you think that consumers are all capable of changing the style on the fly of vector tiles?

13. ***Geometry***:
    - What do you think of vector tiles and the importance of support of different geometries?
    - Are attributes or attribute data more important than geometry?

- How important is analysis for you? Or is the main goal visualization or do you also wish or need to do operations/calculations/computational tasks?

14. ***Preservation of topology***:
    - What challenges or opportunities do you see for the preservation of topology with vector tile technology?

15. ***Simplification and generalization***:
    - What challenges or opportunities do you identify regarding vector tiles and generalization?

16. ***Future possibilities and expectations***:
    - What can be done with vector tiles that has not been done yet? What do you think is the role of vector tiles for cartography? Can it become more important? Expectations and future of vector tile technology.
    - Can anyone be involved with vector tiling web mapping?

3. Conclusion

- Summarize findings
- Completing the interview & thank