MSc thesis in Geomatics

Integrating 3D Functionality into a Web Application for Sharing Geo-information

Stein Köbben

June 11, 2024

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

Stein Köbben: *Integrating 3D Functionality into a Web Application for Sharing Geo-information* (2024)

Supervisors:	Dr.ir. Martijn Meijers (Delft University of Technology)
	Dipl.-Ing. Eftychia Kalogianni (Delft University of Technology)
External Supervisor:	Erik Meerburg (B3Partners)
Co-reader:	Dr. Mila Koeva (University of Twente)

Abstract

Geo-information has many applications in various sectors and is used by public agencies as well as the private sector. A key method for sharing and visualising geo-information is through web viewers. Extensive infrastructure exists for creating web viewers to publish 2D geo-information without the need of coding, much of which is based on OpenLayers.

3D geo-information is becoming increasingly important for many use cases, such as urban planning, asset management, and disaster management. Consequently, there is an increasing need to publish 3D geo-information in web viewers without the need of coding. Since applications to do this for 2D geo-information are widely in use, it would be useful to be able to effectively integrate 3D functionality into an existing application.

This research investigates the needs for integrating 3D functionality into an existing OpenLayers based GIS web application that allows the user to publish geo-information in web viewers. This includes both the technical needs, e.i. how to integrate the necessary code, and the user needs, e.i. what features does the user need to make the application useful and user-friendly.

A proof of concept was developed to investigate the technical needs. The proof of concept is an extension of the existing application Tailormap. Tailormap is based on OpenLayers and allows users to publish geo-information in web viewers without the need of coding. 3D functionality is integrated into Tailormap to allow the users to publish 3D vector data, such as buildings, alongside 2D OpenLayers maps. This integration was achieved by incorporating Cesium functionality to load and visualise 3D Tiles datasets.

To ascertain the user needs, interviews were conducted with Tailormap users and the created proof of concept was tested with them to gather feedback. The interviews and testing sessions revealed that the most critical type of 3D data for publishing in a web viewer was vector data, specifically buildings. Furthermore, specific functions were identified that make the application useful for the users.

The developed technical methodology and the identified user needs are primarily applicable to integrating 3D functionality into Tailormap but can also serve as guidelines for integrating 3D functionality into other OpenLayers-based GIS web applications for publishing geo-information in web viewers.

Acknowledgements

This master's thesis is the final work in my Geomatics studies. I have enjoyed this project a lot and I learned more than I could have imagined. I want to take the opportunity to express my gratitude to everyone who helped me during this project. First, I want to thank my supervisors from the TU Delft Martijn Meijers and Eftychia Kalogianni. They have guided me through the entire process of the master's thesis, and they provided a lot of feedback and ideas. This project would not have been possible without them. I want to thank Mila Koeva for all the useful feedback as the external reader.

I also want to thank my external supervisor Erik Meerburg. He organised the collaboration with B3Partners which made the project possible, and he added a lot of value and knowledge to the process. I want to thank everyone at B3Partners for the opportunity and for all the support. I got all the help I needed with understanding Tailormap and implementing the proof of concept.

I would like to express my gratitude to all the people I have interviewed at the municipality of Gouda, the safety region Brabant Zuid-Oost, Stantec, Antea, and the Kadaster. They took the time to help me with my user research and they expressed a lot of enthusiasm and interest, which gave me a lot of extra motivation.

Lastly, I want to thank all of my friends and family. They gave me a lot of support and also provided some much-needed fun and distraction during my thesis. I especially want to thank my parents, my sister, and my girlfriend for all the support.

Thank you all.

Contents

List of Figures

List of Tables

Acronyms

2D	two-dimensional
3D	three-dimensional
AHN	Actueel Hoogtebestand Nederland
BIM	Building Information Modeling
CSS	Cascading Style Sheets
DSM	Digital Surface Model
DTM	Digital Terrain Model
GIS	Geographical Information System
HTML	HyperText Markup Language
I3S	Indexed 3D Scene Layers
ID	identifier
JSON	JavaScript Object Notation
LoD	Level of Detail
OGC	Open Geospatial Consortium
TMS	Tile Map Service
UML	Unified Modeling Language
URL	Uniform Resource Locator
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tiling Service

1. Introduction

1.1. Motivation and Context

Geo-information has many applications in various different sectors and is used by public agencies as well as the private sector Castelein et al. [2010]. One important way to share geo-information is through the web. With web GIS applications it is easy to share maps and make them accessible to anyone. It allows people to view the geo-information without the need of file transfers and specific software. Web GIS applications were originally focused on 2D geo-information, but web GIS for 3D geo-information is a topic with rapidly increasing interest. Agrawal and Gupta [2017]

The interest in 3D geo-information has grown a lot over the last few years. More and more use cases and applications are being developed, such as asset management, 3D cadastre, and urban planning Coote et al. [2017]. It has become clear that 3D geo-information is becoming an important tool in many fields of study and private sector developments Biljecki et al. [2015]. Research by EuroSDR has shown that the use of 3D geo-information can be economically viable Coote et al. [2017], and a continuation on this study has shown it creates public value Ho et al. [2018].

For these reasons, sharing 3D geo-information through web viewer is a natural development in the field of GIS. There already existing examples of 3D geo-information being shared through web viewers, such as Kilden Nibio Nibio [2023] which shares a terrain model of Norway, the BRO 3D I3S web services by TNO TNO [2023] which visualises soil models of the Netherlands, and the 3DBAG viewer TUDelft3d and 3DGI [2023] which visualises the buildings in the Netherlands. However, the existing examples are mostly made by experts in the field of GIS or they are not based on open source software.

In the last decade, there is research being carried out on the development of web platforms that facilitate the dissemination of 3D geo-information (Mete et al. [2018], Guardia and Koeva [2023]). Although more and more studies and prototypes are developed in this direction (Vincent Jaillot and Gesquière [2020], Brovelli et al. [2018]), 3D web platforms for publishing geo-information based on user needs are not explored yet in detail. Publishing, in the context of this thesis research, means using existing datasets of geo-information to create and share a web viewer that shows this data without the need of coding. This includes selecting the data sources, setting up a web viewer with layers from these sources, and possibly adding some styling or selecting which tools and options are available in the viewer. The user, in the context of this thesis, is the one who publishes data in a web viewer.

There is already existing infrastructure for publishing 2D geo-information without the need of coding (B3Partners [2024], esri [2024]). Potential publishers of 3D geo-information most likely already have a working process for publishing 2D geo-information. For these reasons, research into integrating 3D functionality into existing GIS web applications for publishing can add valuable knowledge to the field of GIS.

A lot of the existing infrastructure for publishing 2D geo-information is based on the open source library OpenLayers OSGeo [2023b]. There are existing tools based on Openlayers which allow users to publish 2D datasets in web viewers without the need of coding, such as Tailormap B3Partners [2024]. Tailormap is an open source application that allows the user (the publisher of the data) to select data sources, both from databases and external services, and use these to make a web application. A website is then created on which visitors can view the created web application with maps.

The context and motivation presented in this section lead to the following problem statement: Although the interest in 3D geo-information has grown, and publishing 3D geo-information in a web viewer is more and more useful for various use cases, the research into the needs of the user that publishes 3D geo-information has been mostly focused on specific use cases (Nemoto et al. [2020], Cejudo et al. [2024], Nishanbaev et al. [2021], Mazzei and Quaroni [2022]). Therefore, the objective of this research is to investigate the enhancement of an existing 2D web application for publishing geo-information without the need of coding, Tailormap, to include support for 3D geo-information. This thesis will focus on identifying both the user needs and technical requirements necessary to allow users to publish 3D geo-information within this established 2D OpenLayers-based framework without the need of coding.

To solve this problem statement, a research question will be answered, which is given in section 1.2. Section 1.3 explains the scope of this thesis, and section 1.4 gives the outline of this thesis.

1.2. Research Questions

To solve the problem statement written in 1.1 this thesis will answer the main research question:

How can 3D functionality be integrated into an existing OpenLayers based GIS web application to ensure that publishing is both useful and user-friendly?

To answer this research question, the following sub-questions need to be answered.

1. What types of 3D datasets are available and which are the most relevant use cases for 3D web GIS?
2. How can you make it useful to publish 3D datasets without the need of coding?
   a) What features and options do users need to make publishing 3D data useful?
   b) What are the necessary manual steps users must undertake in the publishing process?

3. How can the code for 3D functionality be effectively integrated into an existing OpenLayers application?
4. What should an effective data flow for 3D datasets in a web GIS environment look like?
   a) How should 3D data be stored?
   b) What processes should be handled server-side and client-side in the management of 3D data?

The answer to this research question consists of two parts. Firstly, a set of technical recommendations on how to build up an extension to an existing OpenLayers based GIS web application that allows the user to publish 3D geo-information in a web viewer. Secondly, a set of capabilities and functions the application (with its extension) should or could have, along with how important they are.

1.3. Scope and Set-up of the Research

In this thesis, the needs for publishing 3D geo-information is investigated. This includes the user needs and the technical needs. The user needs, such as which data types are important and which functions are needed, were investigated through literature study, interviews, and testing sessions with potential users. The choice was made to do this through interviews and testing with selected people from different parts of the field of GIS. The reason for this is that extensive conversations give more and more extensive information about the needs of these users. Through this format, they can give their ideas about what they need and why.

The technical needs were investigated through the development of a proof of concept. Based on this development process technical recommendations were made. These recommendations are specifically about the integration of 3D functionality into an existing OpenLayers based GIS web application for publishing to a web viewer. The recommendations do not necessarily hold for creating a new application from scratch, or for integration with applications not based on OpenLayers.

This research was done in collaboration with B3Partners, who have developed the Tailormap. Recently, there has been a marked interest from clients in incorporating 3D functionality into Tailormap. The value of this thesis for B3Partners lies in pinpointing what their clients require in terms of 3D capabilities in Tailormap and understanding how to seamlessly integrate these capabilities. For the company, this research results in a proof of concept for 3D functionality, an initial version that can be further developed and refined.

It is noted that the company's role in this initiative is twofold. First, B3Partners will provide essential guidance in developing the proof of concept, ensuring that the integration aligns with the technical framework of Tailormap. Second, the company will provide contact with current and potential users who are interested in publishing 3D geo-information, thereby supporting the practical application and evaluation of the new 3D features. This collaboration will not only enhance the functionality of Tailormap but also reinforce its market position by meeting the evolving demands of its user base.

1.4. Thesis Outline

The further content of this thesis is divided into five chapters. In chapter 2 the literature study is presented. This chapter provides context for the research by illustrating relevant background information about the key concepts of this research. Related studies are used to identify (potential) users and use cases of 3D geo-information on the web.

Chapter 3 describes the methodology that was used to answer the research questions. It explains how this thesis goes from the research questions to the answers. This includes the user and application research, the technical methodology, and how these two parts feed into each other.

In chapter 4 the design of the proof of concept is shown and discussed. This chapter shows how the 3D functionality is integrated into the code of the existing application Tailormap B3Partners [2024], and it shows the design of specific functions that were included in the proof of concept. It also discusses the capabilities and limitations of the proof of concept.

The results of all parts of the research are presented and discussed in chapter 5. This chapter provides the information to answer the research questions. This includes the technical guidelines for integrating 3D functionality and the capabilities and functions that are needed to make the implementation useful.

Chapter 6 provides the conclusions for this research. This includes a discussion of the answers to the research questions, the limitations of the research, and the contributions it has made. This chapter also provides recommendations for future work.

2. Related Work

To develop the proposed solution it is imperative to first comprehend the significance of 3D geo-information on the web and identify the key users and use cases for this technology. Additionally, examining the existing research on the requirements for sharing 3D geo-information on the web is crucial. In this section, the results of the literature review that has been conducted are presented. The topics of this literature review are the following.

The value of 3D geo-information is examined to emphasise the need for infrastructure that facilitates sharing 3D geo-information. Previous scientific research is discussed to determine the current knowledge of the technical requirements and the user needs for publishing 3D geo-information in web viewers. Furthermore, potential use cases for 3D geo-information on the web are identified. This exploration helps in understanding the diverse applications of 3D GIS technology and tailoring the solution to meet the specific needs of different user groups. This exploration of needs for specific use cases is continued by analysing existing web viewers that show different types of 3D geo-information. The state of art is examined by analysing existing GIS applications that allow users to publish 3D geo-information in web viewers without the need of coding.

By exploring these topics, the research aims to lay a solid foundation for the development of a 3D web GIS application built on an existing application that is not only technically viable but also aligns with the practical needs of users and leverages the latest advancements in the field.

The relevance of 3D geo-information is discussed below in Section 2.1. The existing research into the needs of sharing 3D geo-information on the web is discussed in section 2.2, after which potential use cases are considered in Section 2.3. In Section 2.4 four existing 3D web GIS applications sharing specific datasets are analysed, after which section 2.5 explores existing solutions for sharing 3D geo-information on the web. Lastly, section 2.6 summarises the main takeaways from the related work.

2.1. Value of 3D geo-information

Between 2015 and 2017 a EuroSDR project analysed the costs and benefits of 3D geo-information for public mapping agencies [Coote et al., 2017]. Two case studies were used to do this: Flood management and urban planning. They found that "The cost-benefit analysis demonstrated in both of the selected use cases that benefits outstrip costs by a multiple of between two and three times even when considering each use cases in isolation." [Coote et al., 2017]. The project shows that, for these two use cases, investing in 3D geo-information is economically beneficial for public mapping agencies. Further research was done, using the data gathered by the EuroSDR project, by Ho et al. [2018]. They looked beyond economic value and analysed the public value created by 3D geo-information, which includes more general benefits for society. They concluded that investment in 3D geo-information can generate public value in many different ways, such as quality of life, safety, and transparency of public organisations [Ho et al., 2018]. This further encourages investment in 3D geo-information and infrastructure for 3D geo-information.

Lei et al. [2023] investigated the challenges associated with urban digital twins. One important challenge that was identified is that "the lack of data sharing network within and between municipalities, public institutions and private companies limits the amount and variety of available data" [Lei et al., 2023]. This project will not be a solution to this problem, but making it easier to publish digital twins on the internet could help improve the amount of available data and help the development of solutions to this problem. Another challenge that was identified is 'participation': not all users of the digital twins are incorporated in the creation process [Lei et al., 2023]. If more 3D geo-information datasets are shared through the internet, the amount of participation and feedback can be increased. Lastly, one identified challenge is the accessibility of software that is needed to view the data [Lei et al., 2023]. When data is shared on a website, no specialised software is needed to view the data.

2.2. Research on Needs for Sharing 3D Geo-information on the Web

There has been a lot of research on the technical needs of GIS web viewers to show 3D geo-information. As far back as 2004, de Vries and Zlatanova [2004] looked into whether the Open Geospatial Consortium (OGC) specifications then were adequate for publishing 3D geo-information on the web. In 2008 Abdul-Rahman et al. [2008] looked into the system components needed for 3D functionality in a web GIS application. More recently, Mete et al. [2018] built a 3D web-GIS viewer based on open source libraries. They tested the functionality and performance of Cesium and they found that the CesiumJS library was a good option for building a 3D GIS web viewer. Furthermore, Guardia and Koeva [2023] looked into the combination of different kinds of data and the combination of different scales in a web navigation model.

The research on the needs of the users publishing the 3D geo-information is mostly focused on specific use cases. A study by Zhan et al. [2021] proposed a method based on 3D Tiles to efficiently visualise BIM models in a web-based GIS viewer. A study by Nishanbaev et al. [2021] explores the needs for sharing cultural heritage data in a web-GIS viewer, which includes the integration of linked open data. The needs for a web GIS solution for disaster management and emergency response was explored by Cejudo et al. [2024], in which the need for real-time data in the 3D GIS web viewer was emphasised. Mazzei and Quaroni [2022] used a 3D web-GIS viewer to visualise the risks of earthquakes by including animations of the impact on certain structures. For the field of Geology Nemoto et al. [2020] have developed a web-GIS platform for the visualisation of geological voxel models. These studies take into account the user needs for a specific use case. They incorporate both the publisher of the data and the visitors of the web viewers. However, they aim to investigate the needs that apply specifically to one use case. Their purpose is not to provide an overview of user needs. An overview of common user needs that could be obtained from these studies, would therefore not be a complete general overview of the needs of users for publishing 3D geo-information in web viewers without the need of coding.

2.3. Users and Use Cases of 3D Geo-information

The purpose of this section is to identify use cases of 3D geo-information and to asses if they could benefit from publishing to a web viewer without the need of coding. To identify the use cases the EuroSDR project was consulted Coote et al. [2017], as well as a study by Biljecki et al. [2015]. These studies identified multiple use cases of 3D geo-information. Some of these use cases that are potentially relevant for 3D geo-information of the web are highlighted here. The discussed use cases are listed below.

- Asset management
- Management of pipes and cables
- Flood management
- 3D cadastre and valuation
- Urban planning

In **asset management**, specifically **management of utilities**, the utilization of 3D geo-information offers a significant advantage by reducing utility strikes. 3D data detailing the locations of cables and other utilities would reduce damage and danger caused by excavators hitting these utilities [Coote et al., 2017]. Sharing this data through a web application could present a viable solution. This approach allows personnel on-site to access up-to-date quality data, without the necessity of specialized GIS software.

In **flood management**, 3D geo-information can be used by municipalities to visualise flood risks. With this, they can inform decision-makers in government as well as citizens [Coote et al., 2017]. Online 3D GIS could be a good way to share these visualisations, since most citizens and decision-makers do not have GIS software and knowledge. If the visualisations of flood risks are shown in a web viewer, any citizen or decision-maker can easily view them.

In the case of **3D cadastre and valuation**, the most important identified benefit is "Allowing citizens to review and contribute to the data held by the tax authorities about their property creates a crowd-sourcing production efficiency for tax authorities as well as improving citizen trust in the authority's data holdings." (Coote et al., 2017 [Coote et al., 2017]). A practical solution to accomplish this could include the use of a 3D web GIS application. The tax authorities could use this to share the data with citizens. The citizens could view the data and perhaps contribute using this web application. 3D cadastre was also named as a use case for 3D city models by Biljecki et al. [2015]. It states that there are ongoing developments towards 3D property registration.

In the **urban planning** use case, informing the citizens is again one of the most important benefits of 3D geo-information. (Local) governments can use it to visualise plans and development to better show citizens the results and impacts [Coote et al., 2017]. They could share these visualisations with citizens with the use of a 3D web GIS application. Urban planning was also identified as a use case for 3D city models by Biljecki et al. [2015]. Specifically "visualisation for communication of urban information to citizenry" [Biljecki et al., 2015] was named as a use case. This further emphasises the urban planning use case identified by the

EuroSDR project. Additionally, a study by Haraguchi et al. [2024] investigated the maturity of city-scale digital twins and their use cases. This research emphasises the potential of digital twins in improving urban planning.

2.4. Examples of 3D Geo-information Datasets on the Web

This section shows four examples of different types of datasets made available in a 3D GIS web viewer. These four examples were chosen because they represent four different types of datasets: a terrain model, a voxel model, vector data, and a point cloud. In subsection 2.4.1 the web viewer showing the terrain model of Norway is discussed. Subsection 2.4.2 analyses the web viewer published by BRO showing soil models. In subsection 2.4.3 the 3DBAG viewer is discussed, and in subsection 2.4.4 shows the web viewer for the AHN point clouds.

2.4.1. Kilden Nibio

Kilden is an online resource made by research institute Nibio in Norway. It has a function to switch to a 3D view. This function then shows a terrain model of Norway, with the option to display many different maps draped over it. The website is shown in figure 2.1. The terrain model is a 2.5D dataset. To display the data in 3D, Cesium Cesium GS [2024a] was used. The application sends terrain to the client based on the zoom level. This means actual terrain data is streamed to the client, but not the whole model with the full detail and full extents that is stored on the server. When zooming or panning, the client requests a new terrain model from the server. The final images are created on the client side. Cesium GS [2024a], Nibio [2023]

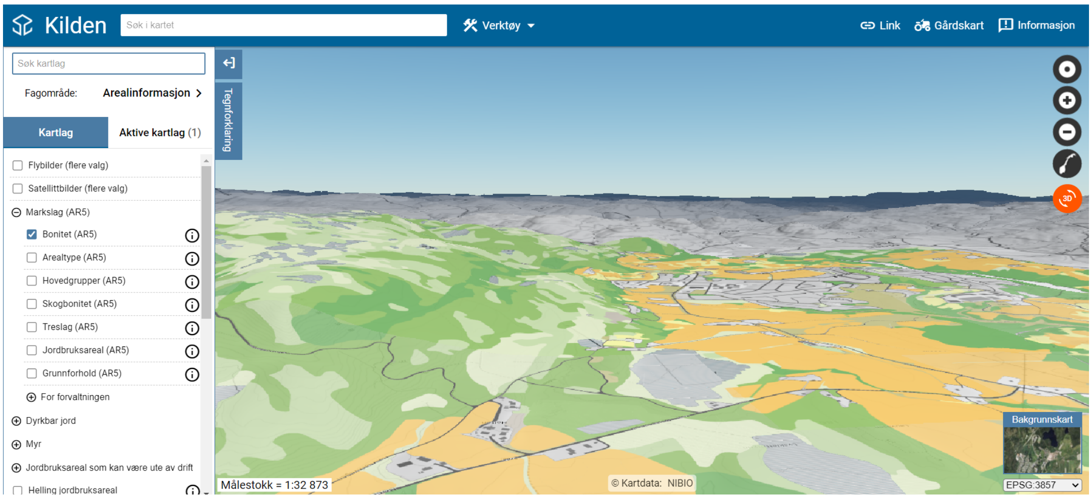

Figure 2.1.: Kilden Nibio [Nibio, 2023]

2.4.2. BRO 3D I3S Webservices

The BRO 3D I3S web services are made by research institute TNO in the Netherlands. The service allow users to view soil models of the Netherlands online in 3D. The website is shown in figure 2.2. The web viewers are made using ArcGIS Scene Viewer Esri [2024], and they use the Indexed 3D Scene Layers (I3S) standard OGC [2023]. They have two different types of data sets: GeoTOP, which is a voxel model, and REGIS II, which consists of 100x100 meter blocks that have layers of different sizes. For the voxel dataset, the entire model is sent to the client when opening the web viewer. For the REGIS II dataset, only a part of the model is sent to the client, and new requests are sent when zooming or panning. The final images are created on the client side. TNO [2023]

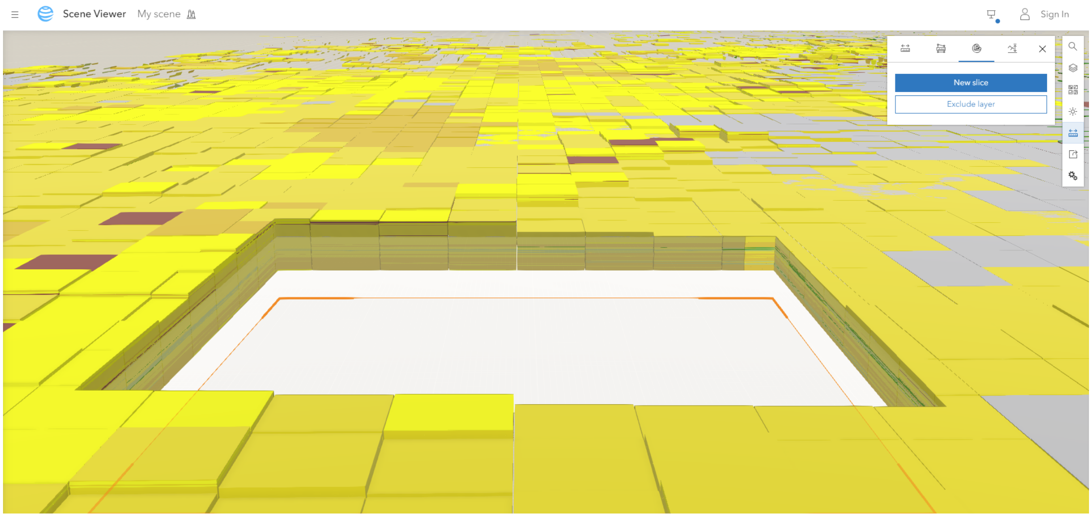

Figure 2.2.: BRO 3D I3S web services [TNO, 2023]

2.4.3. 3DBAG Viewer

3DBAG is an automatically generated 3D dataset of buildings in the Netherlands. An online 3D viewer was created to show this dataset. This viewer is shown in figure 2.3. The viewer displays the 3D model along with a background map from a different source which is obtained via a WMTS. The 3D model itself is streamed to the client using a batched 3D model. Only a part of the model is sent to the client based on the position of the camera, and new requests are sent when zooming or panning. The final images are created on the client side. TUDelft3d and 3DGI [2023]

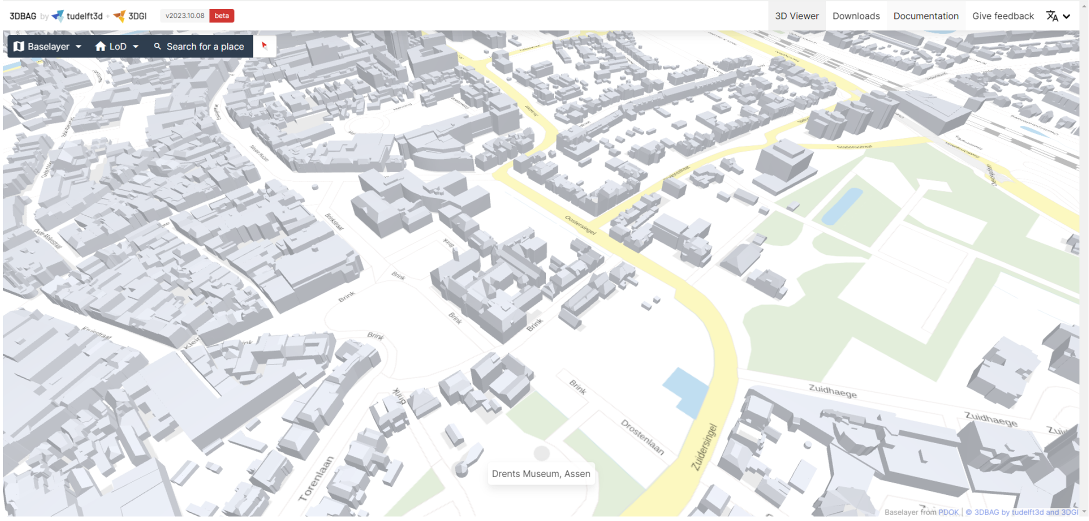

Figure 2.3.: 3DBAG viewer [TUDelft3d and 3DGI, 2023]

2.4.4. AHN Point Cloud

The Actueel Hoogtebestand Nederland (AHN) point cloud is a point cloud dataset covering all of the Netherlands. It is made with LIDAR measurements from aircraft. The dataset is used to create digital surface models (DSM's) and digital terrain models (DTM's) of the Netherlands, but the original point cloud dataset was also made publicly available. The online viewer showing the dataset is shown in figure 2.4. To be able to efficiently stream the point cloud data in parts to the client, the library Potree is used. Schütz et al. [2020], Actueel Hoogtebestand Nederland [2024]

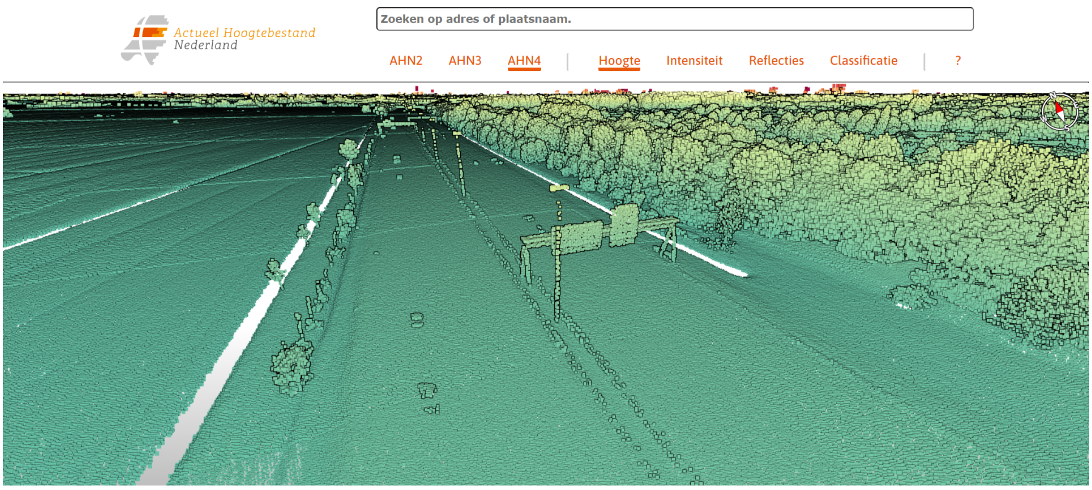

Figure 2.4.: AHN4 in a web viewer Actueel Hoogtebestand Nederland [2024]

2.5. Existing Applications for Publishing 3D Geo-information

This section analyses three existing applications which can be used to publish 3D geo-information in a web viewer. This is done to review the state of art and to see what functionality they have.

2.5.1. MapStore

MapStore is a product made by GeoSolutions, and it is based on open source software. It allows the user to publish geo-information in web viewers, and it also has some 3D functionality. Mapstore is based on the open source libraries OpenLayers, LeafletJS, and Cesium. The user can add a 3D Tileset and a terrain model as a layer to a viewer. The 3D Tileset can be added to a viewer using a URL to the service providing the dataset. There is also some functionality for styling the 3D Tileset. For this, there is a style editor interface in which the user can colour objects. The viewer also provides the ability to perform measurements in 3D, such as distances or areas, the ability to add annotations to the map in 3D, the ability to click objects to see their attributes, and a compass showing where north is while viewing in 3D. GeoSolutions [2024]

2.5.2. Cesium Stories

Cesium Stories is a part of the Cesium Ion services. Contrary to the Cesium libraries used in this research, Cesium Ion is not open source. There is also no integration with OpenLayers. Cesium Stories can be used to publish 3D geo-information that is uploaded in Cesium Ion to a 'story'. The story can include multiple 3D scenes with 2D and 3D geo-information along with added texts and pictures. When 3D vector data is uploaded, it is converted to 3D Tiles which the web viewer can load and visualise. Cesium Stories provides some additional functionality when publishing 3D geo-information. It is possible to colour 3D objects based on their attributes and to click on the objects to see their attributes. Furthermore, in the viewer there is a north arrow that shows which way is north when viewing in 3D, and it is possible to perform measurements in 3D. Cesium GS [2024b]

2.5.3. ArcGIS Online

ArcGIS Online, which is not open source, also provides the ability to publish 3D geo-information in web viewers. It can be used to create a 3D viewer that show both 2D and 3D geo-information. Data types that can be used in the 3D scenes include 3D Tiles, BIM models, and Terrain models. In a 3D viewer, some functions can be added, such as a compass, measurement tools, filtering, and shadow analyses. There is also functionality to style the 3D layers before publishing, and in the viewers it is possible to click objects to see their attributes. esri [2024]

2.6. Main Takeaways From Related Work

In this chapter the value of 3D geo-information on the web as been established. Furthermore, potential use cases of publishing 3D geo-information in a web viewer were identified. These include asset management, management of pipes and cables, disaster management, 3D cadastre, and urban planning.

Some existing examples of web viewers were analysed to obtain technical information and to identify existing use cases and their data types, which include terrain models, voxel models, soil models, point clouds, and vector data of buildings.

Some existing applications to publish 3D geo-information were analysed to determine the state of the art. In all three analysed applications, 3D Tileset is a very important data type, as are terrain models. They also all provided the ability to style the 3D layer, to perform measurements in 3D, to add a compass, and they have the ability to click objects to see their attributes.

3. Methodology

This chapter describes the methodology that was used to answer the research questions stated in Section 1.2. The methodology has been split in two parts: user and applications research to answer sub-questions 1 and 2, and a technical methodology to answer sub-questions 3 and 4. These parts are not separate, they were executed simultaneously and their results fed into each other.

In section 3.1 an overview of the methodology is given and the way the research questions are answered is explained. Section 3.2 explains the user and applications research, and section 3.3 describes how the proof of concept was made.

3.1. Overview

An overview of the methodology and its connection with the research questions and the answers are shown in figure 3.1.

To answer sub-questions 1 and 2, research was done into potential users and use cases of 3D online GIS. To answer these questions it is necessary to find potential use cases of 3D online GIS and discuss them with parties that will potentially publish 3D datasets on the web. Publication options (sub-question 2) also need to be discussed with them. This part of the research was done in three steps: Literature study, interviews, and testing. The outcome of this was a list of capabilities and functions that an application to publish 3D geo-information should or could have, along with their importance. This includes what data types and file types the application should be able to handle, what functions should be included for the publisher of 3D geo-information, and what functions should be included in the web viewer.

To answer sub-questions 3 and 4, a proof of concept was made. This is an application with which one can publish a 3D geo-information dataset on the web. This application was made within an existing Openlayers based application which is used to publish 2D geo-information on the web. This existing application is Tailormap made by B3Partners [B3Partners, 2024]. Tailormap uses Openlayers to display 2D maps. To be able to also display 3D data, Cesium was integrated into the application. Cesium is used within the Openlayers application to display a 3D dataset together with a 2D map. This proof of concept is made for three purposes. First, it proves that an application which can be used to publish 3D geo-information is technically viable and shows that it is possible to extend it with the capabilities and functions described in section 5.4. Second, the developing process of the proof of concept provides answers to the technical side of the research question, specifically sub-questions 3 and 4. These answers consist of guidelines in three parts: recommendations for integrating 3D functionality in an existing GIS web application for publishing, recommendations for integrating 3D functionality in an OpenLayers based GIS web viewer, and recommendations on the data flow and standards. The third purpose is to allow testing with users, the results of which feed into both the user and application research and back into the proof of concept.

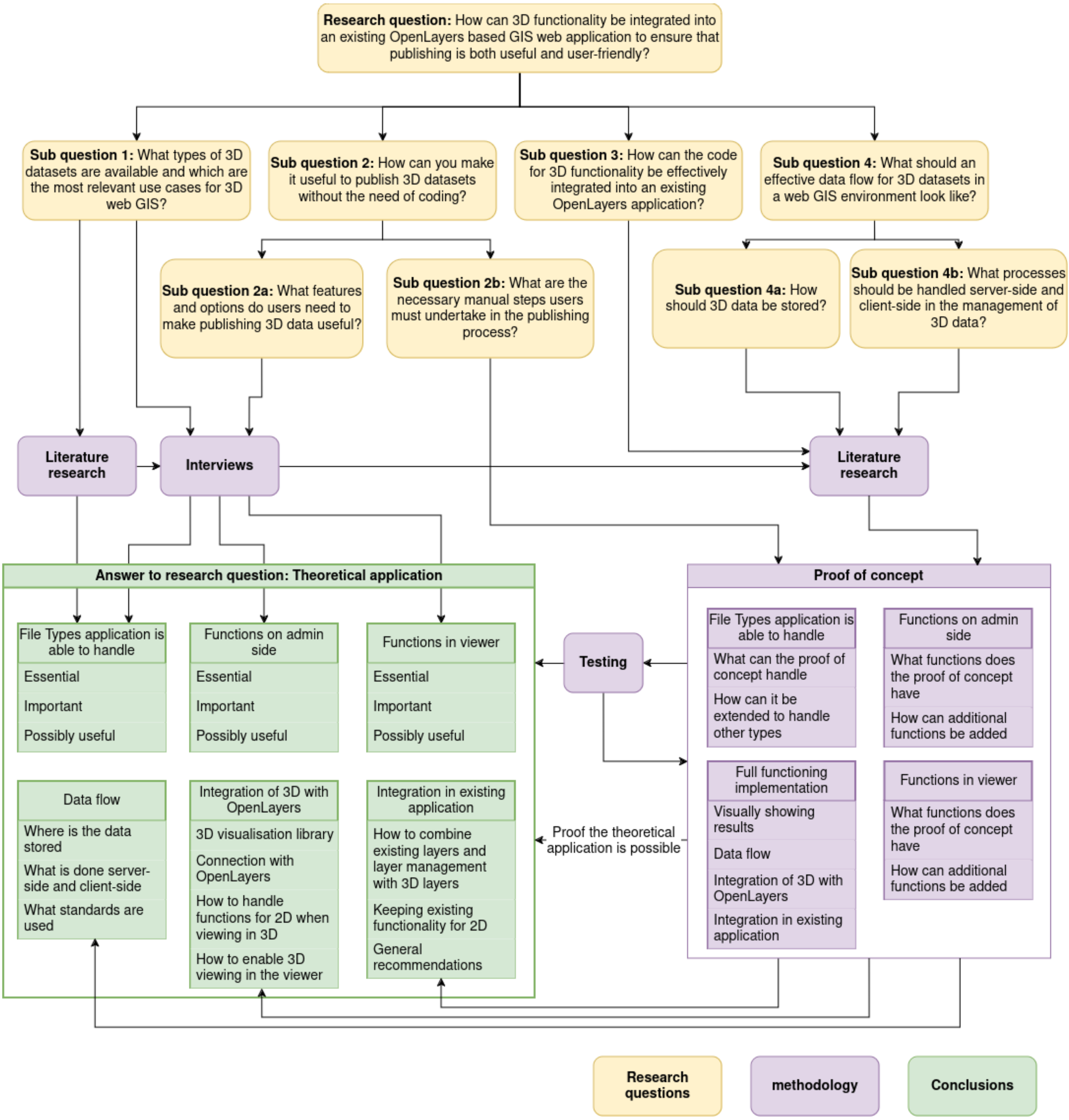

Figure 3.1.: Overview of the methodology and the connection with the research questions and the answers

3.2. User and Application Research

In this section the three steps in the user and application research are described. For each step, it is explained what the focus was and what kind of information was to be obtained. For the interviews and testing it is also explained what parties were interviewed and why.

3.2.1. Literature Study

First, a literature study was conducted to find already identified use cases in other studies. Some existing examples of 3D geo-information in a web viewer were also identified. In these use cases and examples the following things were identified if possible:

- Types of 3D data (point clouds, vector data, voxels, terrain models, etc.)
- Who is publishing the 3D data on the web?
- Who is the target audience for the published data?
- What options could be needed for this application? (viewing, editing, styling, clipping, filtering, etc.)

The findings of this literature study can be found in section 2.3 and section 2.4. The results of this part of the research mainly feed into the part of the conclusions about which types of data are most important to be able to handle. The use cases that are found also feed into which functions are important, since some functions are mostly important for specific types of use cases.

3.2.2. Interviews

Five interviews have been conducted. The interviewed parties are listed below. They are different types of parties to have as much variety in users and use cases as possible. They include a municipality, a safety region, the Dutch Cadastre, and two private companies. In addition to these interviews, there was a Tailormap users day, during which input was also obtained about use cases, data types, and functions.

- Municipality of Gouda
- Safety Region Brabant Zuid-Oost
- Kadaster
- Stantec
- Antea

The selection procedure leading to the choice of these parties was done in consultation with B3Partners. All the interviewed parties, with the exception of the Kadaster, are using Tailormap at the time of writing. With the purpose of obtaining user needs from a varied selection of users, the chosen parties are of different types. A selection of parties was obtained that work on different scales and with different purposes. A municipality works on a municipal scale and is mostly concerned with urban development. A safety region works on a regional scale and is focused on safety-related use cases. The interviewed private companies work mostly on projects pertaining to a single building project. The Dutch Cadastre was added to the selection to provide insight into the developments of 3D geo-information on the web on a national level. Within their organisations, the interviewed persons were GIS experts or working in a GIS-related team.

The purpose of these interviews was to determine what kind of 3D datasets users want to publish in a web viewer, and what functions would be important for the publisher and in the web viewer. In the interviews, the following questions were used as starting points:

- What 3D datasets do they have and what 3D datasets do they plan on obtaining in the future?
- What do they plan to use the 3D datasets for, and with what type of people do they want to share the data?
- Can an online web application be a good solution to share the 3D datasets?
- Who would need to be able to publish or edit the online 3D GIS application, and how much knowledge do they have of GIS?
- What functions (viewing, editing, styling, clipping, filtering, etc.) would they need in the online 3D GIS application?

These questions were used as talking points for all interviews, but more specific questions were also asked based on what the interviewed party does, or on the answers to these questions.

3.2.3. Testing Sessions with Users

The proof of concept was used for testing with potential publishers of 3D geo-information. The purpose of the testing sessions was mainly to better be able to answer sub-question 2. Through the testing sessions, it was determined whether the proof of concept was easy to use and if the process of publishing 3D geo-information made sense. The functions and capabilities that such an application could have were also discussed, and the parties were asked how important certain functions were for them. This was also partly done in the interviews, but it is often easier to understand and determine what functions you need when there is an example application in front of you. Testing was done with the following parties:

- Municipality of Gouda
- Safety Region Brabant Zuid-Oost
- Stantec

During the testing sessions, the users were taken through the process of publishing a 3D dataset in a web viewer. The users were shown each step of the publishing process for 3D geo-information and the different possibilities within the process were discussed. The user was asked for feedback for each step and potential additional functions were discussed. The process consists of adding a data source on the 'Admin - Catalog' page, creating the web viewer on the 'Admin - Applications' page, and checking the created viewer. The discussed steps and functions are listed below.

- **Admin - Catalog - data source** Here, the user enters the connection details for a service with a 3D dataset.
- **Admin - Catalog - database connection** Obtaining a 3D dataset from a database connection is currently not possible in the proof of concept, but the users were asked whether this would be useful for them.
- **Admin - Applications - settings** Here, the user can turn 3D viewing on and off for a viewer.
- **Admin - Applications - layers** Here, the user can add layers, including 3D layers, to a viewer.
- **Admin - Additional functions** The users were asked what additional functions would be useful for them on the admin website.
- **Viewer - Switch between 2D and 3D** The switch allows toggling between 2D and 3D viewing.
- **Viewer - Additional functions** The users were asked what additional functions would be useful for them in the web viewer.

The results of testing were used, together with the results of the interviews and the literature study, to answer sub-questions 1 and 2. The feedback was also used in the further development of the proof of concept.

3.3. Technical Methodology

The answers to the technical side of the research question, specifically sub-questions 3 and 4, are a set of technical recommendations. To be able to find these recommendations, and to show that they are well-founded, a proof of concept was made. This proof of concept was made as an extension to the existing OpenLayers based application Tailormap by B3Partners B3Partners [2024].

In Tailormap the user has an admin website on which they can select data sources and create web viewers. On the Catalog part of the admin website, the user can select data sources either from a web map service or a database connection. They can then create web viewers using layers taken from these data sources. The extension for the proof of concept allows the user to also select data sources for 3D geo-information. When making a web viewer they are also able to add 3D layers taken from these sources, and they have the option to turn on 3D viewing. In the web viewers themselves, the 3D layers are visualised and the user is able to look around in 3D. There is also a switch which toggles between 2D and 3D viewing.

In this proof of concept, choices had to be made in the design and focus of the extension. The most important of these choices are explained and discussed below. First, the choices about the data types are discussed. The choice was made to focus on vector data and to use the 3D Tiles data format. This is explained in subsection 3.3.1. The choices about data storage are discussed in subsection 3.3.2. The choice was made to use data stored on an external server, and to stream the data to the client using 3D Tiles. To visualise 3D geo-information in the viewer, Cesium was used. This is explained in subsection 3.3.3.

3.3.1. Type of Data

There are different kinds of 3D data such as vector data, terrain models (which could be called 2.5D, because there is only a single elevation at each 2D point), point clouds, and voxel models. The choice was made to focus on vector data for this research. In most of the use cases identified in the related work in chapter 2 vector data would be used. Most of these use cases are focused on buildings, and pipes and cables are also stored as vector data. Furthermore, the interviews and testing sessions also determined buildings were essential in most use cases.

There are many different formats to store 3D GIS vector data, such as CityGML OGC [2021] (can store more than just vector data), COLLADA ISO [2022], 3D Tiles Open Geospatial Consortium [2023], and Indexed 3D Scene Layers (I3S) OGC [2023]. The choice was made to use 3D Tilesets for the proof of concept. The main reason for this is its tiling capabilities, which allows to stream the datasets efficiently to a web browser for visualisation. This will be further discussed in the next subsection 3.3.2. I3S datasets also have this capability, but this standard was made in collaboration with Esri communities and is therefore better supported in Esri software, while 3D Tiles was developed with open source communities and is therefore better supported in open source software, such as Cesium which is used in this proof of concept. Furthermore, some interviewed parties that were already using 3D web viewers, were using 3D Tiles. The relevant capabilities of 3D Tiles are discussed in subsection 3.3.4.

3.3.2. Data Storage and Transfer

To store and retrieve 3D geo-information, the data can either be stored on an external server or in a database. For the proof of concept, the choice was made to use connections to external servers. To be able to efficiently visualise 3D geo-information, the 3D datasets need to be streamed in tiles to the client. Sending the entire dataset to the client, a thick client approach, is not feasible. Streaming only images to the client, a thin client approach, does not allow smooth viewing. By using 3D Tilesets hosted on a server, ready-made software such as Cesium, can be used to stream and visualise the 3D datasets, which makes it well suited for a proof of concept. Streaming 3D data in parts from a database is possible by using database formats, e.g. 3DCityDB Yao et al. [2018], and retrieving the data in parts. This would require a far more complex set-up and is therefore less suited for a proof of concept. Furthermore, the interviewed parties have expressed more interest in using 3D Tilesets on a server, than in using 3D data from a database.

For future applications and use cases that require constantly changing or adding to the data, a database connection is a better option. However, in this thesis the focus is only on viewing and viewing related functionality, thus changing and adding to the data is outside the scope.

3.3.3. 3D Visualisation

Cesium Cesium GS [2024a] was chosen as the library to use for 3D visualisation. Other options were also considered. The considered options are listed below.

- Cesium Cesium GS [2024a]
- Xeogl Xeolabs [2022]
- three.js Threejs [2024]
- BabylonJS BabylonJS [2024]

All of these libraries are able to display 3D information on the web. However, Cesium has the most support for geo-information. It is coordinate system aware, and it projects the data on a globe. Since the built product is a proof of concept, it is best to use the library that is already furthest along in visualising 3D geo-information. Furthermore, Cesium is a widely used 3D visualisation tool on the web and therefore has a lot of available support and information. Cesium GS [2024a]

Another reason to use Cesium is the existence of the Ol-Cesium library. This library helps with the combination of OpenLayers and Cesium. This library has the functionality to use an OpenLayers map in a Cesium scene. The relevant capabilities of Cesium and Ol-Cesium are further discussed in the next subsection 3.3.4. OSGeo [2023a]

The utilisation of the Cesium library facilitates the visualisation of an OpenLayers map within a 3D scene along with 3D layers. In the context of this thesis, a 3D layer is an instance of a 3D Tileset or another 3D dataset that can be loaded and visualised using Cesium. A 3D layer containing a 3D Tileset represents the entire tileset including all of its semantic information. Theoretically, one 3D layer can contain tiles of different formats. Cesium's functionality for 3D Tiles is made to load and visualise any tileset that adheres to the OGC specification, so this case would not cause any issues Cesium GS [2024d]. However, the datasets used to test the proof of concept only contain one tile format.

3.3.4. Capabilities of Used Libraries and Standards

This section summarises the relevant capabilities of the libraries OpenLayers, Ol-Cesium, Cesium, and the OGC standard 3D Tiles, which are used in the proof of concept. In Tailormap OpenLayers is used as a mapping library. In the proof of concept, the OpenLayers map is used as input in Ol-Cesium functions to insert it in a Cesium scene. Cesium creates the 3D scene and loads the 3D layers. It loads the data from a server providing a 3D Tileset.

Openlayers

OpenLayers is the mapping library used within Tailormap. It is used to load and render data from WMS, WMTS, WFS, and Tile Map Service (TMS) servers, as well as from databases. The layers from these services and databases can be loaded in the viewer and used to create an OpenLayers map. The OpenLayers functionality allows the users to, among many other things, add and remove layers in the map. This map should be displayed on the ground when viewing in 3D, which can be done with Ol-Cesium. OSGeo [2023b] B3Partners [2024]

Ol-Cesium

Ol-Cesium can create a 3D version of the OpenLayers map using the OLCesium class with the OpenLayers map as input. The OLCesium class has a method 'setEnabled', which can be used to turn 3D viewing on and off. The Cesium scene can be obtained using the method 'getCesiumScene'. With this Cesium scene, the Cesium functionality can be used. OSGeo [2023a]

Cesium

Cesium is a platform for software to visualise 3D data on the web. The CesiumJS library is an open source JavaScript library made for visualising 3D geo-information in web applications. The CesiumJS library is used in the proof of concept. The Cesium scene, which can be created as stated above, can be used to visualise the 2D map and additional 3D data on a 3D WGS84 globe. The CesiumJS library has functions to load 3D Tiles, Cesium terrain models, and other 3D data types. These datasets can be added to the scene and visualised in the web viewer. In the proof of concept, it is used to display the OpenLayers map and 3D Tilesets. Cesium GS [2024d]

3D Tiles

3D Tiles is an Open Geospatial Consortium (OGC) standard. It is made "for streaming and rendering massive 3D geospatial content such as Photogrammetry, 3D Buildings, BIM/CAD, Instanced Features, and Point Clouds" Open Geospatial Consortium [2023]. The standard defines a data structure that is built up of tiles. The tiles can be streamed and visualised separately. The purpose of this is to be able to stream and visualise only the parts of the dataset that are currently needed. This allows using massive datasets in web viewers. A

3D Tiles dataset is called a tileset. Within a tileset, the tiles can be of four different types: Batched 3D Model, Instanced 3D Model, Point Cloud, or Composite. The Batched 3D Model format is storing heterogeneous datasets. This is the tile format which is used for buildings. Within a tile the objects, such as buildings, are features. The tile also contains a batch table with properties (or attributes) of the features. The proof of concept was developed and tested using tilesets with tiles of this format. The other tile formats are not used in this proof of concept but can be used in future developments. The Instanced 3D Model tile format allows using multiple instances of the same model. This can be used for city furniture to use put the same tree or bench in multiple locations. The Point Cloud tile format can be used for streaming point clouds, and the Composite tile format allows combining different formats in one tile. Open Geospatial Consortium [2023]

4. Proof of Concept

In this chapter the technical design of the proof of concept will be shown and its capabilities and limitations will be discussed. There are two main aspects to the design: The admin part and the viewer. The admin module is where the connection with data sources is set up, and where web viewers can be created. The viewer takes the settings and layers that were set up in the admin module and uses them to display the layers. Section 4.1 gives an overview of the design. The Admin module and its connection to the back-end are discussed in section 4.2, and the viewer is discussed in section 4.3. The full code can be found on GitHub [1] [2].

4.1. General Design

This section gives a general overview of the extension to Tailormap which integrates 3D functionality. Tailormap consists of a front-end, 'Tailormap-viewer', and a back-end, 'Tailormap-api'. The front-end contains the code for the admin website and for the viewer. The admin website is where a user can select data sources and set up and publish viewers. The Tailormap-api back-end regulates the connection with the database where the data sources and viewer settings are saved. The code for the viewer uses these layers and settings to create viewers that the user can share.

A functional overview of the extension to Tailormap that integrates 3D functionality is given in figure 4.1. On the admin website, there is a 'catalog' section. In the catalog section, the user can select data sources on an external server by entering the URL and choosing the protocol to use for this source. A new protocol '3D Tileset' was added to facilitate using data sources with 3D Tilesets. In the back-end a new schema 'Tileset3DLayerSettings' was added. This schema is used to save layers that have '3D Tileset' as their protocol.

On the admin website, in the 'applications' section, the user can create web viewers. In the settings form, where settings for a new viewer are set, a new setting 'Enable 3D viewing' is added. This is saved as a boolean in the application settings. The user also can also add layers, which were obtained from data sources in the catalog, to the applications. Some changes were made to allow the user to add a 3D Tileset layer to an application.

The viewer then uses the back-end to obtain all the viewer settings from the database including which layers are in this viewer and the settings of those layers. The 2D layers are used to create an OpenLayers map. This functionality was already in Tailormap. In the extension, the 2D OpenLayers map is used to create a 3D scene in which the OpenLayers map is projected on the ground. The 3D layer settings are used to create the 3D layers using Cesium, which are then also added to the 3D scene. A function was also added to allow switching between 2D and 3D viewing.

[1] https://github.com/B3Partners/tailormap-viewer/tree/cesium
[2] https://github.com/B3Partners/tailormap-api/tree/cesium

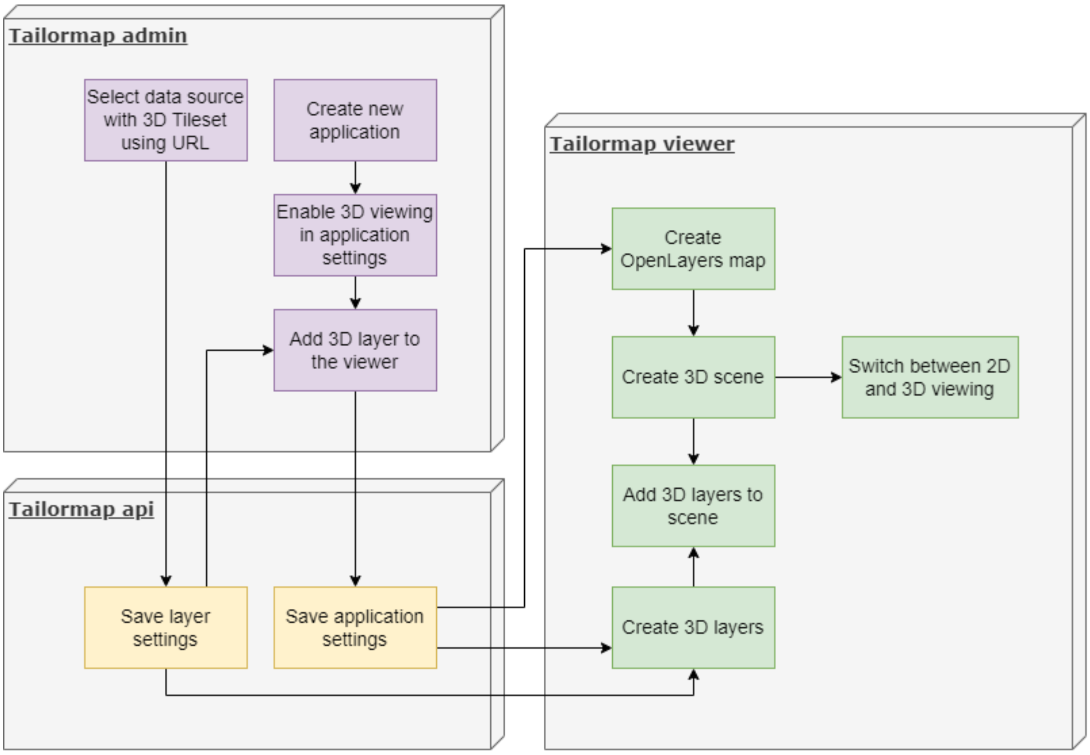

Figure 4.1.: Functional overview of 3D functionality in Tailormap

4.2. Admin and API

The admin side of Tailormap is built up of two modules, admin-core and admin-api. In the admin-core the pages of the website are set up, and the forms for user input are created. In the admin-api models and helpers are set up that are used by classes in the admin-core to ensure efficient communication with the back-end and other parts of the program. An overview of the functionality on the admin side is given in a UML diagram in figure 4.2. This diagram only includes things that were changed or added for 3D functionality, and it was slightly simplified to keep the diagram clear and understandable. Everything in purple text already existed, and everything in black text was changed or added. The forms in the admin-core were changed to allow user input for 3D data sources and the option to enable 3D viewing. The models and schemas in the admin-api and in the back-end were extended for the program to accept and save these new input options. The handling of input for 3D data sources is discussed in subsection 4.2.1. The settings for creating viewers are discussed in subsection 4.2.2, and the regulation of configuration in the back-end is discussed in subsection 4.2.3.

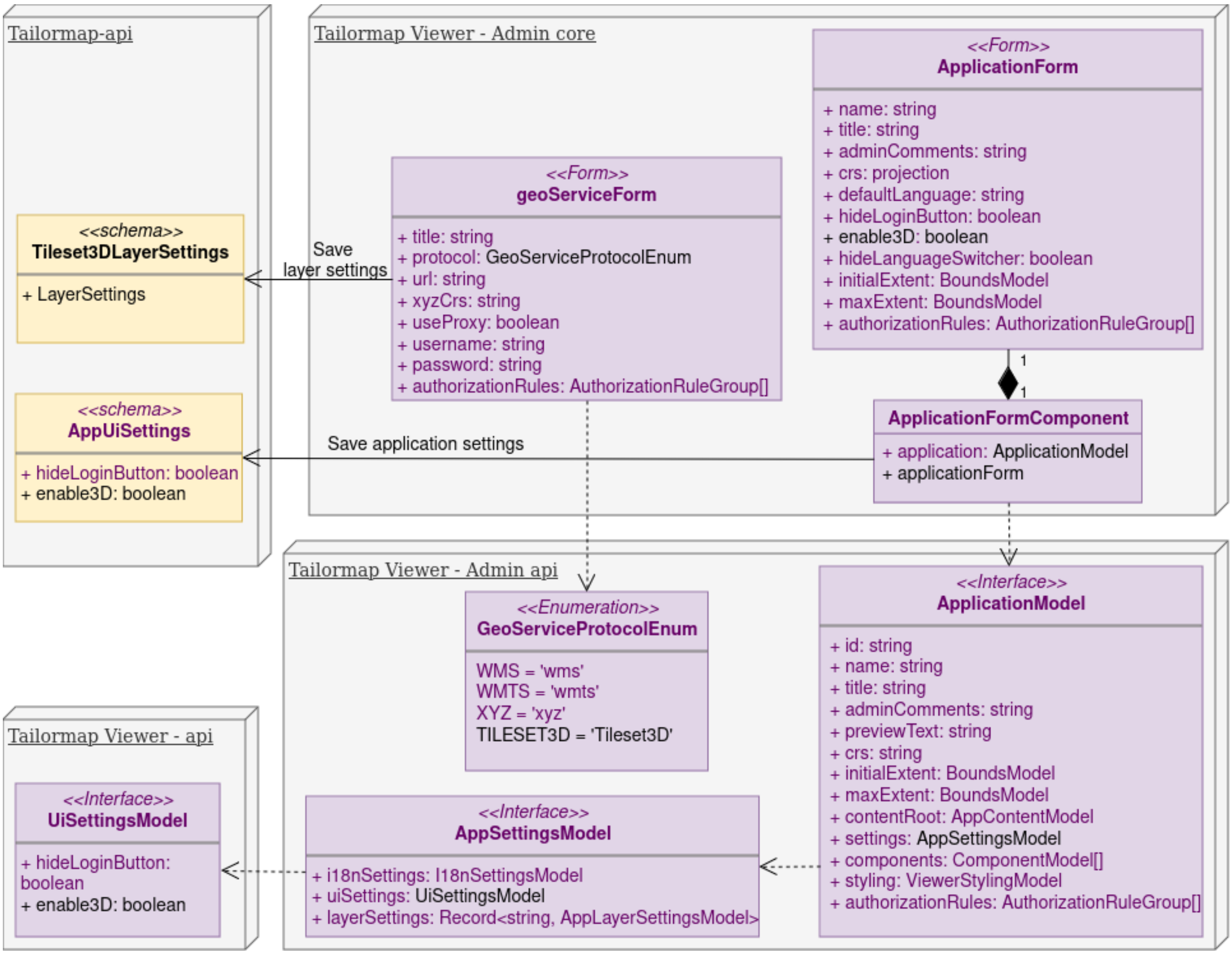

Figure 4.2.: Overview of 3D functionality in admin

4.2.1. Admin - Catalog

In the catalog section, the user can select data sources. Here the 'geoServiceForm' is used to get user input for a connection to an external service, as can be seen in figure 4.2. The user needs to input a title, URL, and a protocol. The protocol determines what type of service the URL is for. The protocol can be chosen from a dropdown list, as shown in figure 4.3. The protocol for 3D Tilesets was added by adding it to the 'GeoServiceProtocolEnum' enumeration. This enumeration is also used in other classes to handle the different services separately. Once a service is saved, its layers are automatically saved as well. In the case of a 3D Tileset, this is just one layer for the entire dataset. There is also a form in the catalog section to change the settings of existing layers. Some small changes were made to make this work for 3D Tileset layers as well. Other 3D layer types, such as terrain models, can be added in the same way. If many 3D layer types are added, it may be better to create a separate form for 3D services.

Every time a service is added or settings are changed updates are sent to the database using the back-end of Tailormap. In the back-end schemas are used to ensure everything is stored correctly. These schemas are further discussed in subsection 4.2.3.

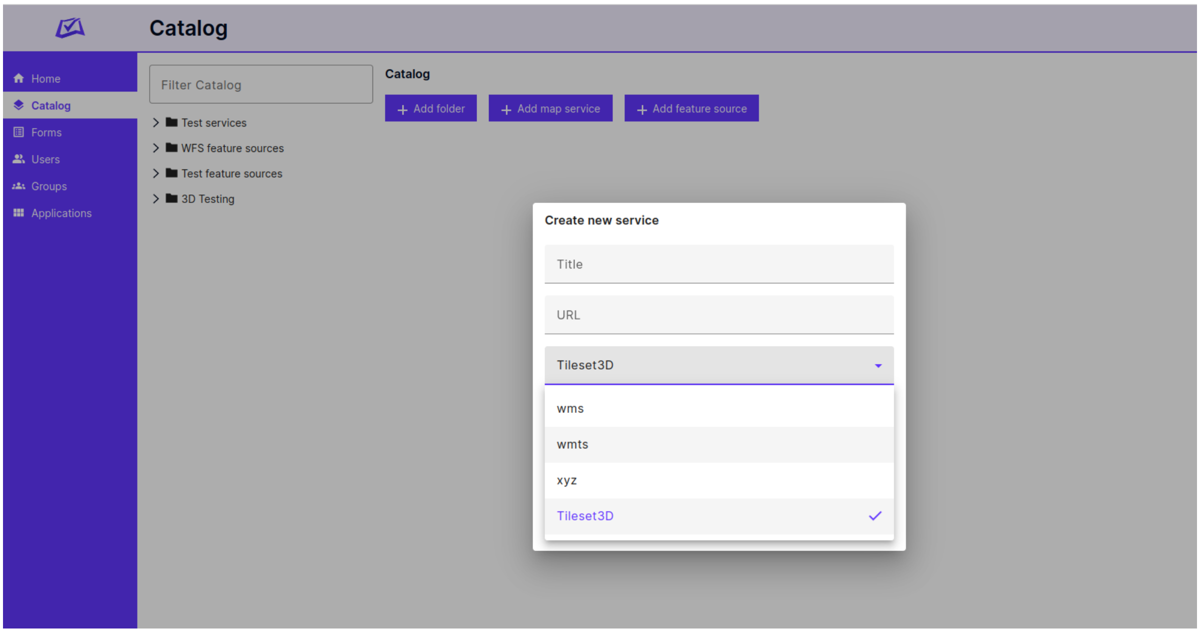

Figure 4.3.: 3D Tileset option in for adding a data source

4.2.2. Admin - Applications

In the applications section of the admin website, the user can set up viewers, which can then be shared. To create a viewer the user needs to set some settings and add layers to the viewer. In the form with the settings, the 'ApplicationForm' in figure 4.2, a new option is added to allow the user to enable 3D viewing in the viewer. The new option is a toggle button for the user, which can be seen in figure 4.4. To add this option to the form, it was added to the JavaScript code of the form as an option to be saved as a boolean, and to the HTML document as a slide toggle.

As can be seen in figure 4.2, this form is an attribute of the 'ApplicationFormComponent' class which also has the attribute 'application'. This attribute is an instance of the 'ApplicationModel' class and contains all the information about the viewer. Whenever the application changes, the ApplicationFormComponent updates it in the database using the Tailormap back-end. The changes in the back-end to enable saving the 3D viewing option are discussed in subsection 4.2.3. The ApplicationModel uses many other models to save all the information, one of which is the 'AppSettingsModel', which contains the 'uiSettings'. The 'enable3D' setting is saved in the uiSettings as a boolean.

Apart from the settings, the user also needs to add layers to the viewer. The user only gets to see the layers that match the projection that is used in the application. The 3D layers of course use a different projection. Some changes were made to the functions that check this to allow the user to also add 3D layers to the viewers.

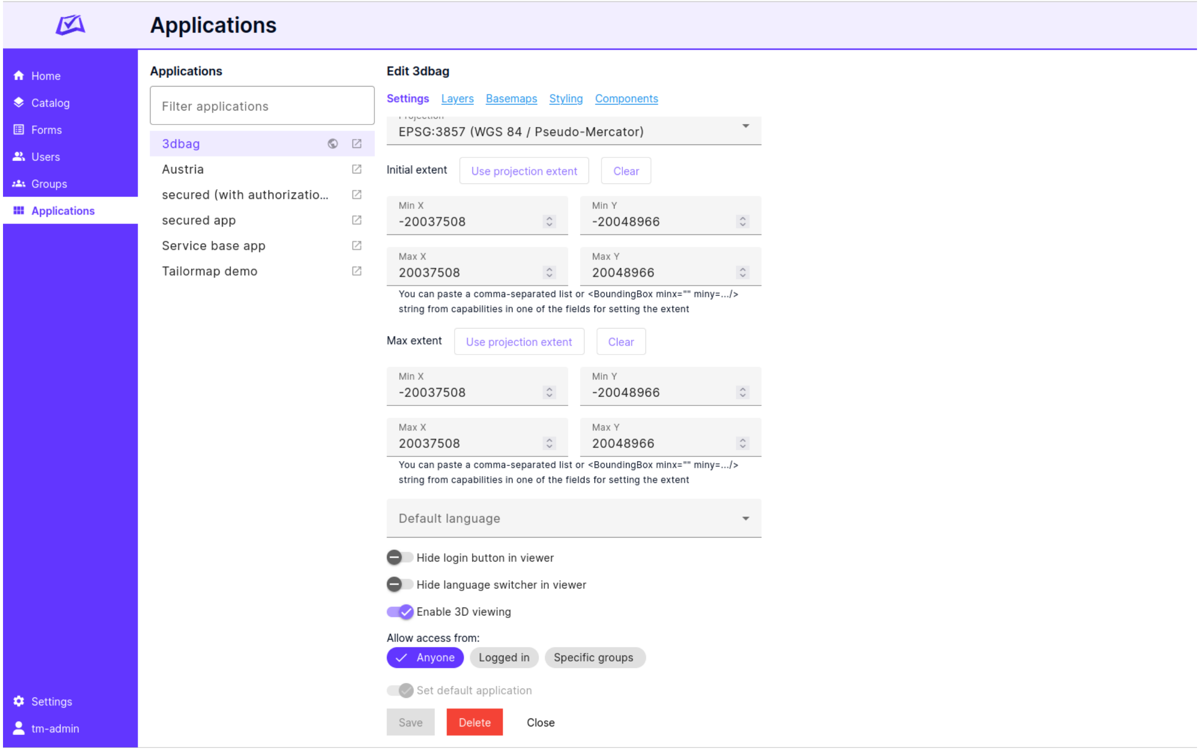

Figure 4.4.: Option to enable 3D viewing in application settings

4.2.3. Back-End

Additions were made in the back-end in three places. First, in the schemas that regulate saving the settings of a viewer, the setting to enable 3D viewing was added. Second, in the schemas that regulate saving layer settings, additions were made to accommodate 3D Tileset layers. Lastly, in the Java code, a method was added to set the layers of 3D Tileset services.

Viewer schemas
A set of schemas is used to check if the information coming from the front-end to save viewer settings is correct and complete. The schemas are then used to save the information in the database. The schema AppUiSettings was extended to include 'enable3D' as a boolean property.

Layers
Every layer type, such as layers from WMS, WMTS, and WFS services, has its own layer settings schema. They inherit some of their properties from the general layer settings schema which includes properties that are needed for every layer, such as title, URL, authorization rules, etc. The new schema for 3D Tileset layers only uses the properties from the general layer settings schema.

In the back-end there is the class 'GeoServiceHelper', which contains methods for loading the layers and capabilities of services. Its method 'loadServiceCapabilities' first checks the type of service, and then sets the layers and capabilities of the service. A new method 'setTileset3DCapabilities' was added, which is called by loadServiceCapabilities if the service protocol is 'TILESET3D'. Since the 3D Tilesets are loaded in the viewer as just one layer, this method just sets the tileset as a layer.

4.3. Viewer

In the core and map modules of the Tailormap viewer, the settings set up on the admin page are used to visualise the layers and create a GIS web viewer. In the core module the 'ApplicationMapService' retrieves the settings and layers that were set up for the viewer from the database. It uses the layers and settings to initialise the 'MapService' which manages the map and its layers. The MapService contains an 'OpenLayersMap' which creates an OpenLayers map ('OlMap') and manages the 2D layers. With the extension it also contains a 'CesiumLayerManager' which creates a 3D scene and manages the 3D layers. An overview is given in figure 4.5 in a UML diagram. Only the classes that are relevant to the extension for 3D functionality are shown and the diagram is slightly simplified to keep it clear and understandable. Everything in purple text already existed, and everything in black text was added or changed.

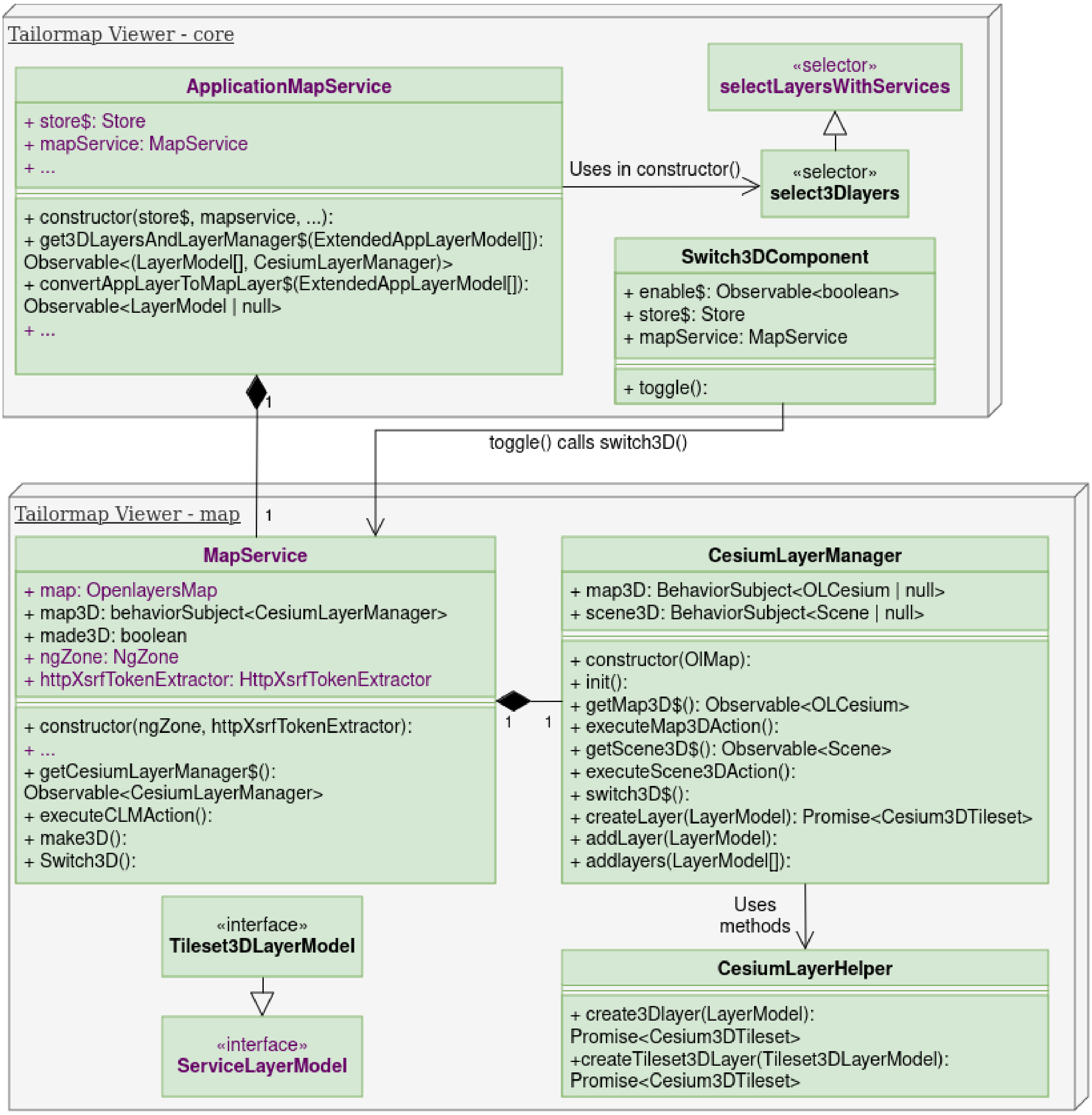

Figure 4.5.: Overview of 3D functionality in viewer

Subsection 4.3.1 further explains the additions to the ApplicationMapService, and subsection 4.3.2 explains the additions to MapService. In subsection 4.3.3 the CesiumLayerManager is discussed. Subsection 4.3.4 describes the design of the switch between 2D viewing and 3D viewing. The projection of the maps is discussed in subsection 4.3.5, and the testing for viewing on a tablet is shown in subsection 4.3.6.

4.3.1. Application Map Service

Once the application and layer settings are retrieved from the database using the back-end, they are stored in the 'store' parameter of the ApplicationMapService. This is an instance of the store class from the library NgRx [2024]. The store "is a controlled state container" NgRx [2024]. It keeps track of the state of the viewer. If changes are made in the admin, the viewer is updated using the store. The ApplicationMapService initialises an instance of the MapService. In the constructor method, which is run when the ApplicationMapService is initialised, the viewer settings are taken from the store and the MapService is used to create a 2D OpenLayers map. Then, the 2D layers are selected from the store and added to the OpenLayers map.

To facilitate 3D functionality the constructor method was extended. It checks in the store for the setting 'enable3D' in the UiSettings. If this is set to true, the MapService is used to initialise a CesiumLayerManager. The new selector function 'select3Dlayers' is then used to get all 3D layers from the store. The method 'get3DLayersAndLayerManager' is used to add the layers to CesiumLayerManager.

Within the get3DLayersAndLayerManager method, the 'convertAppLayerToMapLayer' method is used for all layers to put the layer properties in 'LayerModel' objects, which can be used in the rest of the code. The 3D Tileset layers are put specifically in instances of the 'Tileset3DLayerModel', which is an extension to the 'ServiceLayerModel', which is an extension to the LayerModel.

4.3.2. Map Service

Without the extension for 3D functionality, the MapService only used the OpenLayersMap class to manage the 2D map. With the extension, two new attributes were added, 'map3D' and 'made3D'. The made3D attribute is just a boolean to track whether the CesiumLayerManager, and with it the 3D scene, have been initialised. It is set to false in the constructor method. The map3D attribute contains an instance of the CesiumLayerManager. The new method 'make3D', which is called by the ApplicationMapService, gets the OpenLayers map (OlMap) from the OpenLayersMap (the class defined in Tailormap). It uses this as an input for initialising the CesiumLayerManager. It then also runs the 'init' method of the CesiumLayerManager and sets made3D to true.

The CesiumLayerManager in the map3D attribute is wrapped as a 'BehaviorSubject'. This is a class from the RxJS [2024] library. A BehaviorSubject can be used in multiple places in the program by using 'subscribers'. This means that whenever the BehaviorSubject changes, the subscribers are updated as well. In the case of the CesiumLayerManager, it means the user can add and remove layers while viewing and the layers are also added or removed in the visualisation. Because it is wrapped as a BehaviorSubject, the methods 'getCesiumLayerManager' and 'executeCLMAction' are needed to read and change the CesiumLayerManager.

4.3.3. 3D Layer Manager

The new class CesiumLayerManager is used to create and manage the 3D scene and 3D layers. The class has an attribute 'map3D' which is an OlCesium object wrapped as a BehaviorSubject. OlCesium is a class from the Ol-Cesium library OSGeo [2023a]. It is a reprojection of an OpenLayers map on the Cesium globe. The OlCesium instance is initialised in the init method using the OpenLayers map as input. The OpenLayers map used as input is obtained from a BehaviorSubject, which means the user can turn 2D layers on and off and see the results while viewing in 3D. From the OlCesium object a Cesium Scene is obtained and set as the 'scene3D' attribute of the CesiumLayerManager, which is also wrapped as a BehaviorSubject.

The class has a number of methods. The 'getMap3D' and 'executeMap3DAction' methods are to get and change the OlCesium object. The 'getScene3D' and 'executeScene3DAction' methods are to get and change the 3D scene. The 'switch3D' method uses the executeMap3DAction method to switch between 2D viewing and 3D viewing.

The 'addLayers' method accepts a list of layers as input. It runs the 'addLayer' method for each layer in the list. The addLayer method first runs the 'createLayer' method, which returns a Cesium3DTileset, and then adds the tileset to the 3D scene.

The createLayer method uses the 'create3DLayer' method from the 'CesiumLayerHelper' class. In this method, the other CesiumLayerHelper method 'createTileset3DLayer' is used. This method uses the URL from the layer to create and return a Cesium3DTileset instance. This might seem overly complex, and it is while only 3D Tilesets are used, but it is set up like this to allow easily adding functionality for other 3D layer types. New methods can be added in the CesiumLayerHelper to create, for example, a Cesium terrain. The create3DLayer method can dispatch the layers to the correct method by checking the layer type. This way, the methods in the CesiumLayerManager need minimal changes when adding new layer types.

4.3.4. 3D Switch

The 3D switch allows a user to click a button to switch between 2D viewing and 3D viewing, as visualised in figure 4.6. This function is needed because the 3D functionality should not diminish the 2D functionality. This function was integrated by adding a new module 'Switch3D' to the 'Toolbar' module. The Toolbar module contains all the functions that can be seen on the top of the screen next to the 3D switch button in figure 4.6. The Switch3D module consists of a TypeScript file, an HTML file, and a CSS file.

In the TypeScript file, the Switch3DComponent class is defined. The class has an attribute 'enable'. In the constructor method, the Store and MapService are used as input and assigned to the 'store' and 'mapService' attributes. The store is checked for the enable3D setting, and the enable attribute is set to the same value (true or false).

In the HTML file the button is created, but this is only done if the enable attribute is set to true. Here it is also defined that when the button is clicked, the 'toggle' method is called. The toggle uses the mapService and calls its method 'switch3D'. This switches between 2D viewing and 3D viewing.

Figure 4.6.: Function in viewer to switch between 2D and 3D viewing. OpenStreetMap contributors [2017]

When the viewer is switched to 3D viewing, the 3D Tilesets that the user selected are visualised, with the OpenLayers map on the ground. Some results are shown in figure 4.7 and figure 4.8. Figure 4.7 shows the 3DBAG dataset with OpenStreetMap as the base map Kadaster [2024a], OpenStreetMap contributors [2017]. Figure 4.8 shows a textured data set of the Wilhelminakade in Rotterdam 3D Tiles Nederland and Gemeente Rotterdam [2024], Kadaster [2024b].

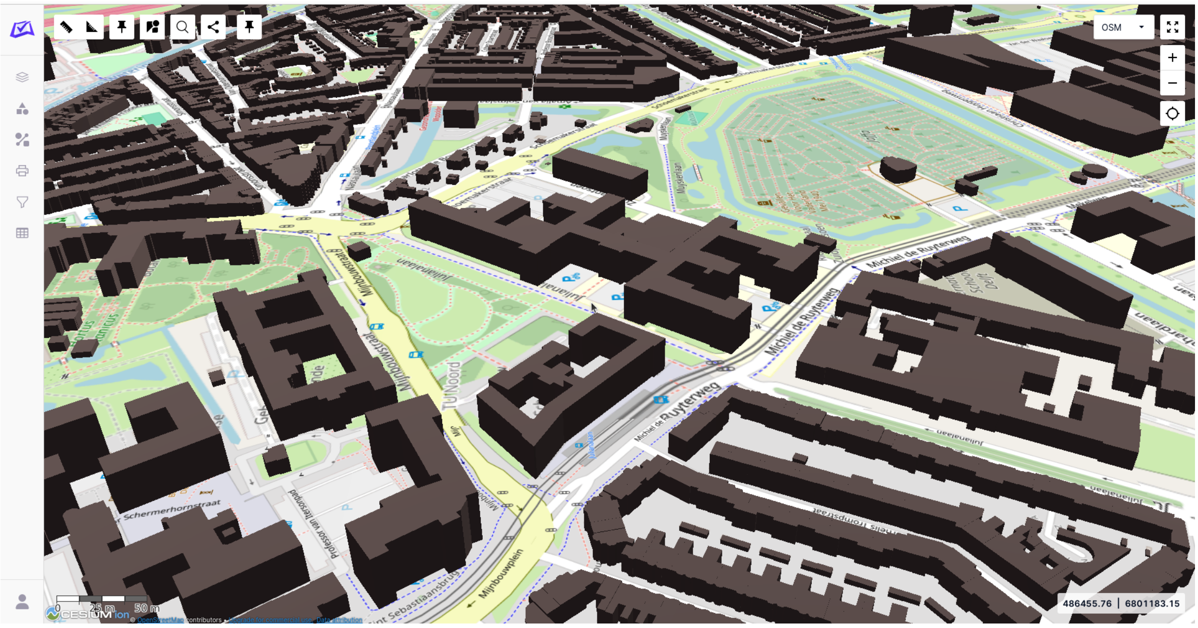

Figure 4.7.: 3D viewing in Tailormap, showing the 3DBAG Cesium3DTileset Kadaster [2024a], OpenStreetMap contributors [2017]

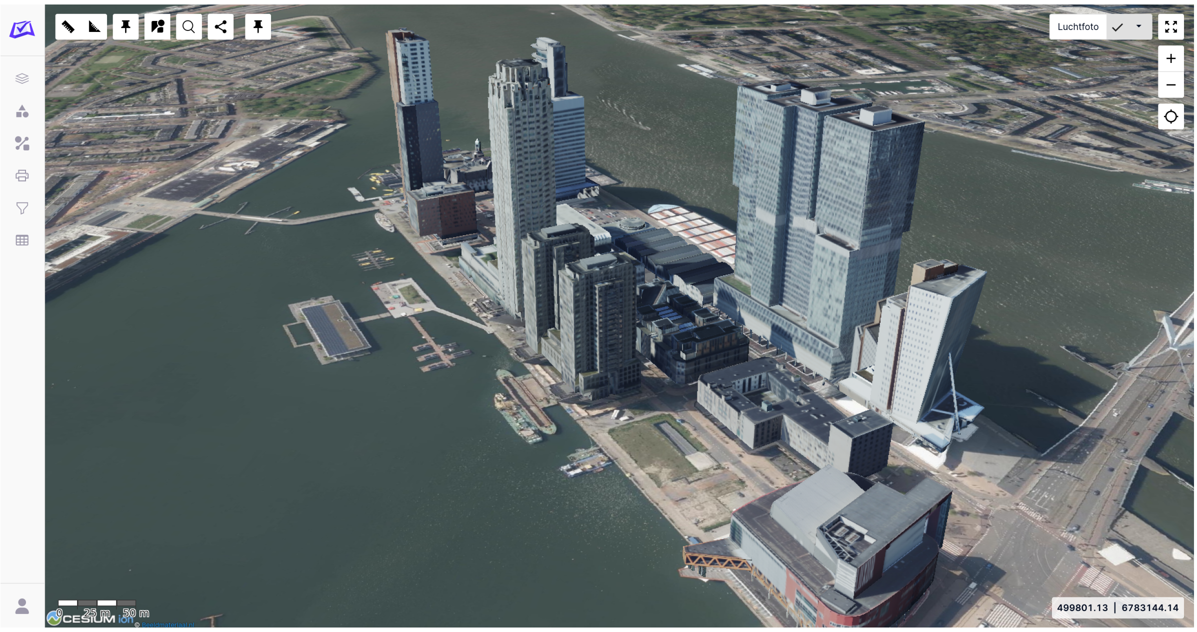

Figure 4.8.: 3D viewing in Tailormap, showing the a textured 3D Tileset in Rotterdam 3D Tiles Nederland and Gemeente Rotterdam [2024], Kadaster [2024b]

4.3.5. Projection of 2D Map

The coordinate reference system used in Cesium is WGS84. Cesium projects 2D maps on the ground, which is the WGS84 ellipsoid if no terrain is in the scene, and the terrain if there is a terrain. Cesium can handle 2D maps in two projections: EPSG:3857 (Web Mercator) and EPSG:4326 (WGS84). Cesium GS [2024d]

This means the view of the OpenLayers map used as input to create an OLCesium object needs to be in EPSG:3857 or EPSG:4326. This means if the map service does not provide the data in EPSG:3857 or EPSG:4326 the layer can not be used in 3D viewing. When setting up a viewer in Tailormap, the user chooses between EPSG:3857 and EPSG:28992 (RD New). For now, the 3D functionality only works with the viewer set to EPSG:3857.

4.3.6. Tablet or Mobile

For some use cases identified in the interview, it is needed to use the viewer on a tablet. Therefore a Tailormap was tested using the Firefox web developer tool 'responsive design mode'. The viewer in responsive design mode can be seen in figure 4.9. All functions in the viewer still worked the same.

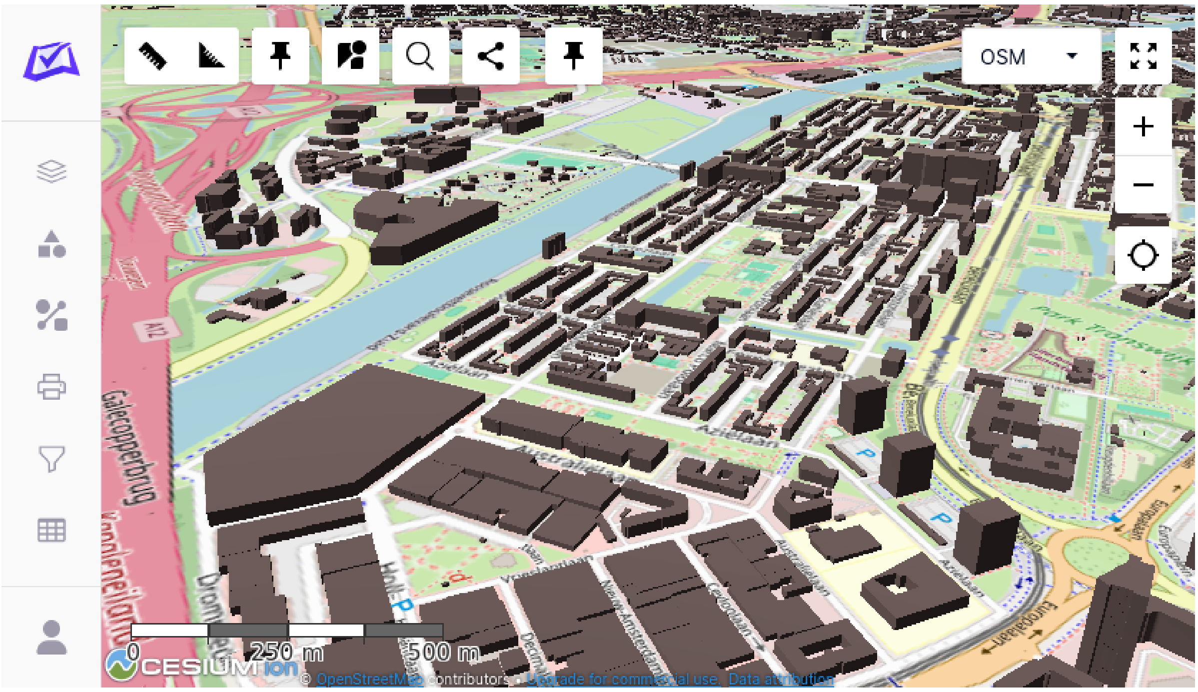

Figure 4.9.: 3D viewing in Tailormap in 'responsive design mode' using Firefox web developer tool, showing the 3DBAG 3D Tileset Kadaster [2024a], OpenStreetMap contributors [2017]

5. Results

In this chapter, the results of the research are presented and discussed. The findings from the user and application research and the technical research are shown and combined. The results of the interviews and the testing are discussed in section 5.1 and section 5.2 respectively. These results, together with the literature study in section 2.3 and section 2.4, provide the information needed to answer sub-question 1: 'What types of 3D datasets are available and which are the most relevant use cases for 3D web GIS?'. In section 5.3 the results of the technical research are presented as a set of guidelines. These results provide the information to answer sub-questions 3 'How can the code for 3D functionality be effectively integrated into an existing OpenLayers application?' and sub-question 4 'What should an effective data flow for 3D datasets in a web GIS environment look like?'. Section 5.4 provides a list of capabilities and functions, along with their importance and how they can be implemented. This is a combination of the results of the user and application research and the technical research. It provides the information needed to answer sub-question 2. The general takeaways from the results are provided and discussed in section 5.5.

5.1. Interviews

This section provides the results of the conducted interviews. The purpose of the interviews was to determine what the interviewed parties want to use the application for. Specifically, the aim of the interviews was to identify the types of data the parties wished to publish and for what reasons, as well as to identify what functions would be needed for those use cases. The different use cases are discussed in subsection 5.1.1. The datasets needed for those use cases are discussed in subsection 5.1.2. Subsection 5.1.3 discusses which functions are needed to make the application useful for the identified use cases.

Different kinds of parties were interviewed in order to get varied input and a better overview of general user needs. The interviewed parties were: a municipality, the Dutch Cadastre, a safety region, and two private companies. There was also a Tailormap user's day during which many users of Tailormap provided input. These results are the main takeaways from the interviews. The full summaries of the interviews can be found in appendix B.

5.1.1. Use Cases Identified in Interviews

Different kinds of use cases for publishing 3D geo-information have been mentioned in the interviews. Furthermore, the use cases they mentioned were on different scales. The most important use case was visualising city development and building plans. Furthermore, important identified use cases were management of pipes and cables, asset management, and disaster management.

Visualising city development and building plans
The most important use case mentioned by the municipality of Gouda was to visualise city development plans to show decision-makers and citizens. Showing this in 3D would make it easier for people to understand the plans and what they mean. This important use case was confirmed during the Tailormap users day by other municipalities and companies. As discussed in section 2.3 this use case was also identified by Coote et al. [2017] and Biljecki et al. [2015].

The same kind of use case was named by Stantec and Antea, although on a slightly smaller scale. It was stated that it is very useful to visualise building projects. They are already using 3D visualisations for this. This helps with the development of the building projects. It is used within the teams from the companies themselves, but also to inform clients and other involved parties. The management of these buildings is also an important use case. Antea is not doing this in 3D currently, but they want to in the future.

Management of pipes and cables
Another important use case mentioned by both the municipality of Gouda and Antea is the management of cables, pipes, and sewage. This would be useful when it is needed to dig into the ground. It was also mentioned it would be useful to visualise damage of sewage systems. For this use case, a viewer would need to have the capability to view objects underground, either by making the ground see-through or by allowing the camera to go underground. This use case was also identified by Coote et al. [2017] as discussed in section 2.3.

Disaster management and risk management
The safety region Brabant Zuid-Oost had different kinds of use cases. The first one is for first responders who respond to incidents such as fires. On the way to the incident, they receive information about the building they are going to. Currently, this information is in 2D with separate layers for the different stories. They would like this information to be in one 3D model in the future. For this use case, the viewer would need to work on a tablet. The other important use case was their datasets of vulnerable buildings which they use to advise municipalities. If these could be visualised in 3D this would help with municipal decision making. Another potential use case for the safety region is to visualise large incidents in 3D, such as a dangerous gas plume or a flood. The use case of flood management was also named by Coote et al. [2017] and the use case of emergency response was explored by Cejudo et al. [2024], as discussed in section 2.3. During the Tailormap users day, other safety regions confirmed these use cases.

Show possibilities of 3D data
For the Dutch Cadastre, the purpose of their 3D datasets is mostly to inform potential users about the possibilities of the 3D data. They have the PDOK 3D Viewer which allows anyone to view the '3D Basisvoorziening' which includes the buildings and the terrain in the Netherlands. They also have a service for retrieving feedback from citizens on the public registries. They want to add 3D functionality to this as well. This is a use case also mentioned by Coote et al. [2017].

In table 5.1 the main use cases are summarised.

Table 5.1.: Most important use cases from interviews

Interviewed Party	**Use cases**
Municipality of Gouda	Show city development plans to decision makers and citizens.
	Visualize underground: pipes, cables, sewage.
Kadaster	Show possibilities of '3D basisvoorziening'.
	Obtainining feedback for '3D basisvoorziening'.
Safety Region Brabant Zuid-Oost	Object information for first responders.
	Visualising large incidents in 3D.
	Coloured 3D buildings to show e.g. vulnerable buildings to advise municipalities.
Antea	vulnerable buildings to advise municipalities.
Stantec	Visualise building project with environment to help development.
	Asset management.
Tailormap users day	Visualise plans to inform decision makers and citizens.

As can be seen in the table, the use case that was mentioned by most parties is to visualise plans, either building plans or city development plans, and use this to citizens and people involved in the project. Another use case mentioned by two parties is viewing cables and pipes underground, for which the viewer needs to allow viewing underground. Asset management was also mentioned by two parties. The Dutch Cadastre had different kinds of use cases than the other parties. The safety region Brabant Zuid-Oost also had different kinds of use cases, but their use cases were confirmed by other safety regions.

The Dutch Cadastre works with 3D datasets on a national scale. The Municipality of Gouda works on a municipal scale and needs mostly the outer shape of buildings. The safety region works on a regional scale, but does need the interior of buildings. The two private companies work mostly on the scale of a single project and also need the interior of buildings. Therefore, the datasets required for the different use cases vary in size and level of detail. The required datasets are discussed further in subsection 5.1.2.

5.1.2. Desired Usable Datasets

For the use cases described above, the interviewed parties need to be able to use different kinds of datasets. The types of data that could be needed are, for example: buildings, pipes and cables, terrain models, or city furniture. There is also a difference in the level of detail of the datasets. To help categorise these levels of detail the CityGML model for levels of detail is used. For buildings, this means Level of Detail 1 (LoD0) is a 2D polygon. LoD1 is a block with a flat roof, LoD2 includes different roof shapes, and LoD3 includes details such as windows and doors. Lod4 also includes the interior of a building. OGC [2021]

The municipality of Gouda is mostly interested in using datasets with buildings. They would want to use this to visualise city development plans. This means they most likely need the buildings in LoD2 or LoD3.

The Dutch Cadastre is interested in datasets covering the entire country. This includes buildings in LoD0, LoD1, LoD2 and a terrain model.

Stantec and Antea are using datasets covering a building project and its surroundings. They want to visualise the outside as well as the inside of the buildings, which means the buildings are in LoD4. Stantec explained they are using a web viewer for this based on MapStore. The building models they use for this come from Building Information Modeling (BIM) datasets and CityGML datasets. They also sometimes include city furniture like signs or trees. They convert all of these datasets to 3D Tilesets before loading them into the viewer.

For Antea cables and pipes are also important datasets. The municipality of Gouda also mentioned this as a potential use case. Cables and pipes would also be vector data like buildings. These could also be stored in a 3D Tileset.

The safety region Brabant Zuid-Oost would need buildings in LoD4 for the use case for first responders. For visualising large incidents, they need information on the different stories of buildings. This can be done in LoD3. Visualising vulnerable buildings can be in LoD2 or LoD3. For visualising floods, they would also need a terrain model.

The most important data types for the interviewed parties are summarised in table 5.2.

Table 5.2.: Most desired data types from interviews

Interviewed Party	Most desired data type(s)
Municipality of Gouda	Buildings (Vector data)
Kadaster	Buildings (Vector data)
	Terrain models
Safety Region Brabant Zuid-Oost	Buildings (Vector data)
	Terrain models
Antea	Buildings (Vector data)
	Cables and pipes (vector data)
Stantec	Buildings (Vector data)
Tailormap users day	Buildings (Vector data)

As can be seen in the table all the interviewed parties need to visualise buildings. The Cadastre uses LoD2 data, the municipality of Gouda would need LoD2 or LoD3 data, and the other parties would use LoD4 data of buildings. Pipes and cables and terrain models are also important for some use cases. City furniture would be something that is nice to have for some use cases.

The buildings in all levels of detail, the pipes and cables, and city furniture can all be saved as a 3D Tileset Cesium GS [2024c]. For the buildings and pipes and cables the tiles would be of the Batched 3D Model format. The tiles for the city furniture can be of the Instanced 3D Model format. This format allows to use the same model, for example of a bench, in multiple places. The loading of these tilesets can be done in the same way and does not require any different functions to load. The terrain models do need to be stored and loaded differently. This will be explained in subsection 5.4.1.

5.1.3. Desired Functions

The interviewed parties were asked what functions they would need in the viewer. They had an overall consensus on what the most important functions were. Some additional functions that could potentially be useful were also named.

Clicking objects to see attributes
All interviewed parties agreed that clicking on objects to see its attributes is needed in a viewer showing 3D geo-information. It did not come up in the interview with Gouda, but later during testing, it was confirmed as a useful function. For most use cases it is mostly about clicking on buildings, but for Stantec and Antea it is also useful to click on objects inside a building.

Colouring objects based on their attributes
Being able to colour buildings based on its attributes is an important function to have, according to four of the interviewed parties: the municipality of Gouda, the safety region Brabant Zuid-Oost, Antea, and Stantec. This function can help convey the important information or message to users of the viewer. When visualising building plans it can, for example, help to show what kind of buildings there are (houses, shops, industry, etc.). The safety region wants to use this to visualise what buildings are vulnerable and for what reasons.

Filtering based on attributes
Filtering based on attributes was mentioned by Gouda, safety region Brabant Zuid-Oost, and Stantec. This function is important for mostly the same reason as colouring based on attributes. It can help to convey the important information by showing only objects relevant to the specific use case the user of the viewer is looking for.

Viewing underground
The use case of visualising pipes and cables was mentioned by Antea and Gouda. For this use case, it is needed to be able to view objects underground. This can be done by making the ground see-through and by allowing the camera in the viewer to move underground.

Other possibly useful functions
Some other functions were also mentioned that could possibly be useful in the future. It could be useful to be able to view from ground level. While this is already possible by navigating there manually, it could be useful to have a function that takes you to ground level looking up automatically. This is related to a function mentioned by Antea which is to be able to walk through a building.

The most important functions based on the interviews are summarised in table 5.3. For each interviewed party the functions that were deemed to be one of the most important are marked.

Table 5.3.: Most desired functions from interviews

Function noted to be one of the most important	Municipality of Gouda	Kadaster	Safety Region Brabant Zuid-Oost	Antea	Stantec
Click on objects to see attributes		X	X	X	X
Colour buildings based on attributes	X		X	X	X
Filtering based on attributes	X		X		X
Viewing underground	X			X	

From the interviews, the conclusion can be drawn that it is essential to be able to click on objects to see its attributes. Colouring and filtering based on attributes are also very important. Viewing underground is important for the use case of pipes and cables. The implementation of these functions is discussed in subsection 5.4.2.

5.2. Testing Sessions with Users

In this section, the results of the testing sessions are discussed. During the testing session, the user was taken through the process of publishing 3D geo-information using the proof of concept. They gave feedback on the steps in the process and desired functions and options were discussed. The process consisted of three main steps: Admin - Catalog, where the data sources are set up, Admin - Applications, where the web viewers are set up, and the viewer. The parties with which testing sessions were held were: The municipality of Gouda, the safety region Brabant Zuid-Oost (VRBZO), and Stantec. These results are the main takeaways from the testing sessions. The full summaries of the testing sessions can be found in appendix C. The purpose of the testing sessions was twofold. First, feedback was obtained from users of Tailormap on the proof of concept to determine whether it was user-friendly. Second, additional input was obtained to determine which functions and capabilities are most important.

The general feedback during the testing sessions was that the proof of concept looked good and was easy to use. The publishing process for 3D geo-information in the proof of concept is very similar to the process for 2D geo-information obtained from a web service. The user only has to enter the URL to the web service containing the 3D geo-information, add the 3D dataset to a viewer as a layer, and turn on 3D viewing. This made the process easy to understand for the users.

Furthermore, it was noted in the testing session that a very simple version that only allows publishing and viewing 3D geo-information would already be useful according to the users. The interviewed parties did of course have a lot of ideas for additional functions they would like. These are discussed below for the admin page and for the viewer.

5.2.1. Admin - Catalog

The consensus of all three parties was that the current option for selecting a data source with a 3D dataset, i.e. from a server as a service, is the best option. Using a database connection is less important, although it could be useful if this means you can add information to the dataset. Using Cesium Ion as a server to retrieve the data from is nice to have, but not important.

5.2.2. Admin - Application

All parties agreed that having a setting to enable 3D functionality or not is indeed important, because you do not need it for every viewer. They were asked what additional functions they would like to have on the admin page. The functions that were mentioned to be important or useful by multiple parties are summarised in table 5.4.

Table 5.4.: Most important functions in admin from testing sessions

Function	**Gouda**	**VRBZO**	**Stantec**
Set up colours based on attributes	Important	Very important	Very important
Choose whether the viewer starts in 2D or 3D viewing		Useful	Useful
Set which attributes can be seen and with which aliases	Useful		Useful
Use polygon from other layer to limit where 3D features are loaded	Useful	Useful	

In the testing sessions, it was concluded that it was better to set up the colours for the building beforehand in the admin than to let users of the viewers do it themselves. This was the most important admin function they all named. Furthermore, it would be useful to have an option which sets whether a viewer starts in 2D viewing or 3D viewing. This was requested by two of the parties. It was also mentioned by the safety region that it would be useful to set at which height and angle the viewer starts if it starts in 3D. Both Gouda and Stantec wanted to set up which attributes of 3D features can be seen in the viewer, and under which names. Another potentially useful function mentioned by the safety region and Gouda is to only load features inside their borders. This could be done by using a polygon from another (2D) layer.

Some other functions were mentioned by one of the parties that could possibly be useful in the future. Combining the 3D layers with other datasets by linking them with ID's could be useful. For example, to colour buildings from a public source based on data from their own files. It would be useful for the safety region to set up scenarios that automatically turn certain layers on or off.

5.2.3. Web Viewer

All parties agreed that it is good to have the switch between 2D and 3D viewing. For Stantec and the safety region, it would be important that the viewer works on phones and tablets. As shown in subsection 4.3.6 this works for the proof of concept. Some additional functions were identified to be important. The functions that were mentioned to be important or useful by multiple parties are summarised in table 5.5.

Table 5.5.: Most important functions in the viewer from testing sessions

Function	**Gouda**	**VRBZO**	**Stantec**
Click objects to see attributes	Very important	Very important	Very important
Switch between 2D and 3D	Very important	Very important	Very important
Filter objects based on atrributes	Important	Important	Nice to have
Viewing underground	Useful	Useful	
Measuring in 3D		Important	Nice to have

The most important function is to be able to click on objects to see their attributes. Filtering objects was also deemed to be important. Viewing underground would be useful for Gouda and the safety region. For the safety region, it is important to be able to measure in 3D, for example to see if a ladder can reach something.

Some other functions were identified that could be useful in the future. Clipping a 3D layer to create a new layer could be useful for Gouda. It would be useful for the safety region to see which way is north when viewing in 3D, and to rotate to look north with a button. In addition to that it would be good to be able to rotate using buttons in general. Another useful function for the safety region is to visualise water for a flood based on a terrain model. Furthermore, it could be useful to give an offset to features from a WFS layer. For example, to see icons at shoulder height when viewing in 3D. For Stantec, it could be useful to see projections of the 3D features when viewing in 2D. Shadow analyses could be useful in the future according to Stantec, and it could be useful to be able to walk through an area at human height.

5.3. Guidelines for Technical Implementation

In this section, the technical side of the research question will be answered, namely sub-questions 3 (”How can the code for 3D functionality be integrated into an OpenLayers application?”) and 4 (”What should the data flow look like?”). These questions will be answered in the form of guidelines on how to integrate 3D functionality into an OpenLayers based application for publishing datasets in a GIS web viewer. These guidelines are based on the literature study and on the development process of the proof of concept (chapter 4). The guidelines are things that are important to do to effectively integrate 3D functionality into an OpenLayers based application for publishing datasets in a GIS web viewer. In addition, the implementations of these guidelines in the proof of concept will be briefly explained to enhance clarity. In subsection 5.3.1 the guidelines pertaining to the technical aspect of the publishing process are explained. Subsection 5.3.2 discusses the technical recommendations related to the integration of 3D functionality in the web viewer. Subsection 5.3.3 addresses considerations pertaining to the data flow.

5.3.1. Integrating 3D Functionality in an Existing GIS Web Application for Publishing

To be able to visualise 3D datasets a connection with the data source is needed. The interviews and testing sessions determined that having an external server as the data source is the most important way to do this. It should be possible to use a URL to connect with a service providing a 3D dataset. The user should be able to set this up in a similar manner as a connection to a web map service (WMS) or a web feature service (WFS), except a different protocol should be selected.
In the proof of concept, this was achieved by adding a 3D Tileset option to the protocol drop-down list in the 'Add map service' menu. The user only has to enter the URL of the service, a title, and possibly authentication credentials. The implementation in the proof of concept is explained more extensively in section 4.2.

The connection with a service as described above is done once by the user (the publisher of the data). For a viewer to be able to load layers from the service, the connection parameters need to be saved in a database (or saved in a different way). A protocol determines what parameters are saved and how the data from the service should be interpreted by the viewer. Different types of services (WMS, WMTS, WFS, etc.) need their own protocols. Services providing 3D data also need separate protocols: A protocol for 3D Tilesets, a protocol for Cesium terrain models, a protocol for point clouds in a '.las' format, etc.
In the proof of concept, a new protocol was created for 3D Tilesets. The settings that were saved for this protocol were only the standard settings (URL, title, type, etc.) that were needed for all service types. This was sufficient because the Cesium library needs only the URL to read a 3D Tileset. However, more settings can be added to this, for example, to save a style for the 3D Tileset (this will be discussed further in subsection 5.4.3.

Integrating 3D functionality into an existing application means the application will also still be used for purposes where only 2D geo-information is required. Therefore, the 3D functionality should not diminish the performance when only using the application viewer for 2D viewing. When the user creates a viewer only meant for 2D viewing, no 3D scene should be created and no 3D layers should be loaded.
In the proof of concept, this was achieved by giving the user the option to enable 3D functionality in the settings for creating a viewer. This setting is saved as a boolean in the viewer configuration. When the viewer is initialised and this setting is set to false, no 3D scene is created. The implementation in the proof of concept is explained more extensively in section 4.2.

5.3.2. Integrating 3D Functionality in an OpenLayers Based GIS Web Viewer

When integrating 3D functionality into an existing application, it is important to make sure none of the existing functionality for 2D geo-information is lost. The interviews and testing sessions determined users want the functionality for 3D to be an addition, not a replacement. This means when a viewer is created that includes 3D viewing it should be possible to switch between 2D and 3D viewing. This way everything that could previously be done in 2D can still be done. Furthermore, it also means that anything that everything that could be displayed on the 2D map should still be visible on the map when viewing in 3D. This can be achieved by having the OpenLayers map as a part of the 3D scene. When the 2D OpenLayers map is updated, it should also be updated in the 3D scene. In the proof of concept this was achieved by using the Ol-Cesium library OSGeo [2023a]. When creating the Cesium scene, the OpenLayers map is input as an RxJS BehaviorSubject to ensure the scene gets updated when the OpenLayers map is updated RxJS [2024]. This is explained further in subsection 4.3.2.

To show 3D geo-information in a viewer, a 3D visualisation library is needed. Based on the research in this thesis, the recommendation is to use Cesium. This is the most widely used visualisation library and the Cesium library makes it easy to load 3D Tilesets. Additionally, there is a library, Ol-Cesium, which helps with the integration between OpenLayers and Cesium. A more extensive reasoning for why Cesium was used can be found in subsection 3.3.3. OSGeo [2023a], Cesium GS [2024a]

2D datasets are loaded in the viewer as different layer types, i.e. a layer type for WMS, a layer type for WMTS, and a layer type for 2D vector data. In order to keep the functionality for 3D data as similar as possible to the functionality of 2D data, the 3D data types should be added as new layer types, e.g. a new layer type for 3D Tilesets and a new layer type for terrain models. This allows the developer to use existing functions and structures in the code of the application. Furthermore, it allows users to work with 3D layers in the same way as 2D layers.

Although the user should be able to work with 3D layers in the same way as 2D layers, the 3D layers do need to be handled differently. OpenLayers is made for 2D layers, so the OpenLayers functions that are used do not work with 3D layers. Therefore, there should be a separate 'layer manager' that creates, adds, and removes 3D layers. The user interface to add and remove layers should be the same as for 2D layers.
The way this is implemented in the proof of concept is explained in section 4.3.

5.3.3. Data Flow

The interviews and testing sessions have shown that the most important way to store and obtain 3D data is to use an external server as the data source. This means the viewer has to obtain the data from a service and get it to the client. Sending the entire dataset to the client at once to do the visualisation on the client side, which is a thick client approach, is not feasible for large datasets. Rendering on the server and only sending images to the client, a thin client approach, does not allow for smooth viewing and navigation in the viewer. Instead, the 3D data should be sent in parts to the client. The rendering still happens on the client side, which means it is still a relatively thick client, and the performance is influenced by the users browser and available memory. This means the services used to obtain 3D geo-information should be services with some form of tiling. This is by far the most commonly used approach (see chapter 2) and has proven effective in the proof of concept.

For vector data, the OGC standard 3D Tiles should be used. For a web viewer to load this a 3D Tileset should be stored on a server with the extension 'tileset.json'. When the tileset is loaded into the viewer, for example by using Cesium, the viewer sends requests to the service to obtain the tiles with data that are currently visible. When the camera is moved or rotated, and other tiles are needed, new requests are sent by the viewer. Open Geospatial Consortium [2023]

5.4. Overview of Functions and their Importance

A GIS web application for publishing 3D datasets needs to have certain capabilities and functions to make it usable and useful. This section describes the capabilities and functions such an application should or could have. The functions are divided into categories of importance. This is meant to answer the "useful and easy to use" part of the research question ("How can 3D functionality be integrated into an existing OpenLayers based GIS web application for publishing **in a useful and easy to use way**?"). The functions and their importance are based on the results of the literature study (chapter 2), the interviews (section 5.1), and the testing (section 5.2). For each function, there is a description of what it should do and why it is useful. For the essential and important functions, there is a description of how they can be implemented.

The functions and capabilities are divided into three categories: Data types, functions in the web viewer, and functions in the admin website. The importance of each function and capability is categorised as essential, important, nice to have, or possibly useful in the future. The essential functions are necessary to make the application useful. The important functions can add a lot of value and allow the user to better tailor the web viewers to convey their information more clearly. The functions categorised as nice to have are functions that some interviewed parties mentioned would be useful, but are not as important as the functions categorised as important. The functions categorised as possibly useful in the future were mentioned by some parties to be interesting in the future, but are not currently a priority.

Subsection 5.4.1 discusses which data types the application should be able to handle. Based on the interviews it was determined that it is essential to be able to publish vector datasets. The most important functions in the web viewer are discussed in subsection 5.4.2. Two functions were determined to be essential in the web viewer. There needs to be a function to switch between 2D viewing and 3D viewing, and it needs to be possible to click objects to see its attributes. Subsection 5.4.3 describes the functions that the user needs in the admin website during the publishing process. The most important capability is to be able to connect with an external server that provides a service containing a 3D dataset. Furthermore, it is essential to be able to set up colours for the objects in a dataset based on the attributes of the objects. An overview of the capabilities and functions is given in table 5.6, after which they are explained in the subsections.

Table 5.6.: Overview of capabilities and functions and their importance

Function / Capability	Importance
Data types	
Vector data	Essential
Terrain models	Important
Point clouds	Nice to have
Voxel models	Nice to have
Function in viewer	
Switch between 2D and 3D	Essential
Click on objects to see its attributes	Essential
Colour objects based on attributes	Important
Filter objects based on attributes	Important
Allow viewing underground	Important
Show in the list of layers which layers are 3D	Nice to have
Show footprints of 3D objects in 2D view	Nice to have
Measuring in 3D	Nice to have
Shadow analyses	Possibly useful in the future
Drape 2D maps over 3D buildings.	Possibly useful in the future
Walk' through an area on the street or building on ground level	Possibly useful in the future
Functions for Admin	
Add a 3D dataset using an external server as the data source	Essential
Set colours of objects based on attributes	Essential
Have the option to turn 3D capabilities on and off	Important
Set which attributes of 3D objects can be seen in an application	Important
Set aliases of attributes of 3D objects for an application	Important
Obtain a 3D dataset from a database connection	Nice to have
Set extents or 2D polygon outside of which data is not loaded	Nice to have
Have an option to start the viewer in 2D or 3D	Nice to have
Have an option to set the initial viewing height and angle	Nice to have
Use another data source to style objects in a 3D data source	Possibly useful in the future

5.4.1. Data Types

The different data types to be discussed are vector data, terrain models, point clouds, and voxel models. Vector data was determined to be essential during the interviews and testing sessions. All parties needed to be able to publish vector data for the application to be useful. Terrain models are needed for some specific use cases, such as flood management, and can give useful context in other use cases. Terrain models are therefore determined to be important. Point cloud and voxel models were determined to be nice to have. These data types can be useful in the future but are currently not as important as vector data and terrain models in the context of publishing 3D geo-information in a web viewer without the need of coding.

Essential

- Vector data

The interviews and testing sessions have shown that vector data is the most important data type for publishing 3D geo-information. In most of the discussed use cases the main focus is on buildings. In other use cases, the focus is on pipes and cables. Both buildings and pipes and cables are usually modelled as vector data. Vector data can be stored in 3D Tilesets. The tiles within the 3D Tileset would be of the Batched 3D Model format, which is specified in the OGC 3D Tiles standard. If it is stored in a 3D Tileset, it can be loaded by Cesium. The implementation of this in the proof of concept can be found in section 4.3. Open Geospatial Consortium [2023]

Important

- Terrain models

In some use cases, such as flood management, terrain models are the most important data type. In use cases where the main focus is on vector data, a terrain model is often a useful addition to add more context. In order to load a terrain model using Cesium, the terrain model needs to be in the Cesium terrain format. To incorporate this in the proof of concept terrain models need their own layer type, just as 3D Tilesets, WMTS, WMS, etc. have their own layer type. When the layer is loaded in the viewer it can be added to the Cesium scene using the scenes 'setTerrain' method. Cesium GS [2024d]

Nice to Have

- Point clouds

Point clouds were not deemed to be important based on the interviews and the testing sessions. There are however some examples of use cases for point clouds in the literature study (see chapter 2). In the interviews, some parties did mention they could have some use cases with point clouds. Point clouds can also be stored in 3D Tiles and loaded into the viewer using Cesium. The tiles within the 3D Tileset would be of the Point Cloud format, which is specified in the OGC 3D Tiles standard. Open Geospatial Consortium [2023], Cesium GS [2024c]

- Voxel models

Voxel models were not mentioned in the interviews or testing sessions. There are some examples of examples of voxel models in web viewer in the literature study (see chapter 2). Voxel models can be saved in an Indexed 3D Scene Layers (I3S) dataset. This can be read by Cesium using the 'I3SDataProvider' class. OGC [2023], Cesium GS [2024c]

5.4.2. Functions in Viewer

This subsection describes which functions are needed in the web viewer. These are functions that a visitor of a viewer can use.

Essential

- Switch between 2D and 3D

For the user, this is a button that can be clicked to switch between viewing in 2D, and only seeing the 2D map, and viewing in 3D which would also show the loaded 3D layers.

Value:
When integrating 3D functionality into an existing application, it is important to keep the existing 2D functionality. Therefore, it is essential to have a switch in the viewer that toggles between 2D viewing and 3D viewing. During the testing sessions, all parties agreed that this is needed, and such a switch is also present in most of the existing applications in chapter 2.

Implementation:
This function has been implemented in the proof of concept. Using the Ol-Cesium library, an OLCesium instance is created with the OpenLayers map as input. This class has a method 'setEnabled' which turns 3D viewing on or off. A button was created in the toolbar in the viewer which calls this method. The full implementation can be found in section 4.3.

- Click on objects to see its attributes

Value:
When viewing 3D objects, it is essential to be able to see the attributes of those objects. A web viewer showing 3D objects is usually made to convey certain information, and while seeing the geometry in 3D adds a lot of value and information, the attributes of the objects usually also contain important information. All the interviewed parties have mentioned this function as something that is essential.

Implementation:
To be able to display the attributes of a feature, such as a building, in a Cesium3DTileset (the instance of a 3D Tileset in Cesium), the 'Cesium3DTileFeature' is needed Cesium GS [2024c]. This can be obtained by using the 'screenSpaceEventHandler.setInputAction' method of the Cesium scene with 'ScreenSpaceEventType.LEFT_CLICK' as the 'type' parameter. This allows selecting a feature in the scene. As the 'action' parameter a function can be used in which the 'pick' method of the Cesium scene is used. The pick method returns the Cesium3DTileFeature. Cesium GS [2024d] Cesium GS [2024c]

The Cesium3DTileFeature provides access to the attributes of the feature. They can be obtained by first using the method 'getPropertyIds' and then looping through the ID's and using the getProperty method. The attributes can then be used to make a pop-up box displaying them. Cesium GS [2024c]

In the proof of concept, this could be implemented by setting an event handler when initialising the Cesium layer manager.

Important

- Colour objects based on attributes

Value:
During the testing session, it was concluded that it is more important to be able to colour the objects from the admin page than to allow users of the viewer to do it. However, the interviews and testing session did show that there is also value to being able to do it in the viewer. Colouring objects, such as buildings, based on their attributes makes it possible to better convey the information in the datasets by putting the focus on certain aspects of the datasets.

Implementation:
Styling of Cesium 3D Tilesets can be done using the 3D Tiles styling language Cesium GS [2024e]. A Cesium 'Cesium3DTileStyle' object can be assigned to the 'style' attribute of a Cesium3DTileset instance. In a Cesium3DTileStyle the style is defined as a JSON object, in which conditions can be set to define the styling of a tileset. A Cesium3DTileStyle can have 'color' properties in which conditions are set to colour features based on their attributes. Cesium GS [2024e]

To allow a user to do this in the viewer, an interface needs to be created where the user can input styling rules. A function then has to be made that converts this input to a JSON object using the styling language and adds the style to the tileset.

In the proof of concept, this could be implemented by adding a new component '3D Tiles styling' to the viewer in which the interface is made. This could then use a new function in the Cesium layer manager which creates and adds the style.

- Filter objects based on attributes

Value:
Filtering objects can help to convey the information better by drawing attention to the objects that are relevant for the current interests of the user of the viewer. This was named as important by most parties in the interviews.

Implementation:
The implementation of this function is mostly the same as the implementation of colouring objects based on attributes. A Cesium3DTileStyle can also have 'show' properties which determine whether features are shown. This can be used to not show features based on conditions. Cesium GS [2024e]

In the proof of concept, the interface for this could be integrated into the existing 'filtering' interface. This could have an extra filter type option '3D Tileset filter'. This interface could then again use a new function in the Cesium layer manager which creates and adds the style.

- Allow viewing underground

Value:
For some of the identified use cases, some objects are underground. The most important example is the management of pipes and cables. To be able to view these objects it needs to be possible to view underground.

Implementation:
This can be achieved by making the ground transparent, and Cesium provides functionality for this. The Cesium scene has an attribute 'globe'. The attribute 'globe.translucency.enabled' should be set to true. The transparency can be set directly by setting 'globe.translucency.frontFaceAlpha' to a value between 0 and 1 (1 is not transparent, 0 is fully transparent). Alternatively, the transparency can be set using a Cesium 'NearFarScalar' to set 'globe.translucency.frontFaceAlphaByDistance', in which case the transparency will be different based on the distance of the camera to the terrain. Cesium GS [2024c]

This can be implemented either by setting this up on the admin side and not allowing a visitor of the viewer to change the transparency, or by adding a slider to the viewer which allows the visitor of the viewer to alter the transparency.

Nice to Have

- Show in the list of layers which layers are 3D

In the list of layers in the viewer, all available layers are shown and can be turned on or off. It could be helpful to show which layers are 3D layers. This would make it clearer to the visitors of the viewers. This can also be done by including 3D in the layer names, or keeping them in a '3D' folder, but displaying it with an icon or by greying them out while viewing in 2D could be helpful.

- Show footprints of 3D objects in 2D view. (e.g. outlines of 3D buildings on 2D view)

This would mean you can see the footprints of buildings from a 3D dataset while viewing in 3D. This can be useful to make navigating the datasets easier.

- Measuring in 3D

Allowing measuring in the 3D viewer can be helpful for some use cases. For example, a safety region could use this to measure where their ladders can reach. Measuring can include distances, area, and angles.

Possibly Useful in the Future

- Shadow analyses

Shadow analyses would allow visualising shadows cast by objects at different times of the day. Shadow analyses can be a useful addition in showing city development plans, or other building plans. Shadows and sunlight are often important for citizens or other stakeholders who see the building plans.

- Drape 2D maps over 3D buildings

This would mean 3D building objects get the colour that a 2D map has at the location of the building object. This would be useful if the data that has important additions to the information in a 3D dataset is stored as a separate 2D map.

- 'Walk' through an area on the street or building on ground level

'Walking' through an area or through a building by viewing from ground level and clicking to go forward could be a useful addition. This could allow visitors of the viewer to get a better idea of what the situation is like in real life.

5.4.3. Functions for Admin

This subsection describes which functions are needed on the admin website. These are functions that the user can use during the publishing process.

Essential

- Add a 3D dataset as a service using an external server as the data source

This means connecting to an external server using a URL (and login details if needed). The 3D dataset is stored on that server. This can be a server made by the user themselves or it can be an external data source. From this service, one or multiple layers can be obtained.

Value:
Interviews and testing have shown that this is the most common way potential publishers of 3D geo-information want to connect to a data source. Users who were already using 3D data currently already have their 3D datasets stored on servers. This function also allows easy combination of data from different external sources.

Implementation:
This function can be implemented in a similar way to how a connection to a 2D web map service is implemented. The connection to the server can be saved as a service from which layers can be obtained. If a layer is then added to a web viewer, the configuration of the web viewer needs to include the layer with its URL and possible other parameters. This function was included in the proof of concept, the details of this are discussed in subsection 4.2.1.

- Set colours of objects based on attributes

Value:
Allowing the user to set the colours of objects before publishing them in a web viewer helps them convey the information that they want to share, and it helps to make the information easier to understand for someone using the viewer. Most of the interviewed parties stated that this is something they needed, and in the testing sessions it was concluded that it is more important to do this on the admin page than to let a user of the viewer do it themselves.

Implementation:
As discussed in the implementation of the viewer function 'Colour objects based on attributes', a style can be defined as a JSON object and added to a Cesium3DTileset. An interface can be added where the user can define styling rules when setting up a viewer. A function is then needed to convert this input to a JSON object with styling rules using the 3D Tiles styling language. This JSON object can be saved as a file or in a connected database and can be added to the viewer configuration. When the viewer is started the style can be added to the Cesium3DTileset.

In the proof of concept this interface can be added to the styling tab of the application settings. In the back-end, the viewer configuration definitions need to be extended to allow a style to be saved.

Important

- Have the option to turn 3D capabilities on and off

Value:
The interviewed parties have stated in the interviews and testing sessions that this function would be very useful for them. They would still want to create viewers that only include 2D functionality and data.

Implementation:
This function can be implemented by giving the user an option to turn 3D functionality on or off when creating a viewer. This setting needs to be saved in the viewer configuration, and when the viewer is loaded, a 3D scene should only be made if the option was set to have 3D functionality. This function was implemented in the proof of concept. This implementation can be found in section 4.2.

- Set which attributes of 3D objects can be seen in an application

This means getting an interface on the admin page where the user can set which attributes of objects in a 3D layer can be seen in the viewer when a visitor of the viewer clicks an objects.

Value:
This function was stated to be important during the testing session by two of the interviewed parties. This allows them to control what information the visitors of the viewers can see. It also helps to put the focus on the information that is relevant for the purpose of the viewer.

Implementation:
When setting up a viewer, the user should be able to get an interface for each layer where they can write the name of an attribute of a feature in the 3D Tileset and check a box that determines it should not be displayed in the viewer. These attributes and whether they should be shown should then be saved with the layer settings. In the viewer, when the attributes are printed, the function that prints the attributes should check them against the saved attributes in the layer setting to check if they should be printed.

A more user-friendly approach would be to give the user a list of attributes instead of letting them write the attributes themselves. However, in a 3D Tileset, the objects in each tile can have different attributes. The attributes of the objects in the tile are saved in the 'batch table' of the tile Open Geospatial Consortium [2023]. This means this approach would require either a separate file that specifies all the attributes in the 3D Tileset, or it would be needed to check all tiles in the tileset for their attributes. This could of course be done only once, after which the attributes could be saved.

- Set aliases of attributes of 3D objects for an application

This means getting an interface where the user can write alternative names for attributes of objects in a 3D layer. This will set the names that someone visiting the viewer will see.

Value:
This function was mentioned together with the function above, which sets which attributes can be seen. Some attributes in datasets have very technical names that visitors of the viewers would not easily understand. This function allows the user to control how the information is brought to the visitors of the viewers.

Implementation:
The implementation of this function would be mostly the same as the function above to set which attributes of 3D objects can be seen. The user would get an interface on the admin page where they can write the attribute for which they want to set an alias, and write the alias. This should then be saved with the layer settings, and the function in the viewer that prints the attributes should print the alias. The more user-friendly approach, e.i. giving the user a list of attributes, again requires looping through the entire tileset, as explained above.

Nice to Have

- Obtain a 3D dataset from a database connection

Obtaining data from a database connection, instead of from an external server, can be useful for use cases where the users often need to alter or add to the data from the viewer.

- Set extents or 2D polygon outside of which data is not loaded

This can be useful if the user is interested in a certain area, such as a municipality or a safety region, and wants to use datasets that cover a larger area. This can be achieved by setting extents, or by using a polygon from a 2D layer to set the boundaries.

- Have an option in application settings to start the viewer in 2D or 3D

For some viewers, the purpose might be mostly to view the data in 3D. For these viewers, the viewer should start 3D viewing. For other viewers, the 3D datasets might only be an addition. In these cases, the viewer should be started in 2D viewing. Therefore, it would be useful to have an option on the admin page to choose whether the viewer starts in 2D or 3D viewing.

- Have an option in application settings to set the initial viewing height and angle

This means the user can set what a visitor of the viewer sees when the viewer is initialised. If the purpose of the viewer focuses on a certain object, it can be useful to set the starting viewing height and angle to put the focus on the intended object.

Possibly Useful in the Future

- Use data from another data source to style objects in a 3D data source

This means using data from a separate, possibly not georeferenced, dataset to style objects in the 3D geo-information dataset by using IDs to link the datasets. An example of this could be colouring buildings based on vacancy. The vacancy data might be in a separate '.csv' file which has building IDs that match the IDs in the 3D dataset.

5.5. Main Takeaways from Results

The most important use cases for the interviewed parties have been determined in the interviews. Furthermore, the data types needed for these use cases have been identified. The use cases, along with the data types needed for them, have been summarised in table 5.7. As can be seen in the table, vector data is the most essential data type and terrain models are important for some use cases.

Table 5.7.: Most important use cases and datasets from the interviews

Use case	Named by parties	Needed datasets
Visualise building project	3	Vector data (buildings)
Management pipes and cables	2	Vector data
Asset Management	2	Vector data (buildings)
Object information first responders	1	Vector data (buildings)
Disaster management	1	Vector data (buildings), Terrain Model
Show vulnerable buildings	1	Vector data (buildings)
Show possibilities 3D data	1	Vector data (buildings), Terrain Model
Get feedback Citizens	1	Vector data (buildings)

Furthermore, it was noted in the interviews that the existing functionality for 2D geo-information should not be diminished by the added 3D functionality. The proposed integration of 3D functionality is meant as an addition. All previously existing functionality should be maintained and the performance of the application when using it for 2D purposes should not be degraded. Additionally, the interviews and testing sessions determined that keeping the publishing process for 3D datasets as similar as possible to the process for 2D datasets keeps the application user-friendly.

To make the application useful, some functions were noted to be essential during the interviews and testing sessions. The essential functions are listed below.

- Allow switching between 2D and 3D viewing in the viewer
- Click on objects to see its attributes in the viewer
- Set up colours based on attributes on the admin website
- Set up connections to services providing 3D datasets

In the proof of concept, the switch between 2D and 3D viewing has been integrated, as has the function to set up connections to services providing 3D datasets. The remaining two essential functions have not been implemented in the proof of concept. A general implementation plan has been provided for the essential and important function that were not included in the proof of concept.

Furthermore, the development process of the proof of concept led to a set of guidelines. The guidelines can be used to integrate the code for 3D functionality into a GIS web application for publishing geo-information in web viewers. The guidelines have been implemented in the proof of concept in Tailormap. This means the guidelines apply best to Tailormap. They can also be used within other GIS web applications for publishing geo-information in web viewers, but the exact technical needs might deviate. The guidelines are listed below.

- Allow user to set external servers providing services with 3D Datasets as data sources, such as services providing 3D Tilesets.
- Create a new protocol for saving the connection parameters for the 3D services. Each type, such as 3D Tiles, terrain models, I3S, etc., should have its own protocol.
- Only load 3D scenes and layers if they are needed, so as to not diminish the performance when only using the application viewer in 2D.
- Keep all existing 2D functionality and create a function to switch between 2D and 3D viewing.
- Use the CesiumJS library to visualise the data in 3D.
- Create a new layer type for 3D datasets in the viewer. Each type, such as 3D Tiles, terrain models, I3S, etc., should have its own layer type. This allows the viewer to visualise the layers using CesiumJS.
- Create a new layer manager for managing 3D layers and the 3D scene.

6. Conclusions and Recommendations

In this chapter, the thesis research is concluded. The conclusions are provided in section 6.1, and the recommendations for future work are given in section 6.2.

6.1. Conclusions

The first part of the conclusion is the answers to the research questions, which are given in subsection 6.1.1. Subsection 6.1.2 discusses to what extent and with what certainty the research questions were answered. The limitations of this research are explained in subsection 6.1.3, and the main contributions of this research are described in subsection 6.1.4.

6.1.1. Research Questions

How can 3D functionality be integrated into an existing OpenLayers based GIS web application to ensure that publishing is both useful and user-friendly?

To integrate 3D functionality into an existing OpenLayers based GIS web application for publishing, the technical guidelines given in the answers to sub-questions 3 and 4 should be followed. The process of publishing 3D geo-information should be made user-friendly by keeping the process as similar as possible to publishing 2D geo-information and by keeping all of the previously existing functionality. The essential functions given in the answer to sub-question 2a should be implemented. In further development, additional functions should be added depending on the application and its users, although it is likely the important functions given in sub-question 2a will be needed.

1. What types of 3D datasets are available and which are the most relevant use cases for 3D web GIS?

Publicly available datasets include mainly vector data with buildings and terrain models, as discussed in chapter 2. Some point clouds and voxel models can also be found (Actueel Hoogtebestand Nederland [2024], TNO [2023]). The important identified use cases of 3D web GIS are the following.

- Urban planning and building projects
- Asset management
- Management of pipes and cables
- Disaster management

These use cases were identified in the literature study, as discussed in chapter 2, and confirmed in the interviews, as discussed in section 5.1. In the interviews and testing session it was determined that vector data with buildings was the most important data type for most use cases. Terrain models were also determined to be important.

2. How can you make it useful to publish 3D datasets without the need of coding?

To make the application user-friendly the process for publishing 3D datasets should be as similar as possible to publishing 2D datasets, as noted in chapter 5. The user should not have to do any coding and should only need to execute the necessary steps, which are given in the answer to sub-question 2b. To make the application useful some functions are needed, which are given in the answer to sub-question 2a.

2a. What features and options do users need to make publishing 3D data useful?

To make the publishing application useful, the users need certain functions. These were determined based on the interviews and the testing sessions, as explained in chapter 5. First, the essential functions are listed, and then the important functions. The functions are discussed in more detail in section 5.4.

Essential:

- Allow switching between 2D and 3D viewing in viewers.
- Click on objects to see its attributes in the viewer.
- Set up colours based on attributes in admin.
- Set up connections to services providing 3D datasets.

Important:

- Colour objects based on attributes in the viewer.
- Filter objects based on attributes in the viewer.
- Allow viewing underground.
- When creating a viewer, have the option to turn 3D capabilities on and off.
- Set which attributes of 3D objects can be seen in an application.
- Set aliases of attributes of 3D objects for an application.

2b. What are the necessary manual steps users must undertake in the publishing process?

The manual steps in the process of publishing 3D geo-information are listed below. The steps were determined by the development process of the proof of concept, which is discussed in chapter 4.

1. The user needs to select the data sources they want to use. This means providing the URLs for the services they want to use, for both the 3D geo-information and the 2D geo-information.

2. The user needs to enable 3D viewing when setting up a viewer. This step is necessary because the application should also still be usable for 2D-only use cases.

3. The users needs to select which layers, both 2D and 3D, they want to add to the viewer.

These are the necessary steps that are always needed. However, some additional functions can be added to enhance the possibilities of the publishing process. For example, an interface for styling 3D Tilesets would be a valuable addition. This and other functions are discussed in 5.4.3.

3. How can the code for 3D functionality be effectively integrated into an existing OpenLayers application?

A working implementation was made in the existing application Tailormap, which is described in chapter 4. In the development process of this proof of concept, some recommendations were determined that can be used as a guideline when integrating 3D functionality into an existing OpenLayers application for publishing. They are discussed in detail in section 5.3, but a list of the recommendations is given here.

- Allow user to set external servers providing services with 3D Datasets as data sources, such as services providing 3D Tilesets.
- Create a new protocol for saving the connection parameters for the 3D services. Each type, such as 3D Tiles, terrain models, I3S, etc., should have its own protocol.
- Only load 3D scenes and layers if they are needed, so as to not diminish the performance when only using the application viewer in 2D.
- Keep all existing 2D functionality and create a function to switch between 2D and 3D viewing.
- Use the CesiumJS library to visualise the data in 3D.
- Create a new layer type for 3D datasets in the viewer. Each type, such as 3D Tiles, terrain models, I3S, etc., should have its own layer type. This allows the viewer to visualise the layers using CesiumJS.
- Create a new layer manager for managing 3D layers and the 3D scene.

4. What should an effective data flow for 3D datasets in a web GIS environment look like?

The 3D datasets should be stored on an external server. The web GIS viewer should request the data from the server in tiles and render it on the client-side.

4a. How should 3D data be stored?

The 3D datasets should be stored on an external server. It was noted in the testing sessions that obtaining data from a service on an external server was more useful than obtaining the data from a database connection, as discussed in section 5.2. For vector data, it should be stored as a 3D Tileset, following the OGC 3D Tiles standard. This format allows streaming large datasets with buildings to a web viewer Open Geospatial Consortium [2023].

4b. What processes should be handled server-side and client-side in the management of 3D data?

The rendering should be done client-side, but tiling services should be used for 3D data. The data should be requested from the server in tiles, based on the position of the camera in the 3D scene. Tiling is needed to allow streaming large datasets and allow visualising them. For 3D vector data, services should be used that apply the OGC standard 3D Tiles.

6.1.2. Discussion

A functional proof of concept has been developed to allow users to publish 3D datasets in a 3D GIS web viewer. This was made by integrating the 3D functionality into an existing OpenLayers based GIS web application for publishing. This has been designed to user-friendly, and the user does not need to write any code. Furthermore, user needs have been found which determine what is needed to make such an application useful.

This proof of concept was made by integrating the 3D functionality in Tailormap. The users that were consulted were users of Tailormap. This means that the answers to the research questions mainly apply very well to integrating 3D functionality in Tailormap. When 3D functionality is integrated into another OpenLayers based GIS web application for publishing, the answers to the research question will still provide useful guidelines, but the exact technical and user needs might deviate slightly. The technical recommendations can still be used as guidelines and give ideas about what to watch out for, but the exact technical needs might be different due to the way the application is build up. The consulted users did come from different types of parties, e.i. a municipality, a safety region, the Cadastre, and private companies, which makes their combined needs more general. However, the users of a different application, that is focused on slightly different use cases, might have slightly different needs. Still, the user needs identified in this thesis provide a general overview of user needs for publishing 3D geo-information.

6.1.3. Limitations

This subsection will discuss the most important limitations of this research. Five different parties were interviewed, but if more parties were interviewed additional information about user needs could have been obtained. The interviewed parties were all Dutch and working mostly with datasets in the Netherlands. Furthermore, the proof of concept and results were using one specific application (Tailormap). Finally, the proof of concept was developed only for 3D Tilesets, other data types have not been tested.

Amount of Users Interviewed

To obtain information about user needs, five parties were interviewed and testing sessions were held with three parties. Additionally, some information was obtained during the Tailormap users day. The parties were different kinds of users (municipality, safety region, Dutch Cadastre, and private companies) to get information about varied use cases. However, more information could be gathered by interviewing more parties.

Interviewed Parties

All five interviewed parties were Dutch, and they all worked mostly with use cases in the Netherlands. This can give biases to the identified user needs that influence which functions and capabilities are deemed to be important. If users from multiple countries would be interviewed, the results might be different. For example, users in other countries might give terrain models a higher priority, since the Netherlands has minimal elevation variation. Furthermore, other use cases might be more important in other countries.

Implementation in One Application

The technical research was done by extending Tailormap to integrate 3D functionality. The technical recommendation obtained from this part of the research can be applied to other OpenLayers based web applications for publishing geo-information. However, they were only tested in Tailormap. In a different application, the technical recommendations might not apply in the exact same way. Furthermore, if 3D functionality is integrated into another application some additional technical recommendations might be found, since other applications might have additional (or fewer) complications.

Data Types

The proof of concept was developed to handle 3D Tilesets. It was tested with tilesets containing vector data, but not with tilesets with point clouds. Although the research has shown vector data was the most important data type, incorporating functionality for other data types would add value to the solution. Especially terrain models would add a lot of value, based on the user research. The architecture of the code was set up in such a way that it is easy to add functionality for additional data types.

6.1.4. Contributions

Three main contributions were made by this thesis. Firstly, Tailormap has been extended with 3D functionality, and this can be used to publish 3D geo-information. Tailormap is fully open source, which means anyone can use it. The proof of concept will be developed further into a full product. Tailormap has many users in the field of GIS, and some of them are likely to use the added 3D functionality in the future.

Secondly, knowledge was added to the field of GIS about the technical needs of integrating 3D functionality into an OpenLayers based GIS web application for publishing geo-information. Recommendations were given that can be used as guidelines when making such an extension.

Lastly, knowledge was added to the field of GIS about the user needs for publishing 3D geo-information to a 3D GIS web viewer without the need for coding. An overview was given of the capabilities and functions the users need.

6.2. Future Work

In this section, suggestions are given for further research to build on the results of this thesis.

User Research

To obtain a more general overview of user needs, more (potential) users should be consulted. If a large number of users is consulted about the capabilities and functions in this research, the importance of those functions could be determined statistically. Furthermore, the users should be from as many different countries as possible, and the type of users (municipalities, private companies, safety regions, etc.) should be varied.

Extend Implementation

The created proof of concept should be developed further into a full product. This means fixing some issues that were not important for a proof of concept. This includes things such as writing checks to not allow users to add 3D Tilesets as a basemap, cleaning the code a bit more, and turning of the buttons that only work in 2D when viewing in 3D. Furthermore, the essential and important functions and capabilities described in section 5.4 that have not yet been implemented should be implemented.

Additional Implementations

It would be useful if the generality of the technical recommendations were tested. This can be done by implementing them by integrating 3D functionality into other OpenLayers based GIS web applications for publishing. This would allow determining to what extent the technical recommendations hold for other applications and how general they are. Furthermore, additional technical recommendations could be found and a more general overview could be made.

A. Reproducibility Self-assessment

This appendix provides a self-assessment by the author about the reproducibility of the conducted research. The template for this appendix was given with the Latex template provided by the MSc. Geomatics GEO2020 course organisers [1]. The template included figure A.1, a list of five criteria to be evaluated, and the section headers. In section A.1 marks are given for the five criteria. An elaboration on the marks and a more extensive evaluation can be found in section A.2.

A.1. Marks for each of the criteria

In this section, a mark is given for five criteria relating to the reproducibility of the research: input data, preprocessing, methods, computational environment, and results. The marks are 0, 1, 2, or 3, the meaning of the marks is explained in figure A.1.

The following grades have been given:

1. input data: 3
2. preprocessing: not applicable
3. methods: 3
4. computational environment: 3
5. results: 1

[1] https://3d.bk.tudelft.nl/courses/geo2020/templates/

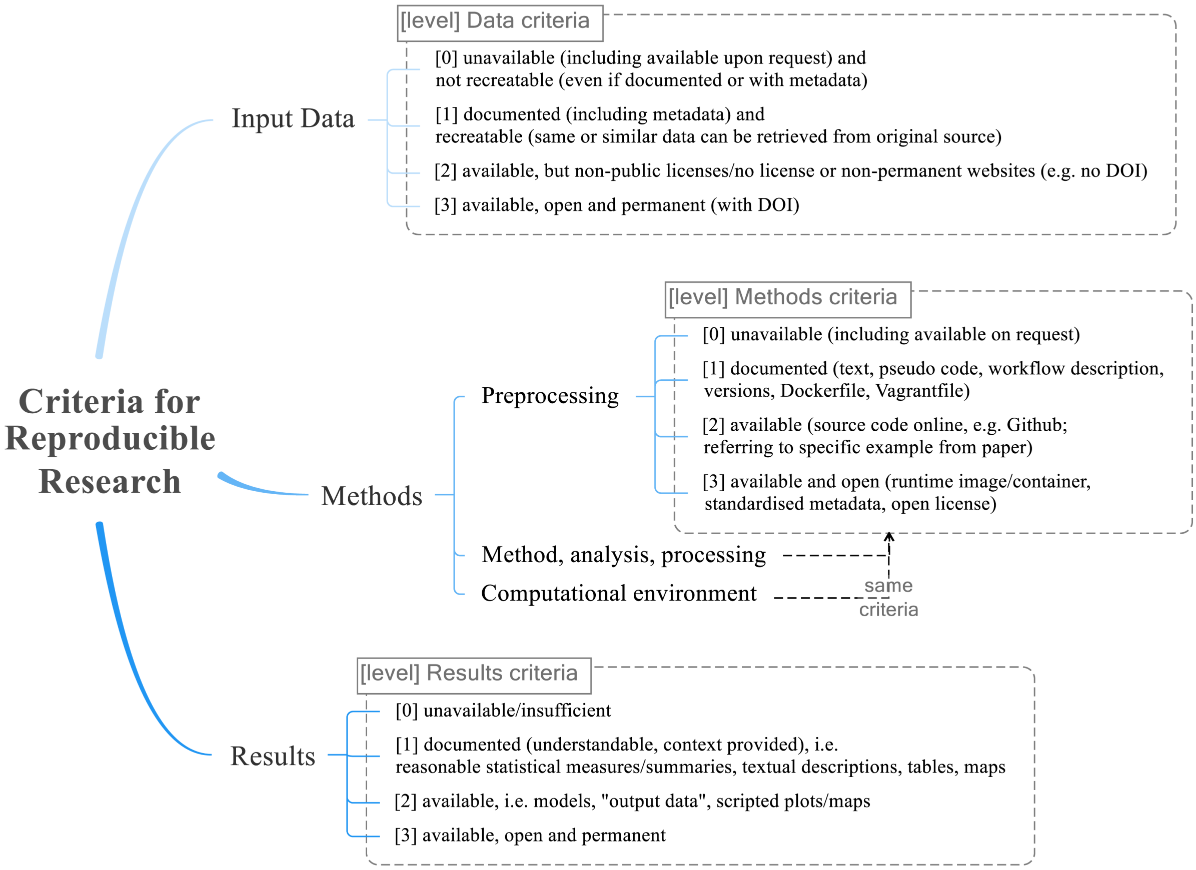

Figure A.1.: Reproducibility criteria to be assessed.

A.2. Self-reflection

In this section the marks given in section A.1 will be elaborated upon. The input data used to develop and test the proof of concept was all open data. The input data will be further discussed in subsection A.2.1. The methods, including the preprocessing and the computational environment, are discussed in subsection A.2.2. All of the code used for this research is available on GitHub under an open licence. The reproducibility of the results is discussed in subsection A.2.3. The results are documented, but they partially consist of measures of importance, which are inherently partially subjective.

A.2.1. Input Data

The datasets used to test the proof of concept are listed below, along with their applicable licence. All of the datasets are publicly available under an open licence. Therefore, the given mark for the reproducibility regarding the input data is 3. Everyone can access and use all of the datasets.

- **OpenStreetMap** The OpenStreetMap data is publicly available under the Open Data Commons Open Database License (ODbL). OpenStreetMap contributors [2017]
- **PDOK Luchtfoto RGB (Open)** The areal imagery is made available by the Kadaster through a WMTS service under a CC-BY 4.0 licence. Kadaster [2024b]
- **3DBAG** The 3DBAG dataset is provided as a 3D Tiles dataset by the Kadaster under a CC-BY 4.0 licence. Kadaster [2024a]
- **Rotterdam Wilhelminakade – Textured** This 3D Tileset is made publicly available by the municipality of Rotterdam and the 3D Tiles Nederland initiative. No specific licence is stated. 3D Tiles Nederland and Gemeente Rotterdam [2024]

A.2.2. Methods

The proof of concept is made in Tailormap. Tailormap is fully open source and all the code is available on GitHub. The version of Tailormap that includes the proof of concept is available in a separate branch [2] [3]. On the GitHub page, a Docker image and a Docker container are provided, which can be used to run the application Merkel [2014]. All the code is publicly available under an MIT Licence. For these reasons, the reproducibility of the methods gets a mark of 3.

The development and testing were done using a laptop running on Windows and a laptop running on Ubuntu. Some software is needed to develop, build, and run the application, but there are no specific requirements on which software to use for this. A guide on how to develop and run Tailormap is provided on the GitHub page. For these reasons, the reproducibility of the computation environment gets a mark of 3. No mark was given for preprocessing since no preprocessing was done.

A.2.3. Results

The results consist of a set of capabilities and functions along with their importance, as well as a set of technical guidelines. The results are documented and justified, so they comply with the criteria for a mark of 1. However, the measure of importance is inherently partially subjective, which reduces the reproducibility. Furthermore, the technical guidelines were determined by the development process of the proof of concept, but another person might draw slightly different lessons from the same development process, which also reduces the reproducibility. For these reasons, the mark for the reproducibility of the results cannot be higher than 1.

[2] https://github.com/B3Partners/tailormap-viewer/tree/cesium
[3] https://github.com/B3Partners/tailormap-api/tree/cesium

B. Interviews

In this appendix the interviews are summarised. The interviews were held in Dutch, so the summaries were translated to English.

B.1. Municipality of Gouda

23-01-2024 Margot Quist, Daniël van den Berg, Anne Jans

What 3D datasets do you have or use within the municipality?

The AHN dataset is sometimes used. In the municipalities we do not have much 3D data. There is elevation data for light poles. There is also some data for cables and pipes, but they are not very accurate.

What 3D datasets do you plan on obtaining in the future?

There are a number of things we would like (better) 3D data for: Trees, water levels, sewage, cables, and pipes. A lot of these things would be underground. For this it would be good if the viewer had an option to make the ground see-through.

What do you plan to use the 3D datasets for?

Digital twins would mostly be used to share plans in urban development.

With what type of people would you want to share the data?

A digital twin would be shared with citizens and colleagues. It would be good if some layers can only be available to some people.

Who would need to be able to publish or edit the online 3D GIS application, and how much knowledge do they have of GIS?

It would be published by people with GIS knowledge, but it should still not be too difficult.

What options (viewing, editing, styling, clipping, filtering, etc.) would be needed in the online 3D GIS application?

Styling and filtering are important. An example of filtering could be colouring building of a certain type. Editing and clipping are not as important now. A useful feature could be being able to look at the data from ground level. Most important is that is works well, more than specific functions.

B.2. Safety Region Brabant Zuid-Oost

23-02-2024 Joachim Tempelaars

What 3D datasets do you have or use within the safety region?

We do not have real 3D datasets, but we have 2.5D. We have object information of buildings with layers for different building layers. We have this data for larger buildings where this information is needed for the fire brigade and other first responders.

What do you use these datasets for?

The fire brigade and other first responders get this information on their way to an emergency. They get it in their car on a tablet.

What 3D datasets do you plan on obtaining in the future and what do you plan to use the datasets for?

A colleague is working on adding a z-component to the existing dataset. This would move the dataset from different layers to 3D. This would allow first responders to better see where important things are like hazardous substances and fire escapes.

Visualising incidents that affect larger areas now happens in 2D. This could be done in 3D in the future. This could help understand risks for different parts of buildings.

Datasets of vulnerable buildings are now in 2D. These are datasets where buildings are coloured according to where vulnerable people live (clustering is done to ensure privacy).This could be made 3D to aid with municipal decision making.

Could an online web application be a good solution to share the 3D datasets?

Yes, this could be a good way to share the data.

Who would need to be able to publish or edit the online 3D GIS application, and how much knowledge do they have of GIS?

People who administer the datasets have a decent amount of GIS knowledge. People who will read the data do not have so much GIS knowledge. It is important that it is easy for them to understand the data when viewing.

What options (viewing, editing, styling, clipping, filtering, etc.) would be needed in the online 3D GIS application?

Clicking on objects to see its attributed and filtering are the most important options.

B.3. Kadaster

30-01-2024 Eric Hagemans
23-04-2024 Richard Witmer, Eric van Mullekom, Caroline Groot

What 3D datasets do you have or use within the Kadaster?

We have the PDOK 3D viewer. This allows all citizens to view the 3D basisvoorziening (base provision) which shows the buildings of the Netherlands in 3D and the terrain model of the Netherlands.

What do you use these datasets for?

The purpose of the PDOK 3D viewer is to inform potential users about the possibilities of the published 3D data.

What do you plan to do with 3D datasets in the future?

There are no plans for new products, but we do plan to develop the current products further. We are developing 3D functonality for the 'verbeterdekaart.nl' viewer, which is used for getting feedback from citizens on the public registries. We also plan to further develop the 3DBAG. In the future we plan to publish a new dataset of the buildings in LoD 2.2 once every year, and we also want to include a terrain model. The 3DBAG product from the TU Delft would then get a more experimental function.

We are also looking into the combination of BIM with geo-information for the purpose of 3D property registration. The combination with large scale maps is difficult and the Kadaster can not show some information about plots and public registries in the way some external parties want.

What options (viewing, editing, styling, clipping, filtering, etc.) are needed in the PDOK 3D viewer?

The viewer is meant to show potential users of the data what data there is. It is not meant as a full GIS application. It does have the function of clicking on buildings to see their attributes.

B.4. Antea Group

02-05-2024 Arjan Wagemakers, Frank Leijssenaar

What 3D datasets do you have or use within Antea and what do you use them for?

Antea is a large company in which a lot is done with 3D data. It is used for building projects. In these projects there are four phases: Planning, design, building, and management. The team that we work in works mostly on the management phase. In the planning and design phases 3D is used a lot to visualise the plans and how they will look in the environment. This means the building itself is visualised, but also the surroundings with other buildings and other objects. The management part is currently mostly being done in 2D.

What do you plan to do with 3D datasets in the future?

There is a wish to move more to 3D for the management phase. If you design and build a building in 3D, you also want to manage it is 3D. There is also a need for 3D viewing for the management of cables and pipes, as well as sewage systems. For the management of sewage systems, currently videos are being made inside to be able to spot damage. Currently you can only view the damage in the videos, but it would be useful if that damage could be shown in 3D in a viewer.

Could an online web application be a good solution to share the 3D datasets?

Yes.

With what type of people would you want to share the data?

It would be mostly used internally for our own applications in the management application, but it would also be used to share with clients for analyses and advise.

Who needs to be able to publish or edit the online 3D GIS application, and how much knowledge do they have of GIS?

For management purposes we would do that. We have a lot of GIS knowledge.

What options (viewing, editing, styling, clipping, filtering, etc.) are needed in the online 3D GIS application?

It is very important to be able to click on objects to see their attributes. It would also be useful to be able to add attributes to existing objects. Colouring objects based on their attributes is very important. Another useful function would be to be able to walk through a building and be able to click objects in the building.

In the future it could also be interesting to be able to see reports of citizens (users of the viewer) on the map. It could also be interesting to show lighting in 3D. Something that would also be very useful for us is being able to make a cross section of an object when clicking on it. We already have some 3D information for a lot of objects in our database, like what the diameter or height is. It would be useful to create 3D visualisation from this in the viewer.

B.5. Stantec

03-04-2024 Anton Vissers

What 3D datasets do you have or use within Stantec?

We are currently using a platform that allows 3D. It uses Mapstore. It is used mostly to show buildings. The buildins come from BIM datasets and CityGML, but everything gets converted to Cesium3DTileset. For some specific cases we also use point clouds, a 3D mesh, and datasets with some small extra things like road signs.

What do you use these datasets for?

These datasets are mostly used to visualise building projects such as an industrial estate. This is used to guide the development of the project. This means it is shared with the project team which can include our colleagues and external people from municipalities or other companies. It is also sometimes used for information evenings with citizens. The datasets are also used for asset management.

What do you plan to do with 3D datasets in the future?

In the future we would like to be able to visualise real time data with the 3D dataset, like real time wind turbine revenues. We would also like to have some standard libraries with 3D objects that can be easily places in a model.

Could an online web application be a good solution to share the 3D datasets?

Yes.

Who needs to be able to publish or edit the online 3D GIS application, and how much knowledge do they have of GIS?

The people from the company that do this have quite a lot of GIS knowledge, but they are not all programmers. They do currently use FME to prepare the 3D Tilesets.

What options (viewing, editing, styling, clipping, filtering, etc.) are needed in the online 3D GIS application?

It is very important to be able to click on things to see their attributes. It is also very important to be to able to colour and filter buildings based on their attributes.

C. Testing

This appendix gives the summaries of the testing sessions. In the testing sessions the users were taken through the process of publishing 3Dgeo-information with the proof of concept. This process consist of selecting data sources on the 'admin - catalog' page, setting up the viewer in the 'admin - applications' page, and looking at the viewer. They were also asked what additional functions they thought would be important. The testing session were held in Dutch, so the summaries are translated to English.

C.1. Municipality of Gouda

24-04-2024 Anne Jans, Daniël van den Berg

C.1.1. Admin - Catalog

Loading the data from a server would be a good way for us to use a data source. Retrieving data from a database connection is less important. Further in the future it could be useful to be able to load BIM models.

C.1.2. Admin - Applications

It is very good that you can turn the 3D capabilities for the viewers on or off.

Additional Functions

It would be very useful to be able to set up colours for buildings based on their attributes from the admin page. It could be useful to set up a polygon, from a different data source outside of which 3D features are not loaded. With this we could set it up so buildings outside of the borders of the municipality are not loaded.

In the future it could also be useful to colour buildings based on data from other datasets, by linking them through some form of ID's. With this we could, for example, colour buildings based on vacancy which we get from our own .csv file.

It would be useful if we could work with tables with attributes for 3D layers in the same way as for 2D layers. We want to be able to set which attributes are visible in the viewer and which are not.

C.1.3. Viewer

The switch between 2D and 3D viewing is very good to have. It would be best if switching back to 2D automatically sets you back to north being up. It could also be good to show in the layer manaer which layers are 3D.

Additional Functions

It is important to be able to click building to see their attributes.

In the future it could also be useful to be able to clip part of the 3D dataset and make a new layer of this.

It would be useful to view underground, this could be done by making the ground see-through.

C.1.4. Overall Notes

Being able to start testing with some simple things in Tailormap would already be useful. If we can load the 3DBAG dataset and combine it with our maps, that would already be very nice.

C.2. Safety Region Brabant Zuid-Oost

06-05-2024 Joachim Tempelaars, Mark Briels, Bob Hulst (Safety Region Zaanstreek-Waterland)

C.2.1. Admin - Catalog

It is good to get the data from a server as a map service. A Cesium Ion connection would be nice to have. A database connection for 3D data would also be useful, mainly to add information. Being able to load terrain model layers is very important, mainly for floods.

C.2.2. Admin - Applications

It is very good that you can turn 3D viewing on and off for applications.

Additional Functions

It would be useful if you can choose whether the viewer starts in 2D or 3D viewing. In addition to that it would be useful if you could choose the initial viewing height and angle for the viewer. You do also want to be able to set the maximum zoom level, just like for 2D.

It is very important to be able to colour objects by their attributes, perhaps using something like a Styled Layer Descriptor (SLD).

It could also be useful to not load features outside of the borders of our safety region.

A useful thing to have for us would be to set up certain scenarios which automatically turn on or off certain layers. For example, when there is a fire we want to see where the fire hydrants and fire station are, or when there is a flood you want to see the terrain and the dykes.

C.2.3. Viewer

The viewer looks good and it is important to have the switch. It would be good if you tested it for tablets.

Additional Functions

It is very important to be able to click objects to see their attributes. It is also important to be able to filter on attributes.

It would be nice if you could see where north is when viewing in 3D. In addition to that it would be nice to automatically rotate to the north by clicking a button, and to be able to rotate with buttons in general.

It would also be important for us to be able to measure distances in 3D, for example to see if a ladder can reach somewhere.

We would also want to visualise water when there is a flood based on a digital surface model. We would want a blue surface where there is water.

In the future it could be useful to give an offset to features from a WFS layer. For example, to see icons at shoulder height when viewing in 3D.

The last thing that could be useful is to view sewage system underground, or waterways. It could be useful to click a waterway and get a cross section.

C.2.4. Overall Notes

The proof of concept looks good and easy to use. It is simple and effective.

C.3. Stantec

29-04-2024 Anton Vissers

C.3.1. Admin - Catalog

Getting 3D Tiles from a server is the most important. Getting it from Cesium Ion specifically would be useful to have. A database connection for 3D datasets is not very important. If it would make the viewer load the data faster it could be useful.

C.3.2. Admin - Applications

The option to turn 3D on or off is important, because you do not want it to be on automatically.

Additional Functions

It is very important to colour buildings based their attributes.

It could be useful to have an option to let the viewer start in 3D or 2D.

It is also useful to be able to work with attributes from the admin in the same way as for 2D layers: setting which attributes can be seen and setting aliases for them.

C.3.3. Viewer

The viewer looks good, and it is nice you can switch between 2D and 3D. It would probably be best if the switch back to 2D made it go back to north being up. It would be good if you can see which layers are 3D in the layer manager. It would be good to test on mobile phones or tablets.

Additional Functions

It is very important to be able to click objects to see their attributes.

Being able to filter objects would be nice to have.

It could be useful to see projections of the 3D features when viewing in 2D.

Measuring in 3D would be nice to have.

Shadow analyses could be useful in the future.

It could also be useful to walk through the area at human height.

Bibliography

3D Tiles Nederland and Gemeente Rotterdam (2024). 3D Tilesets - Rotterdam Wilhelminakade – Textured.

Abdul-Rahman, A., Pilouk, M., Abdul-Rahman, A., and Pilouk, M. (2008). The Web and 3D Gis. *Spatial Data Modelling for 3D GIS*, pages 233–255.

Actueel Hoogtebestand Nederland (2024). AHN-Puntenwolkviewer.

Agrawal, S. and Gupta, R. (2017). Web gis and its architecture: a review. *Arabian Journal of Geosciences*, 10:1–13.

B3Partners (2024). Tailormap.

BabylonJS (2024). BabylonJS.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., and Çöltekin, A. (2015). Applications of 3D city models: State of the art review.

Brovelli, M. A., Minghini, M., and Zamboni, G. (2018). New generation platforms for exploration of crowdsourced geo-data. *Earth Observation Open Science and Innovation*, pages 219–243.

Castelein, W. T., Bregt, A., and Pluijmers, Y. (2010). The economic value of the dutch geo-information sector. *International Journal of Spatial Data Infrastructures Research*, 5(5):58–76.

Cejudo, I., Irigoyen, E., Arregui, H., and Loyo, E. (2024). Emergency Management and Response Through 3D Maps and Novel Geo-Information Sources. In Grueau, C., Rodrigues, A., and Ragia, L., editors, *Geographical Information Systems Theory, Applications and Management*, pages 92–114, Cham. Springer Nature Switzerland.

Cesium GS (2024a). Cesium.

Cesium GS (2024b). Cesium Ion.

Cesium GS (2024c). Cesium ion SDK API reference documentation.

Cesium GS (2024d). CesiumJS API reference documentation.

Cesium GS (2024e). Styling and Filtering 3D Tiles.

Coote, A., Knight, P., Colding, T. S., Home, R., Fröjdenlund, J., Lysell, G., Streilein, A., Kane, P., Brady, K., Wozniak, P., Plá, M., Bayers, E., Ilves, R., Tuokko, J., Rijsdijk, M., Witmer, R., Cantat, F., Crompvoets, J., and Stoter, J. (2017). Official Publication No 68 European Spatial Data Research.

de Vries, M. and Zlatanova, S. (2004). Interoperability on the Web: the case of 3D geo-data. In *Proc. IADIS International Conference e-Society*, pages 667–674.

esri (2024). ArcGIS - Create 3D web apps.

Esri (2024). Scene Viewer.

GeoSolutions (2024). MapStore.

Guardia, M. L. and Koeva, M. (2023). Towards Digital Twinning on the Web: Heterogeneous 3D Data Fusion Based on Open-Source Structure. *Remote Sensing*, 15.

Haraguchi, M., Funahashi, T., and Biljecki, F. (2024). Assessing governance implications of city digital twin technology: A maturity model approach. *Technological Forecasting and Social Change*, 204:123409.

Ho, S., Crompvoets, J., and Stoter, J. (2018). 3D geo-information innovation in Europe's public mapping agencies: A public value perspective. *Land*, 7.

ISO (2022). Industrial automation systems and integration — COLLADATM digital asset schema specification for 3D visualization of industrial data.

Kadaster (2024a). PDOK - Dataset: 3D Basisvoorziening.

Kadaster (2024b). PDOK - Dataset: PDOK Luchtfoto RGB (Open).

Lei, B., Janssen, P., Stoter, J., and Biljecki, F. (2023). Challenges of urban digital twins: A systematic review and a Delphi expert survey.

Mazzei, M. and Quaroni, D. (2022). Development of a 3D WebGIS Application for the Visualization of Seismic Risk on Infrastructural Work. *ISPRS International Journal of Geo-Information*, 11(1).

Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2.

Mete, M. O., Guler, D., and Yomralıoglu, T. (2018). Development of 3D web GIS application with open source library. *Selçuk Üniversitesi Mühendislik, Bilim Ve Teknoloji Dergisi*, 6:818–824.

Nemoto, T., Masumoto, S., Raghavan, V., Nonogaki, S., and Nakada, F. (2020). Development of open source Web-GIS platform for three-dimensional geologic modeling and visualization. *Spatial Information Research*, 28:645–653.

NgRx (2024). NgRx Docs.

Nibio (2023). Kilden Nibio.

Nishanbaev, I., Champion, E., and McMeekin, D. A. (2021). A Web GIS-Based Integration of 3D Digital Models with Linked Open Data for Cultural Heritage Exploration. *ISPRS International Journal of Geo-Information*, 10(10).

OGC (2021). OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard.

OGC (2023). OGC Indexed 3d Scene Layer (I3S) and Scene Layer Package (*.slpk) Format Community Standard Version 1.3.

Open Geospatial Consortium (2023). OGC ® DOCUMENT: 22-025R4 3D TILES SPECIFICATION COMMUNITY STANDARD APPROVED.

OpenStreetMap contributors (2017). Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org.

OSGeo (2023a). OL-Cesium.

OSGeo (2023b). OpenLayers.

RxJS (2024). RxJS API DOCS.

Schütz, M., Ohrhallinger, S., and Wimmer, M. (2020). Fast Out-of-Core Octree Generation for Massive Point Clouds. *Computer Graphics Forum*, 39(7):1–13.

Threejs (2024). Three.js.

TNO (2023). BRO 3D I3S webservices.

TUDelft3d and 3DGI (2023). 3DBAG Viewer.

Vincent Jaillot, S. S. and Gesquière, G. (2020). Delivering time-evolving 3d city models for web visualization. *International Journal of Geographical Information Science*, 34(10):2030–2052.

Xeolabs (2022). Xeogl API Documentation.

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., and Kolbe, T. H. (2018). 3DCityDB-a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data, Software and Standards*, 3(1):1–26.

Zhan, W., Chen, Y., and Chen, J. (2021). 3D Tiles-Based High-Efficiency Visualization Method for Complex BIM Models on the Web. *ISPRS International Journal of Geo-Information*, 10(7).

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.