Cartography M.Sc.

Master thesis

Testing approaches to visualize land cover/land use changes in time series with cloud-based tools

Jesse Friend

Technical University of Munich

Technische Universität Wien
Vienna University of Technology

Technische Universität Dresden

University of Twente
ITC

2021

Testing approaches to visualize land cover/land use changes in time series with cloud-based tools

submitted for the academic degree of Master of Science (M.Sc.)
conducted at the Department of Aerospace and Geodesy
Technical University of Munich

Author:	Jesse Friend
Study course:	Cartography M.Sc.
Supervisor:	Dr.-Ing. Mathias Jahnke
Reviewer:	Prof. Dr. Menno-Jan Kraak
Cooperation:	Dr. Gernot Ramminger, CTO of GAF AG

Chair of the Thesis
Assessment Board: Prof. Dr. Liqiu Meng

Date of submission: 10.10.2021

Statement of Authorship

Herewith I declare that I am the sole author of the submitted Master's thesis entitled:

"Testing approaches to visualize land cover/land use changes in time series with cloud-based tools"

I have fully referenced the ideas and work of others, whether published or unpublished. Literal or analogous citations are clearly marked as such.

Munich, 10.10.2021 Friend, Jesse

Acknowledgements

First, I would like to extend a deep gratitude towards the coordinators of the International M.Sc. Cartography program. The support, help, and insight from the entire staff has been an incredible source of inspiration for the entirety of the two years of this masters program. A big thanks, especially to Julianne Cron for helping establish this environment for all of us to succeed in. The learning atmosphere has been wonderful, and the work you have done to organize the program has been a key factor in that.

I would also like to extend a massive thanks to my supervisors Dr. -Ing Mathias Jahnke, and Dr. Gernot Ramminger at GAF AG. The support and feedback provided along each step of this thesis has helped shape it into what it is now. The insight they offered gave me the opportunity to explore this topic and gain a deeper understanding into this field.

An additional thanks to the team from GAF AG. They have been indispensable to this thesis from the support, ideas, and the willingness to participate in expert interviews and the user study. A further thanks to all my fellow colleagues. Your encouragement and opinions have been vital to completing this thesis as a whole.

A final thanks to my family and friends. The help from you all is something that is nearly impossible to put into words. A dear, warmhearted thanks to all of you.

Abstract

Web GIS applications which are high functioning, efficient, and meet the performance demand of the client are essential in modern cartographic workflows. With more and more complex spatial data being integrated into web applications, such as time related features, it is essential to harmonize the means of data presentation so that the end product is aligned with the needs of the end-user. This thesis presents microservices as a new way of building Web GIS applications which display various time-series visualizations from historical CLC data to explore future opportunities with the coming CLC+ data. Within the scope of this master's thesis, a microservice application prototype was designed and developed to streamline intuitiveness and functionality. After conducting expert interviews and method analyses, a highly usable, utilitarian prototype was deployed which presents various time-series visualizations. Further, an online survey was conducted which highlighted the usefulness and the increase in intuitivity when supplementary statistics are provided to the user in addition to slippy map integration into Web GIS applications. The prototype provides a solution which helps to understand various ways in which current web and spatial analysis methods can be combined to create visualizations that add value to existing spatial data for cartographic web workflows.

Keywords: Web GIS, microservice, land cover/land use, time series, user-centered design

Contents

List of Figures

List of Tables

List of Abbreviations

LCLU Land cover land use
EO Earth observation
CLC CORINE Land Cover
GIS Geographic information system
SOA Service oriented architecture
Ajax Asynchronous JavaScript + XML
API Application programming interface
IoT Internet of Things
GUI Graphical user interface
OGC Open Geospatial Consortium
WFS Web Feature Service
WMS Web Map Service
UCD User-centered design
UI User interface
UX User experience
AOI Area of interest
BBOX Bounding box

Chapter 1

Introduction

As web-based cartographic visualizations are becoming more ubiquitous, ongoing research offers new avenues for continued development. Modern cartography aims to improve understanding of how to harmonize new research within multiple cartographic fields, such as Web GIS design guidelines, processing of earth observation data, and integrating web-based apps into microservice architectures.

Developing methods to present land cover/land use (LCLU) time series in web-based visualizations provides value to cartographers (Yu et al., 2021). When offering earth observation (EO) data products such as raster and vector maps in web-based visualizations, it is essential to maintain readable, intuitive, and aesthetically pleasing visualizations of the data (Farkas, 2020).

The Copernicus Program is a network which collects data through a variety of sources such as EO satellites, and sensors on the Earth's surface (Büttner, 2014). The abundance of data available makes new investigations into earth observation techniques convenient (Bielecka & Jenerowicz, 2019; Kosztra & Arnold, 2019). One derived product from the Copernicus Programme is the CORINE Land Cover (CLC) dataset[1]. The CLC data is available for all of Europe and describes the land cover for the continent in vector and raster products. The products from the historic CLC inventory are used within a wide variety of projects and agencies such as environmental (Szumacher & Pabjanek, 2017), agriculture (Kandziora et al., 2013), community planning (Naranjo Gómez et al., 2020), and urban sprawl (Cerba et al., 2009).

The CLC+ project is currently in development and projected to be the next generation LCLU products for Europe, with a higher spatial and temporal accuracy (Kosztra & Arnold, 2019). The text below aims to describe the ongoing work of CLC+ to provide context for the research of this thesis.

> "The European Environment Agency (EEA) and European Commission DG Internal Market, Industry, Entrepreneurship and SMEs (DG GROW) have determined to develop and design a conceptual strategy and associated technical specifications for a new series of products within the Copernicus Land

[1] `https://land.copernicus.eu/pan-european/corine-land-cover` (Accessed 5 August 2021)

> Monitoring Service (CLMS) portfolio, which should meet the current and future requirements for European Land Cover Land Use (LC/LU) monitoring and reporting obligations. This process of development is nominally called the "2nd generation CORINE Land Cover (CLC)" and the suite of products resulting are referred to as “CLC+”.
>
> The CLC+ suite of products becomes part of the new European baseline for LC/LU monitoring for the decade to come, and the CLC+ Core database (one of the components of CLC+) allows to derive tailored products (so called instances) for support to key EU policy needs through the full policy cycle, as well as to specific needs as expressed by stakeholders in the Member States. Important first use-cases for CLC+ Core instances are around monitoring, reporting and validation to support the implementation of the Regulation on the inclusion of greenhouse gas emissions and removals from land use, land use change and forestry (LULUCF).
>
> The CLC+ Backbone component produces a detailed (...) geometric vector reference layer with basic thematic content (18 classes) and a 12-class 10m spatial resolution raster product. The products will be fully compliant with the EAGLE data model and nomenclature and can be used as input data for the CLC+ Core database. (*Copernicus Land Monitoring Service pan-European component: Technical specifications for implementation of CLC+ based on the EAGLE concept — European Environment Agency*, 2017)"

The high spatial and temporal nature of the CLC+ data presents an opportunity to analyse changes at the Earth’s surface using web based cartographic visualization techniques with more accuracy than the CLC data.

1.1 Innovation Aimed At

The aim of this thesis is to enhance the processing and visualization of changes to LCLU classes in time series through the research and prototyping of Web GIS microservice tools. Additionally, there has not been good access to statistical visualizations for the existing CLC data, so this thesis is also investigating scientifically based methods to provide more accessibility to the data. Compared to the historic CLC Data, the CLC+ Backbone will provide much more data and therefore performant visualisation techniques are needed. Further, creating these tools in user-centered workflows helps to facilitate the design and development (Abras et al., 2004).

To accomplish the goals of the thesis, a framework has been developed to visualize changes to LCLU classes in time series cloud-based tools. This is completed through the research and prototyping of microservice tools and developing a prototype application. The goal of the resulting application is to help communicate land cover information more efficiently and assist in the decision-making process of end-users who wish to statistically compare data from different temporal periods from the map. Furthermore, the prototype aims to provide intuitive, user-center designed cartographic visualizations to help users make informed choices based on historic and current data.

This research could add value to Web GIS microservices with the implementation of a web-based interface that includes interactive maps and other forms of statistical analysis. Integrating microservices into cartographic workflows have been shown to increase efficiency in project workflows by reducing the amount of time necessary to develop new tools and distributing tasks among team members without the fear of breaking a system. Additionally, existing research has presented a need for more research into GIS microservices (Mena et al., 2019).

1.2 Research Identification

1.2.1 Research Objectives

To support this work, research into different avenues of cartographic time series visualization techniques was explored. The input data is derived from historical CLC data of the years 2000, 2006, 2012, and 2018. These datasets are a high-quality source for European land cover data and provides an opportunity for developing visualizations including maps and statistical graphics (Cieślak et al., 2017).

1.2.2 Research Questions

Research Question 1

- How can the concept of digital cartographic visualizations be constructed so that it conveys information in an effective form within the context of developing projects such as CLC+?
  - What are the needs of end-users and clients, who will be using the service?
  - How can a product be designed with intuitive use as a core function?
  - How can user-input directly influence the end-result visualization?
  - What visual aspects are necessary within the final product?

Research Question 2

- How can visualizations of remote sensing derived time series be integrated into a Web GIS digital microservice infrastructure?

- How could integrating cartographic visualizations into microservices improve cartographic project workflows?
- Do web-based cartographic visualizations offer value to ongoing workflows in the remote sensing domain?

Research Question 3

- What are the advantages of creating a visualization application over implementing existing technologies, or other methods such as static maps, reports, and other graphics?

1.3 Hypothesis

Cloud-based cartographic time-series visualizations of remote sensing derived products can be an advantage to include within organizational Web GIS architectures as microservices. Cartographic projects with integrated user-feedback cycles can help improve the continual development of derived products. Additionally, there will be an advantage in the design process for a remote sensing time-series visualization prototype if user-conceptions and ideas are part of the implementation in the sense of user-centered design.

1.4 Thesis Structure

This section provides a brief introduction for the content of the thesis to help with easy navigation of the following chapters.

Chapter 2: Related Work

The related work chapter immediately follows Chapter 1: **Introduction**, and opens by discussing existing research and foundations of *Web and Microservice Architectures*. The next section continues into *Web Interfaces* and how a user interacts with web applications. Section 2.3 follows by exploring the various topics of research on the subjects of *GIS, Geodata, and Implementation into the Web*. Subsequently, Section 2.4 is an examination into work related to *Land Cover and Land Use Data*. Finally, the chapter closes with an investigation into research and techniques of *User-Centered Design*.

Chapter 3: Expert Interview

The **Expert Interview** chapter incorporates concepts from Chapter 2.5: **User-Centered Design** to lay the groundwork for developing expert interviews on topics related to creating a design strategy for a time-series visualization prototype. The results of the interviews are then placed in the context of the design structure of said prototype.

Chapter 4: Time-Series Prototype

Chapter 4: **Time-Series Prototype** discusses the application of technological methods, as well as the case study utilized to create an application prototype. Further, the results are discussed as well as the issues faced in the development of the prototype.

Chapter 5: User Study

The **User Study** chapter discusses how the user-centered design methods can be incorporated into the final stage of development as a task-based survey. The methods, and questions of the survey are also discussed in the context of user-centered design, and the results of the survey are presented.

Chapter 6: Synthesis

The **Synthesis** chapter harmonizes the findings into a discussion on the topic as a whole.

Chapter 7: Conclusion and Outlook

Finally, the thesis closes by discussing the findings, placing them in a larger context, and considering potential future work on the topic.

Chapter 2

Related Work

This chapter provides a sound background on utilizing various approaches to create frameworks to present LCLU geodata in a Web GIS application using a microservice architecture. The sections cover a variety of topics which highlight factors in developing effective frameworks for the utilization of web-based cartographic visualization tools. In order to communicate a thorough background of the topics, descriptions are provided.

Section 2.1 opens by presenting a background of web and microservice architectures in the context of web applications. Section 2.2 then elaborates on the topic of web interfaces, and how to present information to users in a browser in an easy to comprehend manner. Geographic information system (GIS) applications and how to create Web GIS applications with today's available tools are the topic of Section 2.3. Section 2.4 focuses on the topic of LCLU datasets and the applications of said data in scientific context. Finally, Section 2.5 provides definitions, background, and current research into user centered design processes which can be utilized in the field of cartography.

2.1 Web and Microservice Architectures

Every application that is available on the internet must be created through an architecture which is built specifically to store, transmit, and present data (Fielding & Taylor, 2002). The machinery involved in the process is referred to as the hardware. The operating systems, applications, and digital interfaces are referred to as the software of the architecture (Cardellini et al., 1999).

Fielding and Taylor (2002) presents on the classic web application model which involves a user interface which a browser presents to a client. The tools and services within said browser use a method of data transportation to connect to a web server stored on physical hardware to create a common web page. This data transfer protocol makes it possible to send HTML and CSS data back to the user interface which allows the browser client to be interactable (Erl, 2004). The server may be connected to a variety of other systems such as datastores, additional backend processing, and legacy systems such as is displayed in

Figure 2.1. This concept is the backbone of web architecture, including that of Web GIS applications.

Additionally, an important method for Web GIS applications is to apply the Ajax application model. Ajax is shorthand for Asynchronous JavaScript + XML. Ajax removes much of the start/stop nature of the classic web application model. Rather than an entire web page being retrieved when a client uses their browser, an Ajax engine, made by JavaScript, renders the interface and interacts with the server when a user interacts with various features (Garrett, 2007). This method removes the need to interact with a server when a web page is loaded, and allows the engine to asynchronously make requests to the server when it is necessary. This point is crucial with Web GIS applications, where extreme amounts of data must be transported from the server to the browser in efficient ways to present visualizations and further statistical analysis.

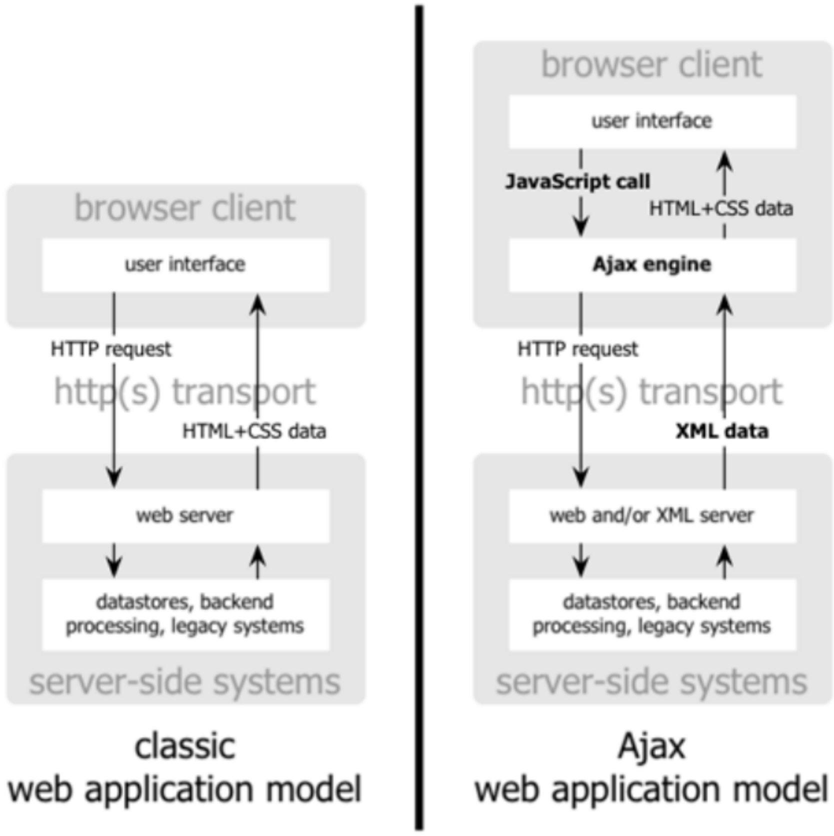

Figure 2.1: The differences between a classic model and the Ajax model (Garrett, 2007).

2.1.1 Web Servers

Each piece of information associated with presenting interactable maps on Web GIS applications must be stored somewhere. The internet is merely a connecting bridge between different computers across the world (Erl, 2004). The places where all of the data and applications available on the internet are kept are referred to as web servers (Arlitt & Williamson, 1997).

Garrett (2007) discusses the specifics of a server model. In terms of hardware, a web server can be described as any computer that stores web server software and the files for a website, such as HTML documents, CSS stylesheets, JavaScript files. After data is uploaded to a server, it is then connected to the internet and allows for data transport to other clients connected to the internet. In the context of Web GIS applications, geospatial data is also stored and retrieved through web servers with a similar model.

A web server has a software component that configures how other clients on the internet can access any files hosted on the server. Generally, the access is through an HTTP software protocol which can be contacted through the domain name of the website. Through this protocol, clients can access data stored within the server (Arlitt & Williamson, 1997).

Garrett (2007) also explains the method to retrieve the data from a server, through an HTTP request made to a back web server. An HTTP request is made by a client using the server's web address to transport the desired resources directly to a users device. The data returned from the server can then be used to present as things such as application interfaces, data which is able to be manipulated by services within applications, and other user interactions. The same method is also applied when retrieving geodata for Web GIS applications (Alesheikh et al., 2021).

Further, an important concept with Web GIS applications is using multiple web servers which are distributed across different architectures help applications load faster. This is due to the size of some geodata, and the speed in transferring data from the server to a client's browser. This dynamic load balancing allows clients to schedule requests across various servers to meet scaling and data availability needs for applications (Bryhni et al., 2000).

2.1.2 Web Applications

Fielding and Taylor (2002) describe a web application as the term used for the access of applications from a browser client over an internet network. Web GIS applications are developed with specific languages which can present interactions, request data, and provide services to users. These languages are often some combination of HTML, CSS, and JavaScript, but also include other variations of web-friendly iterations of these languages such as Angular[1] and React[2].

Web applications are sometimes built using a service oriented architecture (SOA) strategy. With the increased adoption of the internet worldwide over the course of the twenty-first century, a need arose for organizations to adopt an architecture strategy which is not dependent solely on one or a few servers to serve the necessary data for the increasingly complex web applications (Erl, 2004). From this need came the development of SOAs. Alesheikh et al. (2021) shows how this can be applied in Web GIS, and the application of this concept is elaborated more in Chapter 2.3.2: **Web GIS**

SOA helps increase the capabilities of a web application by utilizing the distribution of data among various servers (Xiao et al., 2016). As shown in Figure 2.1, an application uses HTTP requests to collect and present data to a browser client. These interactions are the

[1] `https://angular.io/` (Accessed 14 September 2021)

[2] `https://reactjs.org/` (Accessed 14 September 2021)

foundation for creating applications. By storing data in various servers, and only presenting the relevant information for the services being utilized, applications are able to function at a fast speed compared to if everything was stored in one location and requested at the same time (Bryhni et al., 2000). The use of distributed servers for Web GIS applications are discussed in the context of this thesis in Chapter 2.3.2: **Web GIS**.

Xiao et al. (2016) show how SOAs also help in regard to the ability of developing and delivering applications for continuous development processes. This is because of the need to only work on individual services within an entire application stack when the need arises to update said service. When this is the case, developers can focus their attention towards updating the application code, server, and the associated data connections so that the service can still run reliably. When not using SOA, a monolithic application must be tested based upon the dependencies of all of the aspects included in the development to ensure a successful update.

2.1.3 Application Programming Interfaces and Microservices

Application Programming Interfaces

Another aspect of web application development that is extremely beneficial to Web GIS applications are application programming interfaces, also known as APIs. Qi et al. (2020) describes an API as a software intermediary which allows multiple applications to communicate. With the increase in applications which connect to the web of things, many large organizations have provided their own development functions as an API which can enrich application functionality. This includes enterprises such as Google which provides the full functionality, and customization options of Google Maps to developers who wish to utilize the Google Maps API[3] as a feature within their application. In the context of Web GIS, APIs allow developers to share pre-built application structures[4], geospatial data analysis[5], and data visualization techniques[6] with a single line of code importing a library.

Microservices

Developing microservice architectures is an emerging trend within web application development to alleviate some of the traditional problems that arise within existing web service oriented development architectures. Lewis (2012) first presented on the idea of microservices as a scalable architecture in java, and research has continued on the topic (Mendonça et al., 2019).

Microservices are a concept built on the idea that rather than creating an entire software package that is dependent on all distributed parts to run simultaneously, many individual

[3] `https://developers.google.com/maps/get-started/` (Accessed 14 September 2021)

[4] `https://openlayers.org/` (Accessed 14 September 2021)

[5] `https://gdal.org/` (Accessed 14 September 2021)

[6] `https://geoviews.org/` (Accessed 14 September 2021)

applications, called microservices, are able to execute individually within the main application (Dragoni et al., 2017). Recently, Mena et al. (2019) has presented the possibility to develop microservice applications dedicated to working with geospatial data from a variety of different Internet of Things (IoT) devices. This thesis is building off of the current developments in microservices to explore additional applications of microservices in Web GIS such as web visualizations.

The microservice architecture is differentiated from a SOAs due to the fact that a microservice should be "able to evolve independently and choose its own architecture, technology, and platform, and can be managed, deployed and scaled independently with its own release life cycle and development methodology," (Xiao et al., 2016, p. 63).

The main goal of a microservice is to accomplish singular tasks within the overall scope of the software, as is shown in Figure 2.2. By developing a Web GIS application into multiple smaller applications, known as microservices, the entire application does not need to execute, instead different aspects of the entire application are executed when engaged with by the user (Thönes, 2015). Each microservice is completely independent from any other microservice, and only draws upon a server or database if it's individual task is called upon within the application.

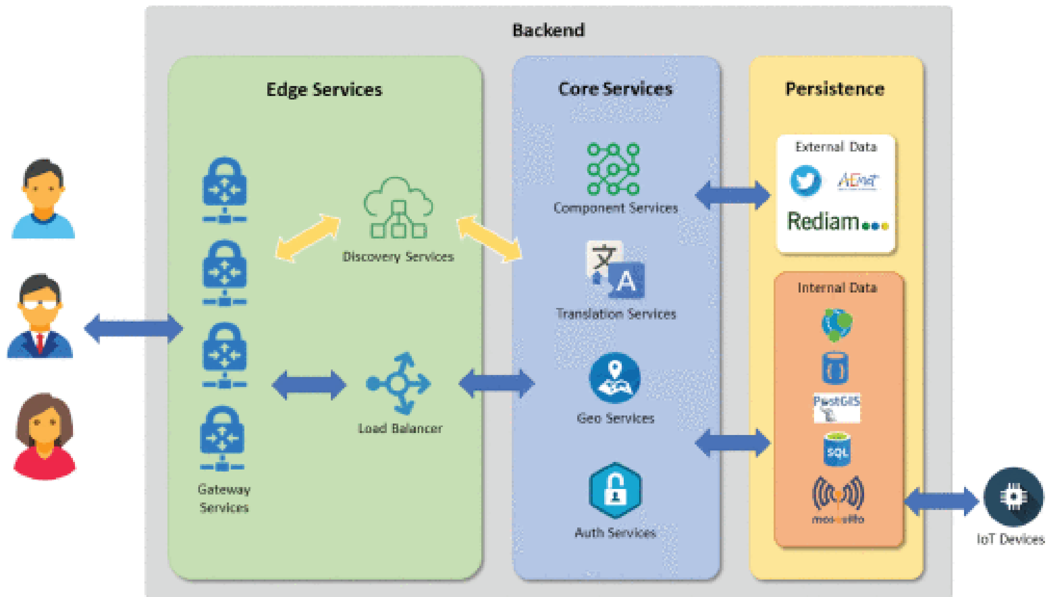

Figure 2.2: Example of a traditional microservice architecture (Mena et al., 2019).

The microservice concept is built around the idea of the scope of context for the microservice application task. Which means that each microservice does not share it's resources with other microservices within the overall scope of the application. Each individual microservice can then be scaled, modified, and updated independently within the combined application (Mendonça et al., 2019).

A few issues with microservices, are discussed in Mazzara et al. (2018), mostly related to the size of development teams and organizational systems. In regards to development teams, small groups of developers are generally dedicated to a few small microservices so development/maintenance costs can become very high. Another issue mentioned is the complexity of organizational architectures when employing microservices. The ability to orchestrate increasing amount of microservices can become a challenge of efficiency as more are added.

2.1.4 Conclusion

The techniques for Web GIS application and architecture development have been in continual development ever since the advent of the internet. Offering simple monolithic map applications is not enough to meet the growing amounts of geospatial data. Distributed servers and Ajax data transfer methods have been key to helping to grow the possibilities of Web GIS applications. Now, the current trend of changing every process within a pipeline into an independent application has been driven by the growth and adoption of APIs and microservices. The same trends can be seen in the field of geography, and implementing GIS and geodata into the internet. This niche of geospatial microservices in the larger aspect of microservices has not been studied to the extent as other microservice offerings, however, there exists a space for further conversation on the topic (Mena et al., 2019).

2.2 Web Interfaces

When developing Web GIS applications, some form of interface is necessary for users to have some level of awareness of the spatial recognition within the application, as well as some inherent recognition of the available features to the user. This section explores important concepts in the field of interfaces such as front-ends and the newly developing trend of micro-frontends, which is a further development of microservices, discussed in Chapter 2.1.3: **Application Programming Interfaces and Microservices**. These concepts are the foundation for presenting geospatial data to users whenever building web based cartographic visualizations such as time-series.

2.2.1 Web Frontends

A frontend of a Web GIS application is the graphical user interface (GUI) that the user actually sees and interacts with when visiting a web page. Jansen (1998) describes a GUI as a visual user interface on a digital template. GUIs present the opportunity for users to interact with programmed components through the use of various tools such as graphical icons, text based commands, audio indicators, and typed commands, among others. It allows user to easily navigate through complex software relations using simple visual cues and intuitive design templates such as a mouse cursor or scroll bars (Hurtienne & Blessing, 2007). Designing GUIs which are intuitive are essential for Web GIS apps, where users may not have a strong frame of reference for familiarity to tools and features.

Garrett (2007) discusses the idea of a frontend as generally being composed of HTML, CSS, and JavaScript. These work in composition with each other to produce the application interface. HTML was introduced as a markup language, allowing different text to be highlighted and linking different documents. CSS is a stylesheet language which formats how HTML or XML documents should be presented on the web page. The CSS file describes how the different components, text, media, and elements should be presented to the user. JavaScript is a programming language which can manipulate elements within the other documents to create dynamic applications and functions in Web GIS applications (Pavlenko et al., 2020).

As discussed in Section 2.1, the development of Ajax has since presented the opportunity for Web GIS applications to generate on the client-side rather than on the server-side. This allows web pages to dynamically load only when users interact with certain features. This change brought about a paradigm shift in the field of web development, as developers shifted away from single page applications and moved towards dynamically loading web pages. In terms of Web GIS applications, this presents opportunities to load massive amounts of data asynchronously, thus increasing the speed of the application.

Figure 2.3 below displays a typical frontend development pipeline and how HTML, CSS, JavaScript, and Ajax interact with each other and the backend of an application to deliver a usable application to the user.

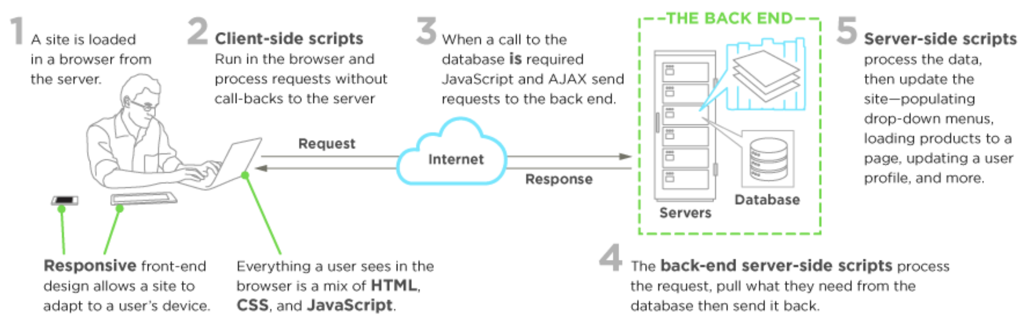

Figure 2.3: A frontend development workflow to present applications.[7]

A frontend commonly runs in a browser, on the client end. When creating a frontend, a developer must also consider how an application must be able to run effectively in different browsers, such as Chrome [8], Safari [9], and Firefox [10] among others. Other factors such as the operating system and all of the different devices that may be used to access the application also play a factor in how a frontend is presented when the application is used.

2.2.2 Micro-frontend

The deployment of Web GIS applications with monolithic single page frontends can create the same issue as monolithic applications such as was discussed in Chapter 2.1.2: **Web**

[7] `https://www.upwork.com/hiring/development/front-end-developer/` (Accessed 20 July 2021)

[8] `https://www.google.com/chrome/` (Accessed 14 September 2021)

[9] `https://www.apple.com/safari/` (Accessed 14 September 2021)

[10] `https://www.mozilla.org/firefox/new/` (Accessed 14 September 2021)

Applications. It again, like with services, becomes an issue of scalability, maintenance, and deployment of applications (Pavlenko et al., 2020). Micro-frontends have been presented as a concept to attempt to relieve some of the known issues when developing web applications. A micro-frontend utilizes the same independent application concept of a microservice. It allows for streamlined development of tasks which involve the presentation of information collected from other resources into a frontend aspect of an application.

Mena et al. (2019) discusses the concept of micro-frontends in terms of Web GIS applications. The possibility to independently deploy Web GIS micro-frontends with no dependencies to the greater organizational architecture presents opportunities to increase efficiency, application loading time, and efficiency.

Like microservices, micro-frontends do not rely on the entire application to run simultaneously, instead, the application can run independently and only utilize the server load necessary for the individual need served by the frontend (Cerny et al., 2018). This concept saves development teams time and load balancing due to the ability to have individual team members tasked with developing various tasks, without the need to continuously update and test the entire application (Cherradi et al., 2017).

M. Geers presents the following foundational needs of micro-frontends (Geers, n.d.):

- **Technology Independence:** Every micro-frontend must be able to run independently from each other, thus the technologies for each application must be chosen with considering the impact of other applications
- **Keep code isolated:** Ensure that the code from all micro-frontends are isolated from each other so that everything is self contained.
- **Establish Micro-frontend Prefixes:** Micro-frontends must have the same naming conventions within each system so that there is clear ownership and code collisions are avoided.
- **Prefer Native Browser Features:** Utilize the Browser Events for communication rather than a local API.
- **Site Resiliency:** Every feature should still be useful, regardless of whether the JavaScript failed or not.

Previous work suggests that performance efficiency, compatibility and maintainability, and security are all increased when employing micro-frontend approaches (Rantanen, 2015). These findings show the benefit of employing these same strategies in Web GIS applications. This is due to the isolating factors inherent in developing user interfaces as a component of an application, separate from all other tasks within a combined application. It has has broad, beneficial effects on quality, far outweighing the potential negative aspects such as performance issues, and lack of adoption in the larger scope of web development.

2.2.3 Conclusion

The ongoing developments within web interfaces is much like that of the changes occurring in backend development: everything is becoming more modularized. Each feature within an interface is becoming a small application in itself, or is a feature called from an API. The same can be seen with map interfaces and Web GIS applications. An important facet in these changes is to have every new application remain independent of each other. Like the research on Web GIS microservices, there is still large room for research into micro-frontends, however preliminary findings point to an improvement in development methods such as single page and monolithic applications.

2.3 GIS, Geodata, and Implementation into the Web

The capabilities of a GIS are continually expanding. Cartographers can now integrate geodata into web applications with ease through the use of various standards to complete complex geographic analysis outside of a traditional GIS (Agrawal & Gupta, 2017). These new changes present the opportunity to take advantage of another evolution in web development, microservices, discussed in Chapter 2.1.3. It is now possible to create geospatial microservices which offer users a source of rich information and features without the need to install native applications. This increase in flexibility provides cartographers with an opportunity to create complex Web GIS applications for nearly any topic of interest.

2.3.1 GIS Applications

The ideas of location, interactions between humans and nature, and the distribution of natural and man-made resources and features, are at the core of scientific inquiry in geography (Agrawal & Gupta, 2017). Humans have found, and continue to find ways to communicate these conceptions in forms that effectively convey the spatial distribution of features in various media. With the development of new technologies and hardware, cartography has been adapted to a digital format, and has continued its growth in this media (Brovelli et al., 2016; Neene & Kabemba, 2017; Nowak et al., 2020). One of the largest growth has been in the form of developing GIS applications for the use in different technologies such as web and mobile devices.

Lee and Kang (2015) show that applications that have a foundation in GIS are incredibly widespread across many forms of technology today due to the idea that nearly every human communication, natural and unnatural feature, and resource contain a geospatial component. Most current technologies and applications integrate some sort of geographic aspect into the system (Goodman et al., 2019).

The existing principles of GIS software have also been incorporated into modern web applications. Alesheikh et al. (2021) helps to define this application of GIS as Web GIS. This transformation has come in part due to the wide and freely available sources of geospatial data now on the internet (Zhou et al., 2014). This topic is essential in developing web applications which can serve as a vehicle to present cartographic visualizations, such as the LCLU time series in this thesis.

2.3.2 Web GIS

One consideration in creating Web GIS applications, is the amount of spatial data which accompanies the program. Voland and Asche (2017) has shown the issues with incorporating the increasing amounts of real-time data into Web GIS applications can cause problems with application performance. Thus the storing and accessing of data is a major component of developing highly effective Web GIS services.

GeoServer[11] is an open sourced web server software which has been designed to share geospatial data across applications on the web, and in GIS software (Youngblood, 2013). It is effective in the field of Web GIS to allow programs to access geodata in extremely efficient manners. Iacovella (2017) discusses how it was developed and can be used in accordance with the Open Geospatial Consortium (OGC) to help serve data in a variety of forms following the various standards for geospatial data, such as Web Feature Service (WFS), Web Map Service (WMS).

WFS

Peng and Zhang (2004) presents how a WFS client can access and download any stored vector data and implement it into software, maps, and perform tasks. The vector data is accessible through a variety of different formats such as KML [12], GML, and JSON. WFS provide more operations available to the client such as GetFeature, which provides flexibility in geodata analysis, especially on web maps.

When using requests such as GetFeature from a WFS, the additional information retrieved, often in the form of KML data, is available to the JavaScript within an application to perform functions. KML data is a file format based on the XML standard used to display and transmit geographic data (Peng & Zhang, 2004). Zhao et al. (2008) shows how complex actions can be performed such as area analysis and feature analysis without having to load the actual features from the GeoServer into the web application. Serving large amounts of actual features would be extremely expensive, so the ability to serve KML data from a WFS is essential in applications which request information on a very large amount of features.

WMS

A WMS is designed to serve raster spatial data. The OGC WMS standard allows clients to serve spatially referenced map data as images. The images are not maps themselves, but allow for portrayal and design styles suitable for computer screens. Boulos and Honda (2006)

[11] `http://geoserver.org/` (Accessed 5 October 2021)

[12] `https://www.ogc.org/standards/kml` (Accessed 14 September 2021)

uses another functionality of a WMS which allows for clients to specify GetFeatureInfo queries for the data at given locations. This allows users to get additional information to the images from the server without loading more images.

Geospatial Microservices

Mena et al. (2019) presents a way geospatial researchers have begun to incorporate practical applications of microservices in Web GIS applications and data. In their example, a web map which acts as a service and displays real time data from various sensors in the field, such as temperature, air pressure, wind speed, etc. that is coded independently from the rest of the web application. The benefit being that when a user chooses to engage a web map, only the relevant database and server will be called upon.

Voland and Asche (2017) is another example of developing a framework for microservices for spatio-temporal geodata collected from IoT data sources. This work was focused on creating aesthetically pleasing cartographic visualizations for car and traffic related data. The work deals with massive amounts of data in microservices, and discusses the importance of performing more research on the topic. This development architecture helps to preserve the overall load balance of the application, especially when accessing large amounts of data from a variety of servers and Geoservers for Web GIS applications.

2.3.3 Conclusion

GIS applications have come a tremendous distance since their advent. With developing means of creating maps such as Web GIS applications, high numbers of users access maps in various forms everyday on their devices. To solve issues in data transportation, solutions such as GeoServer came about to meet the evolving needs by serving geodata in a fast and effective format. This is an important facet when creating applications which are dealing with massive amounts of geodata.

These new developments can be combined with the shifts in web development architectures such as microservices, to present complex geospatial analysis with incredible ease. There exists a large space in research to study the various forms and methods for creating these new geospatial microservices.

Because the application of microservices to geospatial applications in Web GIS is relatively new, there are not as many specific examples utilizing these technologies. Nonetheless, the examples of microservice and API development in Chapter 2.1.3: **Application Programming Interfaces and Microservices**, and the work above on geospatial microservices help present a groundwork for development of time-series visualization applications which utilize data from a variety of sources and servers from across the internet for this thesis.

2.4 Land Cover and Land Use Data

An analysis into LCLU datasets is beneficial for establishing a background in relation to presenting LCLU data in Web GIS time-series visualizations. Land cover helps to describe the natural or man-made cover over the surface of the earth, while land use refers to the purpose for which land is used. LCLU data is often available very cheap or free, and at a high enough quality to create effective analysis models (Castelluccio et al., 2015; Giri, 2012). Visualizing the data in maps and other statistical methods provides an additional benefit to understanding the underlying trends in an easy to understand format (Freitas et al., 2011).

2.4.1 LCLU Datasets and Broad Applications

There are many free to use LCLU datasets available for use on the internet. Some of the popular land cover datasets include the Copernicus Land Monitoring Service[13], Global Land Cover Service[14] from the USGS, and MODIS[15] from NASA. These land cover datasets have been derived from remote sensing data, and often have both a publicly available raster and vector dataset for downloading from some sort of web inventory.

Because of the ability to use LCLU data over specific time periods, the use of the data is used for many scientific analyses of spatio-temporal changes on the Earth related to climate change. Things from urban topics like monitoring heat islands in cities (Chen et al., 2006; Jiang & Tian, 2010) and planning sustainable urbanization (Dewan & Yamaguchi, 2009; Hecht et al., 2020), to changes in the environment like monitoring change analysis in river deltas (Abd El-Kawy et al., 2011; Weng, 2002).

2.4.2 LCLU Time-Series

Using remote sensing derived LCLU products has an added benefit of a static temporal frequency. Depending on the orbit of a satellite, LCLU products can be updated at certain time intervals which makes it easier to analyse changes over time to specific areas of interest.

Rodriges Zalipynis (2012) has proven the value of using time associated geodata from publicly available remote sensing data sources to create time-series visualizations. In the article, the authors present various air quality statistics drawn from remote sensing derived products. They present their findings in a variety of graphs and plots to show how the time-series visualizations can bring value to the data which is not necessarily seen when simply viewing the individual maps and images for each time period given.

Freitas et al. (2011) created an LCLU time series analysis tool using browser utilities available at the time. Their work found that in addition to interactive maps, supplementary information in the form of statistic graphics was a benefit to the time-series visualizations

[13] `https://land.copernicus.eu/` (Accessed 5 October 2021)

[14] `https://www.usgs.gov/centers/eros` (Accessed 5 October 2021)

[15] `https://modis.gsfc.nasa.gov/data/dataprod/mod12.php` (Accessed 5 October 2021)

they created. Figure 2.4 shows an image from their research which contains the interactive map aspect, as well as additional statistical measures which the authors were able to draw further conclusions from for the location selected. Incorporating their findings in a Web GIS application helped to increase the availability of the utilities and data found within their tools to the user.

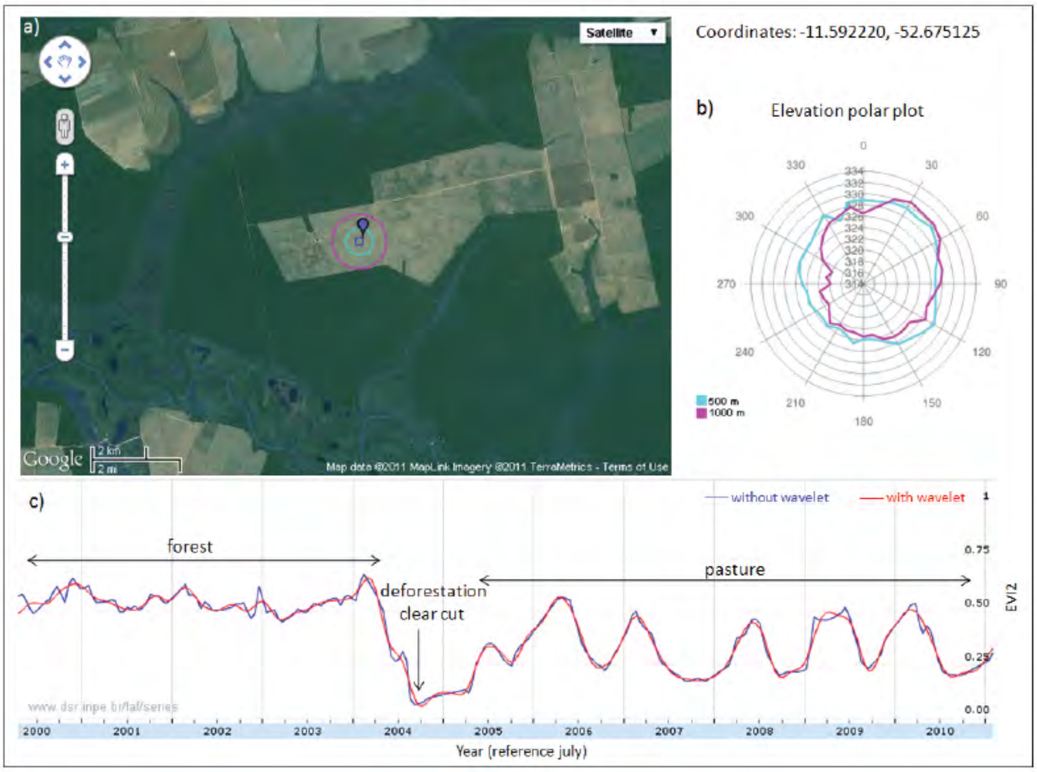

Figure 2.4: An image from the time-series developed by Freitas et al. (2011)

2.4.3 Conclusion

Understanding changes to land cover over time with time-series visualizations presents an opportunity to understand location specific shifts that could be an indication of larger systemic developments such as climate change. Further, adding additional statistical graphs and visualizations helps users understand the inherent data more than simply presenting maps and images of raster data. By incorporating these visualizations and time-series maps into Web GIS applications, rather than just allowing the LCLU data to be available for downloading, researchers have shown it helps to make the data be more available to users for analysis.

2.5 User-Centered Design

User-centered design (UCD) is a process which continually collects and integrates user feedback into the end product of a project. There are multiple stages defined along the development cycle in which a member of such projects pause to gather ideas, feedback, suggestions, etc. from the identified end-user group to continually improve project development goals. The motivation behind UCD is to create a usable end-product which successfully addresses the issue faced by the user group.

The term 'user-centered design' was first used in the 1970s, and then popularized by Don Norman in the 1980s by the work done within his research laboratory at the University of California San Diego (Rogers & Sharp, 2002). User-centered design then exploded in use after the publication of a co-authored book titled: User-Centered System Design: New Perspectives on Human-Computer Interaction by D. A. Norman and Draper (1986).

According to D. Norman (1998, p. 188), there are four main tenets of UCD that help place the user at the center of the design process.

- Make it easy to determine what actions are possible at any moment.
- Make things visible, including the conceptual model of the system, the alternative actions, and the results of actions.
- Make it easy to evaluate the current state of the system.
- Follow natural mappings between intentions and the required actions; between actions and the resulting effect; and between the information that is visible and the interpretation of the system state

The work by Norman and the researchers that followed emphasized the importance of incorporating users to totally understand their needs and wants in the end-product. After the publication of their ideas, users became a central part of the development process. Their involvement lead to more effective, efficient and safer products and contributed to the acceptance and success of products (Rogers & Sharp, 2002).

The concept of User-centered design is based around two distinct methods, investigative and generative. Investigative looks into what users currently know about the desired end-product through the use of surveys and interviews (Soloway et al., 1996). Generative methods are done with the goal of helping to bring new ideas to the table. Things such as brainstorming are employed to accomplish this task.

Once the end-users become identified and an examination into the needs has been concluded, the designers behind a project can begin developing solutions which can be assessed later by the users.

2.5.1 User-Centered Design Process

The user-centered design process normally follows four iterative stages. Figure 2.5 displays how each stage is evaluated in the fourth stage to continually redesign and improve the product (Rogers & Sharp, 2002).

- **Understand the context of use:** Identify the individuals who are ultimately going to use the product, the context under which they will use it, and the environments of the end product.
- **Specify user requirements:** Determine all product requirements and goals so that the final design meets all user-needs.
- **Design solutions:** Develop a design for a final product based on findings from the previous stages.
- **Evaluate against requirements:** Evaluation through usability testing with users of each previous stage of design.

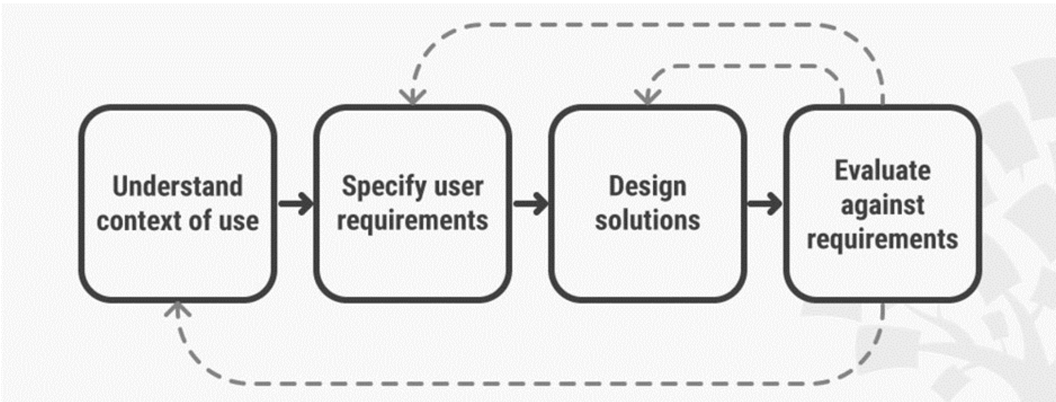

Figure 2.5: A traditional user-centered design process pipeline.[16]

The end goal of UCD is to have all tasks within a product, such as an application for example, match a user requirement. This helps to develop products where the user feels comfortable, and they are able to logically navigate through the different features with ease.

2.5.2 User Interface/User Experience

User interface (UI) is a term used to refer to the the aesthetic experience of a digital product, while user experience (UX) refers to the product design and the full experience of a user for both a digital and physical product (Galitz, 2007). In this thesis, UI/UX will only be discussed in the context of digital products.

UI design is strictly considering the interaction, feel, and style of a product. The topic is related to making a user interface feel intuitive (Galitz, 2007). When creating a UI, a designer must consider the different icons, colors, fonts, imagery, and spatial distribution of screen elements. The end goal is to create a product which visually guides users through

[16] `https://www.interaction-design.org/literature/topics/ux-design?page=8` (Accessed 20 July 2021)

a products functionality. Another aspect of UI is to have a unified design model so a user feels comfortable across the various pages, and features of a product(Rogers & Sharp, 2002).

The study of UX ties directly to the work done by D. Norman et al. (1995, p. 155) in the 90's on user experience design. In this work the topic is explained as such:

> "User experience encompasses all aspects of the end-user's interaction with the company, its services, and its products."

UX has a focus strictly on the experience a user has from the moment they engage in a product until they stop. A goal of UX is to design structures for users to navigate through the product, without encountering any pain points (Blackler, 2018). Although the topics of UI and UX can be difficult to distinguish at times, a key difference between the two is that UI is the sensory experience of the app, while UX is the functional experience of the app.

2.5.3 Intuitivity

The demand for user intuitivity when operating a product is extremely high (Naumann et al., 2007). Intuitive use can be referred to as an aspect of human-machine systems. It is the subconscious interaction process between a user and a digital system which is guided by a user's intuition. Intuitivity can be understood through the *prior knowledge* and *subconscious application* (Naumann et al., 2007).

Many of the concepts used to define intuitivity can be vague, product specific, and hard to define in a concrete manner (Naumann et al., 2007). Due to this, the method for understand the inherent intuitivity of an application, researchers must rely on UCD process to understand a user's sentiments towards a product (Blackler, 2018). Understanding the intuitive interaction of an application is important because it directly leads to improvements in the user interface. When a product developer incorporates the feedback of the product users back into the application, it becomes inherently more intuitive to the user base (Blackler, 2018).

2.5.4 UCD in Cartography

UCD can also be applied to maps and other cartographic visualizations. R. E. Roth et al. (2017) present the topic of *interface success* to describe whether or not an interactive web map works as they should. With the general public being exposed to more and more interactive maps and visualizations, they posit that the collective map-savviness is increasing, and thus the need for more integrated UCD techniques in cartographic development workflows is necessary. R. Roth et al. (2015) outline *Three U's of Interface Success*, discussing how to create design strategies for Web GIS cartographic interfaces.

- **Usability:** How easy is it to use an interface to complete the objectives at hand?
- **Utility:** How useful is the interface to complete the objectives at hand?
- **User base:** Who is the target group that will be using the interface?

Understanding these three defining factors, along various stages of development help to deliver a successful interactive web map. R. Roth et al. (2015) also present three interface evaluation categories: expert based methods, theory based methods and user based methods. The authors also recommend to combine evaluation methods to understand different aspects of design expectations throughout the development of interactive maps.

2.5.5 Conclusion

When developing Web GIS applications, it is important to incorporate UCD into the design process so that users have a high level of usability with the application. Especially with geospatial applications, UI and UX design helps to provide an interface that allows users to know where they are in the application, and what they are doing. Further, designing applications with intuitivity as a core facet of a product ensures that a user base is content when using the application. When all these topics all work in conjunction with each other to create a design model, it helps deliver a highly usable product.

2.6 Summary

With the related work established, the primary tasks of this work are laid out as such:

- Develop a workflow to access, read and analyze the relevant input data obtained through the CLC historical inventory
- Deploy a Web GIS microservice application which theoretically could be implemented into a larger web cloud architecture without dependency issues.
- Provide a frontend which allows accessing, reading, analyzing, and visualizing the comparison of all inputs in a cartographic and graphically pleasing output
- Perform a UCD based user study to understand the user sentiment towards the end-product

In order to complete all of the above tasks with UCD as a core component, the next step after the related work review was to complete a requirements analysis. To do this, multiple expert interviews were completed to gain insight into the necessary components of a Web GIS time-series visualization. The methodology, and applied case study for the work is outlined in the next chapter, Chapter 3: **Expert Interviews**.

Chapter 3

Expert Interviews

3.1 Methodology

The goal of this thesis was to develop a prototype application which can be deployed as a microservice in a larger cloud-based infrastructure. To design the web-architectures behind the final prototype, research had to be completed to inform the development strategy with the necessary requirements before the work could be started. This was done by first doing a related work review, see Chapter 2, to understand the current trends and strategies to develop an application of this type. After the related work, a requirements analysis in the form of expert interviews were then conducted from individuals who are experts in the field of GIS to gain insight into the **user base** of a potential application R. Roth et al. (2015). Information gathered from both the related work review and expert interviews were then assembled to understand the context of development for this prototype.

User-Centered Design Integration

In Chapter 2.5.4: **UCD in Cartography**, the work by R. Roth et al. (2015), on evaluating interactive web map interfaces is introduced. On pages 273-276 the authors discuss the three evaluation categories; expert-based, theory-based, and user-based. The three evaluation categories are then broken down into 13 total specific methods to evaluate the interface. One of these thirteen methods is interviews.

On R. Roth et al. (2015, p. 275) the interview method is specified as good for the three points below:

- the user needs and expectations are poorly known
- the software supports a small number of highly specialized users or a small set of user profiles
- transitioning an interactive map to a new application domain

At the onset of the development of the prototype, the end-users and their needs were very poorly known so the first point helped lead to choosing this method to include in the methodology. The second point could not be assessed due to the unknown user-base, so this will need to be assessed further in the requirements analysis. As established in Chapter 2.3: **GIS, Geodata, and Implementation into the Web**, there have been some examples of geospatial microservice examples, however, there is still space for more work in this area. Because this is true, the third point also holds true. For these reasons, interviews are a method which applies to a requirements analysis for the time-series visualization prototype.

There are multiple interview types which adhere to map interface evaluation. These include structured interviews, semi-structured interviews, unstructured interviews, and contextual inquiry. To allow the interviews to drift a bit from point to point, semi-structured interviews were chosen to employ in the methodology of this thesis.

3.2 Case Study

To fully complete a requirement analysis to design the prototype with the needs of the end-users of the customer's (GAF) CLC+ project in mind, four expert interviews were conducted with professionals in the field of cartography. The goal was to gather viewpoints from various experts in different specialties in the field of developing Web GIS applications. The interviews were conducted with: a GIS front-end developer, a GIS backend software engineer, the CLC+ raster product project manager, and an expert and manager in GIS emergency management dashboards.

Each interview was conducted one-on-one over video chat. The questions were focused on discussing the needs of the end-user, technical aspects to include, and potential data sources. With the expert's consent, each interview was recorded and the findings from the interview were gathered and studied to be implemented in the final end product.

The interviews were conducted with a semi-structured, and generative design to bring new ideas to the table, as mentioned in Chapter 2.5 **User-Centered Design**. A list of questions was prepared to give a general outline of the conversation in advance, however, the topics were set up to allow the participants to diverge from the main topic if they had more ideas about a certain topic and wanted to get into the specifics of it.

3.3 Results

Some of the main findings of the expert interviews are summarized below and placed in the context of the research questions. The questions were derived to address the research questions outlined in Chapter 1.2.2, **Research Questions**. The questions focused on understanding the various experts ideas on:

- Get to know the potential user base of such an application and what their needs are for a visualization

- Web development strategies such as microservices, languages to employ, open sourced libraries, etc.
- Understanding the project goals and end products of the CLC+ project
- What the existing data from the CLC inventory looks like
- Web visualization strategies for maps and other graphics

The full transcriptions for each of the four interviews can be found in *Appendices A-D*.

Research Question 1

- How can the concept of digital cartographic visualizations be constructed so that it conveys information in an effective form within the context of developing projects such as CLC+?
  - What are the needs of end-users and clients, who will be using the service?
  - How can a product be designed with intuitive use as a core function?
  - How can user-input directly influence the end-result visualization?
  - What visual aspects are necessary within the final product?

Expert Feedback

"The typical user of these products will be an expert(see Appendix C)."

"The main need for the end user is to deliver raster GeoTIFF products for experts and public (see Appendix B)."

"The information content that you get is probably the most important concept for the end user. Working stable and fluid is probably more important than having the nicest possible design if you have to make a compromise(see Appendix B)."

"It is more important to get the best data instead of visualizations(see Appendix D)."

Research Question 2

- How can visualizations of remote sensing derived time series be integrated into a Web GIS digital microservice infrastructure?
  - How could integrating cartographic visualizations into microservices improve cartographic project workflows?

- Do web-based cartographic visualizations offer value to ongoing workflows in the remote sensing domain?

Expert Feedback

"Creating visualizations as microservices, that helps. The point here is that you have a lot of data and you can scale microservices quite easily. So if you want to do some visualization services with big data these microservices can be scaled and this could be done much easier. Yeah all in all it can make the code more reusable so it runs for other projects (see Appendix A)."

"Another benefit is that you can have the code which is doing the visualizations on the fly in the microservice. Everything is much lighter (see Appendix A)."

Research Question 3

- What are the advantages of creating a visualization application over implementing existing technologies, or other methods such as static maps, reports, and other graphics?

Expert Feedback

"So the projects that we've had lately are ones where the user has wanted to be integrated. So that the user can also be active, and getting the full picture, and getting the full capabilities where they can choose the time frame for example. The users also get more informed so they can bring in their own ideas (see Appendix A)."

"It's getting more democratic and more accessible with open source tools (see Appendix B)."

"Some projects in the field are more desirable to have a finished product where their user feels in control of the project and the end product is more 'theirs' (see Appendix C)."

3.4 Discussion

3.4.1 Design Strategy

Research Question 1 can be summarily broken down to the question, "What is needed in this prototype?", both in terms of data and functionality. The objective in answering the question was to obtain the needs of the end-users who would be operating on a visualization associated with the CLC+ project. One of the core ideas taken from the interviews is that any visualization for a project such as CLC+ would need to prioritize the **usability** and **utility** of the application over the appearance. This is supported by multiple experts that stated the potential end users would mostly be experts in the field, so the appearance is not as important as the capabilities. Feedback such as this was integral to drive the development direction towards ensuring additional functionality of data analysis when users engaged with the application.

Research Question 2 is aimed at gathering the technical aspects needed to develop the prototype, and the possibilities of integrating this product into larger cartographic workflows. The experts knowledgeable in microservices conveyed how lightweight the application would be when built merely as a microservice. A significant hurdle when developing GIS applications is the size of geospatial data. Multiple experts also indicated the benefit of developing cartographic visualizations as microservices is related to the ability to handle large amounts of data simultaneously from various servers around the internet. The feedback from the experts also directly influenced the choices made to create the development stack of the application.

The goal of Research Question 3 was to gain an understanding as to why developing tools such as this prototype could be helpful in the first place. The consensus from the experts is that tools such as this offer value to users for multiple reasons. One reason being that the unified presentation of large amounts of geospatial data in an interactable tool offers the user an opportunity to explore different datasets with relative ease. This is an important aspect when dealing with time-related land cover data, and provides an avenue to compare similarities and differences of locations without having to invest large amounts of effort into the process. Another point multiple experts expresses was that users feel more empowered over their actions when they can explore data in visualizations such as this. This interactability helps users feel a sense of ownership of the results when they can actively define the parameters from which the statistics are drawn from the data source.

3.5 Summary

Utilizing UCD to complete a requirements analysis by means of expert interviews, in concert with a review of the related work, has helped provide a basis for understanding of the technical needs, user needs, and end-user group for a potential time-series visualization built using microservice technologies. The knowledge gained from the expert interviews bolstered the design strategy for the application prototype in terms of web architectures, as well as UI/UX strategies. The experts helped provide a foundation of knowledge which directly contributed to answers to the research questions of the thesis.

Chapter 4

Time-Series Prototype

4.1 Methodology

The methodology of developing time-series visualization tools in cloud-based architectures, such as microservices, serves to provide a foundation for future research into similar topics. The desired outcome is to design a Web GIS architecture which can be replicated by readers of this thesis. Thus the design strategy is explained below in broad topics, and the specifics of the prototype design are outlined in Chapter 4.2 **Case Study**.

Web GIS Microservice

The basic foundation of Web GIS applications is the same as what is outlined in Chapter 2.1 **Web and Microservice Architectures** for normal web applications. It is necessary to design a workflow which stores, transmits, and presents data (Fielding & Taylor, 2002). The first step in doing so is to create a web server which can host the files which compose an application. Most servers have a hardware component and some operating system, generally Linux[1], and a software component such as Apache[2] which allows for HTTP data transfer requests as outlined in Chapter 2.1.1 **Web Servers**.

To actually create the application a "technology stack", or set of computer languages, frameworks, and tools are chosen for the development process. In Chapter 2.2.1 **Web Frontends**, Garrett (2007) presents how HTML files are the visual interaction framework for users when they enter a webpage, and that the other files in the server interact with the containers within the HTML files to create the interface that defines the UI/UX of a web application. CSS/SCSS files within the web server manipulate the *div* elements in the HTML file to create the design of an application such as colors, fonts, and size of elements

[1] `https://www.linux.org/` (Accessed 3 October 2021)

[2] `https://www.apache.org/` (Accessed 3 October 2021)

on the web interface. In Chapter 2.1.2 **Web Applications**, common web-development languages are introduced such as JavaScript, Angular, and React which actually integrate functionality into web applications. JavaScript is often used within Web GIS applications because of the wide-range adoption within cartographic web APIs. To assist in creating web applications, Parcel[3] can be used to automatically package a project to uploaded to the web server used to host the application.

There are various JavaScript libraries which help to create slippy maps easily through public APIs. Some of the popular ones include Leaflet [4], OpenLayers [5], and Mapbox [6]. Leaflet and OpenLayers are open-sourced libraries which allow for easy integration of many forms of geodata.

The geodata used in the application prototype come from a variety of web servers across the internet. Once it is requested in the JavaScript using Ajax calls, further manipulation with JavaScript allows for additional methods of analysis and visualizations such as the time-series tools in the work by Freitas et al. (2011) and Rodriges Zalipynis (2012). Since the web map libraries are designed and accessible as an API, it increases the ease in isolating geospatial microservices from the overall stack. Once the API is added to an application, further functionality can be developed using JavaScript to enhance the UI/UX to increase the interface success among users (R. E. Roth et al., 2017).

Utilizing APIs is essential to maintaining a microservice architecture, as well as micro-frontends. As Xiao et al. (2016) explain, the independence of an microservice from the rest of an organization's technological stack allows each microservice to be updated, maintained, and scaled with a higher level of ease than by creating monolithic applications or services as a part of SOAs. Throughout the case study of this prototype, many open-sourced APIs are included to help increase the abilities of the application as a whole, while reducing the time needed to develop every functionality independently.

4.2 Case Study

4.2.1 Data Description

As mentioned in Chapter 1, **Introduction**, the primary data source for this research is from the CLC inventory.

Both historical vector and raster products from the CLC inventory are utilized within the context of this research to analyze Web GIS microservice capabilities in creating time series tools. All of the CLC data is available from the official Copernicus Land Monitoring

[3] `https://parceljs.org/` (Accessed 14 September 2021)

[4] `https://leafletjs.com/` (Accessed 14 September 2021)

[5] `https://openlayers.org/` (Accessed 14 September 2021)

[6] `https://www.mapbox.com/` (Accessed 14 September 2021)

Website [7].

The CLC raster products are freely available from the Copernicus Land Monitoring website via a WMS protocol. The layers from 2000, 2006, and 2012 2018 were chosen to normalize the six year temporal period within the visualizations of maps and statistics. The dataset from 1990 was omitted due to the ten year period between it's release and the subsequent CLC product in 2000. The layers were then all integrated in to the web application as tiled image layers to increase the efficiency of loading time for the entire application.

The CLC vector products are not publicly available via a WFS, so further steps needed to be taken to incorporate the functionality of the vector data into the application.

The feature label layer, as well as the transportation layer are accessed direct from the ArcGIS Rest Services Directory[8]. Both of the layers are added to the OpenLayers map by creating a TileLayer, and adding it to the map within JavaScript.

Other open source OpenLayers extensions used include the sidebar-v2[9], and the ol-layerswitcher[10].

4.2.2 Study Area

The CLC dataset is pan-European, coordinated by the EEA, and uses the uses the EEA38+UK countries[11]. Figure 4.1 displays a highlight box over the study area that is the foundation for the prototype. CLC data includes forty-four land cover classes (*Pan-European — Copernicus Land Monitoring Service*, n.d.).

4.2.3 Technical Aspects

The web application which presents the time series visualization prototype was written with the JavaScript runtime engine Node.js [12], CSS/SCSS, and HTML documents. OpenLayers v6.6.1 was chosen as the JavaScript library to present the maps in the prototype due to the impressive amount of functionalities in the API. Since the library is designed and accessible as an API, it increases the ease in isolating geospatial microservice from an overall organizational stack.

Every WMS layer used was added to the map via a XYZ OpenLayers protocol. XYZ requests serves tile data on continually updating URLs which include the coordinates of the user's location on an interactable map. Serving WMS data as tile services separate an entire georeferenced map image into a specified number of smaller tiles. When adding georeferenced map tiles to an application, the speed of delivery is dramatically increased

[7] `https://land.copernicus.eu/pan-european/corine-land-cover` (Accessed 14 September 2021)

[8] `https://server.arcgisonline.com/arcgis/rest/services` (Accessed 20 September 2021)

[9] `https://github.com/Turbo87/sidebar-v2` (Accessed 10 August 2021)

[10] `https://github.com/walkermatt/ol-layerswitcher` (Accessed 10 August 2021)

[11] `https://www.eea.europa.eu/countries-and-regions` (Accessed 27 September 2021)

[12] `https://nodejs.org/en/` (Accessed 10 August 2021)

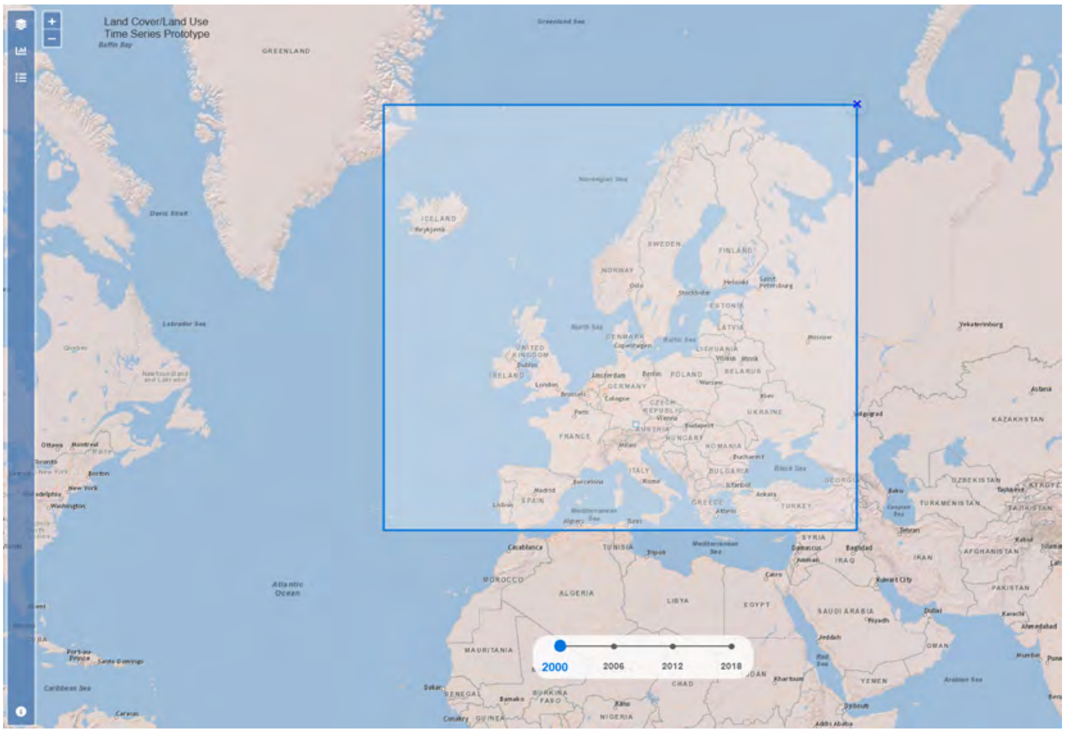

Figure 4.1: A general view of the coverage of the CLC dataset.

because the application can dynamically access only the tiles which are relevant to the current context within the application.

When a user clicks at any location on the map, a pop-up opens over the coordinates of the click. The pop-up displays the land cover class at those coordinates for each of the four years colored square in a time series. Each square in the pop-up is colored the same as the land cover class which it represents. This is completed by using the GetFeatureInfo operation of the different WMS layers at each click location and presents the data via an HTML pop-up.

A GeoServer v2.1.8 was created to serve the CLC vector layers via a WFS due to the increased flexibility vector data analysis offers. Serving the vector data via a WFS also allows users to explore more of the data. One such example is a feature in the application which executes when a user holds the control/command key down and drags a bounding box across the map. When the key is released, the prototype provides a time series in the form of yearly statistics in the side panel. This data was included in part due to the feedback from multiple experts mentioning the increased abilities of a WFS, such as the quote below from the CLC+ raster product manager.

> "We have a vector product, which is made of polygons. And with this there are 18 classes of attributes also attached to that like building height, composition of the land cover, then other attributes from other Copernicus products like phenology or tree cover density. I think that's much richer in terms of information content and also then probably something that is, let say, more options to play a bit. (see Appendix B)"

The statistical data are retrieved from the project-built GeoServer using a GetFeature url to the WFS with jQuery v3.6.0[13] Ajax calls. Each call gathers all of the features that fall within the user defined bounding box. The actual vector layers are never added to the map. The features of the layers are only queried within the area of interest (AOI) from the server when the user defines a box on the map. This speeds up the application loading time. When adding each WFS layer directly to the map, the performance of the application is nearly unusable so an effort to avoid adding unnecessary data was made. The Ajax data response provides KML data of the features, where it is parsed to be used for the statistical analysis in the visualizations within the side bar of the map. When the side bar is opened, users can view the change over time in terms of land cover percentages for the feature coverage across each year. Visualizations appear to a user when an AOI is created in the form of both Chart.js v3.5.1[14] pie charts, and the actual numeral statistics visualized in a table with Grid.js v5.0.1[15].

To create a time series slider for the WMS layers, the following steps were taken. First, a *div* element was created inside the map div element in the HTML file. In this *div*, the current year of the slider was specified. Then, within the JavaScript file a jQuery function is configured to recognize when the input year is changed by moving the slider. On this change, the opacity parameter of the desired WMS layer corresponding to the time slider year is changed to "0", while the opacity value for the other years are set to "1" instantaneously.

The application was then packaged using Parcel for a public URL. The resulting files were added to a web server to host the application on a domain accessible to others on the internet.

4.3 Results

Figure 4.2 displays general view of the coverage of the CLC data across all of Europe when seen in the application. In this figure additional features can be seen, such as the sidebar on the left which houses the layer switcher, statistics panel, legend, and about panels. Other buttons include zoom in and out, fullscreen, and the draggable time slider on the bottom of the screen.

A zoomed in view of Munich, Germany can be seen in Figure 4.3. In this figure the draggable time slider on the bottom of the interface allows users to move the blue dot across the four years of data where A=2000, B=2006, C=2012, and D=2018 in the image.

Another feature implemented is the ability to perform a mouse-click at any given point in the CLC coverage to ascertain the land cover class for the associated coordinates for each of the four years. As seen in figure 4.4, a small pop-up appears which displays the coordinates of the mouse click and the land cover classes for the four years. When the mouse is hovered over the small colored squares for each year, a hover text appears which displays text with the class name. The squares have the same color value as the land cover class the user clicks on.

[13]`https://jquery.com/` (Accessed 10 August 2021)

[14]`https://www.chartjs.org/` (Accessed 7 October 2021)

[15]`https://gridjs.io/` (Accessed 7 October 2021)

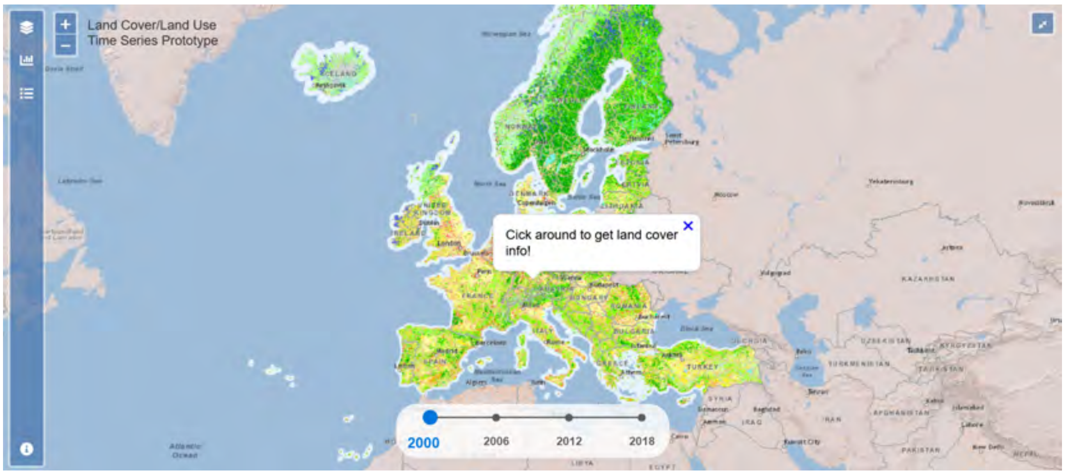

Figure 4.2: The overall view of the CLC coverage across Europe in the application.

Figure 4.5 presents the layer selector panel open. The layer selector panel was included to allow users the ability to remove the transportation network layer, as well as the borders and labels layer to see just the CLC layers for any location without any other distraction. There is also the ability to change the base map in this panel so that when the map view extends beyond the coverage area, users can decide the aesthetic of the base map.

The third button from the top on the sidebar opens the legend panel, which can be seen in Figure 4.6. This panel provides users a full list of the classes and the associated colors of the land cover types in the CLC dataset.

The statistics panel is open in Figure 4.7 and 4.8, and presents users the opportunity to create an AOI by which they can generate yearly land cover statistics with. This is done by holding shift and dragging and dropping a rectangle over the area they would like to see. The data can be presented as a pie chart, as in Figure 4.7, or as a queryable table, as in Figure 4.8. The data in the pie chart is colored the same color values as the associated land cover classes in the AOI.

The application also has an about panel on the sidebar, shown in Figure 4.9, which provides a brief description of the project, as well as providing the source for the data used within this service.

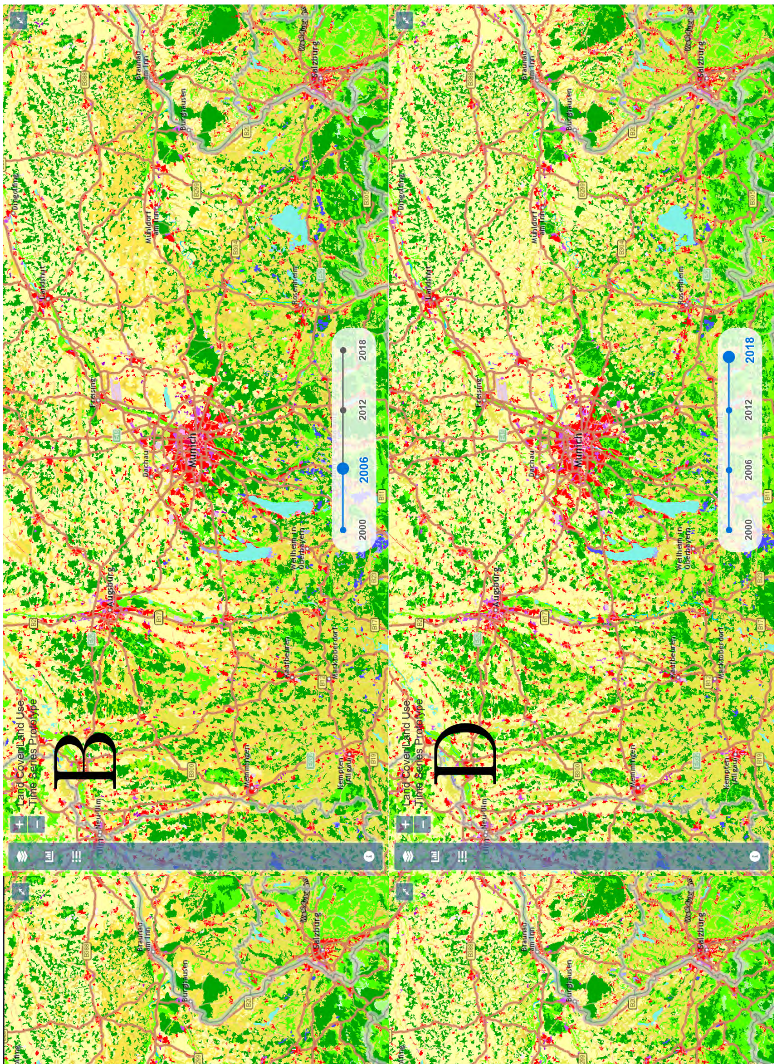

Figure 4.3: Four images which show the effect moving the time slider has over Munich, where A=2000, B=2006, C=2012, and D=2018.

Figure 4.4: Results of clicking to see coordinate class changes over time.

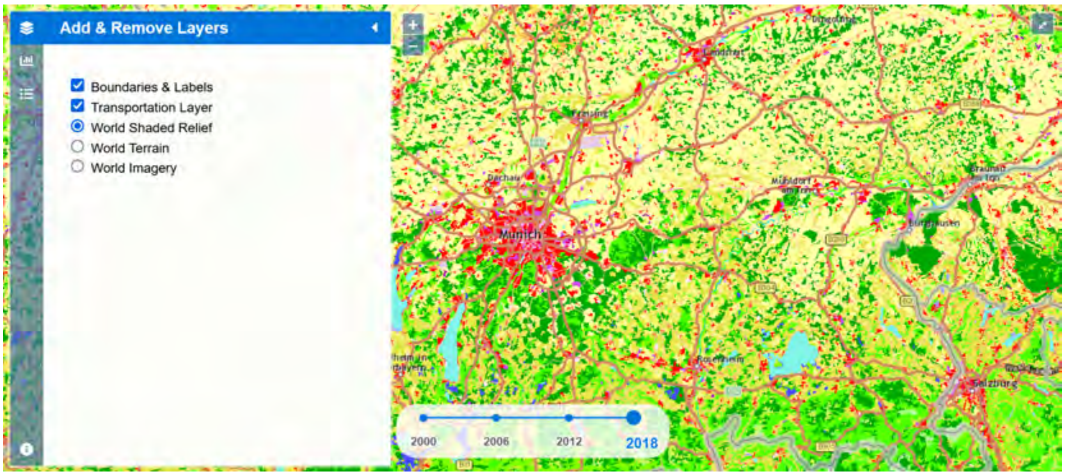

Figure 4.5: The layer selector panel open on the application.

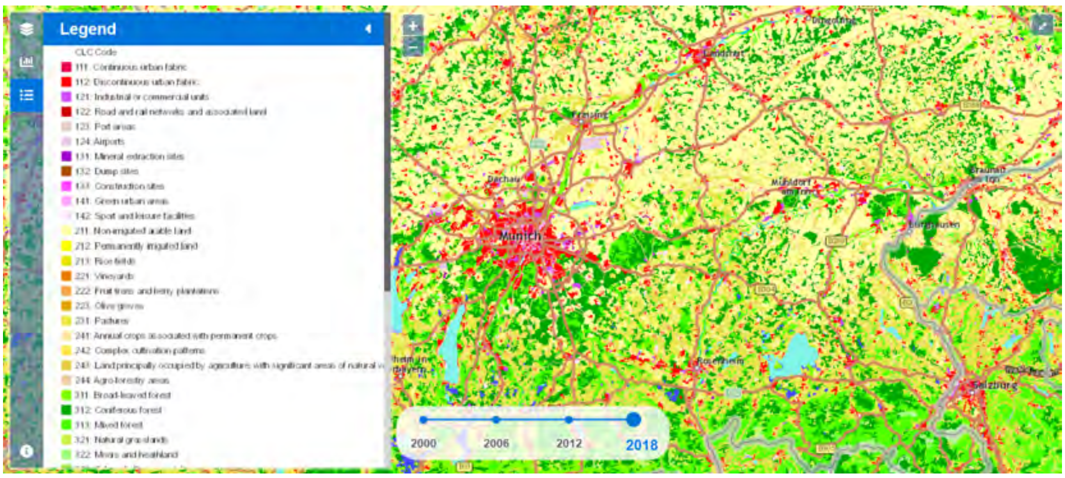

Figure 4.6: The legend panel open on the application.

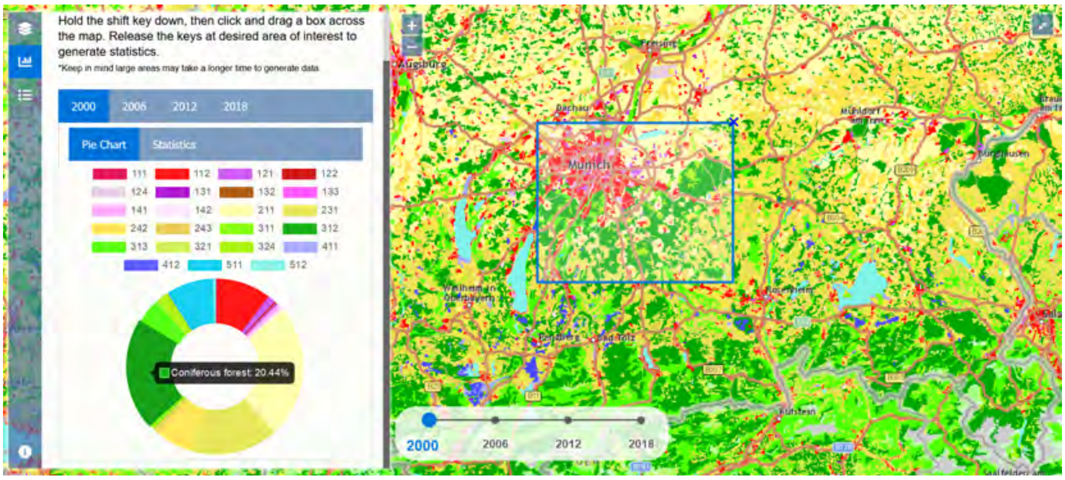

Figure 4.7: The pie chart aspect of the statistic tool in action over an area around Munich.

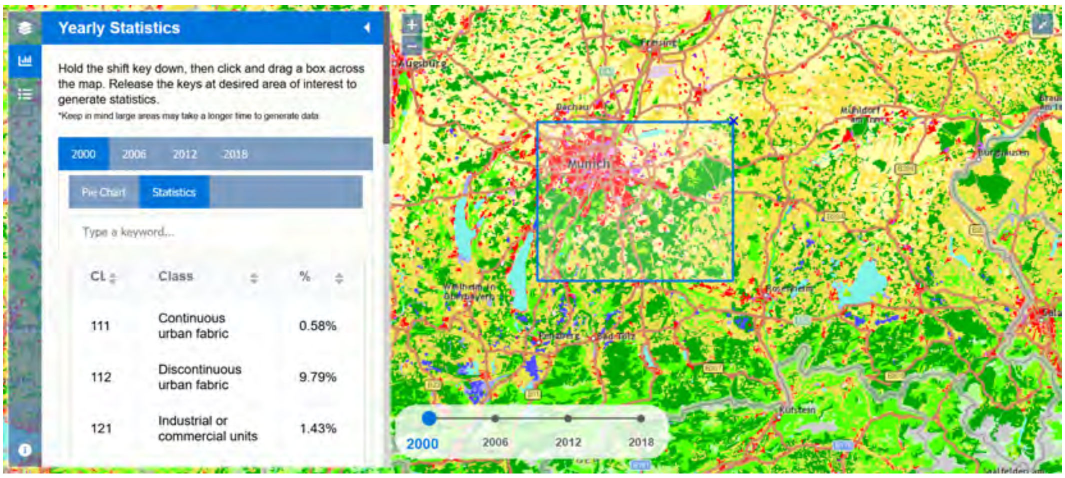

Figure 4.8: The queryable table tab open within the yearly staistics.

Figure 4.9: The about panel open in the application.

4.4 Discussion

4.4.1 Technology

The technology stack for the microservice prototype application is composed of HTML, CSS/SCSS, and JavaScript. These were chosen to allow for easy implementation of a web map API into the application. JavaScript also allows for wide-ranging implementation of APIs, and functionality development due to it's object oriented nature.

The information collected in the interviews helped to inform the decisions made to choose the development stack. Such as when one expert said this, "for front ends there is Node.js which is widely used with the OpenLayers library. It's is a good tool to actually visualize the maps you want to create," (see Appendix A). Upon further research, OpenLayers was chosen to employ as the slippy map JavaScript library for the application due to feedback from the expert interviews in the requirements analysis. This choice was led in part by other expert feedback like:

> "here on OpenLayers.org you can get an impression of what they are doing. So, it is basically a library to create virtual interactable maps like these. I always use the API, which has a super strong library, so you can search for so many things you might want use in your application," (see Appendix C).

The GeoServer was built to serve the WFS vector data to this application because it is the industry leading open-sourced server to transmit, share, and edit geospatial data. Chart.js and Grid.js were used to implement as the tools to present the statistic visualizations because of the simplicity of integration with the KML feature data response from the GeoServer.

4.4.2 Issues

The largest issue that was faced in development was in the creation of a WFS GeoServer for the vector CLC data. As mentioned in Chapter 4.2.1 **Data Description**, the CLC vector products are not publicly available via a WFS, so a GeoServer was prepared for the application to access the vector layers over the internet via WFS standards. The GeoPackage[16] available for download from the official CLC inventory are not compatible with GeoServer, so the GeoPackage for each year included in the case study were first added to a geodatabase. Once the geodatabase was created, the layers were uploaded into GeoServer, and a WFS layer was then configured for each year's vector layer. The vector layers, and all features within were then available for querying from within the prototype via web WFS standards.

Once the WFS were created, further data quality problems arose. The first, being that the OGC Intersect Filter Operator[17] built directly into the GeoServer for the thesis was not functioning when requesting data from any URL attempted. This presented an issue in

[16] `https://www.geopackage.org/` (Accessed 8 October 2021)

[17] `https://docs.geoserver.org/master/en/user/filter/filter_reference.HTML` (Accesed 8 October 2021)

querying data into the application. The work around for the application was to use the Bounding Box (BBOX) Filter Operator[18], however, this also created an issue. If features fell anywhere inside of the BBOX, and continued outside of it, they were still included in the overall feature area calculation for the statistics visualization. See Figure 4.10 to see an example of this. The light yellow feature crosses the AOI (A) in the top right of the box, however, the feature's coverage (B), extends far beyond the area of the AOI. With the BBOX filter, the area of this entire feature would be included in the statistics calculation. To circumvent this issue, a boolean statement was created which only allows features with an area less than 1.5 times greater than the area of the AOI to be included in the calculations. As the purpose of this thesis was to test the overall visualization of LCLU data in a microservice, and not the filter operators, this was acceptable for the completion of the user-study, outlined in Chapter 5: **User Study**.

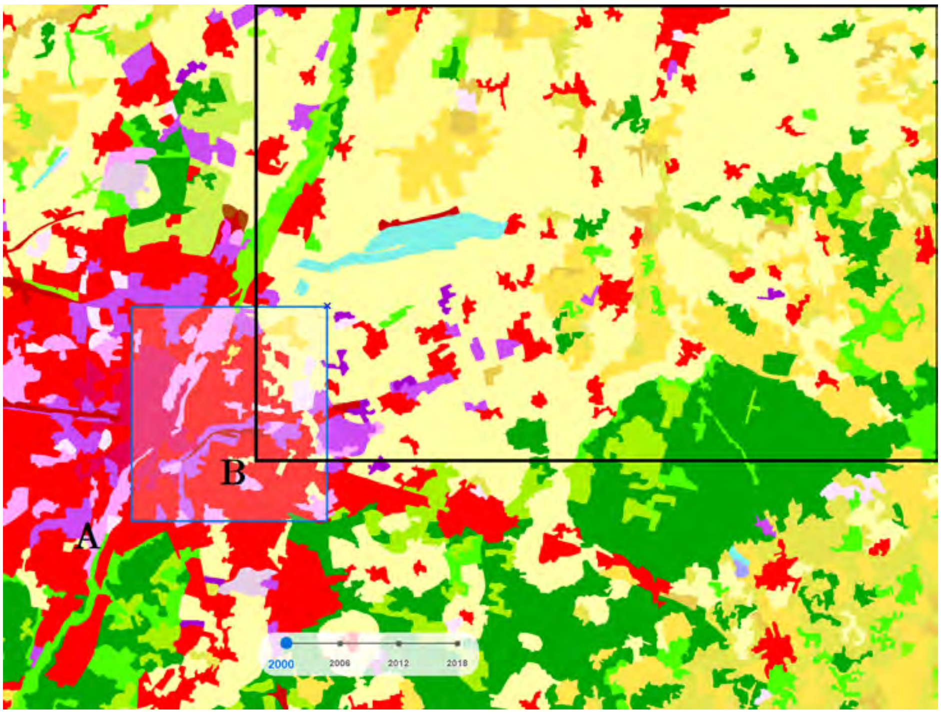

Figure 4.10: An image of the application which shows how a feature which is inside the AOI (A) in the top right of the box can have a coverage (B) which extends far beyond the AOI.

4.5 Summary

A time-series visualization application using microservice cloud-based tools was built for this thesis following the work of others outlined in Chapter 2: **Related Work**, as well as

[18]`https://docs.geoserver.org/master/en/user/filter/filter_reference.HTML#bounding-box-operator` (Accessed 8 October 2021)

advice of experts in the field of Web GIS in Chapter 3: **Expert Interviews**. This chapter shows a functionally rich application which features various time-series visualizations can be built utilizing open-sourced APIs, publicly available data, and existing server technologies.

To further incorporate UCD into the development workflow of the application, the next chapter outlines a user study which was conducted to understand the application in terms of the **usability** and **utility** of the application. Further interest was placed in understanding the intuitivity of the UI/UX of the prototype from the participants, as well as understanding any additional needs of the end user.

Chapter 5

User Study

5.1 Methodology

The research done on understanding UCD for interactive maps by R. Roth et al. (2015) is a building block for evaluating the **usability and utility** of an application. The authors recommend a hybrid approach to incorporating methods to evaluate interfaces throughout design and development. As expert and theory based methods had already been utilized to influence the design strategies, user-based is the third method available to deploy in a development workflow.

Some related user-based methods include: questionnaires, entry/exit surveys, blind voting, and cognitive workload assessment (R. Roth et al., 2015, p. 275). A method that provides value to evaluation at the deployment stage of design and development is an online user-study survey (Buttenfield, 1999). Surveys are good to employ in the evaluation process when characteristics of the targeted audience are not fully known, the investigators cannot be physically present, and multiple versions of an interface need to be evaluated.

5.2 Case Study

User Study

To gather user-feedback on the prototype for the customer (GAF), a summative online survey was created which comprises of twenty-one questions in total. The user study was conducted on the basis of multiple goals. The first being to create an investigative survey, discussed in Chapter 2.5 **User-Centered Design**, to understand the usability of the prototype and gain any insights from users into potential future design implementations. Another goal was to create a task which participants of the survey needed to complete using the functionality of the application. In the context of this thesis, the survey developed for the user study is helping to understand the value the additional time-series statistic generators can also add to the application.

Two different versions of the survey were sent out to users. The only difference between the two surveys is the link to the application which is included in the survey each participant gets sent. One version includes the full functionality of the prototype, and the other does not include the statistic panel. This means the statistic generator and associated graphic visualizations that come from the statistics panel, such as Figures 4.7 and 4.8 are not included in the task. Figure 5.1 shows an image of the view over Munich, Germany that both versions of the application open up to. In the figure, an AOI can be seen as a semi-transparent white feature over the study area. The AOI is pre-defined for the task outlined below.

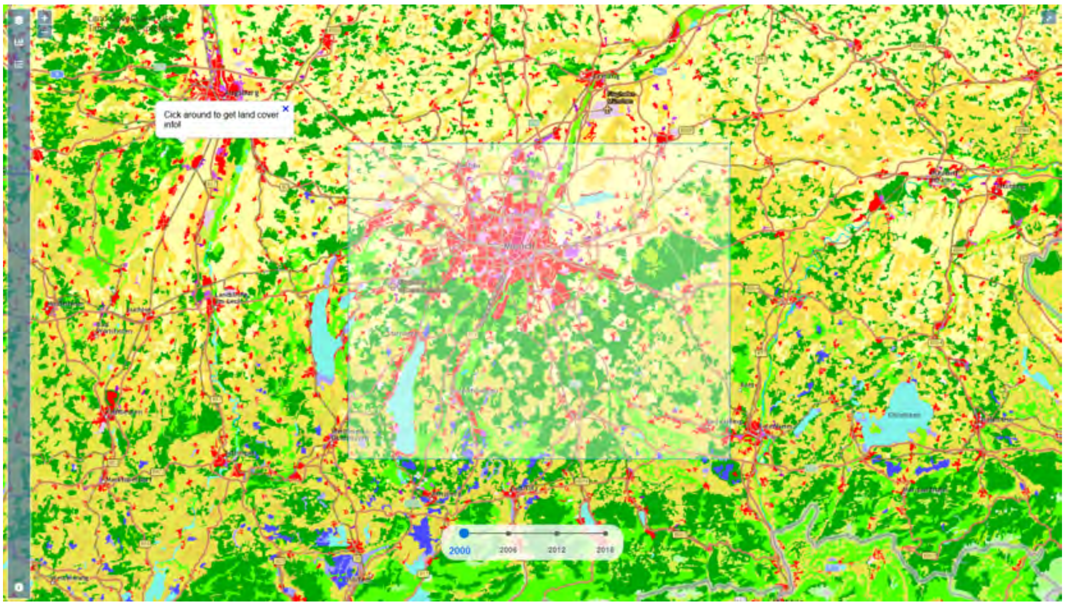

Figure 5.1: The application view when users open the link included in the user study.

The study opens by having the participant agree to a consent form. It then provides three optional background questions on the participants relation to data visualizations, geospatial data, and age. The survey then provides 3 current headlines from papers, which provide context to the effect climate change is having on European weather events. It then gives an excerpt from a European plan to increase carbon uptake of the continent by increasing the forest cover across the entire union. Following this, a user task is described, and the participants are provided a link to the hosted prototype, depending on which group they are a part of.

The user task was to use the built in functionality of the two versions of the app to approximate the percentage of coniferous forest for the years 2000, 2006, 2012, and 2018 around the AOI outlined above across a region in and around Munich (Figure 5.1).

Participants then answer twelve questions which are designed to help understand the usability and utility of the application, as well as the relation to the research questions. A final question allows participants to provide final feedback, and any additional development ideas for the prototype. The questions are listed below with context to their inclusion.

User Study Questions

The full user study and the included questions can be found as a document in *Appendix E*. A description of the questions, and the reason of their inclusion is outlined below.

Question 1 was included for consent to including the participants results in this thesis. Questions 2-4 were included for additional background analysis of the answers.

Participants were then given a instructions to open the application in their browser and perform a task to find the percentage of coniferous forest in an AOI for each year of the case study (see Appendix E, p. 87). Questions 5-8 are all included to directly provide an input for the participant to complete the task. In the analysis, the difference of the input values for each participant to the reference truth, or truthy value (referred to as such because of the potential variation in BBOX placement for each user of the app), for each year were then averaged by participant. All of the resulting values for all of the participants were then averaged for both surveys to make a comparison of the abilities based on the added statistic visualizations.

Questions 9-20 were all given a quantitative answer structure for the participants to provide an answer to help understand different aspects of the **usability** and/or **utility**. Each question had an exclusive scale of 1 to 5 as shown below:

Highly disagree 1 | 2 | 3 | 4 | 5 Highly agree

Table 5.3 highlights whether the questions were included to analyse the **usability** or **utility** of the map interface and associated visualization tools.

Question	Usability	Utility
9	X	
10	X	
11	X	
12		X
13		X
14	X	X
15	X	
16	X	
17	X	X
18		X
19		X
20		X

Table 5.1: The rationale for including the questions in the user study.

Additionally, the questions were grouped to understand various aspects of the application. Questions 9, 13, 15, and 17 were grouped to provide insight into the user opinion on the functionality of the application. Questions 12, 14, 18, and 19 were grouped to help see how time-series visualizations such as this can help provide insight into dynamic Earth systems such as climate change. Finally, questions 10, 11, and 16 were grouped to gain knowledge on how intuitive the application is to the user.

The final question, *Do you have any other suggestions for possible features?*, was included to allow participants an open entry form to enter in any additional features, as well as to provide a space for any feedback on the application.

5.3 Results

In Figures 5.2-5.5, bar graphs are introduced which present some of the results of the user study. Within these graphs, the two surveys are compared side-by-side. The blue bar titled "Full Function" refers to the responses to the survey that included all of the features of the application. The red bar titled "Statistics Removed" refers to the survey that did not have the statistics panel included. Figure 5.2 shows the results of the background questions submitted by the participants for questions 2 (A), 3 (B), and 3 (C).

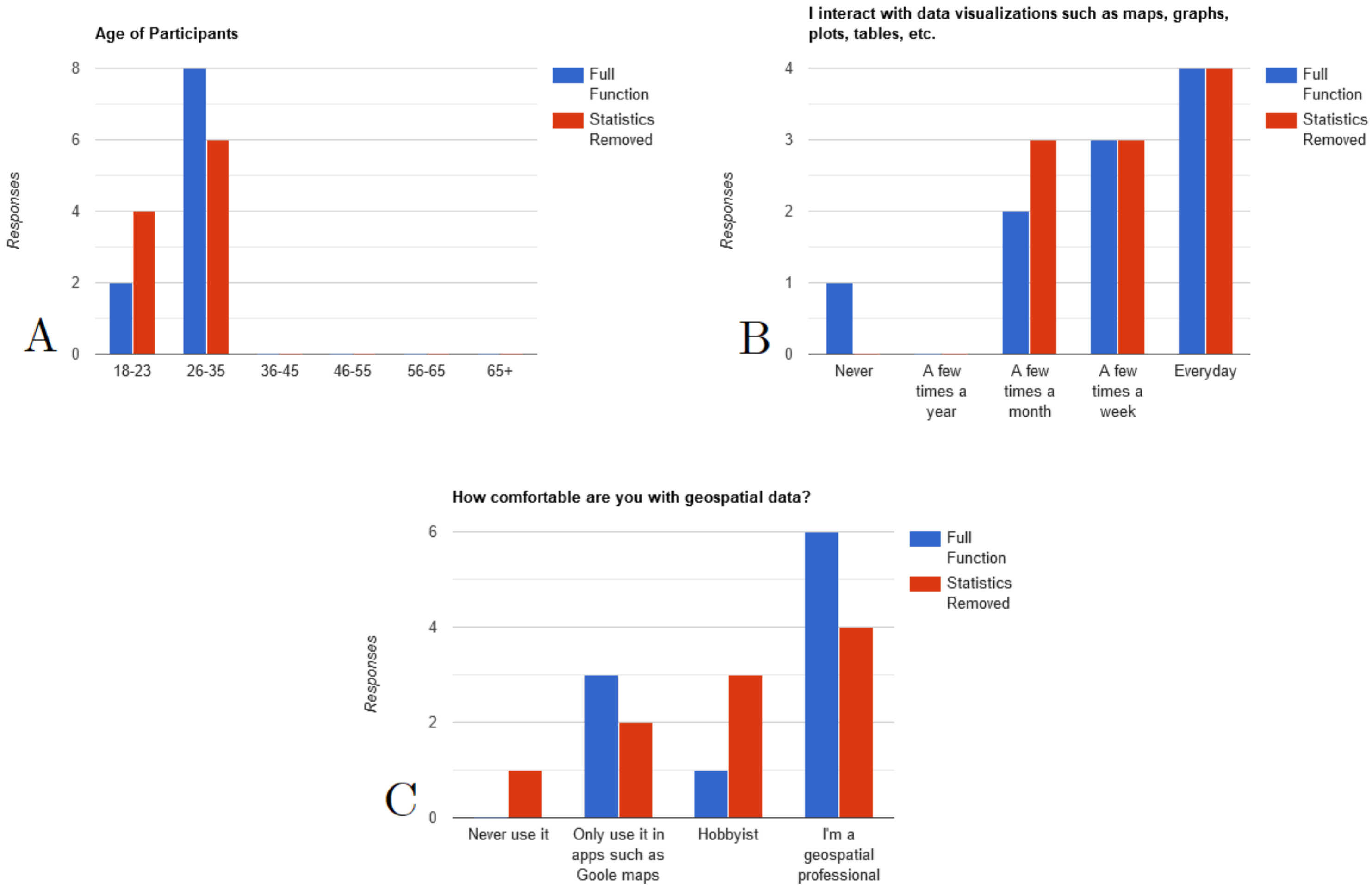

Figure 5.2: The background of the participants of the user study.

Tables 5.2 and 5.3 present the results of the task for the reduced functionality survey, and full functionality, respectfully. The final column in each table presents the average difference each participant had from the truthy values. ∅ ∆ refers to the average percentage of the differences for each participant. The values in in each of the columns relating to the yearly value input for each participant refers to a percentage value out of the total AOI the user defined in the task.

Participant	2000	2006	2012	2018	∅ Δ
1	20	19	12	10	6.30
2	30	29	24	23	17.55
3	N/A	N/A	N/A	N/A	N/A
4	18	17	9	8	4.05
5	40	38	28	25	23.80
6	20	19	15	14.5	8.18
7	N/A	N/A	N/A	N/A	N/A
8	100	70	50	20	51.05
9	12	11	7	6	.05
10	30	28	25	24	17.80
Average Δ					16.10

Table 5.2: The results of the user task in the reduced functionality survey.

Participant	2000	2006	2012	2018	∅ Δ
1	12.78	11.1	5.96	5.95	0
2	12.81	11.06	5.96	5.95	-.003
3	12.73	11.05	5.88	5.87	-0.065
4	12.82	11.08	5.98	5.97	.015
5	12.81	11.07	5.97	5.95	.003
6	35.6	32.55	30.05	29.35	22.94
7	12.76	11.06	5.97	5.95	-.013
8	12.68	11.08	5.92	5.07	-.26
9	N/A	N/A	N/A	N/A	N/A
10	12.81	11.06	5.97	5.96	.003
Average Δ					2.513

Table 5.3: The results of the user task in the full functionality survey.

Figure 5.3 displays the results of questions 9 (A), 13 (B), 15 (C), and 17 (D) which were grouped to provide insight into the user opinion on the functionality of the application.

The results to questions 12 (A), 14 (B), 18 (C), and 19 (D) are shown in Figure 5.4. These questions were put together to understand the user sentiment towards change analysis using the functionality of the application.

Figure 5.5 present the results to questions 10 (A), 11 (B), 16 (C) and 20 (D) in the survey. These questions are displayed together to gain insight into the intuitivity of the application.

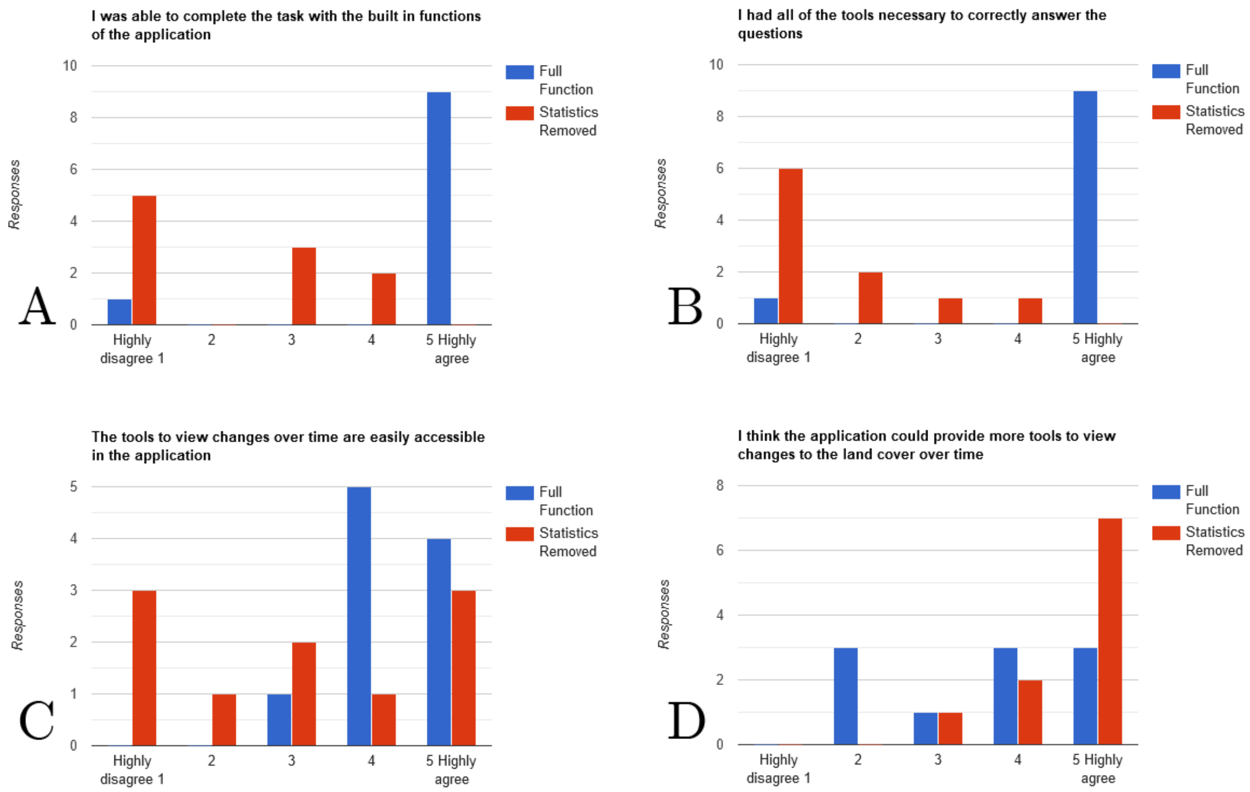

Figure 5.3: The questions grouped together for insight into application functionality.

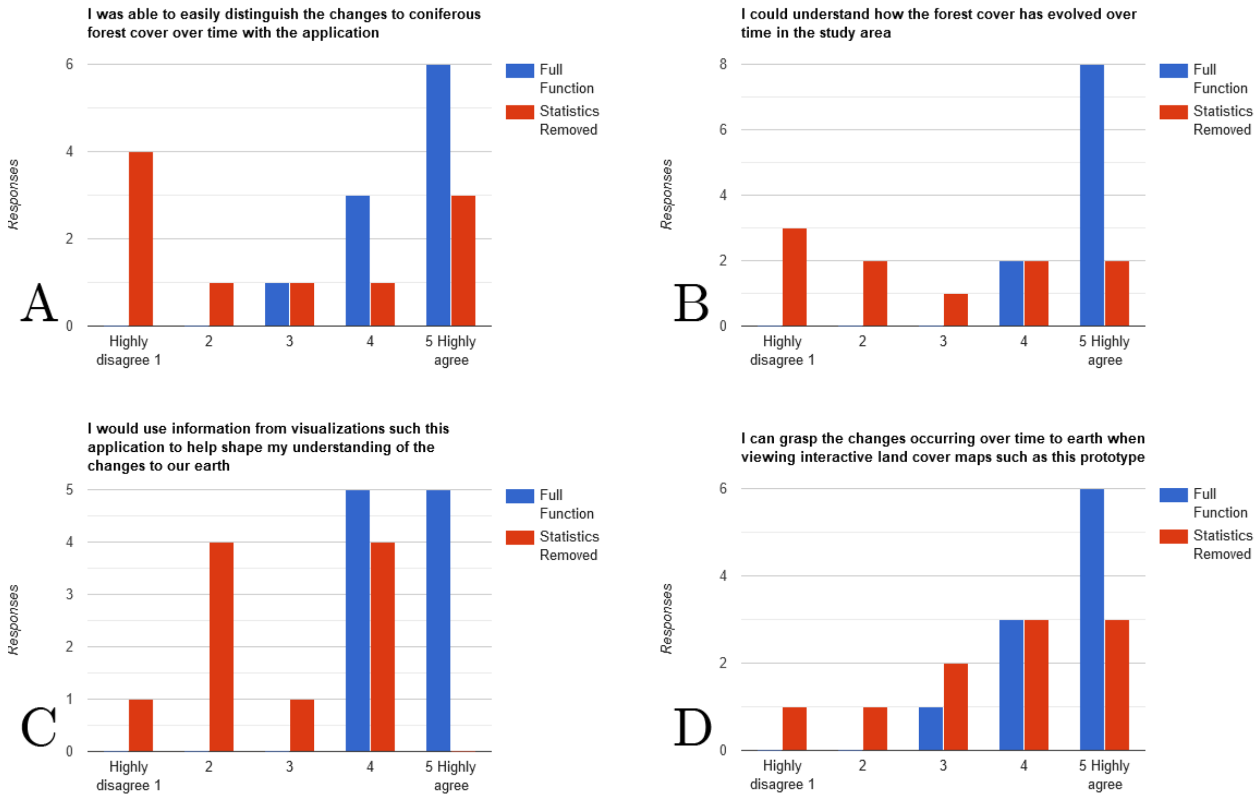

Figure 5.4: The questions grouped together for change analysis with the application.

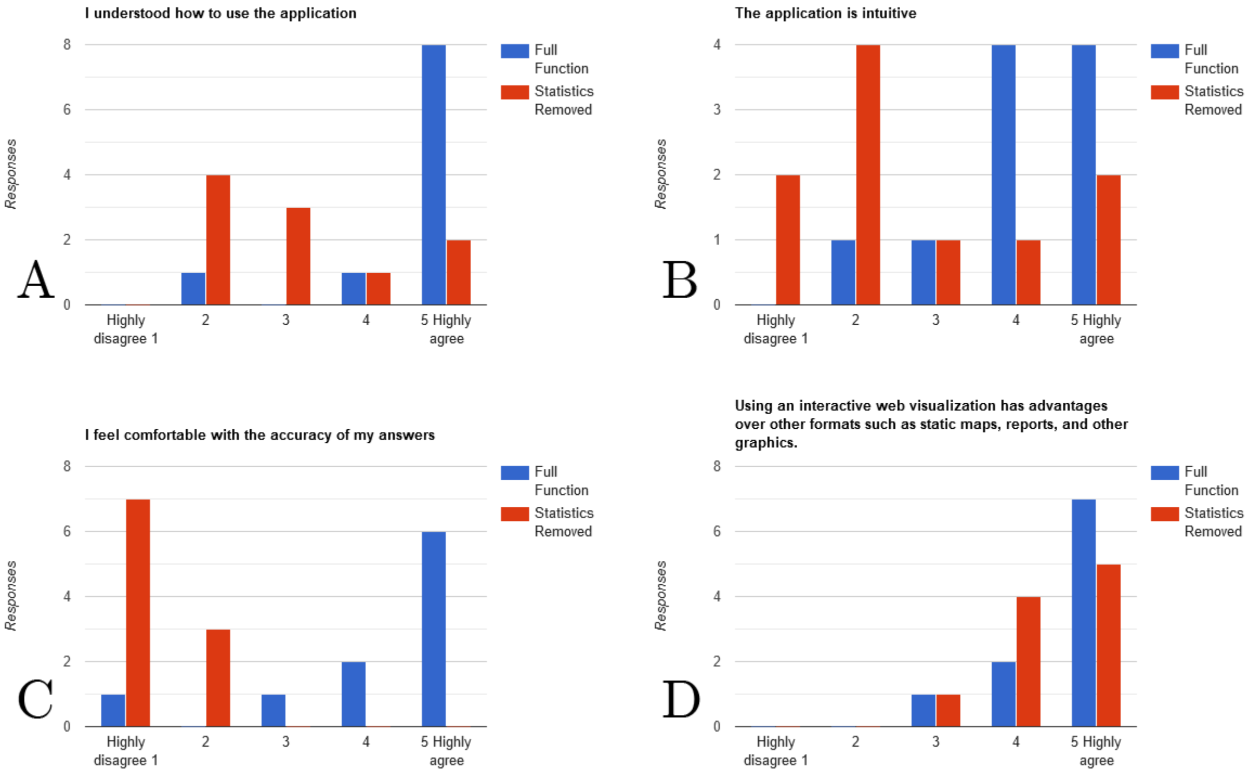

Figure 5.5: The questions grouped together to gain insight on the application intuitivity.

5.4 Discussion

Understanding the **usability** and **utility** of the application is key to keeping the **user base** at the center of the design strategy, according to user-centered design principles outlined in Chapter 2.5: **User-Centered Design**. Questions for the survey were created to gain an idea of the user-sentiment towards the functionality, visualizations, data, and the changes over time. Both versions of the survey included all of the same questions to gain insight into the general sentiments between the two versions of the app.

Knowing the user sentiments in terms of *interface success* for the visualization of time-series LCLU data helps to provide feedback for future projects in the field of Web GIS microservices. The results from Chapter 5.3 are discussed below.

Task Results

When looking at the results of the user studies, the benefit of including further statistical graphic visualizations is in full agreement with the work in Freitas et al. (2011). The additional statistics add present a much more in-depth understanding of the data than maps alone. The group which had the full functionality of the application for outperformed the reduced functionality group. The average Δ for the group with the statistics generator was 2.53%, while for the group without statistics was 16.10%. Additionally, participant 6 in the full funcitonality group reached out to the author afterwards to disclaim that they had mistakenly marked the AOI of their statistic generator far too large. With the anomaly of their entries removed, the Δ of the group is just -.04% which could be due to user differences in perception on a small scale when outlining their AOI in the application.

Another aspect of the task results which is notable is related to the participants who could not complete the task. Although the reasons for not entering the percentages are unknown, two participants in the reduced functionality survey chose not to enter any values, and one participant in the full functionality survey chose not to enter any values for the four years. One possible reason for the lack of participation in the reduced functionality group is the lack of interest in working out the values with merely visual indications of the classes on the map. For the participant who did not enter values for full functionality group, the reasons are also speculation. Perhaps they did not read the full task description, which outlines the statistic generator on the side panel. None of the three participants who did not enter values left additional feedback in the last open entry form.

Functionality

Questions 9, 13, 15, and 17 were grouped together to generally understand more on the topic of the functionality of the application. This is directly related to both the **usability and utility**, and whether the users felt the application was sufficient for completing the task in terms of the quantity and accessibility of the tools for the task of the survey.

As seen in Figure 5.3, in the full function group, 90% responded that they were both able to complete the task with the functions of the application (A), they had all of the necessary tools to answer the questions (B), and that the tools were easily accessible (C). Contrast this with the statistics removed group, which 80% were either neutral or highly disagreed that they had they were able to complete the task functions (A), 90% believe that they had all of the necessary tools (B), and 60% believe that the tools are easily accessible in the application (C). This is a clear indication that the additional statistical time-series visualization increase the **usability and utility** of the application.

Change Analysis

Questions 12, 14, 18, and 19 are grouped to help provide insight into the **usability** in terms of change analysis in the underlying data. The context of the user study was on seeing the changes to carbon uptake of the Earth's surface to directly influence the ability to combat climate change. Tools such as this prototype can add value to the monitoring of such projects on a municipal, country-wide, and global scale.

When analysing the user responses, the results are again very clear in favor of the full function group which includes the time-series statistic generator. 90% could easily distinguish the changes to coniferous forest cover over time (A), 100% of the group could understand how the forest cover has evolved over time (B), and 100% responded that they would also use visualizations such as this application to help shape their understanding of changes to our earth (C). Again, the statistics removed group generally responded on the contrary. There were 60% that were either neutral or disagreed that they could easily distinguish the changes to the coniferous forest (A), 60% which were neutral or disagreed that they could understand how forest evolved over time in the study area (B), and again 60% that were either neutral or disagreed that they would use information from visualizations such as this application to shape their understanding of the changes to our earth (C). Perhaps the starkest indication of these three comparisons is the last statistic mentioned regarding using an application such as this to shape their understanding. It is clear that the direct statistic values provided a strong understanding of the changes occurring to the land cover of the AOI.

Intuitivity

Questions 10, 11, and 16 were included in the survey to help answer the RQ1 subquestion: How can a product be designed with intuitive use as a core function? As outlined in Chapter 2.5.3: **Intuitivity**, the concept in regards to a product's use is incredibly important. This is because understanding the user sentiment on intuitivity of the application can directly lead to important improvements in the interface (Blackler, 2018). This concept is related to the definition of **usability** provided by R. Roth et al. (2015) in Chapter 2.5.4: **UCD in Cartography**.

The results agree that the version of the application with the full functionality is much more intuitive when a statistics time-series is included to understand the land cover data more. In response to "The application is intuitive", 80% of the full function group agreed with the statement (B), and 60% of the statistics removed group disagreed while a further 10% were neutral (B).

In the full function group, 90% understood how to use the application (A), and 80% feel comfortable with the accuracy of their answers (C). Contrast this with the statistics removed group, where 60% were either neutral or disagreed that they understood how to use the application (A), and a full 100% disagreed with the statement, "I feel comfortable with the accuracy of my answers," (C). These findings point to a major increase in intuitivity for Web GIS applications that include additional features which draw out more information from slippy maps, such as statistical analysis graphics.

Potential Issues

One potential issue that could be found in this data, is the lack of age diversity in both groups of users (Figure 5.2, A). All of the respondents to the survey were between the age of 18 and 35. This could lead to misrepresentation of things such as intuitivity, usability, and utility of the application. If more age groups were represented, then a more thorough understanding of how various populations interact with the type of visualizations presented in the application could be found.

5.5 Summary

The results of the user study are clear. Including additional time-series visualizations, such as the statistic pie-charts and tables, into a LCLU Web GIS application is beneficial to the **usability**, **utility**, and also the intuitivity of the application. The statistics helped users understand the data, changes to the data, as well as the application as a whole much deeper. Providing these visualizations add value to LCLU map interfaces and increase the *interface success*.

Chapter 6

Synthesis

The development of microservice web applications has been an emerging trend to create web services which offer an opportunity for streamlined development, code isolation from dependencies, and an abundance of web APIs for enriched application functionality. In terms of Web GIS, there is still a large space for research into the topic. Due to the growing abundance of geospatial data, the opportunity to develop Web GIS microservices and micro-frontends which can be implemented into large organizational web architectures provides a modular approach to integrate additional geospatial applications into more web services. In addition to slippy maps, further statistical analysis in the form of various visualizations impart a growth in understanding for the user in the inherent data displayed on the map interface aspect of a Web GIS application.

For this thesis, a microservice Web GIS application was developed which presents LCLU data in various time-series visualizations such as time sliders, pie-charts, and tabular data. The development tasks of the application integrated UCD methodology for each step to deliver a product, which results suggest has *interface success* for the end-user. The expected results were achieved through the successful creation of a Web GIS microservice which offers multiple time-series visualizations for LCLU data.

The development tasks were accomplished with the help of a review of the related work in the field to gain a grasp of current strategies used to create Web GIS microservices. Additionally, the expert interviews conducted for the thesis lay out a framework of the technological and user needs of the end product in the context of GAF's CLC+ project. From these requirement analyses, various methods such as utilizing existing cartographic and visualization APIs helped create a functional application which meets the needs of the end user. Further, because the application is built with microservice ideology, the application could be integrated into any larger organizational web architecture system.

Cartographic UCD evaluation methods for map interfaces can be broken into three methods: expert based, theory based, and user based. Combining the three methods conveys the most thorough understanding of an interface to help the user remain at the center of the design process. For this thesis all three were completed in the form of: a theory-based review of the related work, expert interviews to understand the product and end user, and a user-based survey to gain insights into user sentiment of the end product. The results of the user survey suggest that the end product is highly intuitive, contains sufficient tools and visualizations for a firm understanding of the data, and meets the needs of the end users.

The results of this thesis would agree with Mena et al. (2019), that there is an inherent value in developing Web GIS applications as microservices. The applications serve data fast and reliably, due to the fact that only the relevant files and servers are called upon when the user engages with the different tools within the map interface. Further, the results of Chapter 5: **User Study** leads to agreement with Freitas et al. (2011) in the discussion that the visualization of remote sensing data as time-series in web map interfaces, is more valuable when additional statistic visualizations are added to the interface.

The research questions were answered throughout the work of the thesis. The results and discussion within the context of Chapter 3: **Expert Interviews**, help lay the groundwork for the direct answers to the questions in the broader results from the expert feedback. These answers to the research questions directly led to many of the design and development strategies of the Web GIS microservice developed as a part of this thesis.

Further, the hypothesis that microservice cartographic time-series visualizations, built with UCD principles in mind, are advantageous to include in organizational Web GIS architectures can be confirmed based on the results of the user study. A majority of the respondents of the survey say that they find the application highly functional, see value in a time-series visualization such as this, and would use these visualizations to inform themselves on various time-dependent events such as climate change. The UCD methods, also directly led to high intuitivitiy, and an overall satisfaction of the product. These, among other survey results point to high positivity in the Web GIS microservice prototype, and the potential to include services such as this in larger organizational web architectures.

In the context of this thesis, the research questions and hypothesis were successful, however there could be certain biased factors which could have lead to the achieved results. As discussed in Chapter 5.: **User Study Discussion**, the survey results were completely from participants between the ages of eighteen of thirty-five. This age bias could have led to a different understanding of inherent web intuitivity among the users. The results from these participants could be different from those in different age groups, such as 65+ for example, which may not have the comfort and understanding of the sub-conscious know-how in navigating web interfaces. The results of the user study were based on these respondents, so future work could search for a larger diversity in age groups for respondents to understand a larger sub-set of the population.

An additional factor which could be considered when evaluating the results of the user study are in the number and the type of visualizations within the app. The decision to only include the time-slider, pie-charts, and tabular time-series visualizations was due to a feedback in the expert interviews, time restrictions for the thesis, and the author's results when searching for possible APIs to integrate. Certainly, other research could potentially utilize more and different visualizations to create time-series and get different results.

Chapter 7

Conclusion and Outlook

7.1 Conclusion

This thesis has helped provide a foundation and framework for the processing and visualization of changes to LCLU classes in time-series in the form of a Web GIS microservice prototype. In the context of existing CLC data, these visualization tools have helped provide a better understanding, increased accessibility, and an overall satisfaction with the data based on the scientific results placed within this thesis. Developing microservice tools which give a more thorough understanding and grasp of the CLC data give an opportunity to develop tools which can be deployed alongside the upcoming release of the CLC+ dataset. Offering time-series visualizations helps the user understand changes to the dataset more, and operating the change tools in the interface gives the user a deeper feeling of ownership over the application and the data within.

A microservice operates as a part of a larger web application architecture as an independent service without further dependencies in the organizational technology stack. Additionally, microservices use a modular approach to development which highlights the value of APIs, and distributed servers across the internet. Because Web GIS applications often operate by utilizing a variety of servers and data sources, it presents a great opportunity to utilize the microservice framework to integrate more Web GIS services into organizational web architectures.

Although coordinating the development and maintenance of an entire organizational architecture structure built from microservices can become a very daunting task, the benefits of code independence and streamlined development have been shown to outweigh the negatives.

The framework for design and development strategy was completed using a hybrid methodological approach. First, a theory based assessment of the related work was laid out. Then expert interviews were conducted to create a basis of understanding of the **user base** and their needs for such a product. A Web GIS microservice was then built to present the time-series visualizations. Finally, a user study was completed to understand the **usability** and

utility of the product. Incorporating UCD as a core component of design and development was essential to achieving the results of this thesis.

The user study was designed to create a real world task to evaluate the functionality of the time-series visualization tools, in the larger picture of Web GIS microservices. The consensus was overwhelmingly positive for the full functionality version of the app, and provides a framework for future Web GIS microservice tools. The results of the user study also suggest that time-series visualization tools offer value to integrating into data inventories such as the CLC data store. A majority of users responded that tools such as the application prototype developed for this thesis would help to shape their understanding of dynamically changing events, such as climate change, and that they would utilize such tools to help understand the data more.

7.2 Outlook - Future Trends in Frameworks for Cloud-based Cartographic Visualization Tools

The outlook for cloud-based cartographic visualization tools is promising. Due to the rise in the number of easily implementable cartographic APIs which offer a rich amount tools, the growth of Web GIS microservices is slowly being encouraged from an open-source perspective. Modular design, and micro-frontends are becoming more used, and this makes it easier to integrate maps, and various cartographic visualization tools into existing web applications as stand alone micro-frontends and microservices.

There are still topics within the topic of Web GIS microservices and visualization which could be researched. Things such as studying the value of modularized maps as a component of web and micro-frontends, researching how different forms of statistic visualizations can be integrated into slippy maps, and the testing the speed performance of Web GIS microservices compared to established Web GIS applications.

One additional note is that during the development of this thesis, the Copernicus Programme released their own land cover and land cover change dashboard[1]. The dashboard also incorporates additional statistic visualizations in the form of pie-charts and bar graphs. One difference between the CLC dashboard and the thesis prototype, is that the dashboard allows users to select individual countries to view localized statistics, as well as highlighting changes within each state in the country. Another feature of the dashboard which is not included in this prototype is the grouping of similar land cover classes to help make the data more digestible to the user. The dashboard highlights how many of the features and design strategies researched in this thesis can be applied to the real existing CLC dataset.

[1] `https://land.copernicus.eu/dashboards/clc-clcc-2000-2018` (Accessed 1 October 2021)

References

Abd El-Kawy, O. R., Rød, J. K., Ismail, H. A., & Suliman, A. S. (2011). Land use and land cover change detection in the western Nile delta of Egypt using remote sensing data. *Applied geography*, *31*(2), 483–494. (Publisher: Elsevier)

Abras, C., Maloney-krichmar, D., & Preece, J. (2004). User-Centered Design. In *In Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications.* Publications.

Agrawal, S., & Gupta, R. D. (2017, November). Web GIS and its architecture: a review. *Arabian Journal of Geosciences*, *10*(23), 518. Retrieved 2021-07-21, from https://doi.org/10.1007/s12517-017-3296-2

Alesheikh, A., Helali, H., Behroz, H., St, D., Valy, A., Sq, T., & Iran. (2021, August). Web GIS: Technologies and its applications.

Arlitt, M., & Williamson, C. (1997, October). Internet Web servers: workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, *5*(5), 631–645. (Conference Name: IEEE/ACM Transactions on Networking)

Bielecka, E., & Jenerowicz, A. (2019). Intellectual structure of CORINE land cover research applications in web of science: A europe-wide review. , *11*(17), 2017. Retrieved 2021-08-18, from https://www.mdpi.com/2072-4292/11/17/2017 (Number: 17 Publisher: Multidisciplinary Digital Publishing Institute)

Blackler, A. (2018). *Intuitive Interaction: Research and Application*. CRC Press. (Google-Books-ID: y9ZiDwAAQBAJ)

Boulos, M. N. K., & Honda, K. (2006, January). Web GIS in practice IV: publishing your health maps and connecting to remote WMS sources using the Open Source

UMN MapServer and DM Solutions MapLab. *International Journal of Health Geographics*, *5*(1), 6. Retrieved 2021-08-19, from https://doi.org/10.1186/1476-072X-5-6

Brovelli, M. A., Minghini, M., & Zamboni, G. (2016). Public participation in GIS via mobile applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, *114*, 306–315. (Publisher: Elsevier)

Bryhni, H., Klovning, E., & Kure, O. (2000, July). A comparison of load balancing techniques for scalable Web servers. *IEEE Network*, *14*(4), 58–64. (Conference Name: IEEE Network)

Buttenfield, B. (1999). Usability evaluation of digital libraries. *Science & Technology Libraries*, *17*(3-4), 39–59. (Publisher: Taylor & Francis)

Büttner, G. (2014). CORINE land cover and land cover change products. In I. Manakos & M. Braun (Eds.), *Land use and land cover mapping in europe: Practices & trends* (pp. 55–74). Springer Netherlands. Retrieved 2021-08-19, from https://doi.org/10.1007/978-94-007-7969-3_5

Cardellini, V., Colajanni, M., & Yu, P. (1999, May). Dynamic load balancing on Web-server systems. *IEEE Internet Computing*, *3*(3), 28–39. (Conference Name: IEEE Internet Computing)

Castelluccio, M., Poggi, G., Sansone, C., & Verdoliva, L. (2015, August). Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. *arXiv:1508.00092 [cs]*. Retrieved 2021-09-23, from http://arxiv.org/abs/1508.00092 (arXiv: 1508.00092)

Cerba, O., Charvat, K., & Ježek, J. (2009, January). Data Harmonization Towards CORINE Land Cover. *Univ. of West Bohemia, Pilsen, Czech Republic,(accessed on 6 Nov. 2012).[Online]. Available: www. efita. net/apps/accesbase/bindocload. asp*.

Cerny, T., Donahoo, M. J., & Trnka, M. (2018, January). Contextual understanding of microservice architecture: current and future directions. *ACM SIGAPP Applied Computing Review*, *17*(4), 29–45. Retrieved 2021-08-19, from https://doi.org/10.1145/3183628.3183631

Chen, X.-L., Zhao, H.-M., Li, P.-X., & Yin, Z.-Y. (2006, September). Remote sensing image-based analysis of the relationship between urban heat island and land use/cover changes. *Remote Sensing of Environment*, *104*(2), 133–146. Retrieved 2021-09-23, from https://www.sciencedirect.com/science/article/pii/S0034425706001787

Cherradi, G., El Bouziri, A., Boulmakoul, A., & Zeitouni, K. (2017, January). Real-Time HazMat Environmental Information System: A micro-service based architecture. *Procedia Computer Science*, *109*, 982–987. Retrieved 2021-08-19, from https://www.sciencedirect.com/science/article/pii/S1877050917311420

Cieślak, I., Szuniewicz, K., Pawlewicz, K., & Czyża, S. (2017, October). Land Use Changes Monitoring with CORINE Land Cover Data. *IOP Conference Series: Materials Science and Engineering*, *245*, 052049. Retrieved 2021-04-21, from https://iopscience.iop.org/article/10.1088/1757-899X/245/5/052049

Copernicus Land Monitoring Service pan-European component: Technical specifications for implementation of CLC+ based on the EAGLE concept — European Environment Agency [Negotiated Procedure]. (2017). Retrieved from https://www.eea.europa.eu/about-us/tenders/negotiated-procedures/copernicus-land-monitoring-service-pan

Dewan, A. M., & Yamaguchi, Y. (2009, July). Land use and land cover change in Greater Dhaka, Bangladesh: Using remote sensing to promote sustainable urbanization. *Applied Geography*, *29*(3), 390–401. Retrieved 2021-09-23, from https://www.sciencedirect.com/science/article/pii/S0143622809000058

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: Yesterday, Today, and Tomorrow. In M. Mazzara & B. Meyer (Eds.), *Present and Ulterior Software Engineering* (pp. 195–216). Cham: Springer International Publishing. Retrieved 2021-08-19, from https://doi.org/10.1007/978-3-319-67425-4_12

Erl, T. (2004). *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services. 2004.* Prentice Hall PTR Upper Saddle River, NJ, USA.

Farkas, G. (2020). Possibilities of using raster data in client-side web maps. *Transactions in GIS*, *24*(1), 72–84. Retrieved 2021-08-19, from https://onlinelibrary.wiley.com/doi/abs/10.1111/tgis.12588 (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/tgis.12588)

Fielding, R. T., & Taylor, R. N. (2002, May). Principled design of the modern Web architecture. *ACM Transactions on Internet Technology*, *2*(2), 115–150. Retrieved 2021-09-24, from https://doi.org/10.1145/514183.514185

Freitas, R. d., Arai, E., Adami, M., Ferreira, A. S., Sato, F. Y., Shimabukuro, Y. E., ... Rudorff, B. F. T. (2011). Virtual laboratory of remote sensing time series: visualization of MODIS EVI2 data set over South America. *Journal of Computational Interdisciplinary Sciences*, *2*(1), 57–68.

Galitz, W. O. (2007). *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. John Wiley & Sons. (Google-Books-ID: Q3Xp_Awu49sC)

Garrett, J. J. (2007). Ajax: A New Approach to Web Applications. *undefined*. Retrieved 2021-09-24, from https://www.semanticscholar.org/paper/Ajax%3A-A-New-Approach-to-Web-Applications-Garrett/c440ae765ff19ddd3deda24a92ac39cef9570f1e

Geers, M. (n.d.). *Micro Frontends - extending the microservice idea to frontend development.* Retrieved 2021-09-24, from https://micro-frontends.org/

Giri, C. P. (2012). *Remote Sensing of Land Use and Land Cover: Principles and Applications*. CRC Press. (Google-Books-ID: WZkEO9yZxS4C)

Goodman, S., BenYishay, A., Lv, Z., & Runfola, D. (2019, January). GeoQuery: Integrating HPC systems and public web-based geospatial data tools. *Computers & Geosciences*, *122*, 103–112. Retrieved 2021-08-19, from https://www.sciencedirect.com/science/article/pii/S0098300418305326

Hecht, R., Behnisch, M., & Herold, H. (2020, August). Innovative Approaches, Tools and Visualization Techniques for Analysing Land Use Structures and Dynamics of Cities and Regions (Editorial). *Journal of Geovisualization and Spatial Analysis*, *4*(2), 19. Retrieved 2021-09-23, from https://doi.org/10.1007/s41651-020-00060-9

Hurtienne, J., & Blessing, L. (2007). Design for Intuitive Use-Testing image schema theory for user interface design. In *DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07. 2007* (pp. 829–830).

Iacovella, S. (2017). *GeoServer Beginner's Guide: Share geospatial data using Open Source standards*. Packt Publishing Ltd. (Google-Books-ID: 7RhKDwAAQBAJ)

Jansen, B. J. (1998, April). The graphical user interface. *ACM SIGCHI Bulletin*, *30*(2), 22–26. Retrieved 2021-09-24, from https://doi.org/10.1145/279044.279051

Jiang, J., & Tian, G. (2010). Analysis of the impact of land use/land cover change on land surface temperature with remote sensing. *Procedia environmental sciences*, *2*, 571–575. (Publisher: Elsevier)

Kandziora, M., Burkhard, B., & Müller, F. (2013, June). Mapping provisioning ecosystem services at the local scale using data of varying spatial and temporal resolution. *Ecosystem Services*, *4*, 47–59. Retrieved 2021-08-19, from https://www.sciencedirect.com/science/article/pii/S2212041613000211

Kosztra, B., & Arnold, S. (2019, November). *Implementation of CLC+ based on the EAGLE concept – additional support for further development of CLC+ databases (CLC+ Backbone and CLC+ Core)* (Tech. Rep.). Retrieved 2021-08-20, from https://land.copernicus.eu/eagle/files/documents-and-report-2019/clc-implementation-semantic-composition-ontologies-task2-deliverable-d3-v3-2-2019

Lee, J.-G., & Kang, M. (2015, June). Geospatial Big Data: Challenges and Opportunities. *Big Data Research*, *2*(2), 74–81. Retrieved 2021-08-19, from https://www.sciencedirect.com/science/article/pii/S2214579615000040

Lewis, J. (2012, March). *Micro services—Java, the unix way,*. Kraków, Poland.

Mazzara, M., Khanda, K., Mustafin, R., Rivera, V., Safina, L., & Sillitti, A. (2018). Microservices Science and Engineering. In P. Ciancarini, S. Litvinov, A. Messina,

A. Sillitti, & G. Succi (Eds.), *Proceedings of 5th International Conference in Software Engineering for Defence Applications* (pp. 11–20). Cham: Springer International Publishing.

Mena, M., Corral, A., Iribarne, L., & Criado, J. (2019). A Progressive Web Application Based on Microservices Combining Geospatial Data and the Internet of Things. *IEEE Access*, *7*, 104577–104590. Retrieved 2021-04-21, from https://ieeexplore.ieee.org/document/8782470/

Mendonça, N., Jamshidi, P., Garlan, D., & Pahl, C. (2019, November). Developing Self-Adaptive Microservice Systems: Challenges and Directions. *IEEE Software*, *PP*.

Naranjo Gómez, J. M., Lousada, S., Garrido Velarde, J., Castanho, R. A., & Loures, L. (2020, July). Land-Use Changes in the Canary Archipelago Using the CORINE Data: A Retrospective Analysis. *Land*, *9*(7), 232. Retrieved 2021-08-19, from https://www.mdpi.com/2073-445X/9/7/232 (Number: 7 Publisher: Multidisciplinary Digital Publishing Institute)

Naumann, A., Hurtienne, J., Israel, J. H., Mohs, C., Kindsmüller, M. C., Meyer, H. A., & Hußlein, S. (2007). Intuitive Use of User Interfaces: Defining a Vague Concept. In D. Harris (Ed.), *Engineering Psychology and Cognitive Ergonomics* (pp. 128–136). Berlin, Heidelberg: Springer.

Neene, V., & Kabemba, M. (2017). Development of a mobile GIS property mapping application using mobile cloud computing. *Development*, *8*(10), 57–68.

Norman, D. (1998, January). The Invisible Computer : Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution / D.A. Norman.

Norman, D., Miller, J., & Henderson, A. (1995, May). What you see, some of what's in the future, and how we go about doing it: HI at Apple Computer. In *Conference Companion on Human Factors in Computing Systems* (p. 155). New York, NY, USA: Association for Computing Machinery. Retrieved 2021-09-23, from https://doi.org/10.1145/223355.223477

Norman, D. A., & Draper, S. W. (1986). *User Centered System Design; New Perspectives on Human-Computer Interaction*. USA: L. Erlbaum Associates Inc.

Nowak, M. M., Dziób, K., Ludwisiak, \., & Chmiel, J. (2020). Mobile GIS applications for environmental field surveys: A state of the art. *Global Ecology and Conservation*, *23*, e01089. (Publisher: Elsevier)

Pan-European — Copernicus Land Monitoring Service [Land Section]. (n.d.). Retrieved 2021-09-28, from `https://land.copernicus.eu/pan-european`

Pavlenko, A., Askarbekuly, N., Megha, S., & Mazzara, M. (2020). Micro-frontends: application of microservices to web front-ends. *J. Internet Serv. Inf. Secur.*, *10*(2), 49–66.

Peng, Z.-R., & Zhang, C. (2004, June). The roles of geography markup language (GML), scalable vector graphics (SVG), and Web feature service (WFS) specifications in the development of Internet geographic information systems (GIS). *Journal of Geographical Systems*, *6*, 95–116.

Qi, L., He, Q., Chen, F., Zhang, X., Dou, W., & Ni, Q. (2020). Data-Driven Web APIs Recommendation for Building Web Applications. *IEEE Transactions on Big Data*, 1–1. (Conference Name: IEEE Transactions on Big Data)

Rantanen, J. (2015, August). Isolation Mechanisms for Web Frontend Application Architectures. Retrieved 2021-09-22, from `https://aaltodoc.aalto.fi:443/handle/123456789/17825` (Accepted: 2015-09-18T09:11:17Z)

Rodriges Zalipynis, R. A. (2012). Representing Earth remote sensing data as time series. Retrieved 2021-05-31, from `http://ea.donntu.edu.ua:8080/jspui/handle/123456789/19731` (Accepted: 2013-05-15T12:55:55Z Publisher: Donetsk National Technical University)

Rogers, Y., & Sharp, H. (2002). *Interaction Design: Beyond Human-Computer Interaction* (J. Preece, Y. Rogers, & H. Sharp, Eds.). John Wiley. Retrieved 2021-08-19, from `http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471492787.html`

Roth, R., Ross, K., & MacEachren, A. (2015, March). User-Centered Design for Interactive

Maps: A Case Study in Crime Analysis. *International Journal of Geo-Information*, *4*, 262–301.

Roth, R. E., Çöltekin, A., Delazari, L., Filho, H. F., Griffin, A., Hall, A., … van Elzakker, C. P. (2017, October). User studies in cartography: opportunities for empirical research on interactive maps and visualizations. *International Journal of Cartography*, *3*(sup1), 61–89. Retrieved 2021-10-05, from https://doi.org/10.1080/23729333.2017.1288534 (Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/23729333.2017.1288534)

Soloway, E., Jackson, S. L., Klein, J., Quintana, C., Reed, J., Spitulnik, J., … Scala, N. (1996). Learning theory in practice: Case studies of learner-centered design. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 189–196).

Szumacher, I., & Pabjanek, P. (2017, April). Temporal Changes in Ecosystem Services in European Cities in the Continental Biogeographical Region in the Period from 1990–2012. *Sustainability*, *9*(4), 665. Retrieved 2021-08-19, from https://www.mdpi.com/2071-1050/9/4/665 (Number: 4 Publisher: Multidisciplinary Digital Publishing Institute)

Thönes, J. (2015, January). Microservices. *IEEE Software*, *32*(1), 116–116. (Conference Name: IEEE Software)

Voland, P., & Asche, H. (2017). Geospatial visualization of automotive sensor data: A conceptual and implementational framework for environment and traffic-related applications. In O. Gervasi et al. (Eds.), *Computational science and its applications – iccsa 2017* (pp. 626–637). Cham: Springer International Publishing.

Weng, Q. (2002, March). Land use change analysis in the Zhujiang Delta of China using satellite remote sensing, GIS and stochastic modelling. *Journal of Environmental Management*, *64*(3), 273–284. Retrieved 2021-09-23, from https://www.sciencedirect.com/science/article/pii/S0301479701905092

Xiao, Z., Wijegunaratne, I., & Qiang, X. (2016, November). Reflections on SOA and Microservices. In *2016 4th International Conference on Enterprise Systems (ES)* (pp. 60–67).

Youngblood, B. (2013). *GeoServer Beginner's Guide*. Packt Publishing Ltd. (Google-Books-ID: mbpsw5QJF54C)

Yu, H.-J., Li, S.-Y., & Zhang, T. (2021). A Space-Time Interactive Visualization Approach for Managing Remote Sensing Data. *Journal of Information Science & Engineering*, *37*(2).

Zhao, T., Zhang, C., Wei, M., & Peng, Z.-R. (2008). Ontology-Based Geospatial Data Query and Integration. In T. J. Cova, H. J. Miller, K. Beard, A. U. Frank, & M. F. Goodchild (Eds.), *Geographic Information Science* (pp. 370–392). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zhou, W., Chi, C., Wang, C., Wong, R., & Ding, C. (2014). Bridging the Gap between Spatial Data Sources and Mashup Applications. *2014 IEEE International Congress on Big Data*.

Appendices

A Expert Interview 1

EO/GIS software engineer

First, could you give me a short introduction to yourself, education background, and work history?

Yeah, I've studied geoinformatics. I've been working for five years at GAF now, and am developing cloud based infrastructure for processing of geodata.

What things do you consider important for a visualization tool used for remote sensing/ geospatial derived data?

They should be easy to use and the data should be useful for the graphs and graphics too.

Do you consider the design style or the capabilities of a visualization tool more important?

Hmm, what do you mean?

So for example would you rather have a front end that maybe looks really nice but can't do as much or something that has more capabilities but doesn't look as good.

Ok, I prefer something that has more capabilities but I think it depends on the use case quite often. Something that is looking good can be more important.

Who will be the end-users of the CLC+ project and any visualizations associated with it?

That's an interesting question. The people who do politics will have it to make decisions. So these are people that are needing information, but the looks may be more important. But yeah, there are also researchers using it that may need more capabilities for extracting data for example.

What are some of the aspects a user would need to see land-cover/land-use changes as a visualization for the CLC+ project?

Uhm, I think what the user wants to see is a chart with the land cover/land use, and the chart with the land use objects. This segmentation. Both of these. And there is the data extracted from the segments which is the important thing, where the user sees something.

Because that's the problem with CLC+ is it is so overwhelming with how much data you have, and what is useful from it.

And the time series for CLC+ is directly where you could see year over year changes for LCLU.

Yeah, I think there was one feature where we wanted to extract NDVI time series, but I think this is something that [redacted] could tell you. I think in the beginning of CLC+ there were big plans for that and then we saw that there was so much data, and it was so overwhelming to extract it and use it. And then with the user, I think that one of the big problems with time series analysis is that you have so much data that nobody can deal with it.

Yeah that makes sense.

So I think during the process of the project there were a lot of changes.

Yeah that makes sense.

And then I think there is also something after the CLC+ that is more focused on time series products.

I see. So we kind of talked about visualizations that are necessary to include on a front-end service for CLC+. I wanted to also ask if any diagrams/graphs add value to some sort of front end service?

Yeah I mean some graphs, and for example if you have a time series and click on one of the land use objects from the segmentation, that you get a time series out of this. I mean this can give you certain information out of the certain field. I think the problem is that if you are a focused user, you aren't looking at one object but bigger issues like how much forest was lost, etc..

So maybe some sort of service where you could choose a land cover and see the percent change over the area you are looking at.

Yeah something like that, and then you should be able to select an area you are interested in too.

Are there any challenges that could arise in connecting a front-end visualization to the databases associated with CLC+?

Yeah I mean it's a lot of data, so I think that can be complicated. One of the plans of the project was that we would have these big data cubes that are really efficient that we could attach to, and extract time series information. I mean, maybe there was a use case which was not foreseen when these cubes were constructed so maybe they are not optimized for certain tasks.

Ah, what do you mean by big data cubes?

The plan is that all of the CLC+ data goes to some kind of hdf5 data type that it will be stored in and this format should be extremely efficient to extract data and it is some kind of table structure.

Ah could you explain that a little bit more.

Yeah so it will be big tables like this and each of the segments will have an id and the geometry stored and then all of these attributes, I don't know, like which part of the segment is covered with water, which part is covered with trees, and so on. These attributes are changing all the time, so it's a little bit complicated. Normally there should also be a time series, like for two years with Sentinel 2 NDVI. I'm not entirely sure if this attribute is still there. That's how this data looks like. In theory it should be extremely easy to make queries based on your geometries which are intersecting your AOI and you can do some visualizations from these geometries.

That could be helpful, if once you select an AOI, if other visualizations like graphs or data load about the data in the AOI load as well.

That's true, yeah.. What is also interesting about CLC+ is that all of the data is stored as polygons. I think you can also get the classification as a raster file. The ultimate goal is the tables though, where all of the classifications are already stored as segments.

Going on to the next questions, do you have any experience with microservice infrastructures in the past?

Yes

Do you think having visualization tools built as a service can provide value to a project?

Yeah as microservices, that helps. The point here is that you have a lot of data and you can scale microservices quite easily. So if you want to do some visualization services with big data these microservices can be scaled and this could be done much easier. Yeah all in all it can make the code more reusable so it runs for other projects.

So something like this, aside from scaling, it's good to know the ideas could be used in other projects too. We kind of talked about this one already, but if you have anything else to add - In your eyes, how can visualizations of remote sensing derived time-series be integrated into a microservice architecture?

You can have the code which is doing the visualizations on the fly in the microservice.

So more efficiency, maybe?

Yeah, not necessarily though. Efficient code can also be running without a microservice, it's just running on a machine.

Could you think of any challenges one faces in creating a visualization as a service or maybe any drawbacks?

Sometimes the non-microservice architecture is more efficient because it runs on the same computer. And especially with big data you really have to think about how you distribute your code. If you want your code really close to the data, or if you want to copy the data somewhere to your microservice. That can be a problem with microservices.

That makes sense. I guess then on the contrary, do you think there are any advantages instead of using a server.

Yeah, I think scaling. Reusability of code, among others

What are some things you consider "best-practice" when involving a front-end component?

Hmm ok that's also difficult. (laughing)

Yeah, so what are some things you would think are essential when involved in a front end.

Hmm you mean from a programmer point of view, or a user.

Yeah maybe from your perspective.

You so you use git and it's a good idea for the front and backend to have mockups so that you can develop independently from each other. That's the idea with microservices...yeah and so if you have a few more people involved you can make it so that some people create the front end and the back end so that the development is not so loaded.

What languages do you think are good to use right now that could be easy to adapt for microservices?

I think for front ends there is node.js which is used widely with the OpenLayers library. It's is a good tool to actually visualize the maps you want to create. But I am more from the data perspective so we are normally using python or r for the visualizations. It's quite easy to produce more complicated graphics. I think something like tables or some simple graphs you can visualize with some front end languages. But I can imagine if you want to do more complicated graphs or diagrams then it's a good idea to use one of these languages. They are easy to adapt but also more professional and better for production than if you use c++ or java.

Ok that's good to know all of the perspectives helps too. And then one last question, it's kind of a two part question - What trends do you see coming along in the fields of visualizing remote sensing data, and geodata visualization?

Ok, so the projects that we've had lately are ones where the user has wanted to be integrated. So that the user can also be active, and getting the full picture, and getting the full capabilities where they can choose the time frame for example. The users also get more informed so they can bring in their own ideas.

Yeah that makes sense. So thanks again for everything, that was my final question. Feel free to reach out again if you have any more final thoughts. It's been really helpful.

Thanks, yeah I would be interested to see the result.

Yeah, gladly, I'll send it out once we are completed.

B Expert Interview 2

Raster product project manager

First Could you give me a short introduction to yourself, education background, and work history?

Yeah, I'm [redacted]. I've been working at GAF for 3.5 years. I originally studied physical geography and geoinformatics in Dresden, Germany. I did my masters thesis on remote sensing of landslides. I did a PhD then in Enschede and Strasbourg, France on the remote sensing of Landslides, image correlation, image classification, change detection, and so forth. A few post-docs and then straight into industry.

So who will be the end-users of the CLC+ project and any visualizations associated with it?

So I think one end user is the European Environmental Agency. And what they would use it for would be to support the downstream products like LULC CO2 footprint. So there is an obligation for reporting by all those who signed the Paris agreement. So further, they want to derive legacy products. So to explain it, there is currently Corine, which is one of the main land cover products and there is a time series dating back to I think '96. And this is all done manually by the member states. They employ people who sit down and interpret images to update these datasets every six years. The idea is to get to an automation of updating these maps so the raster product would be one input, thus providing the land cover. Land use should still come from the member states since it is not really possible to mine land use from remote sensing images alone.

Do you think the design style or the capabilities of a visualization tool are more important?

Oof I don't know. Like the public that you would reach with the land cover products are not like the tik tok community so it's typically expert users. So it should look nice, but I guess the information content that you get is probably and that it works stable and fluid is probably more important than having the nicest possible design if you have to make a compromise.

What visualizations are necessary to include on a web-application that visualizes the raster data and allows the user to access the product?

Yeah, I mean to be frank it's a fairly simple product. It's a geotiff with 11 different codes and considering this I think there is only so much you can do with it. And also the richness of the information is limited. I think what might be more interesting, is that we have a vector product, which is polygons then. And with this there are 18 classes of attributes also attached to that like building height, composition of the land cover, then other attributes from other Copernicus products like phenology or tree cover density. I think that's much richer in terms of information content and also then probably something that is, let say, more options to play a bit. Like okay what could be a meaningful visualization for the user and how to look at this type of product from different angles. And I mean also, the raster product which will be an end product is mono-temporal and when you say a visualization of time series, which will be one of your topics.

Yeah.

So for the moment it will be just one single timestamp, and you are using time series as input data, but there will be one set of reference data from 2018.

Oh okay, will the rasters be continually updating, for example each year with new datasets.

Yeah, well that's the plan in follow up projects every three years. Eventually with yearly updates, but with the scope of the project it may not be coming any time soon.

Okay yeah, I can imagine it is such a big project and new updates will take time to materialize. So do you think any sort of diagrams/graphs, using any sort of different attributes from the vector classes could add value to visualizations.

Yeah, I guess if you could sort of click on the polygons in some sort of map, depending on where you are, getting something even as simply as a pie chart showing the composition of the land cover coming from the raster product. Or showing the phenology values. I don't know if maybe the phenology values are maybe time series coming from another Copernicus service. If it's time series then maybe that could be another thing that could be visualized.

Yeah that makes sense, I think that would be good to consider.

And I mean we have EO data time series which we use in production. It's not foreseen to attach this data to the polygons or the raster. But we use it in production for interpretation. For example we are looking at the interpreted samples, looking at NDVI time series to see the behavior of the sample in time. I mean even though this is not part of the product the data is there and for visualization/testing purposes it might still be interesting to play around with it and see okay yeah I click at that polygon or that pixel and I get a time series for the different bands for example.

Yeah, especially just seeing how to play out these visualizations as a service, for example here in CLC+, it could be reused later in the future. Could you talk a little about the data?

Yeah for the raster classification data they use Sentinel 2 time series, and they are being pre-processed on the cloud and then interpolated to equidistant time steps and we download it. So we have a bunch of data lying around, and then we have 10 x 10 km patches. And there is a time series for every pixel and every band.

Could you imagine challenges that could arise in an end product visualization for a project which deals with this sort of data?

Yeah, I think if you think of the vector product this is really like millions and millions of polygons in the end. In a usual GIS it's a bit clunky to work with it. Just to visualize it, just a fraction of it. Out of 3 million polygons you pan, wait for it to load, pan and wait. I guess to have something that you want to visualize in the browser, that's your goal right, to have some sort of web front end?

Yep, yeah exactly.

Yeah I could imagine doing that in the browser efficiently is a challenge just due to the sheer size of the data.

Yeah I was thinking for the first prototype just to do a first attempt at a visualization and storing the data in sort of server and then pulling in the different vector and raster data to visualize it and then building other visualizations of off that so that users could have access to the data and see other graphs and diagrams as well. But as you mentioned the *size of the data will definitely be an issue. So for the CLC+ project, the end goal you said will be an 11 class raster.*

Mmhhm

So I'll start closing up, what trends do you see coming along in the fields of visualizing remote sensing data and geodata ?

Well I guess it's getting more democratic and more accessible with open source tools becoming available. Maybe also shifting a bit from the classical front end of GIS GUI to doing stuff more in python doing stuff with an API pulling in stuff and doing dynamic plots with geopandas for example. These are kind of my perception because I like the open source tools quite a bit.

Yeah I think it's definitely really interesting, especially to see how much people are embracing the open source mentality, especially for the future. Yeah, so thanks again for all of the information. If you have any more tips or things you'd like to say in closing feel free.

Yeah, so I guess time series are getting more and more important as we get more of this data and we need to use it so, let's say, I can't speak much for the end user. I probably have a bit of a lack of perspective for an end user, but for production it's really important to have these time series and we really helped ourselves by plotting the NDVI profiles or plotting the normalized density snow index to see if that's permanent snow cover or at least permanent enough to classify it as such. Or how much time of the year is the surface of the earth covered by water. Also there's one app I'd like to mention which you might like to see. So there is a guy from Google Engine who created a prototype, where you could see image chips. You just mark and AOI and within a few seconds you have the image chips for the whole area with various indexes and views. Things like that if you have such tools for an interpreter to be able to pull all of the data and save it visualized in a nice way where you have it appear like that with the click of a button. It makes a difference there.

Yeah, that would be great. If you by chance have it, that would be great to see.

Yes, no problem.

Well thanks again for all the information, it will be a great help.

Yeah, I'd be happy to see the final product when you are finished.

I can definitely send you a link. Well, have a nice rest of your day and I'll get back to you when it's finished.

Thanks, you too. I'm looking forward to it.

C Expert Interview 3

Web GIS front-end developer

Thanks again for taking your time today to help.

You're welcome.

So I'm just going to start getting into the questions. This is a semi-structured interview so I have a few set questions, but we can talk about any other topics that come up along the way. Especially as a front end developer, your opinions on things related to visualizations and maybe some best practices related to creating them.

Ok

So maybe you could start with a short introduction to yourself, education background, and what you do at GAF.

Sure, so I've been studying, I don't know the term in English - geoinformatik - in my master's and before that I didn't do anything related to it, it was just geo-science. Five years ago I joined GAF directly as a front end developer. At that point we used the framework Angular.js in the beginning for front end development with OpenLayers library which is a good tool to actually visualize maps. Then I think after maybe one year we switched to a new version of Angular which was a big hassle because they redesigned everything because we had to switch from JavaScript to TypeScript. But what we are basically doing in our project is visualizing geodata with a time series. But we are mostly showing index data, like NDVI, or specific areas of interest which you can select from drop-downs so you can filter where you want to see. Also things like specific indices, maybe I can also show it afterwards if you'd like to see.

Yeah, I think that would be helpful, because the data for my case study is similar to what you're mentioning, and the end-users for the CLC+ project could benefit from having different indices available like what you're mentioning. So when you're creating the front end visualizations. You said you do a lot of geodata things, does that include earth observation remote sensing things?

Yeah, maybe I could share my screen and show you directly.

Yeah, that might help a lot.

Okay so this is now our current project. It is used for a project where the end product is a front end visualization like this. For example, you are able to select various states from the country and it zooms to the respective state and highlights it. It can also go down to further administrative units if you are interested in it. There is also a data layer drop down which has various EO data. Some are pre-processed and some are derived directly from Sentinel Hub. We have specific stylings configured in our sentinel hub account which results in an image which shows the NDVI index.

Ah so this is how you are able to create the time series slider below.

Yeah, exactly. So it is a little bit complicated because first you have to somehow collect all of the scenes, because sometimes they are overlapping and 3 or 4 times a day. So we have configured our account so that we can optimize which images we get for each day. And we also have this time slider, as we call it, which is basically an automated slider which will automatically fetch the image of the day which is on the slider. So you can also have a small "film" running to see the time series in action.

Sure, I see.

So this is also written with Angular.js, but Angular which is basically the same thing just the newer version of it. And all of the map stuff you see here is from OpenLayers. You can integrate it into TypeScript without any issues.

Is OpenLayers from OpenStreetMaps, like OSM.

Not necessarily, they provide OSM data for example as various layers for example. Here on OpenLayers.org you can get an impression of what they are doing. So it is basically a library to create virtual interactable maps like these. I always use this page which is basically the API so you can search for the things you might use in your application. There is also another library called Leaflet. I've never used it, so I can't give an impression on which is better for which use case, but we have always been happy with OpenLayers.

Sure, this has been able to meet your needs for all of your projects.

I think it even provides a lot more than we have actually used up to now and it is constantly maintained so it is very up to date to the needs of the community.

So with the TypeScript, if someone has experience with JavaScript would it be easy to make the transition to TypeScript.

So TypeScript is basically more object oriented than JavaScript and as the name implies typed. So if you have experience with object oriented programming you should be pretty comfortable with it. I think it also makes things easier because with JavaScript it's always a little bit difficult because you can pass in anything and have to check it afterwards if the object is the correct type, and these things get easier when you are using TypeScript. And if you use the angular framework, everything above version two you are required to use TypeScript.

Would it be possible to view the code for something like this.

Yeah of course, let me find something that would be a good example...So this is basically a template for the map layout. The mat-stuff is something related to angular. It's called angular material. It's a library of components which are pre-built that you could customize. The first bit is what draws the side panel in the app. It's similar to the concept of bootstrapping, which is similar to how Angular works. Yeah so here we have all of the different components which are inside the layout. For example the side bar, infor bar, nav bar, and the actual map. This next part is a TypeScript class which describes the map component. There is a specific way to set up the class. There are specific lifecycle hooks in Angular. Things like ngafterhooks means that this section only triggers once all of the other elements have already loaded. This means that all of the other elements are already existing in the DOM and you can act on it. There is also a hook called ngonit which is called when this class has been initialized but maybe not everything has already been rendered on the map. There is always a problem on the front end of asynchronous behavior that sometimes when you fetch something from the server which will then only appears with a delay, then this will take care of that. So then if you have this ngonit hook and call something, like getting an element on the screen by its id, maybe the element hasn't been rendered yet at this point in time. That's why you would have to use afterviewinit so you could be sure that all elements are there.

Sure, yeah. Could you go to the part where the data is called from the server so I could get an idea of what that looks like?

Yes.... So here we have a generic dataservice. It is basically a very generic getrequest which fetches anything from the server. In our application we have something called a geoproxy in the backend which just delegates some of the requests we send to an external source to get EO data to our backend. Apart from that, OpenLayers will actually take care of that. So for example if you start the time slider, it would constantly request data from that server which would appear here (geoproxy). So this is the request url which goes through our backend. We have just appended all of the request parameters that we need, in this case a WMS, a web map service. In this case we also pass in the polygon geometry of the unit selected in the drop downs. We also pass the time stamp of the date on the time slider and then it would return the image which is then shown on the map.

Sure, so for each time stamp it's a new request.

Yeah exactly, for each time stamp.

Do you think it would be possible to do a custom AOI as the geometry parameters in the requesturl? For example if a user wants to look in very specific parts of the data that they define themselves?

Yes, I can show you something. We have a tool where you can actually digitize a polygon on the map where you can define your area of interest and it will be saved in a table. In that case you already have some, and you can just use those. With Sentinel Hub it is easy to send a geometry and get back a clipped image for that. If I run this it takes some time because it is fetching a lot of time stamps.

Ah wow.

It should load at some point, ah yeah. We also have in addition some statistics data here as a graph. It shows the minimum, maximum and mean NDVI values throughout a year. The same thing we saw before with the min units, and here it is also possible to create your own geometry of any kind on the map and retrieve the data for that area.

Ah, okay. So could one say users are wanting more interactivity in the final product.

Yeah. Some projects in the field are more desirable to have a finished product where their user feels in control of the project and the end product is more "theirs."

So who would you generally say are the end-users of a project like this?

I mean I guess the typical user will still be an expert.

Thanks so far this has been so helpful. Just looking back to some of the questions I have, there's just some more general questions here I've been asking in the other interviews too. Are there any challenges you have when creating time series such as this. For example with this application, were there any challenges that came up?

It was actually quite difficult to get to the data that you want to display because there are several steps from actually the raw data from Sentinel Hub server to what we want to show here. Similar to what I showed you with the filtering earlier. So we also needed to transform it so that it could be displayed on the time slider here. We also spent quite some time designing this time scale because we have different types of data. There is the data from sentinel, but also Modis data, and chirps which is precipitation data. With the different sources we had to put them together in a convenient way so that we could display them together on this time slider. So there was quite a bit of thinking and transformation involved to get it prepared for a good result for the UI at the end.

Are there any things you think could add value to the app which are not included yet.

We are currently in the process of restyling it. I think there are a lot of things you could add (laughs). One thing I think we could see improved on is the crowdedness of the app. When there are so many panels and graphs open the map becomes hidden a bit. So it is good to have a concept on how the elements of the UI will behave in specific situations so that the users have minimal issues in different types of use cases. For example one view where you are just looking at the map and maybe one view where you are just looking at the tabular and plot data so it is not squashed together.

Well thank you this has been so helpful, I think I'll be able to use a lot of this information in the prototype I'm developing. Thanks again for your time.

Ah, I'm glad to help.

Have a nice afternoon.

D Expert Interview 4

GIS Emergency Dashboard Management Expert

Could you give me a short introduction to yourself, your work history, and what you do now?

Yeah, so as I said, I am working now. I studied cartography at the University of Applied Sciences in Munich. I finished my studies in 2004. From 2005 until January of this year, I was working at GAF in the image processing department. The last 8 years, since 2012 I was working on the Copernicus Emergency Management Project. The last position was as the project leader at GAF. Since February of this year I've been working at the Fire Department in Munich. At the moment the focus is on the virus situation here in Munich.

So are a lot of the tasks in GAF, and now based on visualizing the data?

Yeah.

Ah ok, do you have any experience in creating frontend visualizations for websites from these remote sensing products.

Yeah, that's true. It is mostly working with remote sensing data, optical data, as well as radar data. Also with elevation models. Combination of different models, comparing models, and so on.

Ah I see, that sounds interesting. So how do you present the data when you create these different visualizations?

It is generally the standard software. When I was at GAF it was mostly with ArcGIS, here at the fire department it is ArcGIS Pro, ArcGIS online tools, dashboards and so on.

Oh cool, so when you are doing these visualizations what sorts of things do you consider to be important to include when you are working with remote sensing data like this.

Yeah so at GAF we were working with emergency management so it was important to get post event and pre event data very fast just to get the information about what has happened. At the moment, the focus is mainly on statistical data. It is not working with geodata as much, but is mainly with statistical data on the current covid-19 situation here in Munich.

Ah sure, so especially with the Covid data, are you guys creating any sort of time series with the data?

Yeah, the focus is on time-related data, and how things are relevant over time but not in combination with geo-information so we are not presenting the situation by administrative boundaries. It's more statistical data.

So when you create the time-series visuals do you consider things such as how would you want people to read them? And are there any other things you consider?

Sorry could you repeat that?

Yeah of course. So when you are creating the visualizations, especially with time series related information, is there anything you think about as you are creating so maybe the user or the person who is reading the data can understand it clearly.

Yeah, it really depends on the layout, let's say layout, of the dashboard for example. You can make a histogram situation showing like a "cake" for example. It really depends on the data though. Normally we are working with diagrams and using a slider you can scale it somehow and see the time changes over time.

Yeah that's similar to the idea I'm working on creating with this project is to create a time slider to see some visual data and then hopefully build out some other features to see more graphics if there is time at the end.

Ah yeah. So in the Copernicus EMS, we are working with some kind of time series because there is often a demand for monitoring different situations such as how water levels have changed during a flood situation. During a fire, because it is highly dynamic it is important to know how things have changed over time.

So when you do these time-series, are most of the tools to see the changes over time built around sliders? My question is if there are any other methods you've seen employed to help visualize these.

Yeah, so there is a very interesting tool, I don't know if you know it. In ArcGIS online there are story maps that can help show time series and changes over time that can help show changes over time.

Ah that's an interesting point. I have some experience with story maps but never considered it for this project. I can look more into that. So when you do these projects such as Copernicus for example, are there any challenges that come up which present difficulties when presenting time related data.

Yeah, definitely. So we are producing static maps just for printing and so the problem was how can we visualize several time steps, With flood areas for example, a combination of four or five situations in one area can be hard to visualize in one map.

So in this situation would you consider using layers, or something like a story map to scroll through and see the different changes.

Yeah, it would be easier with some kind of story map for sure because you can scale it over time. With printing it becomes very difficult to view changes when you have 3 or 4 different layers.

Sure, I can definitely understand. So I'm kind of changing subjects here, but when you were involved with any of these projects, were there any web-front ends involved? Or was it mostly services such as ArcGIS and built in software to help present the data.

We haven't before at GAF or here at the fire department, we are focusing mainly on ESRI products.

Oh sure, yeah. So when you are in the process of creating the visualizations, like the different graphs and maps for example, how do you consider the end needs of the user or the reader in the development of the visuals?

So for sure we are considering the customer in the end. Right now it is our colleagues (laughing), so there is a very close connection on how we decide to present it to them.

Yeah, in that situation it is a lot easier. You don't have to produce it for the public in the end.

Yeah that is true. The data we are producing is not for the public, it is a very close discussion with our colleagues on how we can present the data.

So when you are doing these do you think it is more important to have better design, or more capabilities to present the data?

It is more important to get the best data instead of visualizations.

Ok, great. I think that's mostly the question associated with the core work I'm doing. But just curious if you see any trends coming in visualization of remote sensing data or geodata in general.

Yeah, so a very good example is the ESRI products that have had a huge change. A few years ago we were only working with standard GIS software. Since then we have had the opportunity to use so many of the online tools like ArcGIS Online. There are many different visualization tools like the story map, like a dashboard for example. Other things like 2D and 3D visualizations have been really helpful.

Yeah, I've had the opportunity to use them and it's pretty great to see the possibilities. Okay great, thank you for your time. It really helps with my project. I'll let you know when I finish with the prototype if you'd like to test it out.

Yeah, definitely. Thanks, have a good day.

Thanks, you too.

E User Study Survey

Land Cover/Land Use Time Series User Study

Hello and thanks for helping and taking the time to participate in this study!

The answers submitted here will help aid the research for the M.Sc. thesis “Testing approaches to visualize land cover/land use changes in time series with Web GIS microservice tools” at the Technical University of Munich and GAF AG.

The survey presents a real world example of changes to the landscape over time, then will ask you to perform a task using the web tool built for this research. There will then be a section with questions aimed to help understand your impression of the application.

The survey should take 15-25 minutes and your participation is entirely voluntary. Your responses are 100% anonymous and the information collected will only be used for the M.Sc. thesis outlined above. Raw data will not be passed onto third parties. Should you wish to withdraw from the study at any moment, feel free to contact me.

If you have any questions, please do not hesitate to contact me at jesse.friend@tum.de.

* Required

1. By clicking “I agree” below you are indicating that you are at least 18 years old, have read and understood this consent form, and agree to participate in this research study. *

Mark only one oval.

- ◯ I agree
- ◯ No, I do not agree

Background info

2. I interact with data visualizations such as maps, graphs, plots, tables, etc.

Mark only one oval.

- Never
- A few times a year
- A few times a month
- A few times a week
- Everyday

3. My age

Mark only one oval.

- 18-25
- 26-35
- 36-45
- 46-55
- 56-65
- 65+

4. How comfortable are you with geospatial data?

Mark only one oval.

- Never use it
- Only use it in apps such as Google Maps
- Hobbyist
- I'm a geospatial professional

The role of land use in preventing climate change

With many current extreme weather events occurring in the last year, the conversation on adapting systems to mitigate climate change has been thrust into the forefront of media and politics again.

The role of land use in preventing climate change

Many countries are focusing on increasing carbon sink potential as a mean of lowering the emission balance. Implementing methods of monitoring these changes to land cover are essential to meet the goals. The European Union is making an ambitious plan to increase the carbon uptake by increasing the forest cover across the entire union.
https://www.euractiv.com/section/climate-environment/news/eu-draft-puts-spotlight-on-improving-carbon-sinks-to-tackle-climate-change/

One method of monitoring these changes over time can be presented as visualizations. The intent of a visualization is to utilize data in formats such as graphs, plots, tables, and maps to display information in easy to digest manners.

Munich, Germany forest change over time

You will be provided a link to an application which helps to visualize land cover for all of Europe. Please follow the instructions below to use the application to gather some information.

1. Open the link below in a web browser
2. Explore the application, it's sidebar, and the built in functions
3. Navigate back to the highlighted area over Munich, Germany
3. Use the built in functionality, legend, or any other tool to attempt to determine percentage of coniferous forests for the highlighted area in 2000, 2006, 2012, and 2018
4. Fill in the percentage for each year in the response options below.
5. Once you are done, feel free to examine the rest of the application.
6. Continue to the next page

Application link
129.187.45.33/TimeServer/userstudy.html

5. Percentage of coniferous forest cover for 2000 *

____

6. Percentage of coniferous forest cover for 2006 *

____

7. Percentage of coniferous forest cover for 2012 *

8. Percentage of coniferous forest cover for 2018 *

Please answer the following questions based on your impression of the application.

A link to the application has been provided again below to help aid in answering the questions

Application link
129.187.45.33/TimeServer/userstudy.html

9. I was able to complete the task with the built in functions of the application *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

10. I understood how to use the application *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

11. I feel comfortable with the accuracy of my answers *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

12. I was able to easily distinguish the changes to coniferous forest cover over time with the application *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

13. I had all of the tools necessary to correctly answer the questions *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

14. I could understand how the forest cover has evolved over time in the study area *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

15. The tools to view changes over time are easily accessible in the application *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

16. The application is intuitive *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

17. I think the application could provide more tools to view changes to the land cover over time *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

18. I would use information from visualizations such this application to help shape my understanding of the changes to our earth *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

19. I can grasp the changes occurring over time to earth when viewing interactive land cover maps such as this prototype *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

20. Using an interactive web visualization has advantages over other formats such as static maps, reports, and other graphics. *

Mark only one oval.

	1	2	3	4	5	
Highly disagree	◯	◯	◯	◯	◯	Highly agree

21. Do you have any other suggestions for possible features? *