INF FAKULTÄT FÜR INFORMATIK

Konzeption und Umsetzung einer mobilen Plattform zur Visualisierung von hochauflösenden 3D-Szenarien im Kontext der Infrastrukturentwicklung

Philipp Schladitz

MASTERARBEIT

eingereicht im Studiengang

Computervisualistik

Betreuer:

Prof. Dr.-Ing. Bernhard Preim

Dipl.-Ing. Nicole Mencke

Oktober 2017

Erklärung

Ich erkläre eidesstattlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen entnommenen Stellen als solche gekennzeichnet habe. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Magdeburg, am 1. Oktober 2017

Philipp Schladitz

Inhaltsverzeichnis

Kurzfassung

Die Planung und Entwicklung von Infrastrukturprojekten ist ein komplexes Themengebiet, wie beispielsweise der Bau des Flughafen Berlin Brandenburg zeigt. Dabei rufen Fehlschläge, hohe Kosten und eine geringe Transparenz einen Unmut in der Bevölkerung hervor, welche sich oftmals digital in sozialen Netzwerken äußert. Die Digitalisierung kann jedoch eine Verbesserung in der Infrastrukturplanung bewirken, wenn aktuelle mobile- und webbasierte Technologien genutzt werden.
Das Ziel dieser Arbeit ist daher die Konzeptionierung eines mobilen Systems zur Visualisierung und Planung von Infrastrukturprojekten. Es wird dabei die Frage aufgeworfen, inwiefern aktuelle Technologien, wie WebGL und glTF die effiziente Darstellung im Web ermöglichen und auf welcher Art und Weise die großen Datenmengen eines geographischen Informationssystems sicher mobil gespeichert werden sollen.
Um diese Frage zu beantworten, wurde ein System konzipiert, welches auf Basis eines Content-Delivery-Networks geographische Daten speichert und effizient an Nutzer verteilt. Zugleich lassen sich die Daten mithilfe einer Webapplikation über ReactJS und CesiumJS anzeigen und verändern. Die konzipierte Plattform wurde auf unterschiedlichen Endgeräten verschiedener Betriebssysteme getestet und mit der Engine Unity verglichen. Die Ergebnisse zeigen, dass die konzipierte Plattform als webbasiertes und mobiles geographisches Informationssystem (GIS) verwendet werden kann. Außerdem bietet es innovative Möglichkeiten zur Bürgerbeteiligung und transparenten Projektplanung durch Integration von Social-Media Features, welche mit einem traditionellen desktopbasierten GIS-System nicht möglich ist.
Auf dieser Grundlage ist es also empfehlenswert für die Planung von Infrastrukturprojekten webbasierte Systeme einzusetzen, wenn das Ziel eine kooperative Zusammenarbeit oder Bürgerbeteiligung darstellt. Eine Planung im Sinne einer CAD-Software mit detaillierten technischen Zeichnungen ist aufgrund der Bandbreitenbegrenzungen nicht zu empfehlen. Für weitere Forschungen ist außerdem die Sicherheit bezüglich der Datenaufbewahrung in einem Content-Delivery-Network zu untersuchen. Zusätzlich sollte eine tiefergehende Untersuchung spezieller Interaktionstechniken für mobile Hardware in der webbasierten Infrastrukturplanung stattfinden.

Abstract

The planning and development of infrastructure projects is a complex subject area, as shown, for example, by the construction of the Berlin Brandenburg Airport. Failures, soaring costs and a lack of transparency bring about feelings of resentment among the public, which often take to social networks to express their opinions. Digitization, however, can lead to an improvement in infrastructure planning if current mobile and web-based technologies are used. The goal of this work is therefore to design a mobile system for visualizing and planning infrastructure projects. This raises two questions: To what extent do current technologies, such as WebGL and glTF, enable efficient presentation on the web and how should the mobile storage of large quantities of data of a geographical information system be carried out securely. To answer these questions, a system was designed that stores geographical data and efficiently distributes it to users on the basis of a content delivery network. At the same time, data can be displayed and modified via ReactJS and CesiumJS using a web application. The platform that was designed was tested on a variety of devices with various operating systems and compared to the Unity engine. The results show that the platform designed can be used as a web-based and mobile geographical information system (GIS). Web-based GIS systems also provide opportunities for public involvement and transparent project planning by integrating social media features, something that is not possible with a traditional desktop-based GIS system. Based on this, it is therefore advisable to use web-based systems for planning infrastructure projects if the objective is cooperative collaboration or public involvement. Planning using CAD software with detailed technical drawings is not recommended due to bandwidth limits. For further research, security with respect to data retention in a content delivery network can be examined. In addition, an in-depth examination of special interactive technologies for mobile hardware in web-based infrastructure planning should be carried out.

Kapitel 1

Einleitung

1.1 Motivation

Zur Sicherstellung unseres Lebensstandards sind gut ausgebaute Infrastrukturen von essentieller Bedeutung. Energie-, Kommunikations- und Verkehrsinfrastrukturen sind kritisch für den wirtschaftlichen Erfolg von Unternehmen. Nach einer Studie des *Instituts der deutschen Wirtschaft Köln* beklagten 64% der insgesamt 3300 befragten Unternehmen im Herbst 2013 Beeinträchtigungen durch Mängel im Straßenverkehr. Bis zu 23% der Unternehmen beklagten sogar deutliche Beeinträchtigungen. Andere Geschäftsbereiche, wie Kommunikationsnetze und die Energieversorgung sind in ähnlichem Maß betroffen [14, S. 18, Abb. 2-5]. Ein beachtlicher Teil der Bundesstraßen wurde im Rahmen der Zustandserfassung und -bewertung der Fahrbahnoberflächen von Straßen als sanierungsbedürftig klassifiziert. 20,9% der Bundesstraßen wurden dabei als dringend reperaturbedürftig eingestuft [14, S. 24, Abb. 3-2]. Im detaillierten Vergleich der Infrastrukturmängel im Straßenverkehr werden vor allem Baubetriebe in den Geschäftsabläufen gestört. Dies lässt sich durch ein hohes Transportaufkommen in diesem Branchenbereich erklären. Der schlechte Zustand der Straßennetze der Landes-, Kreis- und Kommunalstraßen führt zu einer hohen Belastung, sowie erhöhten Stauanfälligkeit in der Nähe von Ballungsgebieten und hat weiterhin Verspätungen bei Anlieferungen in allen Branchenzweigen zur Konsequenz [14, S. 21-22, Abb. 3-5].
Die Studie zeigt deutlich, dass die Planung und langfristige Wartung von Infrastrukturprojekten verbessert werden muss. Die Herausforderungen liegen dabei in einer Modernisierung und Sanierung sowie im Ausbau von Verkehrs-, Strom-, Energie-, Kommunikations- und Breitbandnetzen unter erschwerten Bedingungen durch Unterfinanzierung [14, S. 71, Tabelle 6-2]. Digitale Lösungen können einen wichtigen Schritt zur Lösung des Problems darstellen. Hierzu existieren bereits Werkzeuge, wie virtuelle Stadtmodelle, Virtual Reality Szenarien und Geographische Informationssysteme, wel-

che zumeist nativ über Desktopsysteme ausgeliefert werden. Webbasierte interaktive Systeme bieten eine Vielzahl von Vorteilen gegenüber klassischen Desktop-Programmen. Sie sind plattformunabhängig und bieten seit Einführung des Web 2.0 inherent kollaborative Möglichkeiten [93].

1.2 Zielsetzung

Das Ziel dieser Arbeit ist die Identifikation von Ansätzen zur Realisierung einer webbasierten und mobilen Plattform zur Bereitstellung von hochauflösenden Szenarien zur Planung, Wartung und Unterstützung von Infrastrukturprojekten. Weiterhin ist zu überprüfen, inwieweit mobile 3D-Webanwendungen aktiv zur Lösung der oben genannten Infrastrukturprobleme beitragen können.
Um dieses Ziel zu erreichen, sind Anwendungsszenarien zu beschreiben und anhand dieser Anwendungsszenarien allgemeine Eigenschaften und Kriterien für die Plattform abzuleiten. Diese Analyse ist weiterhin im Hinblick auf mobile Systeme und den limitierten Ressourcen und Datenübertragungsraten im Web durchzuführen. Die aufgestellten Anforderungen bilden die Basis für das Konzept und die dementsprechend entwickelte Architektur. Die Architektur umfasst hierzu das Design einer geeigneten Serverinfrastruktur auf Basis moderner Server- und Clienttechnologien, grundlegenden Interaktionsmöglichkeiten sowie geeigneten Visualisierungsverfahren im Kontext der Infrastrukturentwicklung. Das Konzept soll zusätzlich im Rahmen einer Analyse der prototypischen Umsetzung getestet und ausgewertet werden. Diese Tests umfassen die Performanz, Sicherheit und Stabilität bzw. Vergleiche zu bestehenden Plattformen. Schlussendlich ist ein Ausblick auf Verbesserungsmöglichkeiten zu geben.

1.3 Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in fünf thematische Bereiche: Grundlagen, Verwandte Arbeiten, Konzeptionierung, Implementierung, Evaluation.

Kapitel 2 beschäftigt sich mit zugrundeliegenden Themen der Arbeit. Hier sind Infrastrukturprojekte und Planungsphasen definiert sowie Grundlagen zu wichtigen Themen rund um 3D GIS- und Webanwendungen erläutert.

Kapitel 3 setzt sich mit verwandten Arbeiten und Projekten auseinander, die zur Visualisierung von Infrastrukturprojekten eingesetzt werden.

Kapitel 4 beschreibt das Konzept der Arbeit. Zu Beginn sind drei Anwendungsfälle dargestellt. Daraus lassen sich Anforderungen für ein allgemeines System ableiten. Das Konzept stellt schlussendlich eine webbasierte Archi-

tektur vor, welche mithilfe moderner 3D-, Backend- und Frontendtechnologien die Zielstellung erfüllt.

In Kapitel 5 wird auf Implementierungsdetails des vorgestellten Konzepts eingegangen.

Kapitel 6 präsentiert die Ergebnisse des Anwendungsfallbeispiels und bewertet das Konzept hinsichtlich des Erfolgs. Zum Schluss wird eine Empfehlung im Hinblick auf die Zielstellung und zukünftiger Forschungsthemen gegeben.

Kapitel 7 fasst die Ergebnisse der Arbeit zusammen, geht auf Probleme innerhalb des Konzeptes ein und gibt einen Ausblick auf zukünfigte Fragestellungen anhand der Erkenntnissen der Arbeit.

Kapitel 2

Grundlagen

In diesem Kapitel werden die benötigten Grundlagen der Arbeit geschaffen. Zu Beginn des Kapitels wird der Begriff Infrastrukturentwicklung erörtert. Darunter fallen wichtige Begriffe, wie Planungsphasen, Großprojekte und Bürgerbeteiligung. Nach erfolgreicher Begriffsbestimmung werden anschließend grundlegende und ausgewählte Aspekte von 2D- sowie 3D-Visualisierungen beschrieben. Unter diesen Aspekten fallen Visualisierungsziele, webbasierte Techniken und Interaktionen.

2.1 Infrastrukturentwicklung

Die Infrastrukturentwicklung stellt ein komplexes Themenfeld dar. Um diese Komplexität einzugrenzen ist daher die Klärung des Begriffs Infrastrukturentwicklung und dazugehöriger Assoziationen notwendig. Die Planung von Infrastrukturprojekten ist außerdem eine wichtige Grundlage für das darauf aufbauende Konzept. Schlussendlich wird der Abschnitt mit den Risiken von Infrastrukturprojekten und Problemen bei der Bürgerbeteiligung und öffentlichen Zustimmung abgeschlossen.

2.1.1 Begriffsbestimmung

Der Begriff Infrastruktur (lateinisch *infra* 'unterhalb' und *structura* 'Zusammenfügung' [85]) wird in der Literatur anhand marktwirtschaftlicher Theorien erklärt [15]. Laut *Reimut Jochimsen* bilden Infrastrukturen eine wichtige Voraussetzung für die wirtschaftliche Entwicklung eines Staates. Zu Beginn wurde der Begriff vorrangig vom Militär geprägt [69]. Die Nato verwendete die Begriffe Infrastruktur und Suprastruktur erstmals um Leitungen, wie Pipelines und Kabel, zu bezeichnen [85]. Später bezeichneten Infrastrukturen auch Kasernern und Flugplätze. Heute wird unter dem Begriff Infrastruktur vor allem der 'organisatorische Unterbau einer Wirtschaft' bezeichnet [69] .

"Der Autor [Reimut Jochimsen - Anm. d. Verf.] definiert die In-

> frastruktur als Summe der materiellen, institutionellen und personellen Grundlagen einer Volkswirtschaft, die dazu beitragen ... das höchstmögliche Niveau der Wirtschaftsaktivitäten, zu ermöglichen."[69]

Materielle Infrastrukturen bezeichnen alle physischen Vorrausetzungen einer Volkswirtschaft, welche für die Produktion von Gütern und Dienstleistungen benötigt werden [17, S. 5]. Institutionelle Infrastrukturen beinhalten alle organisatorischen Einrichtungen und Verfahrensweisen. In der Volkswirtschaft setzen diese insitutionellen Infrastrukturen die Rahmenbedingungen für die Wirtschaft. Das bürgerliche Gesetzbuch bezeichnet ein Beispiel einer institutionellen Infrastruktur [17, S. 6]. Personelle Infrastrukturen umfassen alle quantitativen und qualitativen Merkmale eines Menschen die zur Wirtschaft beitragen und in der Volkswirtschaft als *Humankapital* bezeichnet werden. Zur Entwicklung dieser Merkmale gehört die schulische und berufliche Aus- und Weiterbildung. Personelle Infrastrukturen verursachen dementsprechend Investitionskosten von denen zukünftige Erträge erwartet werden [17, S. 6]. In der Abbildung A.1 im Anhang sind Infrastruktur-Sektoren aufgelistet, welche in der Literatur von Autoren benannt werden. Die wirtschaftlichen (materiellen) Infrastrukturen werden unterteilt in Verkehr, Nachrichten (heute auch Telekommunikationsnetzwerke), Energie, Wasser, Kommunale Einrichtungen und Umweltschutz. Soziale Infrastrukturen sind Bildung, Wissenschaft, Gesundheit, Sport u. Erholung, Soziales, Kultur und Wohnungsbau. Institutionelle Infrastrukturen bilden Recht/Ordnung, Verwaltung und Verteidigung. Diese Arbeit legt ihren Fokus auf materielle Infrastrukturen, wobei auf Verkehrs-, Kommunikations- und Energienetze im Rahmen des Konzepts genauer eingegangen wird. Materielle Infrastrukturen werden allgemein durch zwei wesentliche Merkmale gekennzeichnet.

> "Die erste Eigenheit bezieht sich auf die wesentlichen Voraussetzungen menschlichen Lebens. Jene Bedürfnisse der Wirtschaftssubjekte (vgl. Douglas 1998), die sich aus physischen und sozialen Lebenserfordernissen der Menschen ergeben, werden durch Infrastrukturoutputs (Güter und Dienste) befriedigt, die mit Hilfe der zugehörigen immobilen, nicht zirkulierenden, am Boden fixierten Kapitalgüter erzeugt werden; sie heißen materielle Infrastruktur."[69]

Dienste und Güter, welche auch Infrastrukturoutput genannt werden, liefern dementsprechend physische Bedürfnisse, wie Wasser, durch geeignete Kapitalbestände, wie Reservoire's oder Rohrleitungen. Kapitalbestände und deren Wechselwirkung zwischen Bedürfnis und Output sind in der Tabelle A.2 im Anhang aufgelistet. Das zweite wesentliche Merkmal einer materiellen Infrastruktur bezieht sich auf die *Nichtverfügbarkeit der Infrastrukturgüter für*

einzelne Haushalte und Betriebe aus Produktions- und Kostengründen [69]. Mitunter bedeutet dies, dass Infrastrukturen durch unterschiedliche Marktformen versorgt werden. Beispielsweise wird die Stromversorgung durch Monopole bereitgestellt, wohin gegen der Wohnungsbau durch marktwirtschaftlichen Wettbewerb stattfindet. Die Baulast für Bundesfernstraßen liegt auf der Bundesebene, die der Kreisstraßen hingegen beim jeweiligen Landkreis.

2.1.2 Infrastrukturentwicklung im Kontext von Großprojekten

Um die Infrastrukturentwicklung aktiv unterstützen zu können, ist es wichtig zu verstehen, wie Infrastrukturprojekte geplant und ausgeführt werden. Infrastrukturprojekte werden ab einer gewissen Investitionsmenge als Subtyp von *industriellen Großprojekten* verstanden.

> "Der Begriff 'Großprojekt' ist weder in der Planungspraxis, noch in der Fachliteratur und auch nicht von Seiten des Gesetzgebers klar definiert. [...] Einzig das Investitionsvolumen, also die Summe aller für die Umsetzung eines Projekts aufzuwendenden finanziellen Mittel, wird durch den Gesetzgeber als ungefährer Anhaltspunkt für eine Legaldefinition von 'Großprojekten' angeführt. [...] Weiterhin hat der Rat der Europäischen Union in seiner Verordnung 1083/2006 über die Bestimmungen zum Europäischen Fonds für regionale Entwicklung vom 11. Juli 2006 einen aktualisierten Wert angegeben, der auf 50 Millionen Euro (25 Millionen Euro bei Umweltprojekten) festgesetzt ist."[16, S. 21]

Großprojekte können in verschiedenen Kategorien und Mischformen existieren [1, S. 6]:

- *Großveranstaltungen* werden in zeitlichen Abständen in ausgewählten Regionen und Städten durchgeführt. Hierzu zählen beispielsweise internationale Veranstaltungen, wie die Olympischen Spiele, europäische Veranstaltungen, wie die Fußball-Europameisterschaft und auch bundesdeutsche Veranstaltungen, wie die bundesdeutschen Vorlesetage oder Bundesgartenschau. Die Auswirkung von Großveranstaltungen ist von einer langen Vor- und Nachlaufzeit geprägt, sodass die Planung und Organisation dieser Projekte langfristig spürbar ist. [1, S. 6]
- *Flagship-Image-Projekte* haben die Aufgabe den Bekanntheitsgrad einer Stadt oder Region zu erhöhen und das Image aufzuwerten. Ein häufig anzutreffendes Beispiel ist Wolfsburg, welche sich selbst als Autostadt bezeichnet und zur Stadtgründung als Wohnort für die Mitarbeiter des Konzerns VW konzipiert war. [1, S. 6]
- *Urban-Renaissance-Projekte* betreiben die gezielte Aufwertung von Industrie und Stadtanlagen, um das städtische Leben attraktiv zu ma-

chen. Als Beispiel von Urban-Renaissance-Projekten werden Hafenentwicklungsprojekte, wie die HafenCity in Hamburg[53], oder Urban Entertainment Center angegeben [1, 62].

- *Infrastruktur-Großprojekte* besitzen ein Investitionsvolumen von hundert Millionen bis zu mehreren Milliarden Euro. Einen Großteil der Infrastruktur-Großprojekte bilden Flughäfen, Bahnhöfe, Tunnel und Verkehrsprojekte. Projekte, wie Stuttgart-21 oder der Berliner Flughafen BER, werden auch als Flagship-Image oder Urban-Renaissance-Projekte eingestuft, da diese ebenso weitreichende Veränderungen für das Stadtbild und die Einwohner bewirken. [1, S. 6]

Im Fall der Infrastrukturprojekte werden zur Genehmigung Planfeststellungsverfahren durchgeführt. Diese Verfahren dienen der Zulassung von raumbedeutsamen Vorhaben und stellen im Rahmen einer Abwägungsentscheidung fest, ob und wie das beantragte Projekt zugelassen wird. Ein Planfeststellungsverfahren für Infrastrukturmaßnahmen ist in Sachsen-Anhalt vom Landesverwaltungsamt durchzuführen und bezieht sich hierbei vor allem auf Bundesfern- und Landesstraßen, Ausbau von Ortsdurchfahrten, Hochspannungsfreileitungen, Flughäfen und Landeplätzen, Betriebsanlagen nicht bundeseigener Eisenbahnen, Straßenbahnen und im Landkreis inbegriffenen Trassen. Zusätzlich muss das Planfeststellungsverfahren durchgeführt werden, wenn das Vorhaben einer Umweltverträglichkeitsprüfung bedarf, wovon bei Großprojekten auszugehen ist [67]. Ein Infrastrukturprojekt wird in fünf Hauptphasen ausgeführt: Idee, Planung, Vergabe, Erstellung und Nutzung [23, S. 17]. In der Planungsphase sind vier Hauptaufgaben abzuarbeiten (s. Abb. 2.1, [23, S. 17]): die technische Planung, die Baugenehmigung, die Organisation der Finanzierung und die Bauausführung. In der technischen Planung wird eine Lösung in mehreren Detailgraden erarbeitet. Hierbei spielt die Vorplanung eine wesentliche Rolle, da eine gewisse Anzahl an möglichen Lösungen untersucht wird [23, S. 17]. An dieser Stelle können digitale Modelle den Aufwand erheblich senken. Ziel der technischen Planung ist eine technisch machbare Lösung zu finden, welche sowohl genehmigungsfähig, als auch finanzierbar ist. In der Planungsphase sind zusätzlich die Richtlinien und Regelwerke des Bundes und der Länder zu beachten. Am Ende der technischen Planung steht ein Ausschreibungsprozess zur Vergabe des Projektes an ein Bauunternehmen. Änderungen im Verlauf der Bauausführung stellen eine Störung im geplanten Ablauf dar und führen zu zeitlichen Verzögerungen, weshalb diese vermieden werden sollten. [23, S. 17-19] Im besten Fall können digitale Lösungen alle Bereiche der Infrastrukturprojektplanung unterstützen. VR-Szenarien eignen sich vorallem für die Vorplanung, bei der mehrere Planungsvarianten erstellt, untersucht und verglichen werden. Für die Planung werden im Regelfall Ingenierbüros oder Consultants über externe Verträge oder interne Vereinbarungen an Leistungen gebunden. Hierbei werden sowohl Planungs- als auch gutachterliche Leistungen vergeben.

Abbildung 2.1: Aufgaben in der Planungsphase [23, S.26]

In Infrastrukturprojekten kommen spezialisierte Fachplaner zum Einsatz [23, S. 25]. Darunter sind bspw. Verkehrs- und Streckenplaner zur Planung von Streckentiefbau und Entwässerung, oder Tunnelplaner für Böschungen, Einschnitte und Dämme. Die Planung der Ingenieure wird durch Gutachter bewertet, welche eine unabhängige Stellungnahme der Planung vornehmen [23, S. 25]. Die Gutachter prüfen außerdem auf wichtige rechtliche Grundlagen wie Altlasten, Kampfmittel, Lärm und Erschütterungen, sowie Umweltbelastungen. Eine konkrete Darstellung aller Teilphasen bei der Projektabwicklung eines Infrastrukturprojektes ist in Tabelle A.1 im Anhang dargestellt. Variante 1 (V1) bedeutet, dass die Ausführungsplanung in Verantwortung des Auftraggebers erstellt wird. Variante 2 (V2) stellt die Ausführungsplanung in Verantwortung des Auftragnehmers. Die Phasen I bis V beziehen sich auf die Hauptphasen eines Infrastrukturprojektes, welche oben genannt wurden.

Risiken

Jede Errichtung einer neuen Infrastruktur birgt Risiken. Grundsätzlich können diese Risiken anhand von zwei Kategorien ausgemacht werden. Zum einen kann es sich bei Infrastrukturprojekten um *Greenfield Investments* (auch genannt *Primary Projects* oder *Growth Infrastructure*), als auch um *Brownfield Investmens* (auch genannt *Secondary Projects*) handeln. Als Greenfieldprojekte werden bspw. Start-ups bezeichnet - also solche Projekte, die vorher noch nicht existiert haben, wie beispielsweise der Bau einer neuen Schule. Brownfieldprojekte operieren auf bereits bestehenden Infrastrukturen, bspw. das Betreiben einer gebauten gebührenpflichtigen Straße. Die

Risiko	Erklärung
Arbeitsmarktrisiken (z.T. Reputationsrisiken)	Greenfieldprojekte schaffen zwar primär neue Arbeitsplätze, bei Brownfieldprojekten werden aber im Zuge von Rationalisierungen oft auch Arbeitsplätze abgebaut. Schlechtere Arbeitsverträge nach Privatisierung.
Gegenparteirisiken	Ein Vorteil von Infrastruktur sind langfristige Verträge, deshalb bestehen bei Gegenparteirisiken sehr konzentrierte Klumpenrisiken. Aufgrund hoher Investitionsvolumen kann die Anzahl Gegenparteien tief sein.
Geschäftsrisiken	Schlechtes Management Neue Konkurrenz
Konstruktionsrisiken	Konstruktionskosten sind viel höher als budgetiert. Verzögerungen bei der Konstruktion Viele Infrastrukturprojekte sind irreversibel; limitierte Möglichkeit, ein Projekt zu stoppen.
Leverage	Infrastrukturdeals weisen typischerweise einen Fremdfinanzierungsgrad von 30% bis 90% auf. Dadurch werden Investments mit tiefen Risiken eher zu risikoreichen Anlagen. Änderungen im Kreditumfeld (z.B. Zinserhöhungen) erhöhen Refinanzierungsrisiken.
Markteffizienz	Hohe Nachfrage nach Infrastrukturinvestitionen. Sehr viele neue Fonds im Markt auf der Suche nach Projekten. Kompetitive Auktionen: Risiko besteht, dass zu viel gezahlt wird («Winner's Curse»).
Politische Risiken	Änderungen in der Besteuerung Öffentliche Akzeptanz der Privatisierung Politische Landschaft ist je nach Land, Region und Gemeinde sehr unterschiedlich.
Regulatorische Risiken	Konzessionen können zum Nachteil des Investors geändert werden. Der Staat hat einen Einfluss auf die Preissetzung/Einnahmen aus Gebühren.
Systematische Risiken	Abhängig vom Zustand der Wirtschaft analog Aktienrisiken
Umweltrisiken	Katastrophen Verschmutzungen

Abbildung 2.2: Risiken von Infrastrukturanlagen [22, S.4]

Risiken eines Greenfieldprojektes sind weitaus höher, als bei Brownfieldprojekten. Greenfieldprojekte sind bestimmt von tiefen operativen Kosten und bieten erst nach einer langen Laufzeit einen stabilen Kapitalfluss (s. *J-Curve-Effect*). [22, S.3] Abbildung 2.2 fasst die Risiken detailliert zusammen.

Bürgerbeteiligung

Die frühzeitige Beteiligung der Bürger ist bei einem Großprojekt eine wichtige Grundlage, um spätere Kritik oder Proteste zu vermeiden. Nach einer Studie des *Instituts für Demoskopie Allensbach*, bei der 1771 Personen ab 16 Jahren befragt wurden, ruft der Begriff 'Großes Bauprojekt' bei 54% der Beteiligten negative Assoziationen hervor. Die größte Akzeptanz für Großprojekte liegt beim Bau von Energieversorgungsanlagen mit 85%. Nur 15% befürworten den weiteren Ausbau von Kohlekraftwerken. [37, 11] Der Bau von Flughäfen wird mit 77% ebenfalls stark abgelehnt. Die Kontroversen rund um den Flughafen Berlin Brandenburg (BER), welcher zum Oktober 2011 eröffnet werden sollte, jedoch erst 7 Jahre später fertiggestellt wird, zeigen deutlich, wieso eine Beteiligung der Bürger am Planungs- und Bauprozess dringend erforderlich ist. [79] Ein weiteres Beispiel für Akzeptanzprobleme ist die Verzögerung im Bau des Frankfurter Riederwaldtunnels, der eine Verbindung zwischen den Bundesautobahnen A66 und A661 im Osten von Frankfurt am Main herstellen soll. Die ersten Planungen für den Bau

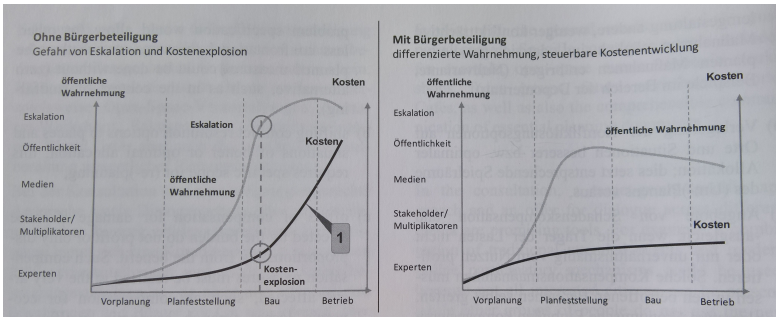


Abbildung 2.3: Korrelation von öffentlicher Wahrnehmung und Projektkosten mit und ohne Bürgerbeteiligung. [10, S.10]

des Tunnels begannen 1989. Jahrelange Proteste und Klagen führten jedoch zu einem Abbruch. Erst im Jahr 2009 wurde das Planfeststellungsverfahren offiziell begonnen, sodass die Fertigstellung zum Jahr 2020 vorgesehen ist [4, S.12]. Um die Bürgerbeteiligung bewusst in das Projektmanagement einzuplanen, wurden vom *Verein Deutscher Ingenieure* Richtlinien konzipiert:

- VDI 7000: *Frühe Öffentlichkeitsbeteiligung für Industrie- und Infrastrukturprojekte*
- VDI 7001: *Kommunikation und Öffentlichkeitsbeteiligung bei Planung und Bau von Infrastrukturprojekten*

Beide Richtlinien geben Empfehlungen zur Nutzung unterschiedlicher Instrumente in der Öffentlichkeitsarbeit und Kommunikation entlang aller Hauptphasen eines Großprojektes (s. Kapitel 2.1.2). VDI 7001 besteht aus zwei großen Abschnitten [10, S.12]:

- Allgemeine Anforderungen an gute Kommunikation und Öffentlichkeitsbeteiligung
- Gute Kommunikation und Öffentlichkeitsbeteiligung in den Leistungsphasen der Ingenieurplanung

Weiterhin sind zwei Prüflisten Teil der Richtlinie, welche als Handlungsrahmen verstanden werden können. Die VDI 7001 hat außerdem zum Ziel, durch ihre Anwendung die Wahrscheinlichkeit von Fehlentscheidungen bei der Planung, Bauausführung, Terminplanung und Kostenkalkulation zu reduzieren. Abb. 2.3 stellt die Korrelation von öffentlicher Wahrnehmung und den Projektkosten in Zusammenhang mit der Bürgerbeteiligung dar. Unzureichende oder keine Bürgerbeteiligung führt vor allem in der Bauphase zu einer Kostenexplosion, da Änderungen im Umsetzungsprozess mit allen Beteiligten (Fachplanern, Gutachtern, Bauherren) abgestimmt werden müssen. [10, S.9-12] Die Richtlinie gibt bei der Vorplanung die Empfehlung eine Stakeholder-Analyse und eine Themenfeldanalyse durchzuführen. Bei der Stakeholder-Analyse wird eine grobe Kategorisierung von drei Typen, die

bei der Öffentlichkeitsarbeit unterschieden werden müssen, angegeben [10, S. 21]:

- Verfasste, dauerhaft organisierte Akteure, wie der BUND, Greenpeace, NABU, Politik und Parlamente
- Direkt betroffene Bürger, Anwohner, lokale Bürgerinitiativen, allgemeine Öffentlichkeit
- Massenmedien, wie Social Media (Facebook, Twitter), Zeitung, Fernsehen

Für diese Stakeholder ist zu ermitteln, welche Positionen sie vertreten, die für das Projekt relevant werden können. Konkreter wird die Empfehlung gegeben, die Hauptargumente der einzelnen Stakeholder zu untersuchen und gemäß des Projektes zu beantworten. Bei der Themenfeldanalyse werden die verschiedenen Aspekte eines Großprojektes systematisiert. Es ist zu erfassen, welche Themen potenziell kommunikative Risiken in sich bergen und welche Themen positiv besetzt sind. Um diese Themen herauszufinden, ist es schon zu Beginn wichtig, dass Sondierungsgespräche und Bürgerdialoge stattfinden, um die Sichtweise der Betroffenen zu verstehen. Beide Analyseverfahren sollten nicht in der Vorplanung enden, sondern während des Projektes fortlaufend angepasst werden. Laut VDI 7001 ist das Kernstück der Themenfeldanalyse eine Themenlandkarte, welche die Aussagen der Stakeholder, Medienberichterstattung und Diskussionen systematisch erfasst und zueinander in Bezug setzt. Abb. 2.4 zeigt ein Beispiel für eine exemplarische Themenlandkarte am Beispiel "Fracking". [10, S.22-23] Fracking ist ein technisches Verfahren zur Förderung von Erdgas [83]. Im Planfeststellungsverfahren wird gesetzlich eine Öffentlichkeitsbeteiligung durch Einsicht in die Planungsunterlagen für einen Zeitraum von einem Monat gewährt. Vom Landesverwaltungsamt wird die Einsicht ortsüblich bekannt gemacht. Grundstückseigentümer, die ihren Wohnsitz außerhalb der Kommune haben, werden darüberhinaus über die Auslegung der Planungsunterlagen schriftlich informiert, sodass Einwende und Stellungnahmen abgegeben werden können. Der Öffentlichkeit wird somit auch im Planfeststellungsverfahren die Möglichkeit zur Mitwirkung gegeben [68]. Im Rahmen dieser Arbeit werden Projekte durch grafische Visualisierungen unterstützt. Zentralisierte 2D-Visualisierungen innerhalb einer Plattform können ebenfalls eingesetzt werden, um die genannten Analysen durchzuführen und für alle Projektbeteiligten eine ergänzende Perspektive zu der 3D-Visualisierung zu bieten.
Die größten Kritikpunkte an 3D-Visualisierungen in der Entwurfsplanung sind laut Prof. Brettschneider die Manipulation durch Schönung von 3D-Visualisierungen beispielsweise durch stimmungsvolle Lichtszenarien, digitalen Hochglanzplänen und Planungsvarianten, die den Erwartungen der Bürger in den Projektphasen nicht entsprechen [37, S.21]. Anhand dieser Kritikpunkte stellt Prof. Brettschneider zusätzliche Regeln auf, damit die Akzeptanz an der Visualisierung steigt [37, S.22]:

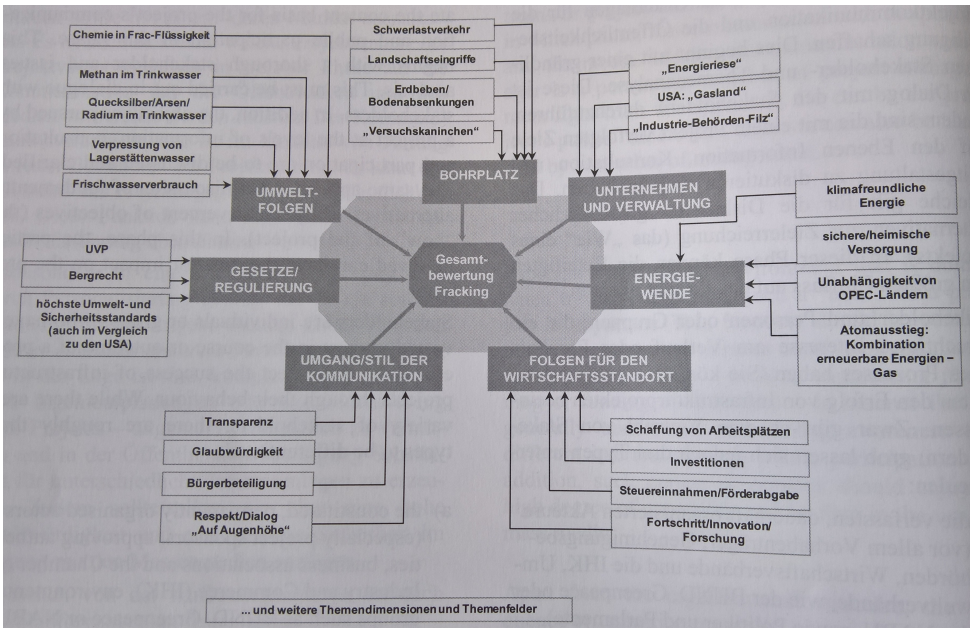


Abbildung 2.4: Exemplarische Themenlandkarte am Beispiel von Fracking [10, S.22]

- Visualisierungen müssen verständlich, d.h. für die relevanten Anspruchsgruppen geeignet sein.
- Sie müssen informativ und an die jeweilige Planungsphase angepasst sein.
- Sie müssen transparent sein, d.h. es muss nachvollziehbar sein, auf welcher Basis sie zustande kommen.
- Sie müssen dynamisch / interaktiv sein, d.h. es müssen Varianten prüfbar und nachvollziehbar sein. Dazu müssen sich die relevanten Parameter interaktiv ändern lassen.
- Sie müssen nicht nur die Baumaßnahme, sondern auch deren Funktionsweise darstellen.
- Sie müssen echt sein, d.h. sie müssen auf maßhaltigen, aktuellen und realen Planungsdaten basieren und die gesamte Planung beinhalten

Die genannten Regeln sind Teil der allgemeinen Visualisierungsaufgaben in der Informationsvisualisierung. Im Rahmen interaktiver Systeme werden im folgenden Kapitel konkrete Umsetzungsmöglichkeiten anhand von grundlegenden Aspekten der Informationsvisualisierung diskutiert. Weiterhin werden Visualisierungsziele und -aufgaben im Kapitel 2.2.1 analysiert.

2.1.3 Virtuelle Städte, Landschaften und Welten

Virtuelle Welten können in einer ganzen Reihe von Anwendungen mit unterschiedlichen Zielen zum Einsatz kommen. Es lassen sich zwei Arten von unterschiedlichen Datenmodellen klassifizieren [9, S.14]:

- Virtual Reality (VR) Modelle, welche durch 3D-Geometrie die Erscheinung der realen Welt repräsentieren.
- Informationsmodelle (IM), welche sowohl die räumlichen, als auch die thematischen Aspekte der Objekte einer Stadt repräsentieren.

Die Visualisierung dieser Modelle ist auf vier unterschiedlichen Herangehensweisen möglich. Methoden und Algorithmen aus dem Bereich der *Computergrafik* zielen auf effiziente und qualitativ hochwerte 3D-Visualisierungen ab. Oftmals sind diese Ansichten reine VR-Modelle zur Präsentation, welche in einem Szenegraphen strukturiert werden. Semantische Informationen stehen hier häufig nicht im Fokus. Eine weitere Herangehensweise nutzt Modelle und Methoden aus dem Bereich der *Simulation und Computerspiele*, welche zusätzlich die Beschreibung physikalischer und kinematischer Eigenschaften für komplexes Objektverhalten in die Visualisierung einbeziehen. Ein grundlegend unterschiedlicher Aspekt der Visualisierung fokussiert sich auf die sogenannte *Bauwerksdatenmodellierung* mithilfe von Computer Aided Architectural Design (CAAD) Anwendungen zur Beschreibung von Gebäuden in einem Maßstab, der es erlaubt ein komplettes Bauvorhaben planen zu können. [9, S.14] Weiterhin existiert der Ansatz der *Geodäsie bzw. Geoinformationssysteme* (kurz GIS), welcher raumbezogene Daten in 2D oder 3D erfasst, organisiert und präsentiert. Diese Daten sind im Gegensatz zu den anderen Ansätzen georeferenziert, sodass Positionen anhand von Koordinaten (Geokodierung) in Bezug auf ein festgelegtes Weltkoordinatensystem konkret bestimmt werden können. [9, S.15] Im Alltagsgebrauch werden GIS-Applikationen vorallem in Navigationssystemen und digitalen Kartendiensten verwendet. Beispiele für bekannte 2D-GIS-Systeme sind Google Maps, Bing Maps und OpenStreetMap [81]. Neben reinen 3D-Modellen (s. Abb. 2.5 für Beispielanwendungen) existieren ebenso Hybrid-Visualisierungen, welche die Vorteile der 3D-Visualisierungen mit den Vorteilen der 2D-Karten verbindet. Die Beschreibung der 3D-Daten ist ein wesentlicher Aspekt der Visualisierung eines virtuellen Modells. Hierzu wurden eine ganze Reihe unterschiedlicher Austauschformate und Datenquellen entwickelt. Im Folgenden werden einige der bekanntesten offenen Formate vorgestellt.

Geography Markup Language

Geography Markup Language, oder auch kurz GML, erweitert die Auszeichnungssprache XML durch Schemen zur Beschreibung von raumbezogenen Objekten. Durch GML können anwendungsbezogene Schemen erstellt werden, welche es Entwicklern erlaubt, auf Straßen und Brücken zu referenzie-

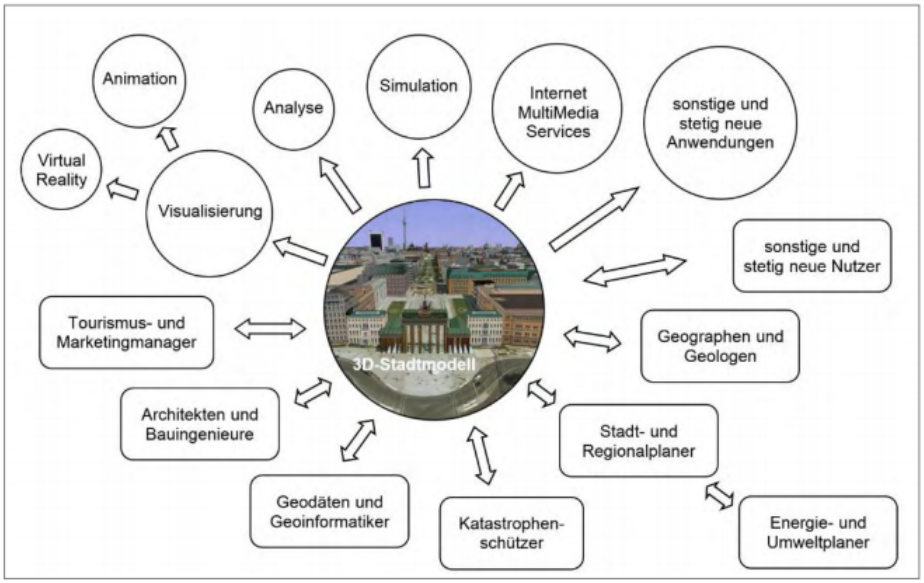


Abbildung 2.5: 3D-Stadtmodelle und ihre Anwendungsfälle [9, S.15]

ren, anstatt auf Linien und Punkte. Zum Austausch von GML-Daten empfiehlt das Open Geospatial Consortium die Implementierung der OpenGIS Web Feature Service Schnittstellen [41].

Keyhole Markup Language

KML, ist ein Dateiformat zur Anzeige von geographischen Daten in einem Browser zur Darstellung der Erde. Entwickelt wurde KML erstmals von Google und fand seinen Einsatz in Google Earth. KML wird hier als Sprache zur Darstellung von Image-Overlays, Layern und Pin-Locations eingesetzt. Standardisiert wurde die Sprache ebenfalls durch das Open Geospatial Consortium. Durch dessen Ähnlichkeit zu GML ist es möglich einfach zwischen beiden Formaten zu konvertieren [42].

GeoJSON

GeoJSON wurde in Anlehnung der Simple Feature Access Spezifikation des Open Geospatial Consortiums entwickelt. GeoJSON beschreibt einfache Strukturen, wie Punkte, Linien und Polygone, sowie mehrteilige Geometrien. Weiterhin ist GeoJSON als Austauschformat gedacht, da es auf der Javascript-Object-Notation (JSON) basiert. Dadurch ist es möglich, dass javascript-basierte Server mit anderen javascript-basierten Anwendungen unkompliziert geographische Daten austauschen können [46].

glTF

glTF ist ein modernes Austauschformat für 3D-Anwendungen, welches nicht auf geographische Daten beschränkt ist. Entwickelt wird glTF (GL Transmission Format) von der KhronosGroup, welche auch für OpenGL, Vulkan und WebGL verantwortlich ist. glTF basiert, wie auch GeoJSON, auf dem JSON Dateiformat - benötigt jedoch zusätzliche externe Daten, wie Binärdaten für Geometrien und Animationen oder Bilddaten, wie jpg oder png für Texturen. glTF wurde mit dem Ziel entwickelt, dass das Format möglichst klein ist, sich schnell laden lässt und laufzeitunabhängig von der Zielapplikation ist. Außerdem soll eine komplette 3D-Szene mit Erweiterungen durch glTF beschreibbar sein. Nachteil des Formats im Vergleich zu Formaten, wie GeoJSON ist, dass das Format nicht auf menschliche Lesbarkeit ausgelegt ist [51].

Digitale Geländemodelle

DGM- und DOP-Daten sind landestypische Datenmodelle, die vorrangig vom Landesamt für Vermessung und Geoinformation bereitgestellt werden. DGM-Dateien beschreiben ein "digitales Gelände- und Höhenmodell"mit einer standardmäßigen Gitterweite von 1m. DGM's können in allerhand von Ausgabeformaten ausgetauscht werden: ASCII (xyz Liste der Gitterpunkte), WINPUT, SCOP-RDH, DXF. Weiterhin werden digitale Geländemodelle vom Landesamt für Vermessung im ETRS89/UTM32 Format bereitgestellt. Die Konvertierung zu anderen Systemen ist jederzeit möglich. DOP-Dateien sind das Pendant der DGM-Daten, um georeferenzierte Texturen über das Geländemodell zu spannen. DOP ist eine Kurzfassung für 'Digitales Ortofoto' und wird ebenfalls vom Landesamt für Vermessung bereitgestellt. Diese Luftbildaufnahmen werden mithilfe von zentralperspektivischen senkrechten Kameras aufgenommen, weshalb sie entzerrt werden müssen, um eine korrekte Repräsentation im digitalen 3D-Raum zu bieten. Diese Entzerrung führt zu einer orthografischen Projektion und gibt dem Luftbild den Namen Orthofoto [72, 73].

2.2 Ausgewählte Aspekte webbasierter Visualisierungen

Dieser Abschnitt untersucht grundlegende Aufgaben und Techniken der Visualisierung multivariater Daten. Außerdem wird neben den Visualisierungsaufgaben zur Förderung des Erkenntnisgewinns eines Nutzers ein ausgewählter Überblick über Interaktionstechniken im 3D-Raum gegeben.

2.2.1 Visualisierungsaufgaben

Die Aufgabe der Visualisierung ist es den Erkenntnisgewinn des Nutzers zu unterstützen und zu fördern. Nutzer können anhand von visualisierten Datensätzen quantitative Aussagen über das visualisierte Themenfeld treffen, welche in tabellarischer oder textbasierter Form nicht möglich sind. Die Ziele des Erkenntnisgewinns können mannigfaltig sein [20, S. 442]:

- das Entdecken von neuen Zusammenhängen oder Besonderheiten (discovery)
- das zuverlässige Treffen von Entscheidungen (decision making)
- die explorative Analyse von Informationsräumen (exploration)
- das Finden von Erklärungen für Muster, Gruppen von Datenobjekten oder einzelne Eigenschaften (explanation)

Zusätzlich können die Ziele in drei Ebenen abgestuft werden [20, S. 442-443]:

- Explorative Analyse: interaktive, ungerichtete Suche nach Informationen, Hypothesen über vorhandene Daten und Strukturen existieren noch nicht.
- Konfirmative Analyse: zielgerichtete Suche nach Informationen, um Annahmen zu bestätigen bzw. Erklärungen zu finden.
- Präsentation: Fakten erkennbar darstellen, damit Dritte sie problemlos identifizieren und verstehen können.

Unabhängig der vom Kontext gesteckten Ziele müssen grundlegende Visualisierungsaufgaben unterstützt werden. Diese Aufgaben wurden im Kapitel 2.1.2 bereits kurz angeschnitten. Eine konkrete Taxonomie von Shneiderman beschreibt die Aufgaben der Visualisierung als Lösung um die gesteckten Ziele des Nutzers zu erreichen [20, S. 443]:

- Overview: Gewinn eines Überblicks über den gesamten Informationsraum, Erkennen von globalen Mustern und Trends.
- Zoom: Heranzoomen von interessanten Informationsobjekten. Betrachten einer kleineren Untermenge der Daten.
- Filter: Herausfiltern von uninteressanten Datenobjekten, Auswahl einer Untermenge von Attributen.
- Details-on-Demand: Auswahl eines Datenobjekts oder einer Gruppe von Daten, um Details zu erhalten, Anzeige von Attributen von Datenobjekten nach Auswahl.
- Relate: Betrachtung von Beziehungen zwischen Datenobjekten, Vergleich von Werten.
- History: Aufzeichnung der Aktionen, um sie rückgängig machen zu können.
- Extract: Extraktion von Untermengen des Informationsraums und der Anfrageparameter.

2.2.2 Visualisierungstechniken

Zur Visualisierung von Datensätzen muss die jeweilige Dimension des Datensatzes beachtet werden. Die Visualisierung von 3-dimensionalen Daten unterscheidet sich inherent von der Visualisierung von univariaten Daten. Im Folgenden werden für verschiedene Datentypen Visualisierungstechniken vorgestellt.

Univariate Daten

Eindimensionale Daten werden auch univariate Daten genannt. Zeitabhängige Variablen sind ein typisches Beispiel, wie die Anzahl eintreffender Emails am Tag [20, S. 450]. Die Visualisierung univariater Daten durch rein textuelle Anzeige kann deutlich schwerer zu erkennen sein, als eine visuelle Präsentation. Typisch sind Visualisierungen durch Balkendiagramme in Relation zu Minimum und Maximum, Kreisdiagramme, Histogramme, oder Streudiagramme. Es werden zusätzlich kreisförmige Anzeigen durch Tachometer verwendet. Bei der Visualisierung von univariaten Daten ist zudem oftmals nicht der präzise Wert von Interesse, sondern dessen Einordnung in einen festgelegten Wertebereich (bspw. die Geschwindigkeitsanzeige bei einem Auto). Zusätzlich zu den klassischen Techniken werden auch Tag-Clouds, oder Word-Clouds genutzt, um eine Anzeige der wichtigsten Daten durch Größe und auffällige Farbe abzubilden. Ein Beispiel für eine Word-Cloud ist in Abb. A.2 zu sehen [20, S. 454-455].

Bivariate Daten

Zweidimensionale Daten, oder auch bivariate Daten, unterscheiden zwei Dimensionen eines Datensatzes. Beispielsweise können zwei ausgewählte Attribute eines Produktes als Dimensionen bezeichnet werden. Landkarten stellen hierbei eine wichtige Spezialisierungsform dar. [20, S. 450] Visualisierungen durch 2D-Streudiagramme (two-axis Scatterplot) tragen für jedes Attribut den skalierten Wertebereich auf einer der senkrecht aufeinander stehenden Achsen ab. Ein Punkt im Diagramm visualisiert ein konkretes Datenobjekt, s. Beispiel in Abb. A.3. Üblicherweise werden die Anzahl der Datensätze pro Punkt durch Größe und Form kodiert, um Überdeckungen mehrerer Datenobjekte auf einer Koordinate zu verhindern. [20, S. 455]

Multivariate Daten

Multidimensionale Daten, oder auch multivariate Daten, sind Datensätze mit mehr als zwei Dimensionen. Relationale und statistische Datenbanken speichern multidimensionale Datenobjekte, wobei n-Attribute einen n-dimensionalen Raum bilden. Bei der Speicherung in Form von Tabellen wird auch von einer relationalen Strukturierung der Daten gesprochen.[20, S. 450-

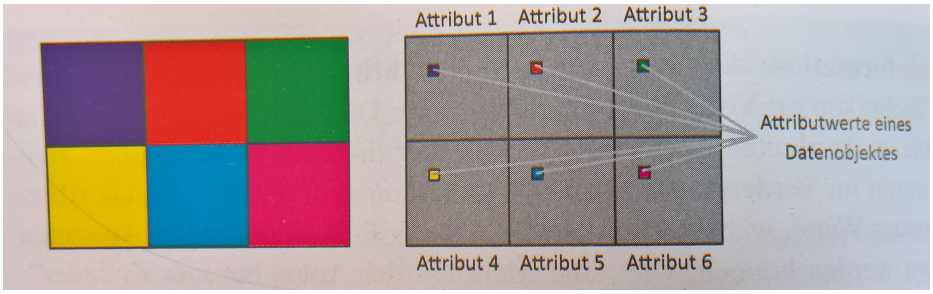


Abbildung 2.6: Grundlegende Anordnung sogenannter Subfenster (rechteckiger Bereiche) für Daten mit sechs Dimensionen. Links eine ikonische Darstellung, rechts eine Verteilung über mehrere Subfenster. [20, S. 466]

451] Die Visualisierung von trivariaten Daten stellt einen Sonderfall der multivariaten Daten dar, da die Visualisierung leicht durch ein 3D-Scatterplot dargestellt werden kann. Dabei wird die klassische XY-Achse um eine dritte Z-Achse erweitert. Bei 3D-Scatterplots treten jedoch einige Probleme auf [20, S. 457]:

- Verdeckungsproblem: Teile der Visualisierung können andere verdecken.
- Navigationsproblem: Techniken müssen bereitgestellt werden, um unterschiedliche Ansichten des 3D-Raums interaktiv auszuwählen. Freie Rotation im 3D-Raum ist mit 2D-Eingabegeräten kompliziert durchzuführen.
- Zuordnungsproblem: Ohne eine manuelle Rotation oder stereoskopische Darstellung ist eine Zuordnung zwischen Datum und Koordinate schwer einzuschätzen.

Um die Probleme zu beheben, werden vor allem erweiterte 2D-Scatterplots mit zusätzlichen Kodierungen, wie Farbe, Form und Größe verwendet, um Attribute auf das Diagramm abzubilden. Eine wichtige Aufgabe für multivariate Daten ist die Entscheidungsunterstützung. Häufig ist ein gezieltes Datenobjekt aus einer großen Menge anhand verschiedener Merkmale zu selektieren. Bei mehr als drei Dimensionen können Attribute nicht mehr ausschließlich auf geometrische Dimensionen abgebildet werden. Um weitere Dimensionen abzubilden, werden Techniken, wie parallele Koordinaten verwendet. Abb. A.4 zeigt ein Beispiel für eine Visualisierung von mehrdimensionalen Attributen eines Datensatzes. Die Linienführung ist in dieser Darstellung jedoch unübersichtlich, weshalb geeignete Interaktionstechniken notwendig sind, um die Usability zu gewährleisten [20, S. 458-461]. Zusätzlich zu den parallelen Koordinaten werden auch sogenannte ikonische Techniken verwendet werden. Diese Visualisierungstechnik bezeichnet die Anordnung von Merkmalen in einem radial- bzw. sternförmigen Muster,

dessen Ursprung für alle Attribute gleich ist. Der Attributwertebereich muss dementsprechend so skaliert werden, dass im Zentrum 0% und im Außenradius 100% abgetragen werden [20, S. 462] Neben Star-Plots (s. Abb. A.5) konnte in einer Studie gezeigt werden, dass die Repräsentation von Attributen durch multidimensionale Icons (metaphorische Icons, Glyphen) statt durch Text die Such- und Filterzeit um die Hälfte reduziert werden konnte. Durch multidimensionale Icons werden nicht nur Zustände, wie passiv oder aktiv visualisiert [20, S. 464]. Die Visualisierung von mehreren tausend Datenobjekten ist durch ikonische Techniken oder parallele Koordinaten aufgrund der Abbildung multipler Attributdimensionen auf räumliche und visuelle Eigenschaften problematisch. Daher wurde die Skalierbarkeit auf Millionen von Datenobjekten durch pixelbasierte Techniken addressiert. Bei diesen Techniken werden Datenwerte auf farbige Pixel abgebildet. Verschiedene Attribute werden dementsprechend in verschiedenen Quadranten visualisiert. Datensätze werden auf sogenannte Subfenster verteilt, wobei die Pixelposition der Subfenster pro Datenobjekt gleich ist (s. Abb. 2.6) [20, S. 466].

Relationen

Hierarchische Visualisierungen sind in modernen Benutzerschnittstellen, Webseiten, sozialen Netzwerken und geographischen Systemen allgegenwärtig. Es existieren zwei Kategorien von Relationen [20, S. 450-451]:

- *Relationen: Hierarchien (Bäume)* stellen eine Mutter-Kind Beziehung dar und vereinfachen Zuordnungen zu Kategorien. Diese werden häufig als Baumstruktur repräsentiert und bilden eine Untermenge von Graphen.
- *Relationen: Netzwerke (Graphen)* werden aus verknüpften Datenobjekten gebildet. Die Datenobjekte bzw. Attribute sind Knoten und die Relationen zwischen ihnen Kanten. Soziale Netzwerke des Web 2.0 werden häufig als Graphen dargestellt.

Die Baumdarstellung stellt eine der klassischen Formen der Visualisierung für Baumhierarchien dar. Diese lassen sich durch Techniken wie einfache Einrückungen, Node-Link-Diagramme, flächenfüllende Verschachtelung und geschichtete Ansätze erzeugen. Einfache Einrückungen sind in den Desktopsystemen vorallem durch Folderbrowser repräsentiert. In diesen Komponenten werden Subfolder durch Einrückung hierarchisch angeordnet, sodass auf direkten Blick der Bezug zwischen Eltern-Kind-Ordnern sichtbar ist. Node-Link-Diagramme visualisieren explizit die Kanten und Knoten des Datensatzes. Dabei kann die Visualisierung der Knoten durch einfache Kreise, jedoch auch durch komplexe Techniken, Glyphen, Bilder und Texte erfolgen. Baumdiagramme sind nicht platzsparend, weshalb die Visualisierung einer großen Anzahl an Datenobjekten besser durch flächenfüllende Verschachtelung oder

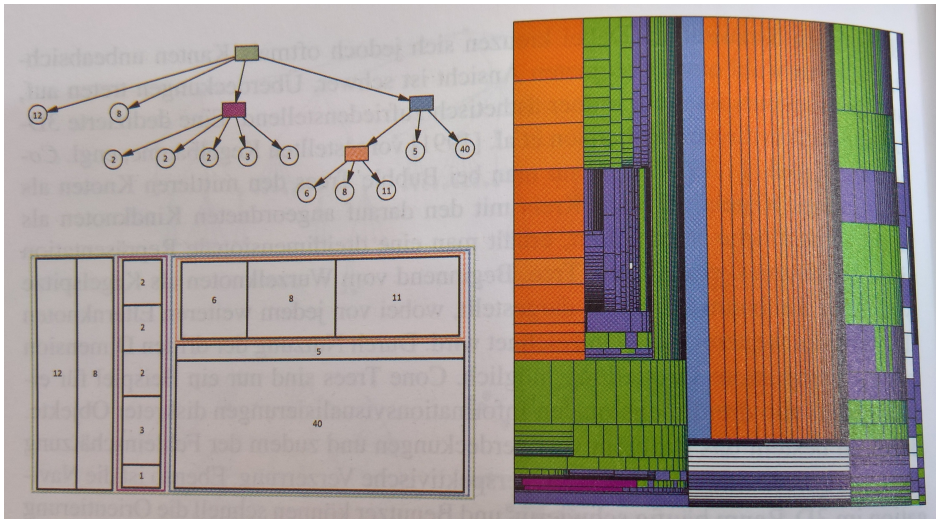


Abbildung 2.7: Konstruktion einer Treemap. (Dateihierarchie, orange: Videos, grün: PPT, hellblau: EXE, dunkelblau: PDF, lila: ZIP, weiß: unbekannt) [20, S. 474]

Abbildung 2.8: Visualisierung einer Dateihierarchie durch ein Sunburst Diagramm [20, S. 479]

.

auch Treemaps geeignet ist. Bei einer Treemap repräsentiert die gesamte zur Verfügung stehende Fläche den Wurzelknoten. Die Kindknoten erhalten einen Teil als rechteckige Fläche, wobei deren einzelne Fläche mit der Größe ihres Teilbaums skaliert (s. Abb. 2.7) [20, S. 467-474]. Die letzte Form der Hierarchievisualisierung für Baumstrukturen stellen geschichtete Ansätze dar. Geschichtete Ansätze sind eine Weiterentwicklung der Treemaps und werden durch Ebenen und geschickte Methoden der Aneinanderreihung dieser Ebenen erreicht. Hierzu zählen die sogenannten Icicle Plots, Tree Ring, InterRing, und Sunburst Diagramme [20, S. 477-478]. Zur Visualisierung von Graphen und Netzwerken bieten sich Node-Link-Visualisierungen, wie Mentionmaps, radiale und hyperbolische Layouts an. Zusätzlich können Matrixvisualisierungen, wie Cluster-Maps, Ontologiegraphen und semantische Graphen eingesetzt werden [20, S. 487-507].

2.2.3 Interaktionen

In der vorangegangenen Sektion wurden Visualisierungstechniken vorgestellt, um Datensätze und Informationsmengen verschiedener Dimensionen zu visualisieren. Außerdem wurden wesentliche Visualisierungsaufgaben im Abschnitt 2.2.1 präsentiert. Um diese Aufgaben als Benutzer durchführen zu können, werden folgende Interaktionskategorien unterschieden [20, S.564-565]:

- *Selektieren*: Etwas als interessant markieren.
- *Explorieren*: Etwas anderes zeigen.
- *Rekonfigurieren*: Eine andere Anordnung zeigen.
- *Kodieren*: Eine andere Repräsentation der Daten zeigen.
- *Abstrahieren/Detaillieren*: Weniger oder mehr Details anzeigen.
- *Filtern*: Etwas unter bestimmten Bedingungen anzeigen.
- *Verknüpfen*: Verknüpfte Datenobjekte oder Informationen anzeigen.

Weiterhin lassen sich für die Manipulation von User Interfaces oder GIS-Kartenanwendungen weitere grundlegende Interaktionen unterscheiden [20, S. 513]:

- *Scrolling*: Ausschnitte eines Dokumentes werden angezeigt und durch Scrolling mit einem Eingabegerät verschoben.
- *Overview und Detail*: Zusätzlich zu einer Überblicksansicht mit geringem Detailgrad werden Detailansichten innerhalb des Dokuments präsentiert. Diese Ansichten können geteilt in verschiedenen Fenstern dargestellt werden.
- *Multiple koordinierte Ansichten*: Dies ist eine Verallgemeinerung der Übersichts- und Detailansichten. Verschieden detaillierte Ansichten können in seperaten Fenstern angezeigt werden.

- *Zoomable User Interfaces*: Darstellung eines Informationsraums in verschiedenen nahtlos übergehenden Detaillierungsstufen.
- *Fokus und Kontext*: Ähnlich wie in Overview & Detail werden detaillierte Ansichten als Fokus betrachtet, wobei der Rest des Informationsraums der Kontext ist. Durch Techniken wie Verzerrung oder Informationsunterdrückung kann der Fokus gleichzeitig mit dem Kontext dargestellt werden.

Touch-basierte interaktive und mobile Systeme besitzen zudem eigene Herausforderungen, die im Folgenenden beachtet werden müssen. Da das Ziel der Arbeit eine Plattform darstellt, die sowohl durch klassische Eingebegeräte, als auch durch Gesten gesteuert werden soll, ist eine spezielle Betrachtung von Problemen bei mobilen Endgeräten notwendig [21, S. 580-585]:

- *Fat-Finger Problem*: Die Größe der Fingerkuppe führt dazu, dass Eingaben verdeckt werden können, oder aufgrund einer zu geringen Größe der Elemente das User Interface (UI) nicht bedienbar ist. Weiterhin besteht das Problem, dass je nach Berührung eine unterschiedlich große Eingabefläche entsteht und die präzise Bestimmung des Mittelpunktes dementsprechend unterschiedlich ausfallen kann. Nach der menschlichen Wahrnehmung kann es daher passieren, dass ein UI-Element berührt wurde, jedoch aufgrund der berechneten Koordinaten nicht ausgelöst wird, da dieses außerhalb des Bereiches liegt.
- *Eingabepräzision*: Die Präzision ist auch aufgrund des Fat-Finger Problems eine wichtige Komponente, die es bei mobilen User-Interfaces zu berücksichtigen gilt. UI-Elemente, die zu nah aneinander liegen, können versehentlich ausgewählt werden. Fehlerhafte Eingaben können daher schnell zu Irritationen führen und die Arbeit mit dem System erschweren.
- *Hovering-Problem*: Mit dem Mauszeiger lassen sich Elemente ohne Drücken berühren und zeigen. Bei touch-basierten Geräten existiert solch ein Hovering-State nicht. Das Fehlen des Hovering-States führt dazu, dass eine wichtige Feedback-Komponente fehlt.
- *Mehrfach-Selektion*: Mit dem Mauszeiger lässt sich nur ein Element selektieren, per Touch können jedoch mehrere Elemente gleichzeitig ausgewählt werden.
- *Mehrfachnutzung*: Die Arbeit von mehreren Nutzern an einem System kann zu Konflikten führen, wenn mehrere Nutzer gleichzeitg ein oder verschiedene Elemente berühren. Weiterhin kann es passieren, dass mehrere Nutzer an den gleichen Texteditoren parallel schreiben. Die Fragestellung ist hier, nach welchen Eingaben priorisiert wird und wie sich die Eingaben auf Korrektheit validieren lassen?
- *Element-Orientierung*: Während für manche Nutzer die Elemente richtig orientiert sind, können vorallem an Tabletops oder Tablets die ge-

genüberstehenden Nutzer durch falsch orientierte Elemente benachteiligt sein.

Die Navigation ist eine der fundamentalen Aufgaben der 3D-Interaktion. Durch geeignete Navigationstechniken lassen sich Gebäude, Stadtmodelle, virtuelle Umgebungen intuitiv und effizient navigieren. Vorallem im Bereich der Infrastrukturentwicklung sind daher Betrachtungen zu grundlegenden Navigationstechniken notwendig. Es wird zwischen zwei verschiedenen Navigationsarten unterschieden: Egozentrische- und Exozentrische Navigation. Die egozentrische Navigaton stellt das 'Ich' in den Mittelpunkt, wodurch die virtuelle Kamera mit dem Augenpunkt korrespondiert und eine Art First-Person-Perspektive eingenommen wird, die sich vorallem in Spielen und Flug- oder Verkehrssimulationen wiederfindet. Die exozentrische Navigaton stellt das Zentrum außerhalb des 'Ich' dar, wodurch diese Technik vorallem für große virtuelle Umgebungen und immersive Virtual-Reality Szenarien geeignet ist. In dieser Variante kann der Nutzer die gesamte Szene greifen, drehen, transformieren und nach Wünschen ausrichten [6, 66, S. 604- 606]. Die Wegfindung durch ein dreidimensionales Szenario kann durch sogenannte kognitive Karten (oder auch *mental maps*) vereinfacht werden. Diese bilden eine mentale Repräsentation des geographischen Raums nach vorstellbaren logischen Zusammenhängen. Eine solche vereinfachte Repräsentation wird genutzt, um sich im Raum zurecht zu finden und geographische Merkmale an wiedererkennbaren Gebäuden, Sehenswürdigkeiten, Landmarken oder Landschaften zu erkennen. Eine solche kognitive Karte entsteht während der Navigation durch den Lernprozess am effektivsten. Für digitale Lösungen bedeutet dies, dass dreidimensionale Szenarien wichtige Merkmale eindeutig, klar erkennbar und von anderen Objekten unterscheidbar darstellen. Wichtige Merkmale sollten daher eine einzigartige visuelle Repräsentation besitzen, um von der mentalen Vorstellung des Betrachters begriffen werden zu können. Als Beispiel wird angeführt, dass Hochhäuser sehr groß und sichtbar sind, jedoch oftmals nicht einzigartig, hingegen Kirchen oder besonders dekorierte Häuser durch einzigartige Merkmale auffallen [66, 86].

2.3 Zusammenfassung

In diesem Kapitel wurden grundlegende Themenfelder in der Infrastrukturentwicklung und Visualisierung darstellt. Zu Beginn wurde der Begriff Infrastruktur erläutert. Von den drei hauptsächlichen Infrastrukturen wird in dieser Arbeit vorwiegend auf materielle Infrastrukturen Bezug genommen. Für diese wurde auch die Planung über Planfeststellungsverfahren und die damit zugehörigen Akteure erläutert. Neben den Projektverantwortlichen spielen Fachplaner und Gutachter eine wichtige Rolle in der Entwicklung. Ein großer Aspekt der Infrastrukturentwicklung stellt die Bürgerbeteiligung dar, welche ausbaufähig ist. Im Bezug auf das Themengebiet der Visua-

lisierung wurden die grundlegenden Aufgaben erörtert und Techniken zur Visualisierung von univariaten, bivariaten und multivariaten Daten vorgestellt. Weiterhin wurden grundlegende Interaktionstechniken und Probleme bei der Bedienung von mobilen Geräten, wie das 'Fat-Finger Problem' dargestellt.

Kapitel 3

Verwandte Arbeiten

Dieses Kapitel beschäftigt sich mit verwandten Arbeiten zum Thema der Visualisierung von zweidimensionalen Karten und dreidimensionalen georeferenzierten Modellen. Es existiert eine Vielzahl von bestehenden Arbeiten, daher wird eine aktuelle Auswahl im Folgenden vorgestellt. Desweiteren wurden Geoinformationssysteme von Firmen wie Autodesk und ESRI entwickelt, welche in diesem Kapitel betrachtet werden.

3.1 Aktuelle Forschung

Die Thematik der Visualisierung von geografischen Informationen ist ein Themenfeld, welches primär in der Geoinformatik untersucht wird [80]. Kartenmaterial wird dabei oftmals mit dreidimensionalen Grafiken zu Geoinformationssystemen verbunden. Diese Systeme werden kontinuierlich weiterentwickelt, sodass neue Technologien, wie WebGL und glTF den Stand der Wissenschaft um neue Datenstrukturen und Algorithmen erweitern.

3.1.1 Aufbau einer Plattform zur Anzeige zweidimensionaler Karten

Eine wichtige Grundlage für das zu konzeptionierende System stellt die Arbeit von Maria A Brovelli dar. In dieser Arbeit wird eine Plattform entwickelt, welche Karten zur Landnutzung und Bodenbedeckung mithilfe webbasierter Techniken darstellt. Das System basiert dabei auf GeoServer als Map-Server, Leaflet als Client-Side Kartenbibliothek und Bootstrap zur Anzeige des Frontends. Das Frontend wurde im Hinblick auf bekannte Desktop-GIS Systeme entwickelt und erlaubt, neben Drag and Drop, die Aktivierung und Deaktivierung von Kartenlayern. Außerdem wird es den Nutzern ermöglicht zwischen mehreren Kartenprovidern, wie OpenStreetMap und Bing zu wählen. Diese Arbeit befasst sich ausschließlich mit den Möglichkeiten zur 2D-Visualisierung von Kartenmaterial im Web. 3D-Visualisierungen werden

in dieser Arbeit nicht behandelt [3].

3.1.2 Webbasierte Visualisierung von 3D-Stadtmodellen

Die Visualisierung von 3D-Stadtmodellen stellt eine fortwährende Grundlage für Forschungsarbeiten dar. Eine der aktuellsten Arbeiten stellt die Visualisierung von Stadtmodellen mit WebGL in den Vordergrund, um eine neue Grundlage für WebGIS-Systeme zu schaffen. Anhand von bekannten Grafikalgorithmen diskutiert die Arbeit Methoden um ein webbasiertes 3D-Stadtmodell effizient in WebGL zu laden. Das später folgende Konzept bezieht sich vorallem auf das in der Arbeit verwendete 3D-Format glTF, welches im Verlauf des Kapitels seperat vorgestellt wird [18].
In einer weiteren Arbeit beschreibt Zhang Liqiang die Applikation von adaptiven Renderingverfahren in Bezug auf 3D-GIS-Systeme. Dazu werden verschiedene Wege zur Darstellung von Stadtmodellen im webbasierten Kontext analysiert und ein Hybridverfahren vorgestellt. Das vorgestellte Verfahren soll als Client-Server-Modell große 3D-Stadtmodelle über einen Remote-Server rendern und dabei Algorithmen, wie image-based und on-demand Rendering effizient kombinieren. Laut Zhang Liqiang erlaubt die erläuterte Vorgehensweise eine fortlaufende Darstellung einer hochauflösenden 3D-Szene in höchster Qualität in Echtzeit. Im Hybridverfahren speichert der Server das Stadtmodell und generiert ausgehend von der Position des Nutzers automatisiert Level-of-Detail Repräsentationen. Weit entfernte Gebäude werden dabei als Bild generiert und an den Client gesendet. Der Client rendert naheliegende Stadtmodelle mit der qualitativ besten LoD-Stufe, weit entfernte Gebäude werden mithifilfe des Bildes gerendert. Dadurch verringert sich die benötigte Datenübertragungsrate. Um Netzwerklatenzen zu vermeiden, werden weiterhin Multithreading-Techniken vorgestellt, welche die großen Datenmengen optimiert verwalten sollen [26].
Jürgen Döllner beschreibt in einer anderen Arbeit immersive Visualisierungsysteme für 3D-Stadtmodelle. Dabei geht die Arbeit vorallem darauf ein, dass Stadtmodelle hohe Anforderungen an die Hardware und Software stellen, und dementsprechend optimierte Rendering-Techniken zum Einsatz kommen müssen, um immersive Städte darzustellen. Döllner geht zusätzlich auf physikalisch korrekte Lichtberechnung ein, was zunehmend durch den Einsatz von PBR-Shadern (Physically Based Rendering) als Ablösung des klassischen Blinn-Phong-Shading zum Einsatz kommt [8].

3.1.3 Übertragung und Darstellung von geographischen Daten

Muhammad Bute befasst sich mit der effizienten Speicherung von GIS-basierten Daten. Die traditionelle Vorgehensweise der Speicherung stellt dabei ein Problem für mobile Endgeräte dar. Bute's vorgestelltes Verfah-

Abbildung 3.1: Visualisierung eines neuen Highways in Norwegen mithilfe von Autodesk InfraWorks 360. Laut der referenzierten User-Story ermöglichte die Nutzung von InfraWorks 360 einen schnelleren und besseren Entscheidungsprozess [44].

ren nutzt hingegen eine Spatial-Database, um auch von mobilen Geräten aus, effizient auf GIS-Daten zugreifen zu können. Eine Spatial-Database ist auf Abfrage und Speicherung von räumlichen Daten optimiert. Das mobile GIS-Framework besteht zusätzlich aus einem Webserver und einem mobilen Client. Der Webserver übernimmt, wie auch in der vorherigen Arbeit die Aufgabe der Aktualisierung, des Uploads und der Bearbeitung der georeferenzierten Daten. Der Client implementiert über eine REST-Schnittstelle Algorithmen und Filterfunktionen, um diese GIS-Daten weiterzuverarbeiten. Das Framework stellt die Digitalisierung des Kartenmaterials im Binärformat in den Vordergrund [5].

3.2 Stand der Technik

Im Bereich der digitalen Infrastrukturplanung wurden bereits unterschiedliche Softwarelösungen für verschiedene Einsatzbereiche entwickelt. In diesem Abschnitt werden Lösungen vorgestellt, welche zur Visualisierung von Infrastrukturprojekten geeignet sein können. Alle vorgestellten Projekte haben den Nachteil, dass sie proprietäre, kostenpflichtige Lösungen darstellen:

InfraWorks 360

InfraWorks ist eine der weltweit meist genutzten Infrastruktur-Design Applikationen (s. Abb. 3.1). Entwickelt von der Firma Autodesk bietet Infraworks Unterstützung im Planungs- und Designprozess in Infrastrukturprojekten im Kontext von realen Planungsprozessen. InfraWorks bietet eine Vielzahl von Features [33, 35]:

- Einbettung georeferenzierter Daten auf Basis von den in Kapitel 2.1.3 erwähnten Austauschmodellen
- Design von Kreuzungen, Kreisverkehrsplätzen, Rampen, Straßen und reversible Spuren (engl. Reversible Lanes)
- Komponentenbasierte Brückenerstellung durch seperate Platzierung und Vernetzung von Balken und Brückendecks
- Analysen von Bodenabsonderungen, Verschmutzungen, Überflutungen (mehrere Zuflüsse möglich)
- Terrain Konturen und Terrain Analysen
- Partielle Anzeige des Modells über iPad

Der Nachteil von InfraWorks besteht im Preis der Software und dem dazugehörigen Abomodell (1575€/Jahr). Außerdem können die Szenarien nur über Windows und iPad bearbeitet und angesehen werden. Eine Version für WebGL steht nicht zur Verfügung [33].

AutoCAD Civil 3D

Das Werkzeug AutoCAD Civil 3D ist ebenfalls eine Software von Autodesk, welche im Gegensatz zu InfraWorks für CAD-Modellierer entwickelt wurde. Daher existiert auch hier nur eine Variante für Windows zu einem Preis von 2290€/Jahr. Die Software besitzt aufgrund der Ausrichtung auf CAD-Modellierer Funktionen, welche in InfraWorks nicht vorhanden sind. So können Oberflächenmodelle durch Grading Groups (grading engl. für Abstufung) automatisiert erstellt und angepasst werden. Außerdem können detaillierte und präzise Pläne sowie Profilansichten von Gebäuden, Bauwerken und Infrastrukturanlagen erstellt werden. Es existieren Funktionen zur hochgenauen Bahntrassenplanung, Kreisverkehrsentwürfen, Kanalnetzen mit natürlichem Gefälle, hydraulischen Analysen, Versatzlängsschnitten und Polygonzügen. Im Gegensatz zu InfraWorks 360 werden jedoch keine komplexen Animationen oder Touren angeboten. AutoCAD Civil 3D ist dafür konzipiert in Zusammenarbeit mit InfraWorks 360 verwendet zu werden. Modelle aus Civil 3D können in InfraWorks 360 verwendet, bearbeitet und gespeichert werden [32, 34].

ArcGIS

Das US-amerikanische Unternehmen ESRI (Environmental Systems Research Institute) pflegt unter dem Namen ArcGIS eine Produktfamilie von Apps und Lösungen im Bereich der Geoinformationssysteme [78]. Die Apps bieten 2D- und 3D-Visualisierungsmöglichkeiten für viele Arten von Infrastrukturproblemen. Unter anderem befinden sich die Software-Lösungen bereits im Praxiseinsatz am Hafen von Rotterdam zur Organisation des Transportaufkommens der Import- und Exportgüter auf dem Seeweg. Ein

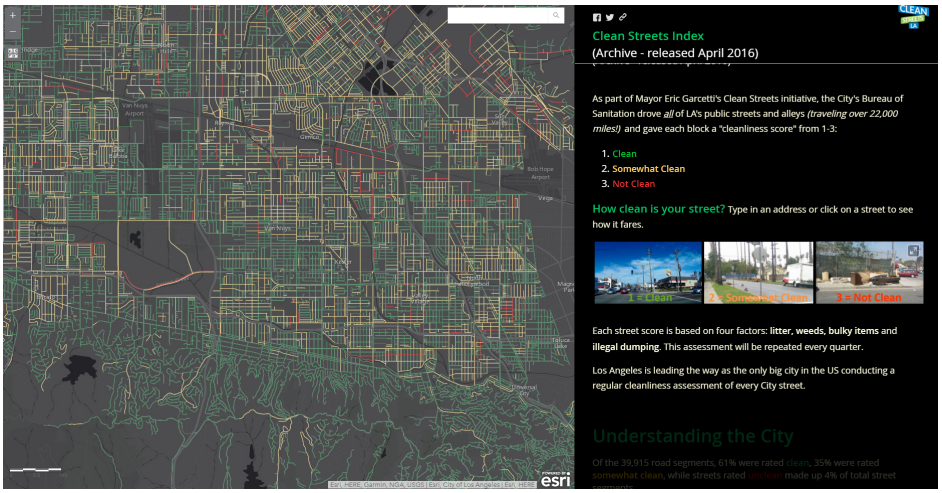


Abbildung 3.2: ArcGIS Szenario zur Darstellung der Sauberkeit des Straßennetzes von Los Angeles [31]

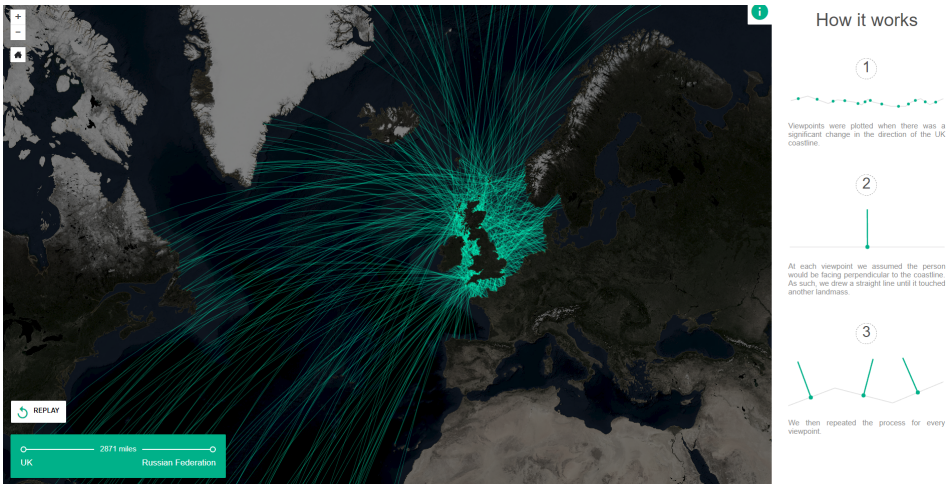


Abbildung 3.3: ArcGIS Szenario zur Darstellung der Sichtpunkte und Linien an der Küste Großbritanniens [58]. Wie an der Abbildung zu sehen ist, leidet die visuelle Darstellung an Informationsüberflutung.

weiteres Praxisbeispiel für den Einsatz von ArcGIS liegt in der Stadt Seattle in der die Operationen der Feuerwehr und Polizei durch ArcGIS-Lösungen kontrolliert werden [57]. Weitere Online-Beispiele sind die Visualisierung des Straßennetzes von Los-Angeles zur Bewertung der Sauberkeit, die Visualisierung von Sichtpunkten an der Küste Großbritanniens und die Darstellung der Tiefseekabel in den Weltmeeren (s.a. Abb. 3.2 und 3.3) [31, 58, 59].

Abbildung 3.4: Visualisierung der Stadt Norrköping in Schweden mit aktiver Bürgerbeteiligung von 176 Kommentaren verteilt auf verschiedene Kategorien mithilfe des CityPlanner's. [27]

Windplanner

Der Windplanner ist ein auf Windenergieanlagen ausgelegtes browserbasiertes Planungs- und Visualisierungswerkzeug. Entwickelt wird der Service von *The Imagineers* mit Sitz in den Niederlanden. Aktuelle Features beinhalten eine Hybridansicht aus 2D und 3D, die Möglichkeit in der 3D-Ansicht zu Satellitenbildern und Panoramen zu wechseln, den möglichen Einsatz von Google Daydream VR und durch den Webansatz ein Support für nahezu alle Plattformen die HTML5, WebGL und CSS3 unterstützen. Außerdem sollen bis Q1 2018 High-Detail-Terrains (mithilfe von Photogrammetrie), Social Sharing und Augmented Reality über das Smartphone in den Service eingebettet werden [55].

Barthauer Basys

Eine weitere Produktreihe stammt von der Barthauer Software GmbH. Diese Firma vertreibt Produkte mit den Namen BaSYS (Netzinformationssystem), BaSYS Regie (Betriebsführung), BaSYS Straße (Straßeninformationssystem), PISA Compact (Sanierungsplanung) und GeoDS (Erstellung individueller Informationssysteme). Alle Produkte stehen parallel für Desktop, Web und Mobile Systeme zur Verfügung und richten sich vorrangig an Unternehmen [49].

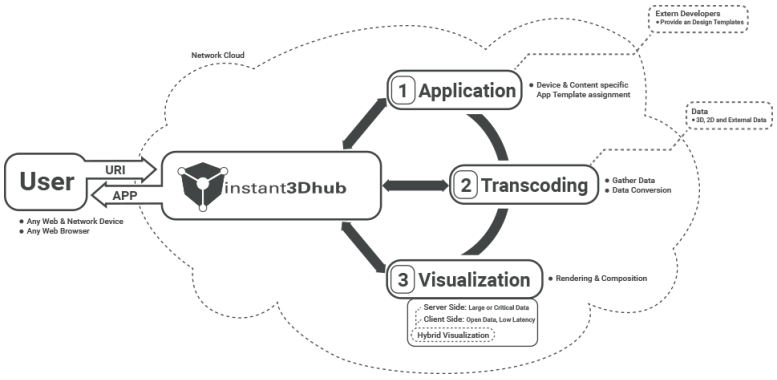


Abbildung 3.5: Instant3DHub Service und Backend-Architektur [54]

CityPlanner

Der CityPlanner ist als Online-Service speziell für den Aspekt des Urban-Planning konzipiert. Hierbei lassen sich 3D-Städte durch Kombination aus CAD und GIS-Daten erstellen und visualisieren. Der CityPlanner setzt als Online-Service ebenfalls auf HTML5 und WebGL. Abb. 3.4 zeigt als Beispiel die Visualisierung der Stadt Norrköping in Schweden. In diesem Beispiel können sich Bürger am Prozess beteiligen; die Visualisierung dient hiermit als Kommunikationselement [27].

Fraunhofer Instant3DHub

Das Instant3DHub, welches vom Fraunhofer IGD (Visual Computing System Technologies Group) entwickelt wurde, ist kein speziell auf Infrastrukturentwicklung ausgelegtes Tool. Vielmehr ist diese Plattform ein Service, um 3D-Szenen über eine interne Serverarchitektur zu rendern und über das vernetzte webVis-Framework einzubetten und zu bearbeiten. (s. Abb. 3.5) Solch ein Rendering-Model wird auch Headless-Rendering (oder Server-Side Rendering) genannt und hat im Vergleich zum Client-Side Rendering einige Nachteile. Einer der größten Kritikpunkte ist ein intensiver Rechenaufwand für die Server bei intensiven 3D-Applikationen und hoher Nutzerlast. Es müssen mindestens 30 Bilder pro Sekunde zu den Client-Rechnern übermittelt werden, um eine akzeptable Bildfrequenz zu gewährleisten, was wiederum eine hohe Bandbreite benötigt. Dies führt bei mobilen Leitungen zu Verzögerungen, weshalb für interaktive 3D-Web-Applikationen das Client-Side Rendering bevorzugt wird [54].

3.3 Programmierschnittstellen in der Computergrafik

Im letzten Abschnitt wurden Apps und Services vorgestellt, die zur Visualisierung von 2D- und 3D-Inhalten im Web geeignet sind. Um 3D-Inhalte rendern zu können, benötigen Webapplikationen Schnittstellen. Hierfür wurde mit HTML5 das Canvas Element eingeführt, welches über ECMAScript (auch bekannt als Javascript) mit der Grafikkarte kommunizieren kann. Die Kommunikation findet über, die von der Khronos Group entwickelte WebGL-API statt. Mittlerweile ist mit WebGL 2.0 am 27.06.2017 eine neue Version der Spezifikation entstanden, welche die Fähigkeiten von WebGL an die Features von OpenGL ES 3.0 angleicht. Web-Applikationen, welche bspw. über Electron [48] zu nativen Systemen kompiliert werden, können somit ohne Kompatibilitätsprobleme die API-Spezifikation von OpenGL ES 3.0 verwenden [52]. Im Bereich der webbasierten Visualisierung sind aufgrund moderner Technologien neue Verfahren entstanden. Diese Verfahren sind eine Kombination aus modernen Internettechnologien [19, S. 555-559]:

1. *Visualisierung als Webservice* bezeichnet ein Verfahren bei der SOAP-basierte oder REST-basierte Serverprogramme entwickelt werden. Auf REST basierte Programme haben sich hierbei durch einfache Integration und simple Entwicklungsmethoden gegenüber den SOAP-basierten durchgesetzt. REST bedeutet Representational State Transfer und ist ein zustandsloses, cache-basiertes Paradigma, welches HTTP-Methoden (GET, POST, PUT etc.) nutzt, um Daten zwischen Client und Server auszutauschen.
2. *Grid-basierte Visualisierung* bezeichnet ein Verfahren, welches sogenannte *verteilte Systeme* zur Lösung von Problemen nutzt. Diese verteilten Systeme sind hochperformante Computerinfrastrukturen, welche die Berechnung vom Client auf den Server auslagern.
3. *Cloud-basierte Visualisierung* ist eine Weiterentwicklung der gridbasierten Visualisierung und nutzt Cloud-Services zur Speicherung der Daten, welche je nach Anzahl an Anforderungen auf Seiten des Servers beliebig skaliert werden können. Beispiele hierfür sind die Services von Amazon AWS, Microsoft Azure oder Google Cloud Storage.
4. *Lokales Rendering* mithilfe von WebGL im Browser verbessert die Interaktivität auf Kosten der Performance. Schwächere Endgeräte, wie Smartphones, können nur einen Bruchteil der Berechnungen ausführen. Der Vorteil des Renderings im Browser liegt jedoch auch in der Latenzzeit, die wesentlich geringer ist, als wenn der Server fertigberechnete Bilder an die Clients verschickt. Die Geschwindigkeit der Berechnungen wird auch hier durch Verbesserung in Javascript, WebGL und neuerdings WebAssembly gesteigert.

Im Folgenden werden weitere Arbeiten vorgestellt, welche auf WebGL auf-

Abbildung 3.6: Anzeige der 1.1 Millionen Gebäude von New York über CesiumJS [39].

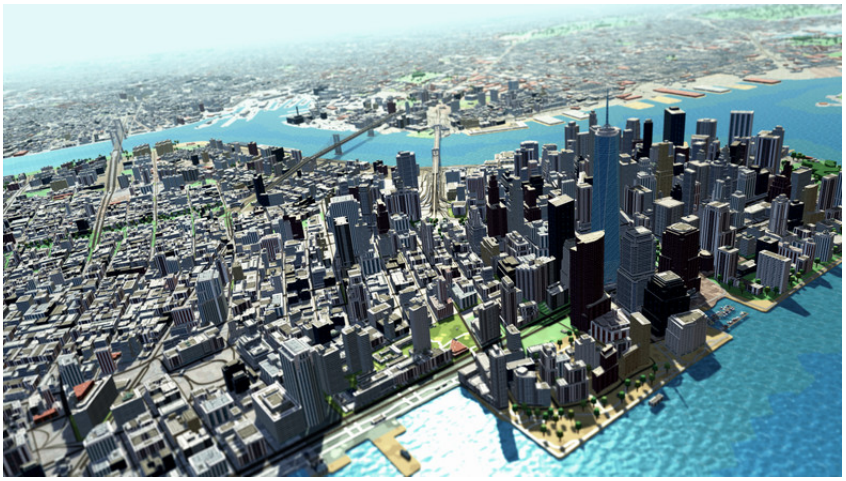


Abbildung 3.7: Beispiel der Anwendung des WRLD Plugins zur Visualisierung einer georeferenzierten Stadt in Unity [94].

bauen und Frameworks zur Entwicklung von webfähigen 3D-Inhalten bieten.

ThreeJS

ThreeJS ist einer der größten Open-Source WebGL-Libraries. Mit über 19 Tausend aufgezeichneten Änderungen im Quellcode (Commits) und 78 veröffentlichten Versionen (Releases) stellt dieses Framework eine stabile Grundlage für 3D-Applikationen im Web dar. Entwickelt wird ThreeJS, welches

2010 erschienen ist, vorrangig über Javascript. Dabei bietet die Library bereits Kombatibilität zu WebGL 2.0 und modernen Features wie Physically Based Rendering [63].

BabylonJS

BabylonJS stellt sich als direkter Konkurrent zu ThreeJS dar. Der erste Commit ist im Juni 2013 veröffentlicht worden. Mittlerweile besteht BabylonJS mit über 7000 Commits und 45 Releases vorrangig aus TypeScript Code, welches eine Obermenge von Javascript ist. Ebenso wie ThreeJS bietet auch BabylonJS einen PBR-fähigen Renderer mit WebGL 2.0 Kombatibilität. Außerdem existiert die Möglichkeit VR-Geräte über WebVR anzubinden [36].

CesiumJS

Spezialisiert auf 3D-GIS-Anwendungen ist CesiumJS eine populäre Open-Source Bibliothek. Die Features reichen von Global Terrain, über Unterstützung modernster Formate wie GeoJSON, KML und glTF bis hin zu der Möglichkeit 2D, 2.5D und 3D-Anwendungen zu erstellen. Die Kartendaten können dabei von Bing Maps, Google Maps, OpenStreetMap und zahlreichen Varianten wie CityGML geladen werden. Abb. 3.6 zeigt ein Beispiel für komplexe Visualisierungen [38].

PlayCanvas

PlayCanvas ist eine auf HTML ausgelegte Game Engine und wurde im Oktober 2011 veröffentlicht. Besonders auffällig ist an PlayCanvas, dass der Editor Closed-Source und die Engine im Hintergrund Open-Source ist. Die Engine hat auf GitHub über 6500 Commits und 552 Releases. PlayCanvas wurde bereits erfolgreich von Firmen, wie Nickelodeon, Disney, Zynga, Mozilla und Facebook für 3D-Spiele Produktionen verwendet. Es ist von einer erhöhten Gefahr von Regressionen auszugehen, da die Engine stark an den dazugehörigen Editor gekoppelt ist. Eine Entwicklung eigenständiger Apps mithilfe der PlayCanvas Engine ist daher riskant [65]. Eine Regression beschreibt eine Art von Fehler, die entsteht, wenn Quellcode, welcher von anderer Software abhängig ist, geändert wird. So können Fehlerbehebungen im PlayCanvas-Editor zu Fehlern in der Engine und auch umgekehrt führen.

Unity

Unity ist in der aktuellen Version 5.6 eine der weltweit führenden Game-Engines. Unity Technologies schließt den Einsatz der Engine für Architektur- und GIS-Visualisierungen nicht aus, daher existieren im Unity-AssetStore GIS-fähige Plugins. Eines dieser Plugins heißt WRLD Unity SDK, welches

als raumbezogen akkurate Karte beworben wird und mit dem sich georeferenzierte Städte und Landschaften erstellen lassen. Abb. 3.7 zeigt ein Beispiel für den Einsatz des Plugins. In der Engine entwickelte Projekte lassen sich als WebGL-Applikation kompilieren, jedoch ist auch die Kompilierung zu anderen Plattformen wie Android, iOS, Linux, Windows möglich [70, 94].

Unreal Engine 4

Der größte Konkurrent zu Unity ist die Unreal Engine 4. Unreal unterstützt eine ähnliche Anzahl an Plattformen, wie Unity, unter anderem auch WebGL. Im Gegensatz zu Unity (C#) werden in Unreal Komponenten über C++ programmiert. Außerdem besitzen Entwickler die volle Kontrolle über den Quellcode. In Unreal lassen sich bereits nativ komplexe Landschaften mithilfe von Heightmaps oder digitalen Höhenmodellen (DGM) erzeugen. Zusätzliche GIS Features wie Koordinatentransformationen lassen sich durch native Bibliotheken, wie GDAL oder GEOS einbinden. Das Einbetten dieser Bibliotheken besitzt jedoch das Risiko, dass die Kompilierung des Projektes durch Emscripten zu Javascript fehlschlägt [56, 61].

3.4 Zusammenfassung

In diesem Kapitel wurden verwandte Arbeiten vorgestellt. Neben den zu Beginn beschriebenden Arbeiten, welche sich vorallem mit der theoretischen Lösung beschäftigen, wurden praktische Visualisierungstools, sowie Frameworks und Engines vorgestellt. Eine wichtige Arbeit stellt die WebGIS-Plattform von Brovelli [3] dar, welche die zweidimensionale Grundlage einer webbasierten Plattform darstellt. In der Konzeptionierung fließen die Erkenntnisse dieser Arbeit ein, um ein dreidimensionales System zu schaffen. Bei den bereits bestehenden Softwarelösungen sind vor allem InfraWorks 360 und AutoCAD Civil 3D als wichtige Grundlage zu nennen. Zur Visualisierung im Web eignen sich moderne Engines auf Basis von WebGL. Spiele-Engines, wie Unity oder Unreal sind nicht geeignet, da diese keine permanenten dynamischen Änderungen der Szenenstruktur erlauben. Unity-Szenarien müssen beispielsweise bereits kompiliert auf den Server geladen werden, bevor diese genutzt werden können.

Kapitel 4

Konzeption einer Visualisierungsplattform

Dieses Kapitel beschäftigt sich mit der Konzeptionierung eines Systems zur Lösung der Aufgabenstellung, welche in Kapitel 1 eingeleitet wurde. Im ersten Abschnitt wird das Ziel der Arbeit anhand eines Vergleichs des Ist-, sowie des Soll-Zustands beschrieben. Darauf folgen Anwendungsfallszenarien, die das System visuell darstellen soll. Die Anforderungen werden basierend auf den Ausführungen in den vorangegangen Abschnitten aufgestellt und das System anhand der Anforderungen konzeptioniert.

4.1 Eingrenzung der Problemstellung

Das Ziel dieser Arbeit ist die Identifikation von Ansätzen zur Realisierung einer webbasierten und mobilen Plattform zur Bereitstellung von hochauflösenden Szenarien zur Planung, Wartung und Unterstützung von Infrastrukturprojekten. Im Folgenden wird dieses Ziel anhand eines Ist-Soll-Vergleichs verdeutlicht und daraufhin allgemeine Anforderungen abgeleitet.

Ist-Zustand

In Kapitel 3 wurden bereits verwandte Arbeiten beschrieben und ein Ausblick auf bestehende Software-Lösungen zur Infrastrukturvisualisierung dargestellt. Es ist dabei erkennbar, dass noch keines der existierenden Lösungen als uneingeschränkt mobile Variante zur Verfügung steht, da der Großteil an Lösungen native Desktopprogramme darstellen (s. [32, 33]). In Bezug auf Plattformunabhängigkeit existieren für einige Werkzeuge Android oder iOS Versionen [33]. Eine uneingeschränkt plattformunabhängige Lösung ist jedoch nicht vorhanden. Die Ansätze und Features bisheriger Interaktionstechniken zur Infrastrukturentwicklung stellen vorallem in Autodesk Infraworks 360 (s. Kapitel 3.2) und ArcGIS (s. Kapitel 3.2) eine ausgereifte Ba-

sis dar. In beiden Systemen sind umfangreiche Werkzeuge zur Bearbeitung, Analyse und Exploration vorhanden. Neben diesen Systemen existieren auch moderne Lösungen für anwendungsspezifische Teilbereiche in der Infrastrukturplanung [27, 55]. Der Windplanner (s. 3.2) zeigt als modernes System, dass die Planung von Windfarmen intuitiv über spezialisierte Interaktionstechniken im Web stattfinden kann. Besonders der Ansatz der kombinierten 2D-Karten mit 3D-Virtual-Reality-Visualisierungen kann ebenfalls für allgemeine Infrastrukturprojekte eingesetzt werden [55].

Soll-Zustand

Der bisherige Stand der Technik soll um eine neuartige Plattform erweitert werden, welche auf den erfolgreichen Ansätzen der bestehenden Plattformen aufbaut und eine allgemeine Lösung zur Infrastrukturentwicklung darstellt. Besonders wichtig ist es, dass das System fortlaufend um neue Interaktionstechniken für spezielle Anwendungsfälle erweitert werden kann. Hierzu ist es fundamental, dass eine strikte Benutzerrollenverteilung im System existiert und zwischen verschiedenen Anwendern des Systems unterschieden werden kann. Den Nutzern werden je nach Kategorie unterschiedliche Rechte auf bestimmte Bereiche des Projektes gewährt. Weiterhin ist es entscheidend, dass die Serverarchitektur eine qualitativ hochwertige Visualisierung für ein großes Nutzerspektrum gewährleistet. Die Nutzer können demnach weltweit auf die Infrastrukturprojekte zugreifen. Hierbei ist tiefergehend zu analysieren, wie die Serverlast verteilt und eine geringe Antwortzeit für Clients gewährleistet werden kann, die mitunter Serveranfragen aus großer Entfernung stellen. Außerdem ist eine Sicherstellung sensibler Daten für nicht-authorisierte Benutzer zu gewährleisten. Da die Visualisierung von Infrastrukturprojekten die Übertragung von großen Datenmengen erfordert, ist die Konzeptionierung eines modernen und intuitiven Front- und Backends notwendig, welches aktuelle Technologien, wie Content Delivery Networks, ECMAScript 2015 und WebGL berücksichtigt. Da die Infrastrukturentwicklung ein breites Themenfeld darstellt, wird im Folgenden das Gebiet auf materielle Infrastrukturen eingeschränkt. Trassenplanungs- und Bundesautobahnprojekte sind tiefergehend Beispiele für die Kerninfrastrukturen im dargestellten Konzept. Besonders Entwickler sollen im Kontext einer allgemeinen Lösung des Konzepts für die Weiterentwicklung der Plattform geeignete Schnittstellen erhalten, welche den Einsatz von 2D- und 3D-Kartenmaterial zur Planung von Szenen und Touren unter Einbeziehung von Social Media Features zur Bürgerbeteiligung vereinfacht.

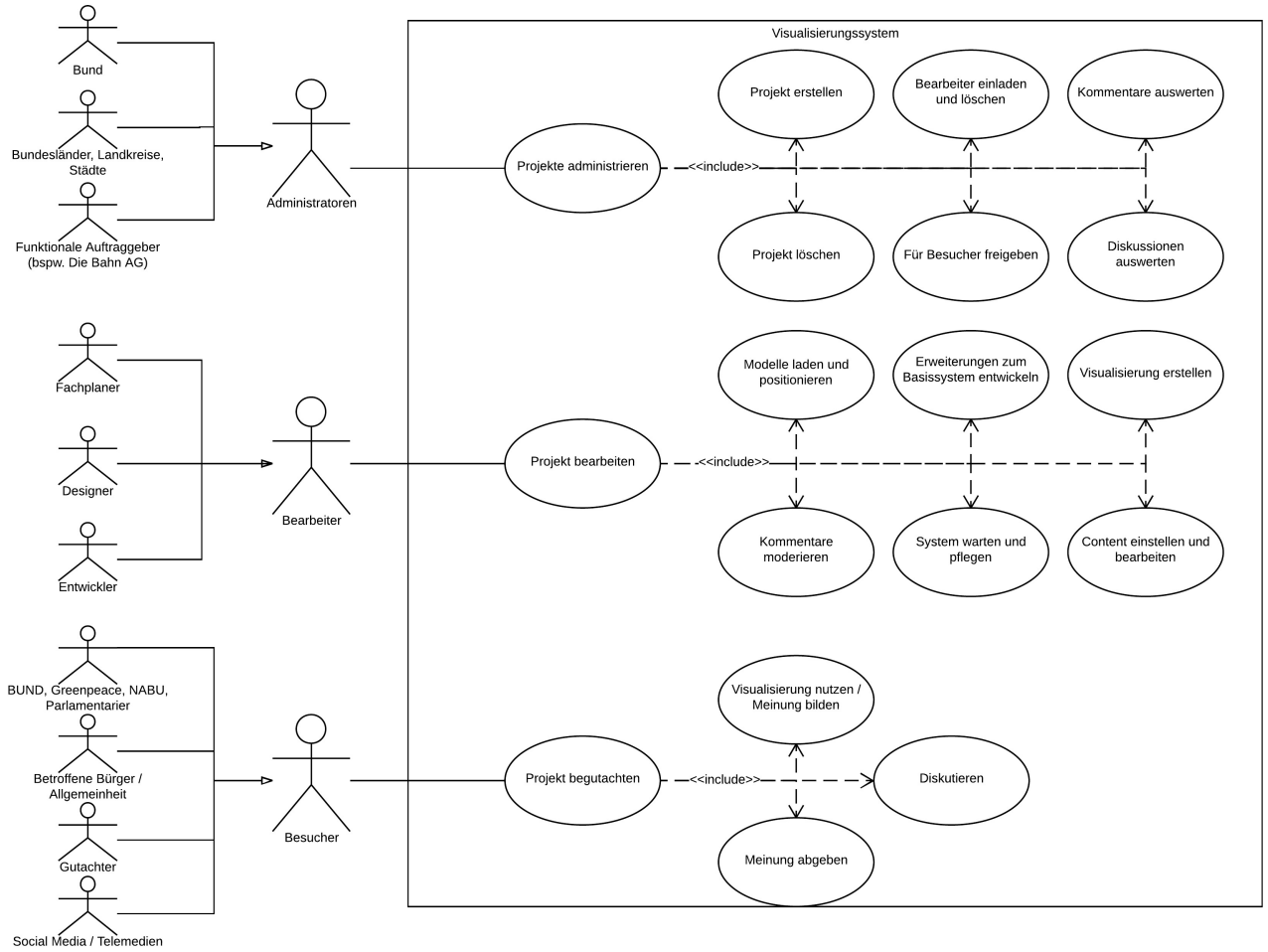


Abbildung 4.1: Anwendungsfalldiagramm der Akteure des Visualisierungssystems. (Eigene Darstellung, Administratoren abgeleitet aus Vergabe von Infrastrukturprojekten s. [24, S.430-436])

Anwender

Das Kapitel 2.1.2 hat die Planungsphasen eines Infrastrukturprojektes behandelt und verschiedene Akteure, wie Fachplaner, Gutachter, Auftraggeber und Bürger beschrieben. Abb. 4.1 zeigt, wie sich die verschiedenen Akteure des Systems im Anwendungsfall verhalten. Es existiert eine große Zahl von Anwendern, welche vom System beachtet werden müssen:

- Bund, Bundesländer, Landkreise, Städte und funktionale Auftraggeber, die im Auftrag einer Gebietskörperschaft arbeiten (bspw. Die Bahn) sind als Administratoren des Systems anzusehen. Diese Anwender haben das Ziel das Projekt der Öffentlichkeit zu präsentieren und im Sinn der Planungsphasen, VDI und Planfeststellungsverfahren die Akzeptanz der Öffentlichkeit zu gewinnen. Weiterhin kann das Feedback genutzt werden, um die Planung positiv zu gestalten und im weiteren Kommunikationsverlauf anzupassen.
- Fachplaner, Designer und Entwickler sind als Bearbeiter eines Infrastrukturprojektes zusammenzufassen. Diese Anwender verfolgen das Ziel das Infrastrukturprojekt realistisch darzustellen, freigegebene Dokumente und Pläne offenzulegen und für Besucher aufzubereiten. Die Bearbeiter besitzen demnach einen großen Einfluss auf die Qualität des präsentierten Infrastrukturprojektes.
- Neben der Allgemeinheit und betroffenen Bürgern existieren weitere öffentliche Organisationen und Personen, die ein Interesse an der Einsicht in Infrastrukturprojekte besitzen. Hier sind vorrangig Gutachter und öffentliche Organisationen, wie Greenpeace, BUND und NABU zu nennen. Außerdem besitzen Social Media Netzwerke und andere Telemedien das Interesse zur Verbreitung wichtiger Projekte.

4.2 Anwendungsfallanalyse

Im Folgenden werden Anwendungsfälle dargestellt, welche das System darstellen soll. Da das System allgemein auf Infrastrukturszenarien ausgerichtet ist, sind die Anwendungsfälle Beispiele für ein breites Themenfeld. Tiefergehend werden nachfolgende Infrastrukturen bezüglich Verkehrs- und Leitungstrassenprojekte beschrieben.

4.2.1 Visualisierungsprojekte geplanter Bundesautobahnen

Das Projekt des Autobahnausbau's der A14 ist ein Kooperationsprojekt des Ministeriums für Landesentwicklung und Verkehr Sachsen-Anhalt mit dem Fraunhofer-Institut IFF Magdeburg [12]. Das Projekt wurde in der Visualisierungsplattform des Fraunhofers (VRAuthor) visualisiert und stellt den Lückenschluss der BAB14 dar. Zu Beginn des Projektes wird mithilfe einer Übersichtsdarstellung, einem Audiokommentar und einer passenden Tour

der Sachverhalt beschrieben. Es wird gezeigt, wie sich die A14 geographisch in Bezug auf benachbarte Bundesautobahnen und Städte einordnen lässt. Aktuelle Daten sind in der Visualisierung nicht eingepflegt. Die A14 soll frühestens zu Beginn des Jahres 2020 befahrbar sein. Die Gesamtstrecke des Ausbaus beträgt in Einbeziehung der durchlaufenden Bundesländer 422 km und kostet knapp 1,3 Milliarden Euro. Vorallem Naturschützer beklagen den Ausbau [76]. In der Verkehrsuntersuchung wurden 1995 verschiedene Varianten aufgestellt. Es existierte eine sogenannte X-Variante, die von Magdeburg nach Lüneburg verlaufen sollte, während die A39 von Wolfsburg nach Schwerin verlängert wird. Diese Variante hat sich nicht durchgesetzt, woraufhin der I-Variante 2004 stattgegeben wurde [76]. Die A14 läuft dementsprechend nach Schwerin und die A39 nach Lüneburg. Bei der Infrastrukturentwicklung im Kontext des Bundesautobahnausbaus sind nicht nur einzelne Autobahnen sondern Kombinationen mehrerer Ausbaustrecken einzubeziehen. Die Streckenabschnitte sind in in Tabelle B.1 im Anhang aufgelistet. In der Detaillansicht eines jeweiligen Streckenabschnitts kann weiterhin analysiert werden, ob Tunnel, Wildunterführungen, Brücken oder ähnliche Strukturen benötigt werden. Der Streckenabschnitt bei Colbitz benötigt bspw. eine Fledermaus- und Wildunterführung und verläuft durch ein Waldgebiet, wodurch Naturschutz-Auflagen erfüllt werden müssen (s. Abb. 4.3) [12]. Laut Bundesverkehrswegeplan ist weiterhin eine Verkehrsprognose über einen Zeithorizont von 15 Jahren durchzuführen. Die Prognosen sollen hierbei Aussagen zu künftigen Verkehrsverpflechtungen, Verkehrsaufkommen, Verteilung der Verkehrsmengen, sowie die Nutzung verschiedener Verkehrsträger beinhalten [4, S. 112-123].

4.2.2 Planung von Leitungstrassen

Das Fraunhofer IFF Magdeburg entwickelte im Rahmen eines Projektes ein VR-gestütztes Werkzeug zur Planung von elektrischen Trassen. Hier ist die Visualisierung der Trassenführung das entscheidende Merkmal. Vorallem bei länderübergreifenden Projekten, welche Hochspannungsleitungen einsetzen und eine große Auswirkung auf die Umwelt besitzen, ist eine korrekte Visualisierung notwendig. Der Trassenverlauf wird in diesen Planungsprojekten in mehreren Stufen festgelegt, um die Akzeptanz von Bevölkerung und Bundesnetzagentur zu sichern. Als Interaktionstechniken sind ähnliche Schritte, wie im Autobahnausbau notwendig. Verschiedene Trassenmodelle sollen hierbei in die Plattform geladen werden können. Ein wichtiger Punkt ist dabei die Georeferenzierung, damit die Modelle an der korrekten Position dargestellt werden. Um eine korrekte Aussage bezüglich der Trassensichtbarkeit zu treffen und insbesondere für Bürger eine Möglichkeit zu bieten, verschiedene Sichtpunkte effizient und informativ einzunehmen, ist die Einnahme von interaktiven Viewpoints eine wichtige Interaktionskomponente. Die Entwickler können hierbei Sichtpunkte vorgeben, die für die geplante Trasse von

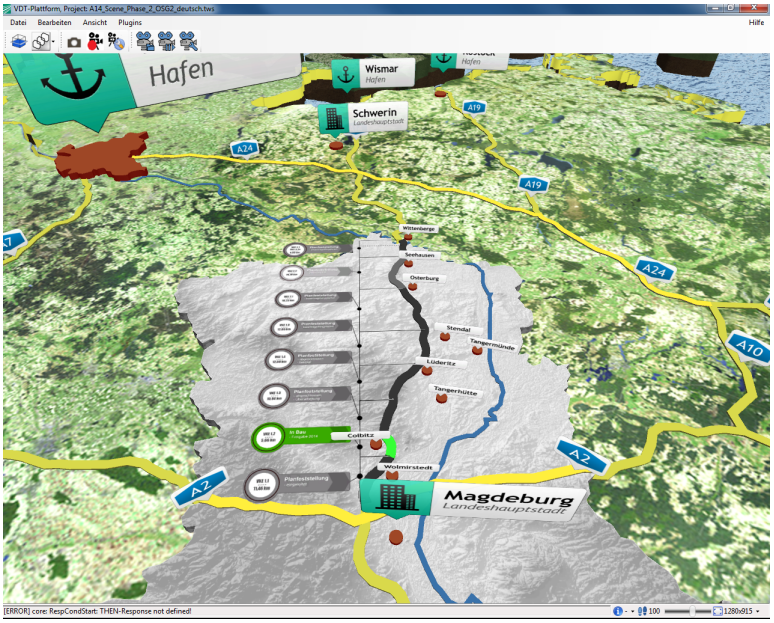


Abbildung 4.2: Detaillierte Anzeige der Planfeststellungsverfahren und Streckenabschnitte der A14 [12].

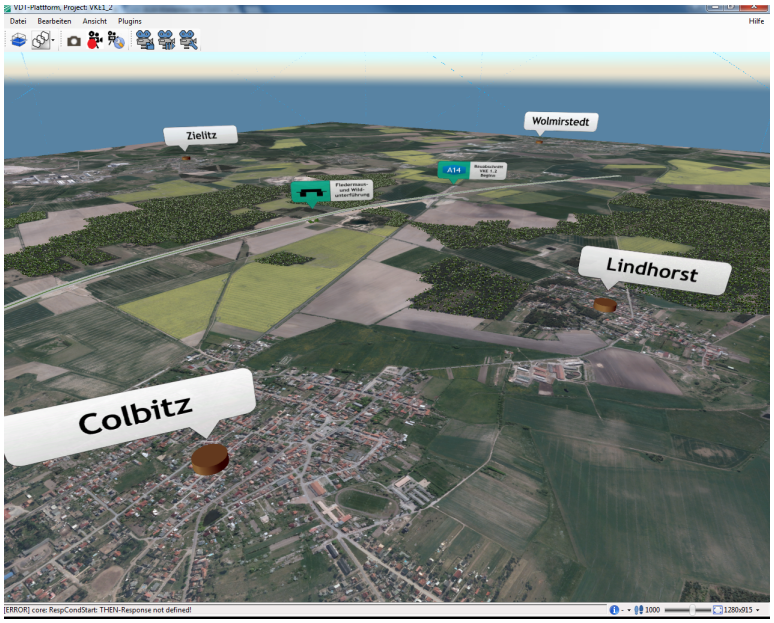


Abbildung 4.3: Detailansicht des Streckenabschnitts in Colbitz - visualisiert in einer seperaten Szene [12].

wichtigem Charakter sind. Durch geeignete User-Interfaces können Besucher daraufhin zu den gesetzten Punkten springen, um einen ersten und schnellen Einblick in die geplante Trassenführung zu erhalten. Mit Social-Media-Features sollte weiterhin eine Bewertung des Verlaufs einer Trasse möglich sein. Eine Studie der Bertelsmann-Stiftung zeigt bei 1003 Befragten deutlich, dass sich Diskussionsveranstaltungen als vielversprechendste Möglichkeit für mehr Mitsprache herausgestellt haben [2, S.15]. Daher ist die Einführung von raumbasierten Social-Media Features eine wichtige Komponente für die weitergehende Infrastrukturplanung.

4.2.3 Transport und Logistik

Eng mit dem Ausbau von Verkehrsnetzen ist die Branche der Transport- und Logistik-Unternehmen verbunden. Die interaktiven Simulationskomponenten, die bereits bei den Prognosen für Bundesautobahnen angeschnitten wurden, sind hier ein zentraler Bestandteil. Hier werden vorallem Simulationen und daraus resultierende Visualisierungen benötigt, die das Transportaufkommen, die Liefermenge, Lieferdauer und die dazugehörigen Wegenetze darstellen. Ein Beispiel für einen solchen Anwendungsfall findet sich im Kooperationsprojekt des Fraunhofer IFF mit dem Unternehmen Stork Umweltdienste GmbH wieder. In diesem Projekt wurde über die bereits genannte Visualisierungsplattform das Betriebsgelände des Unternehmens visualisiert. Weiterhin konnten Warenbestände und Lieferungen über eine Datenbank erfasst und im 3D-Raum präsentiert werden. Zur Beschreibung der Anforderungen im nächsten Kapitel spielt das Anwendungszenario für Transport und Logistik eine untergeordnete Rolle. Dieser Anwendungsfall soll speziell die Erweiterbarkeit des Systems darstellen. Entwickler können Module einfügen, die zur Verwendung im Szenario bereitgestellt werden.

4.3 Anforderungen

Die Anforderungen einer webbasierten Plattform lassen sich in funktionale und nicht-funktionale Anforderungen unterteilen [7, S. 19]. In dieser Sektion werden allgemeine Anforderungen an die Plattform gestellt. In der Tabelle 4.1 sind die funktionalen Anforderungen aufgeführt. Die Tabelle geht dabei, wie im vorherigen Abschnitt beschrieben, von drei Haupt-Akteuren aus (s. 4.1): *Administrator*, *Bearbeiter* und *Besucher*. Administratoren erstellen und verwalten Projekte in eigener Verantwortung. Administratoren haben zusätzlich die Möglichkeit *Bearbeiter* zur Projektbearbeitung einzuladen. Die *Bearbeiter* erhalten dabei bis auf projektspezifische Funktionen (keine Erlaubnis zur Löschung eines Projektes) alle administrativen Rechte. Zusätzlich können Administratoren Projekte zur Beteiligung freigeben. Diese Freigabe erlaubt es anonymen Besuchern das Projekt zu begutachten und über Social-Media-Komponenten, wie in Tabelle 4.1 ID 6 beschrieben, zur Verbesserung des Projektes beizutragen. *Bearbeiter* und *Administrato-*

ren liefern im Kontext der Infrastrukturentwicklung die notwendigen Daten zur Visualisierung des bevorstehenden Bauprojekts. Zusätzlich können Pläne, Dokumente und Tabellen für bevorstehende, abgeschlossene oder aktuelle Planfeststellungsverfahren nach eigenem Ermessen geliefert werden. *Besucher* können sobald ein Projekt freigegeben ist, alle bereitgestellten Daten einsehen. Zusätzlich zu den funktionalen Anforderungen sind nichtfunktionale Anforderungen an das System zu beachten. Die Anforderungen sind in der Tabelle 4.2 dokumentiert.

ID	**Einordnung**	**Beschreibung**
1	Authentifizierung	Das System stellt Interfaces für das Login und Sign-Up zur Verfügung.
2	Authentifizierung	Administratoren können Projekte für authentifizierte Nutzer zur Bearbeitung freigeben
3	Authentifizierung	Besucher können anonym und ohne Anmeldung freigegebene Szenarien begutachten.
4	Authentifizierung	Unangemeldete und nicht authorisierte Besucher dürfen keine Projektdaten ändern.
5	Beteiligung	Besucher können über Social-Media Funktionen zur Diskussion beitragen. Die Beteiligung erfolgt im System über georeferenzierte Kommentare und Diskussionen in einem seperaten Feld zu gewählten Aspekten des Projektes. Die Kommentare können zusätzlich mit Farben zur Verdeutlichung versehen werden.
6	Beteiligung	Das im Juni 2017 beschlossene Netzwerkdurchsetzungsgesetz (kurz NetzDG) richtet sich an soziale Netzwerke. Das System muss aufgrund der Social-Media Komponenten mit NetzDG konform gehen und Moderatoren zur Regulierung der Kommentare einsetzen. Diese Anforderung wird im späteren Prototypen nicht berücksichtigt, da laut Gesetzestext eine Grenze von mindesetens zwei Millionen Nutzern erreicht werden muss. Für ein vollständiges System ist diese Anforderung dennoch zu berücksichtigen [87].
7	Beteiligung	Diskussion mithilfe integrierter Social Media Features, wie Foren, Chats und Kommentare.
8	Schnittstelle	Clients können mithilfe eins Webservices statische Dateien zur Anzeige der Website erhalten.
9	Schnittstelle	Anzeige von Dokumenten, Pläne und Planfeststellungsverfahren.
10	Schnittstelle	Das Backend stellt zusätzlich eine API bereit, welche projektspezifische Ressourcen anfordert und basierend auf der Authentifizierung des Clients die Ressourcen zur Anzeige bereitstellt.
11	Schnittstelle	Die API bietet die Möglichkeiten zur vollständigen Projektverwaltung mit Erstellung, Bearbeitung, Freigabe und Löschung.
12	Schnittstelle	Die API ermöglicht die Speicherung von Projekten. Weiterhin können externe Daten gespeichert und beliebig verändert werden.

13	Visualisierung	Administratoren und Projektbearbeiter können Dateien hochladen, bearbeiten und löschen
14	Visualisierung	Die Visualisierung zeigt gleichzeitig 2D-Kartenmaterial, sowie 3D-Modelle an. Die notwendigen Kartenmaterialien werden von geeigneten Providern angezeigt (bspw. OpenStreetMap).
15	Visualisierung	Darstellung hochauflösender georeferenzierter 3D-Szenarien inklusive Modelle und Texturen.
16	Visualisierung	Zur Visualisierung von georeferenzierten Gebieten und Flächen können Layer erzeugt und in der Szene hinterlegt werden.
17	Visualisierung	Hochgeladene Dateien können zur Visualisierung eingesetzt werden. Modelle können in der Szene frei platziert werden.
18	Visualisierung	Alle Akteure können georeferenzierte Kommentare nach Anzahl, Ort und Farbe filtern und dadurch wichtige Standorte explorieren.
19	Visualisierung	Projektbearbeiter können Objekte zur Selektion freigeben und Texte hinterlegen, welche bei Selektion von Besuchern des Projektes angezeigt werden.
20	Visualisierung	Primäre und sekundäre Elemente können durch visuelle Filter unterschieden werden. Die Karte und alle sekundären Elemente werden dabei durch einen Klick ausgegraut, während primäre Elemente des Projektes visuell hervorgehoben werden.
21	Visualisierung	Es können zur weiteren Verdeutlichung von wichtigen Punkten des Projektes Labels und Markierungen platziert werden
22	Visualisierung	Ein zu Beginn erscheinender Informationsscreen informiert zusammenfassend über alle wichtigen Details des Projektes. Bearbeiter und Administratoren können diesen Screen beliebig anpassen.
23	Visualisierung	Zusammenfassung von Kommentaren zu Clustern für vereinfachte Anzeige einer bestimmten Bewertungsart der Bürger.
24	Visualisierung	Farbskalen zur Bewertung mithilfe der Social-Media Funktionen.
25	Erweiterung	Administratoren und Bearbeiter können das Projekt beliebig erweitern. Zur Erweiterung stellt das System eine gebündelte clientseitige API bereit, welche den Zugriff auf den Visual Globe und den gespeicherten Projektdateien gewährt.

Tabelle 4.1: Allgemeine funktionale Anforderung an ein webbasiertes Visualisierungssystem mit Bezug auf Infrastrukturentwicklung und diesbezüglicher Bürgerbeteiligung.

ID	Einordnung	Beschreibung
1	Qualitativ	Das Backend ist performant und stellt angeforderte Ressourcen in einer akzeptablen und für den Benutzer intuitiven Antwortzeit zur Verfügung. Es ist zu beachten, dass die Architektur der Plattform die Nutzung eines hohen Nutzerstamms berücksichtigt. Das Cache-Verhalten und moderne Technologien, wie Cloud-Storages, sind auf mögliche Nutzung zu prüfen und in der Architektur zu berücksichtigen.
2	Qualitativ	Der Client kann 3D-Ressourcen beliebig anfordern und dadurch hochauflösende Daten durch LOD's ersetzen. Qualitativ hochwertige Modelle können daher später nachgeladen werden. Die Interaktivität wird nicht durch Ladevorgänge blockiert.
3	Qualitativ	Das Backend erlaubt die Anforderung von Ressourcen über weite Entfernungen unter Einhaltung von Anforderung 1. Das System ist für internationale Verwendung ausgelegt und enthält kein unbeabsichtigtes Geoblocking.
4	Qualitativ	Die Performance des Systems wird nur durch die Leistung der vom Client verwendeten Geräte und Verbindungen limitiert. Die Visualisierung verwendet moderne Technologien, wie WebGL zur Anzeige der Daten. Der schnellstmögliche Transfer von Daten ist durch geeignete Datenformate zu gewährleisten.
5	Qualitativ	Das System ist zuverlässig. Ein Ausfall von Systemkomponenten führt nicht zu einem Verlust von sensiblen Projektdaten.
6	Qualitativ	Das System ist sicher gegenüber Fremdzugriff.
7	Qualitativ	Das System ist wartbar und kann nach einem Ausfall innerhalb kürzester Zeit in Betrieb genommen werden.
8	Qualitativ	Das Frontend nutzt moderne Technologien zur performanten Darstellung und ist für mobile Geräte konzipiert.
9	Qualitativ	Das System kann in Echtzeit genutzt werden. Hohe Latenzzeiten und Pings werden durch geeignete Technologien minimiert.
10	Ergonomie	Die ergonomische Gestaltung der Benutzeroberfläche ist zu gewährleisten, damit alle Akteure auf optimaler Weise Infrastrukturprojekte begutachten und bearbeiten können.
11	Ergonomie	Das System soll in kurzer Zeit erlernbar sein und durch intuitive User-Interfaces einen effizienten Einstieg in die Infrastrukturentwicklung ermöglichen.
12	Ergonomie	User-Inputs werden validiert und sind robust bezüglich Fehlern und unangemessen Eingaben. Durch User Inputs können keine datenbankspezifischen Abfragen (auch genannt SQL-Injection) durchgeführt werden. Außerdem führen Nutzereingaben zur geeigneten Fehlermeldungen, welche dem Nutzer zu besseren Inputs helfen.

13	Ergonomie	Nutzer des Systems können Fenster erstellen, anpassen und beliebig verändern. Das Fenster-System muss daher intuitiv, wie ein natives UI-System bedienbar sein (s. InfraWorks 360).

Tabelle 4.2: Allgemeine nichtfunktionale Anforderungen an das zu konzeptionierende System.

4.4 Entwurf der Systemarchitektur

Zur Herleitung einer Software-Architektur werden in diesem Kapitel verschiedene Varianten in den unterschiedlichen Schichten einer Webapplikation verglichen, welche die Anforderungen aus Kapitel 4.3 erfüllen. Daraufhin wird eine Variante für die jeweiligen Schichten herausgearbeitet und in der Implementierung prototypisch umgesetzt. Um konkrete Entscheidungen zu treffen, müssen die verschiedenen Teilaspekte einer Softwarearchitektur für mobile Web-Applikationen betrachtet werden (s. 4.4).

4.5 Allgemeines

Für die Konzeptionierung einer mobilen Infrastrukturplattform existieren verschiedene Varianten. Zum einen ist die Planung einer nativen Applikation für mobile Systeme möglich, zum anderen kann das System auch webbasiert entwickelt werden. Weiterhin existiert der Vorschlag Hybridapps zu entwickeln, die gleichermaßen für das Web und mobile Systeme kompiliert werden können. Alle Varianten benötigen jedoch zum Austausch von Planungsdaten eine Serverinfrastruktur und ein Kommunikationsprotokoll. Sobald das Fundament der Serverinfrastruktur festgelegt wurde, können beliebige Applikationen an das System anknüpfen. Für dieses Konzept wurde daher festgelegt, dass das Frontend mithilfe einer webbasierten Applikation dargestellt wird. Die Erweiterung, um native mobile Applikationen für iOS, Android und Windows Mobile ist jederzeit mithilfe der hier dargestellten Techniken möglich. Als Kommunikationsprotokoll der HTTP-Standard (Hypertext Transfer Protocol) verwendet [84].

4.5.1 Systemübersicht

Wie in den Anwendungsfällen beschrieben, soll der Nutzer des Visualisierungssystems Projekte erstellen, Projekte für Nutzer freigeben, Dateien laden und speichern und das Objektverhalten durch eigene Javascript Erweiterungen steuern (s. Abb. 4.5). Das konzipierte Erweiterungssystem basiert dabei auf einem komponentenorientierten Design, welche ähnlich zu Unity's

Monobehaviour aufgebaut ist. Unity's Monobehaviour ermöglicht eine Einbettung komplexer Systeme durch Austausch von Ereignisnachrichten an essentiellen Punkten der Applikation [71]. Ein komponentenorientiertes Design stellt ein Objekt als einfache ID dar. Dieser ID werden Komponenten angefügt, welche das Verhalten des Objektes steuern. D.h. alle Objekte besitzen eine Liste von Komponenten, hier 'Behaviour' genannt. Diese Komponenten besitzen Funktionen, wie OnStart, OnUpdate und OnDestroy. Der Updatezyklus des Systems beträgt dabei 33 Millisekunden (30 FPS), um einen flüssigen Ablauf zu gewährleisten. Mit den dargelegten Methoden ist es möglich komplexe Vorgänge abzubilden, bspw. bedingungsabhängige Animationen und Kontrollsysteme. Das Visualisierungssystem kann Szenen laden und speichern. Die Auswahl einer Szene ist abhängig vom Projekt und kann entweder vom Besucher ausgewählt oder durch Erweiterungsskripte automatisch geladen werden. Eine Szene wird im JSON Format abgespeichert, welches von Javascript beliebig zwischen Objektnotation und String konvertiert werden kann. Eine Szene speichert folgende Daten:

1. Startposition und -orientierung
2. Liste aller Objekte und Modelle mit Aktivitätsstatus
3. Für jedes Objekt eine Liste von Verhaltenskomponenten
4. Liste aller Marker und Informationen zu diesen Markern
5. Liste von GeoJSON Layern zur Visualisierungserweiterung

Die Abb. 4.6 zeigt, wie das Visualisierungssystem initialisiert wird. Zum Laden eines Projektes werden asynchrone Funktionen ausgenutzt, um Inhalte dynamisch zu laden. Der Nutzer des Projektes kann somit schon mit der Bearbeitung beginnen, während Teile des Projektes noch geladen werden. Die Visualisierungsplattform besitzt 5 wesentliche Komponenten, um die Anforderungen aus 4.1 und 4.2 zu erfüllen. Diese Komponenten sind:

1. *Scene*: Diese Komponente ist verantwortlich zur Darstellung des 2D-Kartenmaterials und der 3D-Modelle. Weiterhin können Komponenten auf diese Komponente zugreifen, um Daten innerhalb der Szene zu ändern. Modelle können in der Szene durch Drag&Drop eingepflegt werden. GeoJSON Layer werden ebenfalls durch Drag&Drop hinzugefügt und sind automatisch bei dem nächsten Szenenstart aktiv.
2. *File-Browser*: Diese Komponente erlaubt die Erstellung, Bearbeitung und Verwaltung projektinterner Dateien. Hier können in einer selbst gewählten Ordnerstruktur Dokumente, Modelle, Bilder, Szenen und weitere Daten angelegt werden.
3. *Discussion-Board*: Zur zentralen Verwaltung der Kommentare und Diskussionen bietet das System ein Diskussionsforum an. Nutzer können neue Threads erstellen, sich austauschen und ihre Kommentare in der 3D-Szene verlinken.
4. *Visualizer*: Eines der zentralen Bestandteile der Plattform ist ein funk-

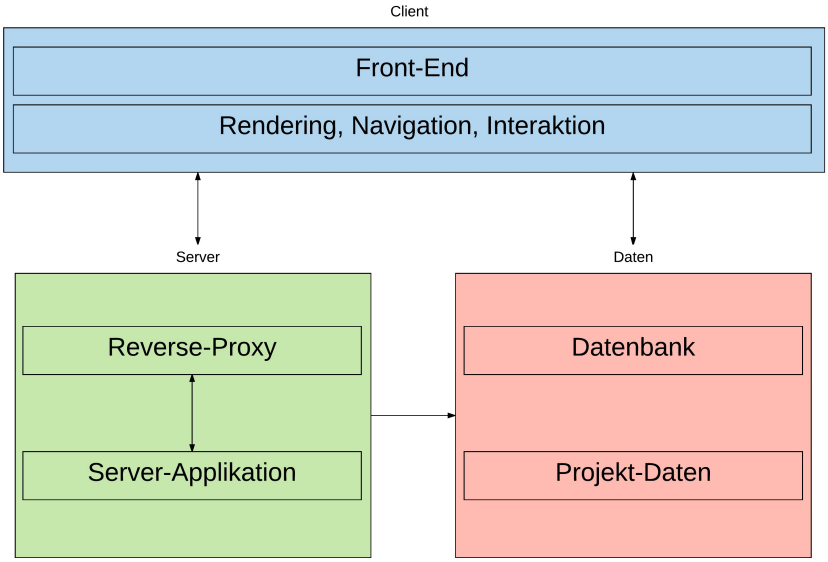


Abbildung 4.4: Schichten der Architektur der konzipierten Visualisierungsplattform (Eigene Darstellung).

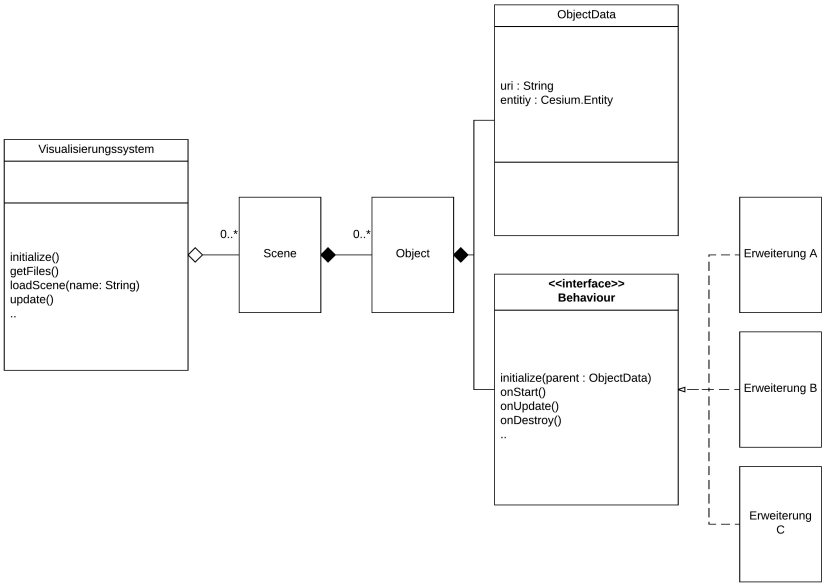


Abbildung 4.5: Übersicht der Erweiterungsmöglichkeit des Visualisierungssystems (Eigene Darstellung).

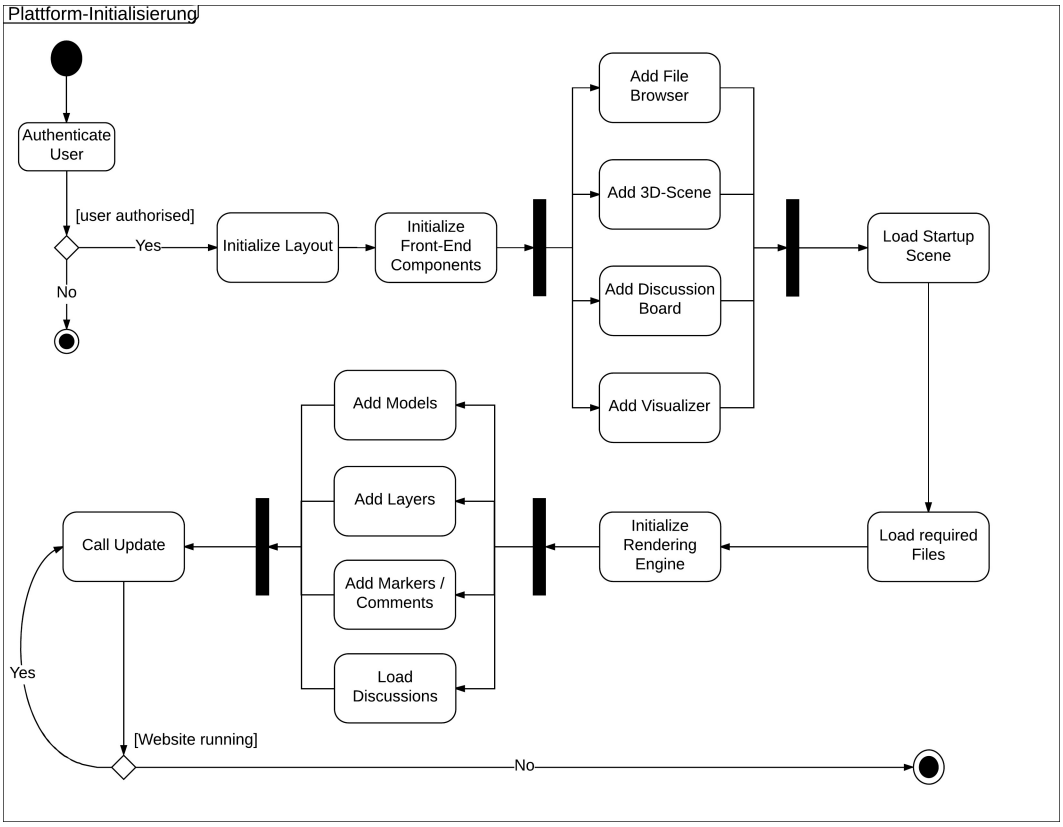


Abbildung 4.6: Aktivitätsdiagram der Visualisierungsplattform, welche die Initialisierung des Systems darstellt (Eigene Darstellung).

tionsfähiges Tool zur Visualisierung von Infrastrukturprojekten. Dieses Tool kann, wie bereits beschrieben, beliebig erweitert werden. Im ersten Entwurf ist das Visualisierungstool auf Straßen und Trassen ausgerichtet. Es können Splines erstellt, bearbeitet und verwaltet werden. Weiterhin können auf Basis dieser Splines Fahrzeuge, Straßen und Trassen platziert werden. Außerdem bietet das Visualisierungstool die Möglichkeit Marker zu setzen, welche in der 3D-Szene durch Billboards dargestellt werden.

4.5.2 Front-End

Lösungsvarianten

Die, für den Nutzer, sichtbarste Schicht stellt das Front-End dar. Eine moderne Webapplikation rendert das Frontend mithilfe von HTML5, CSS3 und Javascript und muss aufgrund der verschiedensten Displaygrößen responsive sein, d.h. die grafische Darstellung der Website wird automatisch an den

Anforderungen des Endgerätes angepasst. Grundlegend wird das Gerüst aus HTML5, CSS3 und Javascript dynamisch auf der Serverseite erzeugt und bei entsprechender Authentifizierung an den Client gesendet. HTML5 wird innerhalb des Servers durch eine Template-Engine generiert, welche benutzerdefinierte Daten in ein Web-Template einfügt. Ein Web-Template-System erlaubt dadurch die Verbindung von statischen Daten, welche die Struktur vorgibt, mit dynamischen Daten, die auf den Benutzer zugeschnitten sind. Web-Template-Systeme zur Erzeugung von HTML5 für Javascript sind beispielsweise *Mustache*, *Pug* und *Handlebars* [77]. Für CSS3 existieren ebenfalls Präprozessoren, die für die Erzeugung von Cascading Style Sheets verantwortlich sind. Dazu gehören *Sass*, *Less* oder *Stylus*.

Javascript wurde als ECMAScript standardisiert. Die aktuellen Sprachfeatures neuer ECMAScript-Versionen werden jedoch nicht von allen Browsern unterstützt, weshalb sogenannte *Transpiler* existieren, die nicht unterstützte Sprachfeatures in eine kompatible Version übersetzen [60] (s.a. Listing B.3 und B.4). Auch hier sind verschiedene Optionen denkbar, unter anderem *CoffeScript*, *TypeScript*, *ES2018* über *Babel* oder *Traceur*. Sprachen, wie *TypeScript* oder *ES2018* unterstützen Module, welche von aktuellen JavaScript-Versionen nicht unterstützt werden. Diese Module müssen mit geeigneten Plugins gebündelt werden. Hierfür stehen bspw. *Webpack*, *Bower* oder *Browserify* zur Verfügung, welche geschriebene Module in Javascript übersetzen und in einer Datei bündeln, die von allen gängigen Browsern eingelesen werden kann. Die Bündelung kann weiterhin konfiguriert werden, um die Performance zu optimieren. Beispielsweise werden Webdokumente, wie HTML, CSS und Javascript-Dateien, wie in den Beispiel-Listings B.1 und B.2 komprimiert.

ID	**Kategorie**	**Framework**
0	HTML-Template-Engine	Mustache
1	HTML-Template-Engine	Pug
2	HTML-Template-Engine	Handlebars
3	CSS-Präprozessor	Sass
4	CSS-Präprozessor	Less
5	CSS-Präprozessor	Stylus
6	Sprache	CoffeScript
7	Sprache	TypeScript
8	Sprache	ECMAScript2015
9	Bundler	Webpack, Babel
10	Bundler	Bower
11	Bundler	Browserify
12	UI-Framework	VueJS
13	UI-Framework	ReactJS
14	UI-Framework	Angular
15	Panel-Framework	DockSpawn
16	Panel-Framework	wcDocker
17	Panel-Framework	GoldenLayout

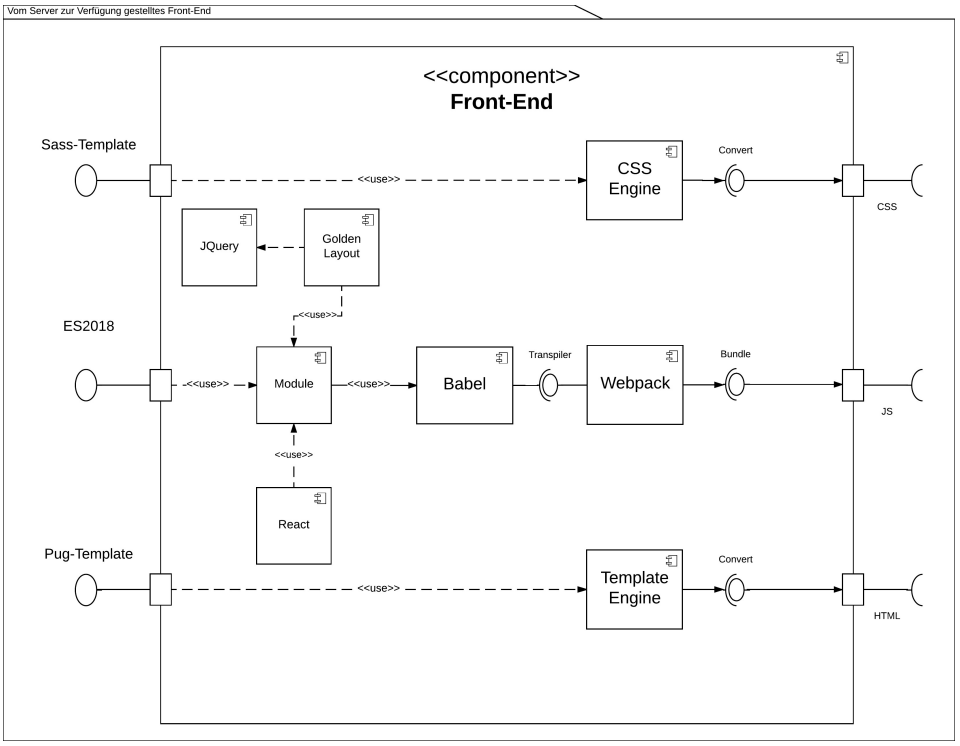


Abbildung 4.7: Komponentendiagramm zur Lösung der Anforderungen für das Front-End. Zuerst werden die ES2018 Module in Javascript übersetzt, gebündelt und minifiziert, gleichzeitig erzeugen Präprozessoren CSS-Sheets und HTML-Dokumente. (Eigene Darstellung)

Tabelle 4.3: Optionen für das Front-End der Webapplikation zur Erfüllung der Anforderungen aus 4.1 u. 4.2.

Neben den genannten Präprozessoren und Template-Engines ist ein UI-Framework für komplexe Webapplikationen entscheidend. Das populärste Webframework ist *JQuery*. Weiterhin ist für Data-Binding, Domain Object Model Manipulation und User-Interfaces ein Framework, wie *VueJS*, *ReactJS* oder *Angular* eine Option. Schlussendlich existiert bisher keine standardisiertes JavaScript-Panel, weshalb auch hier Frameworks notwendig sind, um die Anforderungen der Applikation zu erfüllen. Die populärsten Optionen sind hier *DockSpawn*, *wcDocker* und *GoldenLayout*. Die genannten Optionen für das Frontend sind in Tabelle 4.3 zusammengefasst.

Gewählte Lösung

Um die Anforderungen zu erfüllen und eine zuverlässige und stabile Webarchitektur abzubilden, sind mitunter Erfahrung und Präferenzen hinsichtlich der Programmierparadigmen und Sprachen einzubeziehen. Abb. 4.7 zeigt ein Komponentendiagram für die gewählte Lösung. *ReactJS* (hergestellt von Facebook) gilt mitunter als stabiles User Interface Framework, welches von Websites, wie Facebook, Instagram und Netflix verwendet wird [43, 45, 91]. Für die gewählte Lösung wird daher *ReactJS* verwendet, wobei auch *VueJS* und *Angular* bewährte Frameworks darstellen. Durch Verwendung von React ist es notwendig, dass auch das gewählte JavaScript-Panel Framework *React* unterstützt. *DockSpawn* hat zum April 2015 das letzte Update erhalten, weshalb es aus dieser Kategorie herausfällt [40]. Aus der Dokumentation von *wcDocker* geht nicht klar hervor, ob ReactJS unterstützt wird [74]. Im Gegensatz dazu unterstützt *GoldenLayout* React, weshalb jenes für die gewählte Lösung eingesetzt wird. Weiterhin bedingt die Verwendung von *GoldenLayout* das *JQuery* eingebunden wird. Für die JavaScript-Erzeugung wird in diesem Konzept davon ausgegangen, dass die Sprachfeatures von *ES2018* in allen Browsern in Zukunft umgesetzt werden, da ECMAScript die standardisierte Programmiersprache des Webs darstellt [47]. Daher wird als Programmiersprache ES2018 mit dem Transpiler *Babel* und *Webpack* eingesetzt. Die vorgestellten HTML-Template Engines und CSS-Präprozessoren werden von allen bekannten Server-Technologien unterstützt, weshalb hier die Wahl zwischen Mustache, Pug und Handlebars, sowie Sass, Less und Stylus beliebig ist. Für die gewählte Lösung wird Pug als HTML-Template-Engine und Sass als CSS-Präprozessor verwendet. lin

4.5.3 Rendering, Navigation und Interaktion

Lösungsvarianten

Das *Rendering* stellt einen essentiellen Teil des zu konzipierenden Visualisierungssystems dar. Es existieren im wesentlichen zwei verschiedene Rendering-Optionen. Zum einen kann das Rendering auf das Backend ausgelagert werden (*Headless Rendering*), sodass Geräte mit geringer Leistungsfähigkeit ebenfalls von einer Visualisierung profitieren. Zum anderen kann das Rendering auch durch die bereits vorgestellten Frameworks und Engines direkt über die Grafikkomponente der Endgeräte erfolgen. Weiterhin existieren mehrere Optionen für die Darstellung der 3D-Welt im Client. Eine Option ist die Entwicklung einer neuen Visual-Globe und 2D/3D-Karten-Engine über *WebGL*. Weitere Optionen stellen die in Kapitel 3 bereits vorgestellten Frameworks *CesiumJS*, *ThreeJS* und *BabylonJS* dar. 3D-Engines, wie Unity und Unreal werden in diesem Vergleich nicht als Lösungsvarianten berücksichtigt, da sie primär auf Spiele ausgelegt sind und die Features hinsichtlich von Geoinformationssystemen beschränkt sind. Weiterhin sind die entwickel-

ten 'Applikationen' starr in dem Sinn, das eine Änderung einer Szene nur im jeweiligen Windows-Editor stattfinden kann. Die Anforderungen zeigen jedoch, dass eine Änderung jederzeit webbasiert oder mobil stattfinden muss. Dies ist in Verbindung mit einer nativen Engine nicht möglich.

Gewählte Lösung

Ein Headless-Rendering System führt zu einer erhöhten Last der Server. Bei einer großen Anzahl von Nutzern kann dies zu unbeabsichtigten Einschränkungen in der Stabilität und Bildrate führen. Es ist davon auszugehen, dass die öffentliche Beteiligung der Infrastrukturprojekte sehr hoch ist, weshalb die Serverlast so gering wie möglich gehalten werden sollte und für das konzipierte System ein clientseitiges Rendering-Framework verwendet wird. Um jedoch mobile Geräte zu unterstützen, die über keine dedizierte Grafikkarte verfügen, ist eine Erweiterung des Systems um einen Headless-Renderer denkbar. Für diese Erweiterung sind alle Eingaben des Nutzers an den Server zu übermitteln. Aus den Eingaben resultierend wird anhand des aktuellen Status der Szene ein Bild gerendert. Dieses Rendering muss zur Gewährleistung der Usability eine Bildfrequenz von mindestens 24 Bildern pro Sekunde übertragen. In diesem Konzept wird nicht näher auf das Headless-Rendering eingegangen. Für eine Entwicklung auf Basis eines 3D-Frameworks, wie *ThreeJS*, muss das gewählte Framework erst um georeferenzierte Karten und Visual-Globe Funktionen erweitert werden. Da *CesiumJS* eine stabile und aktiv-entwickelte Open-Source Bibliothek für 3D Globes und Maps darstellt, wird für dieses Konzept *CesiumJS* verwendet. Für die Navigation und Interaktion im 3D-Raum sind typische Techniken zu unterstützen. Die Navigation findet hierbei vorrangig über Maus und Tastatur statt. Über Maus können Objekte selektiert, rotiert und transformiert werden. Weiterhin lässt sich die 3D-Kamera über die Tasten W (vor), A (links), S (rechts) und D (zurück) steuern. Um mobile Geräte zu unterstützen, ist eine rudimentäre Touch-Unterstützung notwendig.

4.5.4 Server

Lösungsvarianten

Die intern verwendete *Server-Applikation* ist für die Logik der Applikation zuständig. Die Hauptaufgabe dieser Schicht ist eine API auf Basis von REST zu bieten, mit der die Clientapplikation kommunizieren kann. Vom Server erhält der Client die notwendigen URL's zur Übertragung der 3D-Modelle, Dokumente und Projekte. Die Authentifizierung, Projektverwaltung und Ressourcen-Akquise geschieht über den Server. Außerdem stellt sie den Knotenpunkt aller anderen Technologien dar, da die Datenspeicherung und Datenbankanbindung von ihr abhängt. Für die Server-Applikation existieren verschiedene Technologien. Häufig verwendete Technologien sind *NodeJS* in

Kombination mit *ExpressJS* oder *SailsJS*, *Django*, *Ruby on Rails*, *ASP.net* und *PHP*. Ein *Reverse-Proxy* stellt eine erste Kommunikationsschnittstelle der Webapplikation mit den Webservern dar. Diese werden vorallem für die effiziente Verteilung statischer Daten, Netzsicherheit und Lastverteilung eingesetzt [92]. Ein einmalig eingerichter Reverse-Proxy erlaubt hierbei die Verteilung der Last auf mehrere Server. Es ist auch hier davon auszugehen, dass die hohe Last der Verarbeitungsschritte dreidimensionaler Infrastrukturprojekte mit Bürgerbeteiligung und Social Media Features vom Einsatz eines Reverse-Proxys profitiert. Open-Source-Lösungen sind hier bspw. der *Apache Traffic Server* (Apache TS), *HAProxy*, *NGINX* und *Varnish*.

Gewählte Lösung

Die Auswahl einer geeigneten Servertechnologie ist vorallem im Hinblick auf Stabilität, Zuverlässigkeit und Performance entscheidend. Aufgrund der Tatsache, dass *PHP* eine stark veraltete Technologie darstellt, wird diese für die Architektur nicht verwendet. *ASP.net* ist auch in der neusten Version '*ASP.net Core*', ebenso wie *NodeJS* und *Ruby on Rails*, Open-Source Software. ASP.net basiert auf auf dem *.NET Framework* und ist nur für Windows Server erhältlich, ASP.net Core hingegen auch für Linux. Da *ASP.net Core* jedoch eine sehr neue Technologie darstellt, die aktuell in 2.0-Preview Status ist, und die Anforderungen eine stabile und zuverlässige Plattform erfordern, ist ASP.net Core aktuell nicht geeignet. Im Bezug auf *NodeJS*, *Django* und *Ruby on Rails* schreibt Ryan Paul:

> To build this streamlined middle tier, the LinkedIn developers wanted to use a lightweight event-driven framework. They also wanted an environment that would be well-suited to handle the aggregation and string interpolation capabilities required for the service. They tested several candidates, including Ruby with EventMachine, Python with Twisted, and the JavaScript-based Node.js framework. They found that Node.js, which is based on Google's V8 JavaScript engine, offered substantially better performance and lower memory overhead than the other options being considered. Prasad said that Node.js "blew away" the performance of the alternatives, running as much as 20 times faster in some scenarios. [64]

Für das vorliegende Konzept wird daher *NodeJS* als Serverframework genutzt. Um Routes effizient bearbeiten zu können, wird NodeJS zusätzlich um *ExpressJS* ergänzt. Die Wahl des Reverse-Proxy fällt anhand der Anforderungen schwieriger, da alle vorgestellten Lösungen flexibel, schnell und stabil sind. Da dieser jedoch während der Implementierung schnell geändert werden kann, wird für dieses Konzept angenommen, dass *NGINX* verwendet wird.

4.5.5 Daten

Lösungsvarianten

Die Datenschicht ist der wichtigste Teil der Visualisierungsplattform. Hierzu gehört neben der Datenbankverwaltung ein System zur Speicherung großer Datenmengen, wie 3D-Modelle, Skripte und Dokumente. Das gesamte Szenario der Autobahn A14 der Fraunhofer VR-Plattform umfasst 11,4 Gigabyte [12]. Zum einen muss diese Datenmenge zwangsläufig durch geeignete Algorithmen verkleinert werden, zum anderen führt die Bearbeitung mehrerer Projekte mithilfe der Plattform dazu, dass auch die Daten, die nicht in der Datenbank erfasst werden das ACID-Prinzip verteilter Systeme zu unterstützen hat [75]. Die Aufgaben sind abgeleitet aus den Anforderungen in 4.1 folglich:

1. *Konsistenz*: Befehlssequenzen zum Speichern und Abfragen von Dateien führen nach Beendigung zu einem konsistenten Datenzustand, falls der Ausgangszustand konsistent war.
2. *Atomarität*: Es wird davon ausgegangen, dass eine Transaktion aus mehreren Befehlen eine atomare Operation im System darstellt. Eine solche Transaktion kann bspw. eine Reihe von Lösch- und Kopierbefehlen darstellen. Eine Transaktion ist nur dann vollständig abgeschlossen, wenn die Befehle sequenziell ohne Fehler ausgeführt werden können. Schlägt eine Transaktion fehl, wird der Ursprungszustand der Daten wiederhergestellt.
3. *Isolation*: Transaktionen können sich nicht gegenseitig beeinflussen oder behindern, d.h. das gleichzeitige Löschen und Lesen von Daten wird durch eine geeignete Rechtevergabe verhindert.
4. *Dauerhaftigkeit*: Nach Abschluss einer Transaktion sind die Daten dauerhaft und zuverlässig gespeichert. Ein Ausfall des Systems führt nicht zu einem Datenverlust.

Zu den gespeicherten Daten des Systems gehören folglich:

1. 3D-Modelle (.obj, .fbx, ..)
2. Texturen und Bilder unterschiedlicher Dateiformate (.png, .jpg, ..)
3. Dokumente (.pdf, .docx, .txt)
4. Benutzerdefinierte Inhalte (.pug, .sass, .js)
5. Erweiterungsmodule (.js)

Zwei primäre Lösungsvarianten können die soeben gestellten Anforderungen erfüllen.

Die erste Variante ist ein Array aus Servern, die auf jeden verwendeten Server Projekte speichert und verwaltet. Abb. 4.8 zeigt ein Beispiel, wie eine solche Architektur aussehen kann. Ein Hauptserver leitet hierbei alle Anfragen der Clients auf den richtigen Datenserver um, auf den das gesuchte

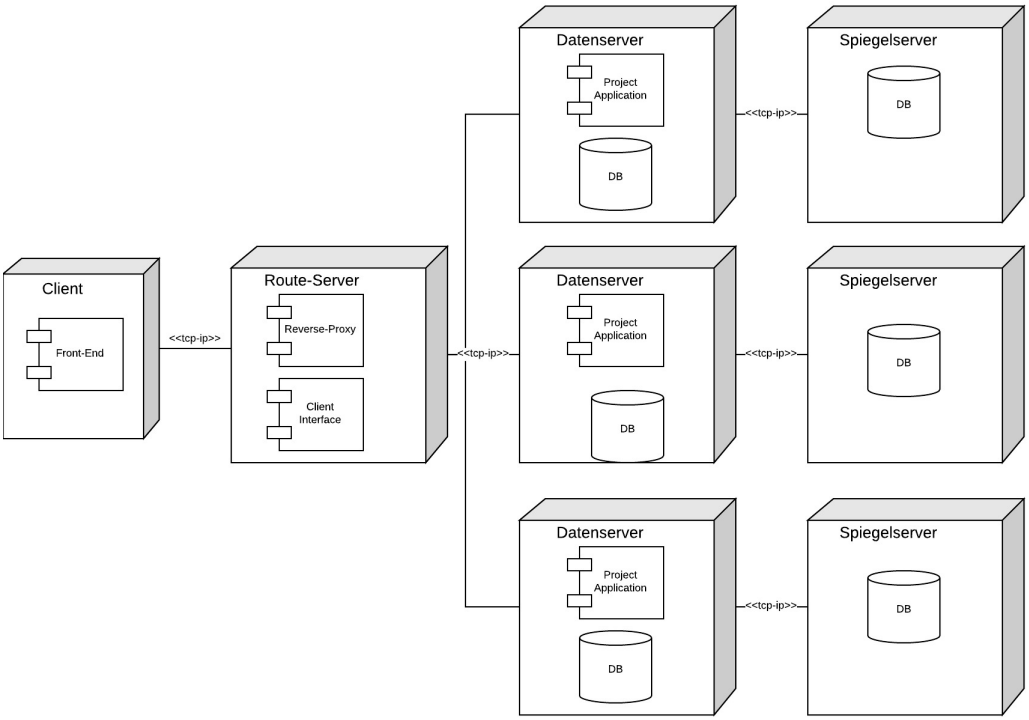


Abbildung 4.8: Darstellung einer Serverarchitektur zur Speicherung von hochauflösenden, benutzerdefinierten 3D-Projekten. Ein Hauptserver verwaltet eine Kette von Datenservern. Jeder Datenserver speichert eine gewisse Anzahl an Projekten und alle Anfragen werden passend weitergeleitet. Die Datenserver werden zusätzlich regelmäßig gespiegelt, um Dauerhaftigkeit zu gewährleisten. (Eigene Darstellung)

Projekt abgelegt ist. Erstellt der Nutzer ein neues Projekt, wird im Hauptserver ein neuer Eintrag in der Datenbank abgelegt und ein Datenserver ausgesucht, der die zurzeit geringste Auslastung besitzt. Die Datenbank im Hintergrund führt dazu, dass die Konsistenz gewährleistet wird. Zur Gewährleistung der Dauerhaftigkeit wird für alle Datenserver ein Spiegelserver eingerichtet, um einem Systemausfall vorzubeugen. Die hier beschriebene Variante besitzt einige herausfordernde Probleme:

1. *Echtzeitfähigkeit*: Das System soll mit Bezug auf die Anforderungen aus Tabelle 4.2 effizient und echtzeitfähig sein. Der Zugriff auf das System aus großer Entfernung ist in diesem Fall nicht echtzeitfähig, da hohe Latenzzeiten abhängig vom Standort der Server zu erwarten ist. Weiterhin führt das Speichern der Projekte auf einzelne Datenser-

ver dazu, dass die Auslastung der Server zu jeder Zeit Schwankungen unterliegt. Bspw. kann die Auslastung eines Servers während der Bearbeitungszeit minimal sein. Werden jedoch genügend Projekte zur Bürgerbeteiligung freigeschalten, kann das gesamte System einbrechen.

2. *Wartungsaufwand*: Der Aufwand zur Wartung dieser Serverarchitektur ist sehr hoch. Während des laufenden Betriebs können Server ausfallen und neue Server in Betrieb genommen werden. Ein Update der Server-API des Systems muss auf allen verwendeten Systemen installiert werden.
3. *Speicherkapazität*: Da ein Projekt nicht in seiner Größe beschränkt ist, ist das Risiko vorhanden, dass die Festplatte eines Datenservers während der Bearbeitungszeit keinen weiteren Speicherplatz besitzt. Dies führt dazu, dass auch alle anderen Projekte des Servers keine neuen Dateien aufnehmen können. Die Beschränkung der Größe umgeht das Problem, jedoch bleiben in diesem Fall Speicherreserven ungenutzt, wenn ein Großteil der Projekte nur einen Bruchteil des Speichers benötigt.

Die zweite Variante zur Lösung der der gestellten Anforderungen nutzt moderne Cloudtechnologien. Abb. 4.9 stellt die Architektur übersichtlich dar. Sogenannte Cloud-Storages speichern und verwalten Projekte auf Basis einer bestehenden Serverinfrastruktur. Jeder Cloud-Storage besitzt dabei eine eigene API, um die Daten zu bearbeiten. Darauf aufbauend stehen sogenannte Content-Delivery-Networks (CDN), welche auf die Cloud-Storages zugreifen und die Daten auf Edge-Standorte spiegeln. Egde-Standorte sind *ein globales Netzwerk* aus Servern in Schlüsselpositionen. Der Zugriff auf ein Projekt wird dabei auf den nächstgelegenden Edge-Standort weitergeleitet. Damit ist ein performanter, latenzfreier und effizienter Zugriff sichergestellt. Für diese Variante ist es weiterhin notwendig, dass nicht der eigene Hauptserver die Projektdaten zum Client sendet, sondern stattdessen den Client für eine gewisse Zeit authorisiert, die Projektdateien direkt vom CDN herunterzuladen. Diese Datenverteilung ist in Abb. 4.10 als Flussdiagramm dargestellt. Bewährte Cloud-Systeme und CDN's sind unter anderem:

1. Google Cloud Storage, welches bspw. von Spotify, Niantic und Airbus Defense & Space verwendet wird [50].
2. Akamai, welches bspw. von Adobe TV, Airbnb und Autodesk verwendet wird [28].
3. Amazon Web Services (AWS), welche die Basis für Netflix, Reddit, Shazam und Soundcloud bilden [30].

Gewählte Lösung

Hinsichtlich der Probleme der Variante 1 wird in diesem Konzept die zweite Variante für die Architektur genutzt. Zusätzlich wird davon ausgegan-

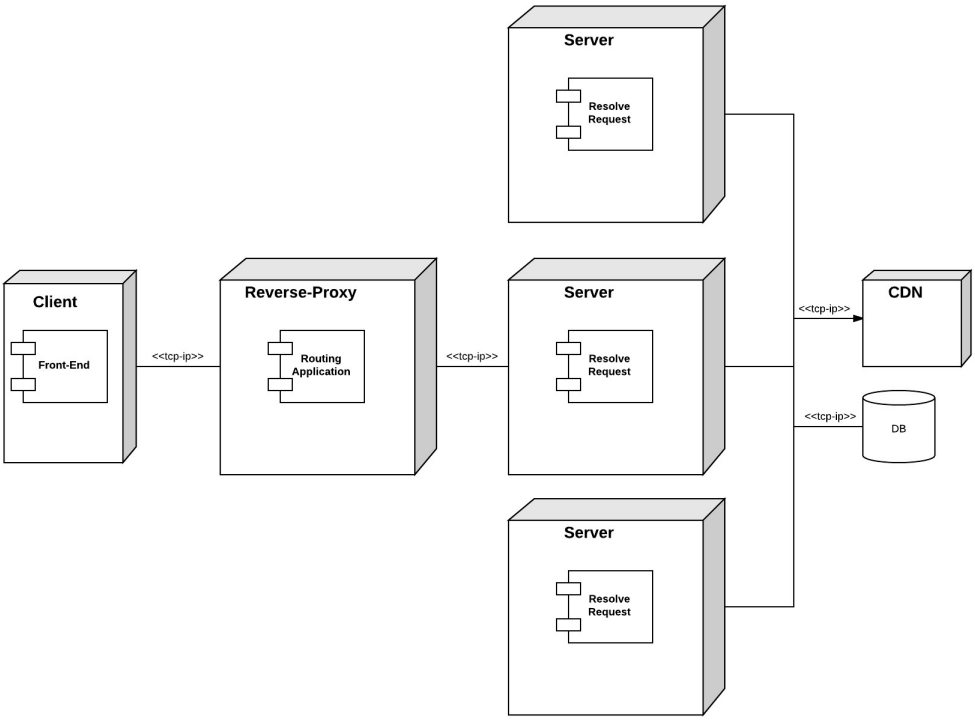


Abbildung 4.9: Darstellung einer Serverarchitektur auf Basis von Cloud-Services. Content-Delivery-Networks werden dabei ausschlaggebend eingesetzt. Die Datenbank und Daten sind von der Logik klar getrennt, um das SOLID Prinzip zu gewährleisten. (Eigene Darstellung)

gen, dass die Datenverteilung vom Cloud-Storage nicht über den Webserver umgeleitet wird, sondern direkt, wie in Abb. 4.10, beim Client stattfindet. Angesichts der drei Cloud-Alternativen ist die Wahl ähnlich der Servertechnologien nicht trivial. Die Tatsache, dass, laut eines Reports von Sandvine Incorporated, Netflix für 35.15% des gesamten Downstream-Traffics in Nordamerika verantwortlich ist und Netflix auf Amazon AWS basiert, stützt sich dieses Konzept folglich auf Amazon Web Services [13, S.4, Fig. 1].

Die verwendeten Technologien für die Datenschicht sind dementsprechend AWS S3 für die Cloud-Speicherung und AWS Cloud-Front als Content-Delivery-Network. AWS S3 speichert Daten als Objekte, die als Buckets bezeichnet werden. In einer Bucket können beliebig viele Objekte geschrieben, gelesen und gelöscht werden. Der Zugriff kann außerdem durch Signaturen kontrolliert werden, sodass der Hauptserver der Website nur authentifizierten Nutzern den Zugriff auf die Projektdateien erlaubt [29]. Für das Konzept

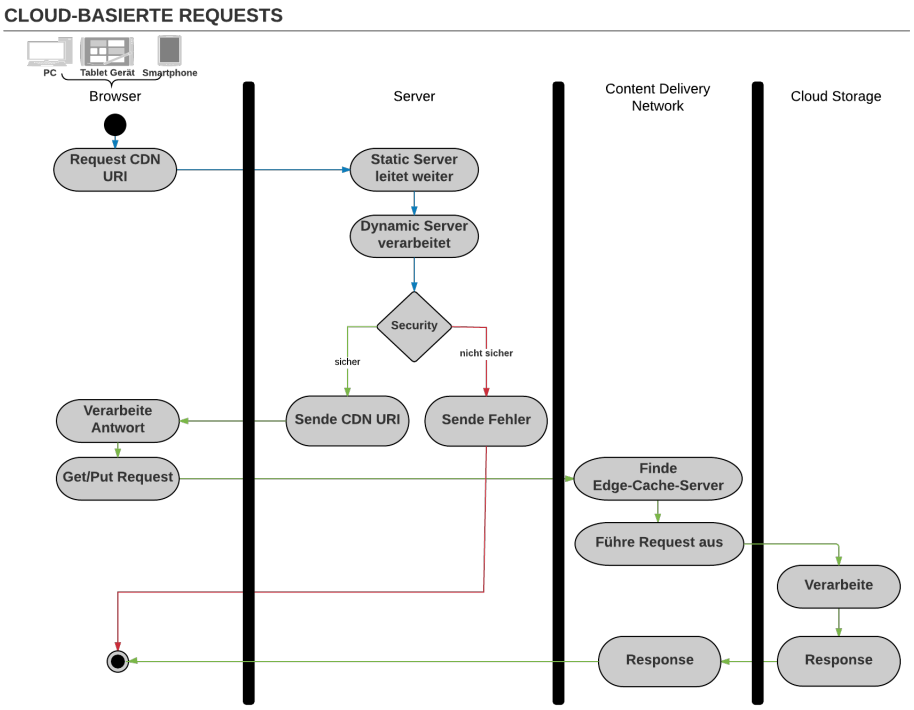


Abbildung 4.10: Flussdiagramm zur performanten Verteilung von Daten über Content-Delivery-Networks. Der Browser führt zwei Requests aus, um eine Datei zu erhalten. Der erste Request geht an den Server, um eine sichere Adresse der Datei im CDN zu erhalten, der zweite Request fragt die eigentliche Datei ab. (Eigene Darstellung)

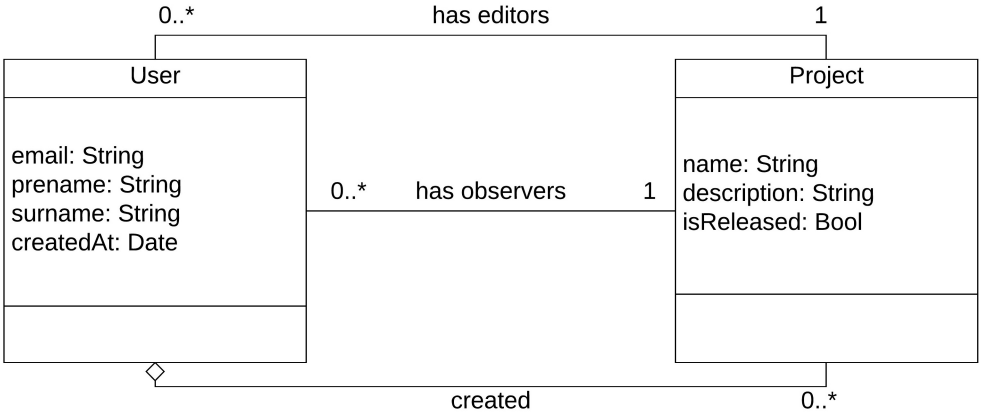


Abbildung 4.11: Simples Datenbankschema zur Verwaltung der Nutzer und Projekte (Eigene Darstellung)

werden simple JSON-basierte Dateistrukturen verwendet. Eine Szene speichert Modelle im glTF-Format, Polylinien, sowie Polygone, GeoJSON-Layer und KML-Layer. Die Modelle im glTF-Format referenzieren wiederum alle benötigten Daten auf Shader, Texturen und binäre Buffer selbstständig. Eine Szene kann im Verlauf eines Projektes zusätzlich um weitere Daten erweitert werden, die nicht in diesem Konzept aufgenommen wurden. Ein Beispiel für solche Daten sind Heatmaps, zusätzliche georeferenzierte Orthophotos und Geländemodelle usw. Im Kapitel 5 wird das Szenenformat anhand einer Beispielimplementierung dargestellt.
Zur Speicherung von Projekten, Accounts und deren Metadaten kann eine Datenbank verwendet werden. Damit alle Server Zugriff auf diese Datenbank haben, ist eine extern eingerichtete Datenbank vorteilhaft. Welche Datenbank genutzt wird, ist in diesem Konzept nicht entscheidend, da das verwendete Projektschema, welches in 4.11 abgebildet ist, sehr simpel gehalten ist. Es kann zwischen zwei Alternativen entschieden werden. Als erste Möglichkeit ist eine traditionelle SQL-Datenbank bspw. Azure mit SQL-Befehlen zu empfehlen. Andererseits ist es im Projektverlauf auch möglich, dass nicht-relationale Daten in der Datenbank abgespeichert werden. Dann ist eine nicht-relationale Datenbank, wie MongoDB oder DynamoDB zu empfehlen.

4.5.6 Zusammenfassung

In der Abb. 4.9 ist die Server-Client-Architektur des Systems anhand der hergeleiteten Lösungen zusammengefasst. Das Frontend wird mithilfe von ReactJS und GoldenLayout Modulen präsentiert. Der Reverse-Proxy NGINX leitet je nach Auslastung eingehende HTTP-Requests an verschiedene Server weiter und liefert statische Webdokumente an den Client. Umgeleitete HTTP-Requests werden von einer NodeJS Applikation bearbeitet, diese liefert authorisierte Zugriffe auf ein Content-Delivery-Network. Projektspezifische Ressourcen können mithilfe dieser authorisierten URL's von AWS S3 und AWS Cloudfront angefordert werden.

URL	**HTTP**	**Result**
/	GET	Gibt die Index-Seite wieder. Wenn der Nutzer authentifiziert ist, wird stattdessen das Dashboard zurückgegeben.
/watch/:creator/:project	GET	Wenn das angegebene Projekt vom angegebenen Nutzer freigegeben wurde, wird eine Read-Only Version des Projektes geliefert.
/sign-in	GET	Gibt die Website zum Log-In wieder.
/sign-up	GET	Gibt die Website zur Registrierung wieder

/dashboard	GET	Erstellt die spezifische Dashboard Seite des Nutzers und listet alle Projekte auf, die von ihm erstellt oder für ihn freigegeben wurden.
/dashboard/:projectName	GET	Gibt das spezifische Projekt zur Bearbeitung wieder.
/getObject?key=keyToFile	GET	Gibt eine signierte Adresse zum geforderten Objekt in AWS S3 zurück. Wenn des Objekt nicht existiert, sende Error 404.
/get/:creator/:project/:filePath*	GET	Gibt eine signierte Adresse zu einer Datei mit dem gewählten Dateipfad zurück. Sendet Error 404 bei Nicht-Existenz.
`/^\/getList\/(.*)/`	GET	Nutzt einen regulären Ausdruck (auch Regex genannt), um eine Liste von Dateien in einem Ordner wieder zu geben.
/putObject?key=name& contentType=type	POST	Erstellt ein Objekt in AWS S3 am angegebenen Pfad.
/sign-up	POST	Registriert einen neuen Nutzer. Die Daten zur Registrierung sind Vorname, Nachname, Email, Passwort und Benutzername. Die Daten werden im Header der Nachricht an den Server übermittelt.
/sign-in	POST	Authentifiziert einen Nutzer anhand der eingegebenen Daten. Bei Erfolg wird dieser auf den Dashboard weitergeleitet.
/sign-out	POST	Löscht die Sitzung eines Nutzers und erlaubt keinen weiteren Zugriff ohne neue Authentifizierung.
/createProject	POST	Erstellt ein neues Projekt mit dem angegebenen Namen.
/deleteProject/:projectName	DELETE	Löscht ein Projekt vom Konto des Nutzers.

Tabelle 4.4: REST-API des konzipierten Systems, welche für die Authorisierung und Ressourcenverteilung zuständig ist.

Untereinander kommunizieren die Komponenten über HTTP-Requests. In der Tabelle 4.4 ist eine Übersicht über die wichtigsten Funktionen des Servers aufgelistet. Darunter fallen, wie in den Anforderungen festgelegt, die Authentifizierung, Projekt-, Nutzer- und Ressourcenverwaltung. Abb. 4.12

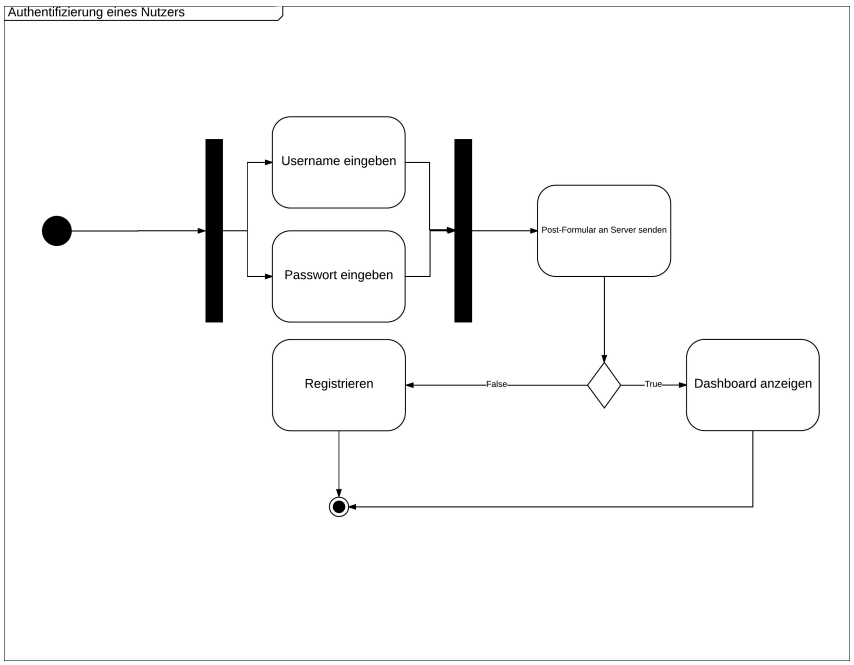


Abbildung 4.12: Aktivitätsdiagramm zur Authentifizierung eines Nutzers. (Eigene Darstellung)

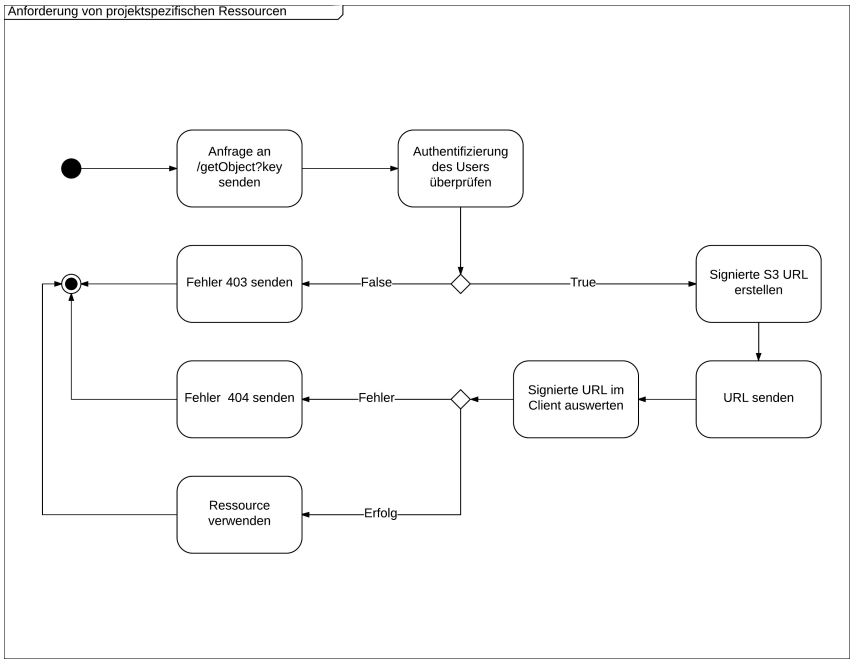


Abbildung 4.13: Aktivitäten zur Anforderungen von projektspezifischen Ressourcen über /getObject?key. (Eigene Darstellung)

stellt die Aktivitäten des Servers zur Authentifizierung eines Nutzers dar. Abb. 4.13 zeigt weiterhin genauer, wie der Server eine Liste von Projektressourcen von AWS S3 anfordert. Für die Verbindung zwischen Projekten und Nutzern werden die Datenbankschema, wie in Abb. 4.11 verwendet. Die vorgestellten Schema sind nach den Anforderungen aus Kapitel 4.3 aufgebaut. Das User Interface des Systems ist geradlinig. Das Mockup B.1 zeigt beispielhaft, wie die UI aufgebaut sein kann. Zentraler Bestandteil des Mockups stellt die Kartenansicht dar, welche eine wechselbare zweidimensionale bzw. dreidimensionale Karte beinhaltet. Beim Wechsel in eine dreidimensionale Ansicht wird ein Visual Globe gerendert; außerdem wechselt die Kamera automatisch von einer orthographischen in eine perspektivische Projektion. Oberhalb der Szenenansicht befinden sich zentrale UI-Funktionen zur Navigation und Bearbeitung der dargestellten Fenster. Weiterhin können zusätzliche Funktionen, über das Navigationsmenu realisiert werden. Innerhalb der Szenenansicht befindet sich ein *Visualizer-Panel*, welches je nach Szenario unterschiedliche Funktionen bereitstellt. Diese Funktionen sind maßgeblich davon bestimmt, welches Ziel das gewählte Szenario verfolgt. Im Mockup sind beispielhaft die Funktionen *kommentieren* und *bewerten* aufgeführt. Links von der Szenenansicht befindet sich der Projektexplorer, welche alle benötigten Projektdaten, Dokumente und Pläne bereitstellt. Weiterhin lassen sich von diesem Explorer aus 3D-Modelle einbinden. Beim Klick auf einer Szenendatei (.scene) wird eine neue Szene automatisch geladen. Unterhalb des Projektexplorers befindet sich das Diskussionsforum, welches, wie im Konzept beschrieben, vorrangig zur Bürgerbeteiligung genutzt wird. Die Kommentare lassen sich georefenzieren und in der Szenenansicht orten.
In diesem Kapitel wurde eine Server-Client-Architektur konzeptioniert, welche die hohen Anforderungen an eine Plattform zur Infrastrukturentwicklung erfüllt. Dabei sind wesentliche Bestandteile, die Übertragung hochauflösender Daten mithilfe einer REST-basierten Anbindung an einen Cloudserver, die Darstellung von 2D- und 3D-Szenen über einen WebGL-Renderer und die sichere Verteilung der Szenarien auf multiple Nutzer.

Kapitel 5

Prototypische Realisierung

Im vorherigen Kapitel wurde eine Plattform zur Visualisierung von Infrastrukturprojekten konzipiert. In diesem Abschnitt wird folglich eine prototypische Realisierung dieses Konzepts vorgenommen. Als erstes wird eine Übersicht über das Projekt dargelegt, anschließend werden serverseitige und clientseitige Komponenten beschrieben.

5.1 Übersicht

Das erstellte Projekt befindet sich in Gitlab unter dem Repository https://gitlab.com/pschladi/vreya als Open-Source Projekt. Das Projekt lässt sich weiterhin unter www.vreya.de besichtigen. Das Projekt gliedert sich in folgende Bereiche:

1. NodeJS Server Code mit Routing Mechanismen, um einen flüssigen Website-Betrieb zu gewährleisten.
2. ReactJS Frontend Code zur Darstellung der Website im Client.
3. PugJS und Sass Dokumente, welche durch NodeJS zu HTML und CSS konvertiert werden.

Weiterhin befinden sich im Projekt einige Konfigurationseinstellungen zur Anbindung an Amazon AWS S3 und zur Konfiguration von Webpack, welches verantwortlich für die Konvertierung von ECMA-Script 2018 ist. Im Anhang unter Listing C.1 ist beispielsweise die Konfiguration des NGINX Reverse-Proxy dargestellt. Diese beinhaltet eine Weiterleitung von HTTP-basierten Anfragen auf verschlüsselte HTTPS-Server. Außerdem werden alle Anfragen, Dokumente und Dateien mithilfe von *gzip* komprimiert. In der Konfiguration ist außerdem erkennbar, dass statische Dateien, wie Bilder direkt von NGINX gesendet werden, während alle anderen Anfragen an den NodeJS Server gesendet werden. Die wichtigsten gerenderten Seiten stellen die Abbildungen 5.1, 5.2 und 5.3 dar. Abb. 5.1 zeigt nach der Authentifizierung eine Auflistung aller Projekte eines Accounts. Außerdem können dort

neue Projekte erstellt, bereits bestehende gelöscht und für andere Nutzer freigegeben werden. Abb. 5.2 zeigt die *Landing-Page*, also die Seite, die der Benutzer beim ersten Start sieht. Bei erfolgreicher Authentifizierung wird diese Seite übersprungen und direkt die Auflistung der Projekte, wie in Abb. 5.1 angezeigt. Abb. 5.3 visualisiert den eigentlichen Kern der Applikation. Aufgrund der prototypischen Realisierung beinhaltet der Editor nicht alle Features, die im Konzept ausgeführt wurden. Die primären Eigenschaften sind folgende:

1. Die Anzeige einer zweidimensionalen Karte bzw. dreidimensionalen Visual Globe.
2. Initialisierung eines Fensterlayouts für individuelle Anpassungen des Editors.
3. Integration eines Datei-Explorers zur Organisation projektspezifischer Dateien und Ordner.
4. Drag and Drop Funktionalitäten zur Anzeige von glTF-Modellen, geojson- und KML-Layern.
5. Automatische Initialisierung und Speicherung von Szenendateien über die Serverarchitektur.

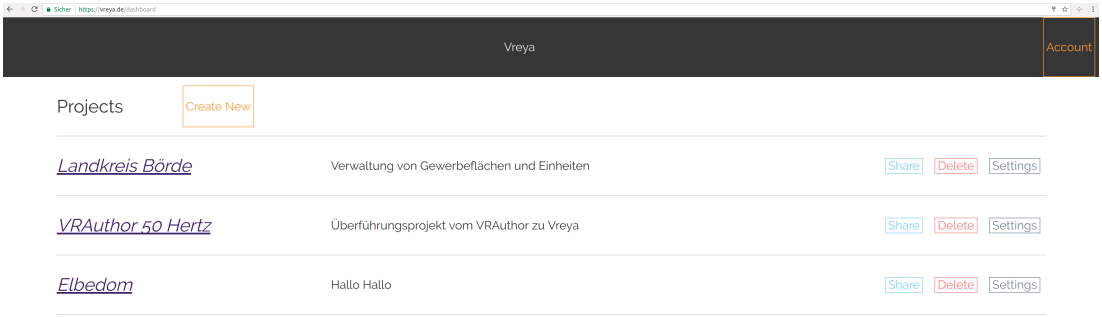


Abbildung 5.1: Auflistung aller Projekte eines Accounts. Funktionen zum Löschen, Freigeben und Ausführen eines Projektes sind ebenfalls in dieser Ansicht vorzufinden. (Eigene Darstellung)

Im Kapitel 4 wurde zusätzlich zu der Serverinfrastruktur ein abstraktes Szenenformat beschrieben, welches auf JSON basiert und die Eigenschaften der zu visualisierenden Daten in einer listenartigen Struktur abspeichert. Im Anhang C enthält das Listing C.2 eine Beispielrealisierung für eine JSON-Szene. Wie im Listing zu sehen ist, werden die Daten in Arrays gespeichert und als Eigenschaften unterschieden und geordnet. Die Übertragung wird mithilfe von Webpacks Minifizierung und NGINX-GZIP-Algorithmus optimiert von den Serverkomponenten an den Client gesendet. Die Clientkomponenten besitzen dadurch jedoch einen zusätzlichen Mehraufwand der Dekomprimierung. Die Kompression beträgt in dem Fall der genannten Szene 314% von

1829 Bytes in der Originalgröße auf 582 Bytes in der komprimierten Version. Für die eigentliche Speicherung der Daten einer Szene sind die Serverkomponenten zuständig, die im folgenden vorgestellt werden.

5.2 Serverseitige Komponenten

Serverseitig steuert eine NodeJS-Applikation unter dem Port 8000 alle eingehenden Anfragen. Das initiale Serverskript unterteilt sich in verschiedene Module zur Verwaltung der zentralen Aufgaben der Serverseite. Die dort referenzierten Komponenten, wie *Login* oder *Editor* sind unabhängig voneinander, sodass eine klare Trennung und damit das Single-Responsibility-Prinzip gewährleistet ist. Einige der wichtigsten Funktionen der Serveranwendung sind in den Listings 5.1, C.3, C.4 und C.5 dargestellt.

```
1  const mongoose = require('mongoose');
2  const passportMongoose = require('passport-local-mongoose');
3  const ProjectSchema = new mongoose.Schema({
4      name: { type: String, required: true, unique: true },
5      creator: { type: mongoose.Schema.Types.ObjectId, ref: 'User',
       required: true },
6      editors: [{ type: mongoose.Schema.Types.ObjectId, ref: 'User' }],
7      observers: [{ type: mongoose.Schema.Types.ObjectId, ref: 'User' }],
8      description: String,
9      released: { type: Boolean, required: true }
10 });
11
12 module.exports = mongoose.model('Project', ProjectSchema);
```

Listing 5.1: Initialisierung der Datenbankschema zur Verwaltung von Projekten.

Listing 5.1 zeigt, wie Projektschema in der Datenbank gespeichert und geladen werden. Ein Projekt ist, wie im Konzept (s. 4.11) beschrieben, eine einfache nicht-relationale Datenstruktur, welche einen Namen besitzt, und auf einen primären Author (hier *Creator* genannt) referenziert ist. Weitere Nutzer werden als Bearbeiter (Editor) oder Besucher (Observer) referenziert und freigegeben. Listing C.4 nutzt dementsprechend das initialisierte Datenbankschema für die Authentifizierung - in diesem Fall für die Registrierung eines neuen Nutzers. Die Registrierung erfolgt durch einen Abgleich aller in der Datenbank enthaltenden Nutzer mit dem Username und der Email. Ist eine Email oder ein Username gleich, so wird ein Fehler ausgegeben und der Client wird wieder auf die Sign-Up-Seite geleitet. Ist die Registrierung erfolgreich, wird der Client auf die Dashboard-Seite weitergeleitet, um Projekte erstellen und bearbeiten zu können. Um die Abfrage aller Projekte eines Accounts zu verdeutlichen (s. Abb. 5.1), ist in Listing C.3 der Algorithmus verdeutlicht. Als erstes wird die Projektdatenbank angefragt, dort werden alle Projekte gesucht, die unter dem Username des Accounts

laufen. Aus diesen Projekten wird nur der Name und die Beschreibung gefiltert, da alle anderen Daten in diesem Fall irrelevant sind. Bei einem Fehler wird die Dashboard-Seite mit der Meldung *'Projects not found'* gerendert. Im erfolgreichen Fall werden die Projekte, der Name und zusätzliche Flash-Nachrichten gerendert, die die Projekterstellung intuitiver gestalten. Das Rendering wird durch PugJS geregelt. Die Seite dashboard.pug wird mit den eben dargelegten Informationen gefüllt und zu einer HTML-Datei konvertiert. Auf den genauen Pug-Code wird im nachfolgenden Abschnitt eingegangen. Eine weitere wichtige Funktion zur Verdeutlichung der Serverarchitektur stellt die Verbindung zum Datenspeicher AWS S3 dar. Das Listing C.5 zeigt, wie Ressourcen angefordert werden. Wie im Konzept beschrieben, geschieht dies über einen HTTP-GET-Route, welcher vom Client an den Server gesendet wird. Als Url /ressource/path/* kommt dabei jeder Dateipfad in Frage. Wichtig ist ausschließlich, dass Keys zu den gespeicherten Dateien das folgende Muster besitzen:

/ressource/path/username/projectname/folder/file.extension

Der Algorithmus erzeugt mithilfe der AWS S3 API eine signierte Url zum Dateipfad. Diese Url ist eine Minute lang gültig und wird an den Client weitergeleitet. Dadurch, dass die Funktion *redirect* genutzt wird, wird die weitergeleitete Url direkt im Client ausgewertet und die angeforderte Ressource direkt von S3 an den Client gesendet. Für weitere Implementierungsdetails des Servers sind die Projektdateien im Repository unter https://gitlab.com/pschladi/vreya/tree/master/server einsehbar. Aufgrund der Tatsache, dass das Projekt als Git-Repository angelegt ist, können sich die hier referenzierten Code-Beispiele zu einem späteren Zeitpunkt verändern.

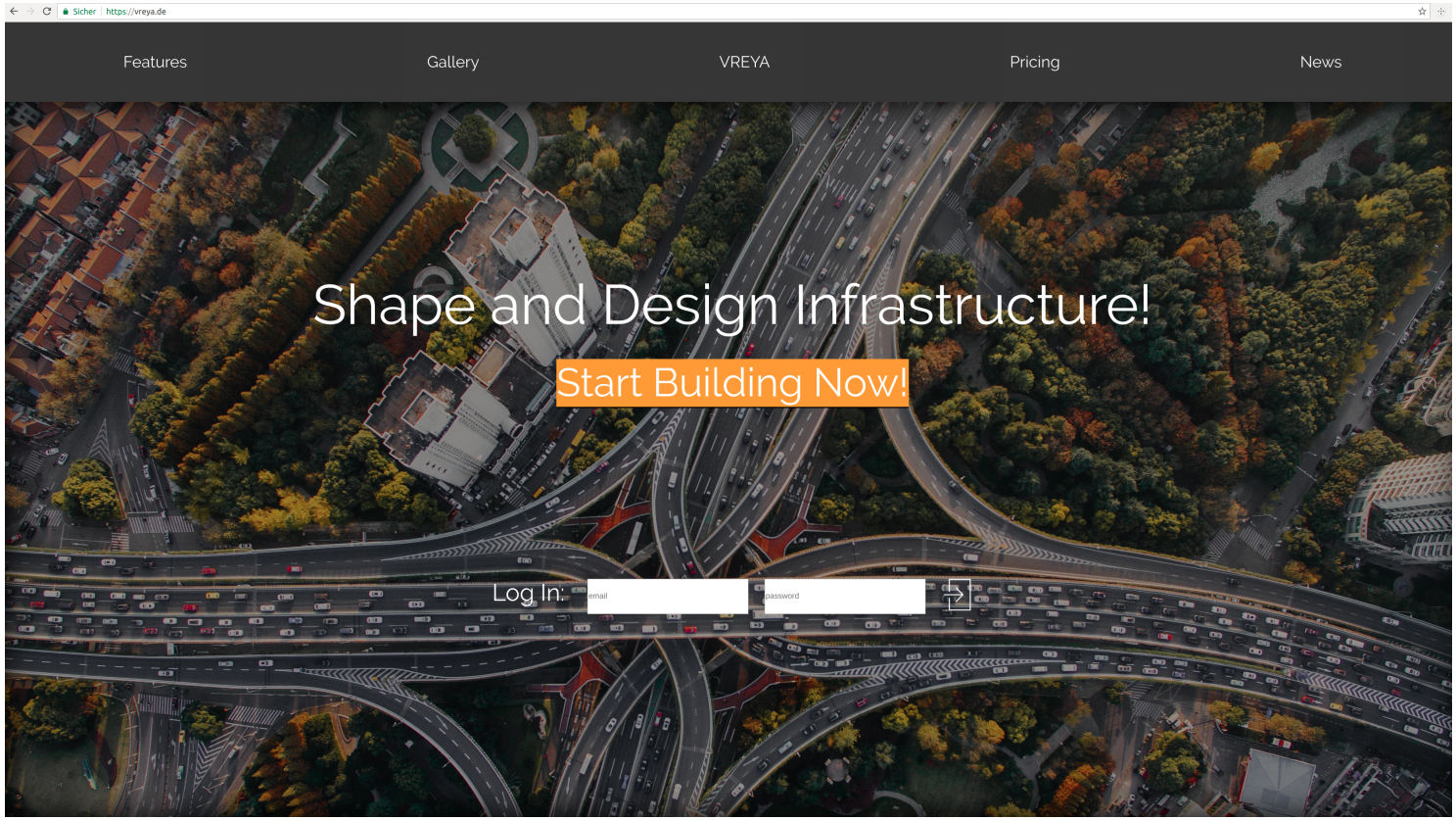


Abbildung 5.2: Initiale Sicht auf die realisierte Webapplikation. (Eigene Darstellung)

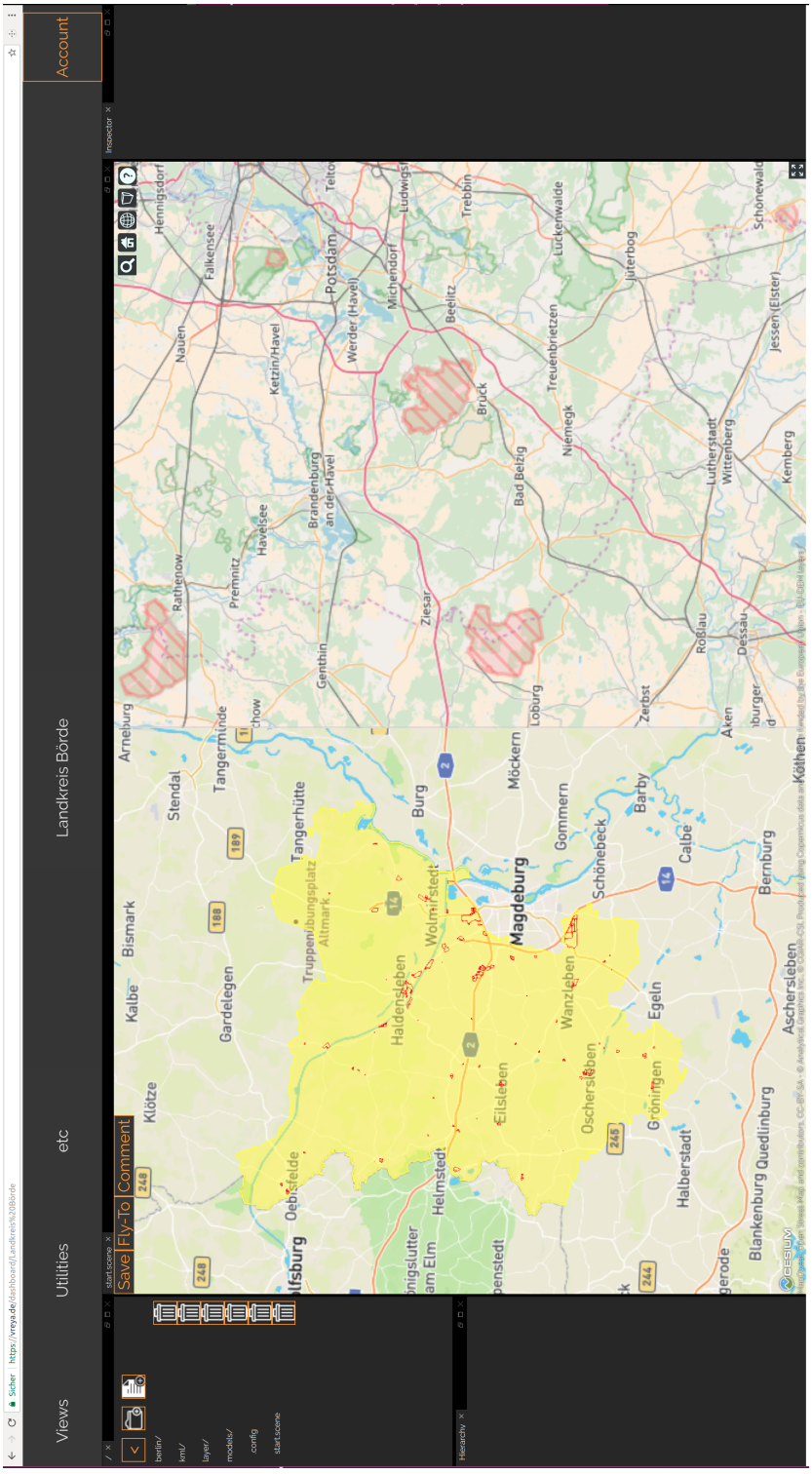


Abbildung 5.3: Darstellung eines ausgeführten Projektes. Wie im konzeptionierten Mockup besteht die zentrale Sicht aus der 2D- bzw. 3D-Szene. Links sind die Projektdateien als Explorer dargestellt. Rechts ist ein sogenannter Inspector, welche Daten zu einem selektierten Objekt abruft. (Eigene Darstellung)

5.3 Clientseitige Komponenten

Auf der Seite der Client-Applikation wird, wie im Konzept beschrieben, eine Kombination aus folgenden Frameworks genutzt:

1. GoldenLayout zur Anzeige eines Fenstersystems, um anpassbare und individualisierte Projekte zur Verfügung zu stellen.
2. ReactJS für eine komponentenbasierte Entwicklung der einzelnen Applikationsteile. ReactJS dient zusätzlich als Schnittstelle für Erweiterungen, da neue Module über React-Komponenten eingepflegt werden können.
3. CesiumJS zur Anzeige eines Visual Globes und als Schnittstelle zu WebGL 2.0.

Wie im vorherigen Abschnitt beschrieben, werden die Seiten über einen PugJS-Compiler mit entsprechenden Informationen zu HTML-Dokumenten verarbeitet. Zwei Beispiele für diese PugJS-Dokumente zeigen die Abbildungen 5.4 und 5.5. Wie in Abb. 5.4 zu sehen ist, werden die HTML-Elemente in Blöcken eingeteilt. Diese Blöcke können wiederum erweitert werden. Der Theme, der in Abb. 5.5 dargestellt ist, stellt die Blöcke in den Kontext der Website. Nur an diesen Stellen sind Ergänzungen zum Template erlaubt. Auf Basis der gerenderten Pug-Dateien werden folgende Client-Komponenten initialisiert:

1. *Application*, welche als zentrales Objekt das Layout und die Szenen initialisiert.
2. *Layout*, welches eine Standardkonfiguration besitzt und im Fall von Änderungen den Local-Storage des Browsers zur Speicherung nutzt. Bei einer erneuten Initialisierung des Projektes wird das geänderte Layout geladen. Dadurch sind individualisierte Projektanpassungen möglich.
3. *Library*, welche als Dateiexplorer des Systems fungiert. Die Library steht eng mit dem Server und AWS-Datenspeicher in Verbindung und verändert primär die Dateien eines Projektes. Durch Drag and Drop lassen sich neue Dateien hinzufügen, außerdem können Dateien und Ordner gelöscht werden. Alle Anfragen werden primär durch den Server verwaltet, jedoch bei korrekter Authentifizierung für Performance-Zwecke direkt durch den AWS-Server ausgewertet.
4. *Scene*, die eine zweidimensionale bzw. dreidimensionale Szene mit projektspezifischen Daten füllt. Die Szene basiert auf CesiumJS und WebGL.
5. *AWSUtility*, welche zentrale Anfragen an den Server in korrespondierende AWS-Anfragen übersetzt. Da AWS zusätzliche Informationen im XML-Format liefert, werden im Großteil der Anfragen die wesentlichen Informationen herausgefiltert und alle anderen Daten verworfen.

Die beschriebenen Komponenten können jederzeit durch das Navigations-

menu in die Szene gezogen werden. Dadurch ist es möglich, dass in einem Projekt auch mehrere Szenenkomponenten dargestellt werden. Dies ist vor allem dann nützlich, wenn verschiedene Positionen im 3D-Raum eingenommen werden. Als zusätzliches Features kümmert sich jede Szenekomponente seperat darum, welche Szene aktuell sichtbar ist. Dadurch ist es möglich im Bezug auf die Infrastrukturentwicklung, dass in verschiedenen Szenen unterschiedliche Infrastrukturvarianten diskutiert werden können. Die Anforderung von Daten übernimmt damit auch jede Szene für sich. Ein Beispiel hier für ist in Listing C.6 dargestellt. Als erstes wird vom Server unter der Adresse */api/ressource/getList?key=$key* eine Liste von Dateien und Ordnern angefordert. Diese Liste, ist wie eben beschrieben, im XML-Format mit vielen zusätzlichen und redundanten Informationen. Der Parameter Key der Url bezeichnet dabei den Pfad der Datei. Wie auch im vorherigen Abschnitt bezeichnet der Key ein Muster aus Username des Erstellers, des Projektnamens und dem schlussendlichen Pfad der Datei mit Dateierweiterung. Die zurückgelieferte XML-Datei wird geparst und nach XML-Elementen vom Tag *Contents* durchsucht. In diesen Elementen sind die Dateien enthalten. Da AWS keine Ordner speichert, sondern die Ordner als Teil der Keys in einer Bucket referenziert, müssen Ordnerstrukturen anhand sogenannter *CommonPrefixes* gefiltert werden. Sobald eine Szenenkomponenten initialisiert ist - im ReactJS-Fall wird dies *Mounting* genannt, kann die eigentliche Szene über CesiumJS initialisiert werden. Die Initialisierung der Szene ist dabei vollständig asynchron. Nutzer können also noch während des Ladevorgangs mit der Bearbeitung beginnen. Das asynchrone Laden hat jedoch den Nachteil, dass mitunter wichtige Elemente der Szene spät angezeigt werden, sollte die Szene oder darin enthaltende Elemente zu groß sein. Listing C.7 zeigt, wie der Ladevorgang generell stattfindet. Zu Beginn wird die Szenenkonfiguration mit der Erweiterung *.scene* geladen. Diese Szenenkonfiguration wird über einen Key im Projektordner gespeichert. Die Konfiguration beinhaltet alle Daten der Szene und referenziert Modelle anhand ihrer Keys im Projekt. Sobald die Konfiguration geladen ist, wird der Header der Komponente auf den Namen der Szene gesetzt und alle bisher enthaltenden Cesium-HTML-Elemente gelöscht. Daraufhin wird Cesium initialisiert. Als Terrain wird das STK World Terrain von Analytical Graphics geladen. Diese Terraindaten sind im 3D-Tile-Format veröffentlicht wurden. Weiterhin werden nach der Initialisierung von Cesium alle referenzierten glTF-Modelle, geoJSON- und KML-Layer geladen.

Erweiterbarkeit

Wie in der Konzeption beschrieben, lässt sich das System mithilfe eigener Programmcodes erweitern. Um die entsprechenden Erweiterungsmechanismen zu nutzen, sind Entwickler daran gebunden, projektspezifische Javascript-Dateien zu erstellen. Diese Dateien werden zum Start des Projekts geladen

```
extends theme.pug

//- additional features of the editor
block linksLeft
    ul.menu
        li
            a Views
            ul.menu
                li#menuItemInspector.draggable Components
                li#menuItemHierarchy.draggable Hierarchy
                li#menuItemLibrary.draggable Library
                li#menuItemScene.draggable 3D Scene
                li#menuItemScenePlay.draggable 3D Scene (Play-Mode)
        li
            a Utilities
            ul.menu
                li Utility1
                li Utility2
                li UtilityTest
                li Hello
                    ul.menu
                        li SuperUtility1
                        li Super Utility2

        li
            a etc
//- main content, initialized by golden layout
block content
    #editor.editor
block css
    link(href='/static/css/editor.css', rel='stylesheet')
    link(href='/static/css/goldenlayout-base.css', rel='stylesheet')
    link(href='/static/css/goldenlayout-dark-theme.css', rel='stylesheet')
    link(href='/cesium/Widgets/widgets.css', rel='stylesheet')
block scripts
    script(src='/cesium/Cesium.js', type='text/javascript')
    script(src='/static/js/vreya.bundle.js', type='text/javascript')
    script.
        const container = document.getElementById('editor');
        const vreya = new Vreya.Application(container, "#{creatorName}", "#{projectName}");
```

Abbildung 5.4: Pug-Code zur Darstellung des Editors. (Eigene Darstellung)

```
doctype html
html(lang='de')
    head
        title= pageTitle
        link(rel='shortcut icon' type='image/x-icon' href='/static/img/favicon-v3.ico')
    body
        header.siteHeader
            section
                block linksLeft
            if projectName
                a(href='/dashboard')=projectName
            else
                a(href='/dashboard') Vreya
            section
                block linksRight
            button#userSettingsToggle Account
        main.main#main
            block content
        section.userSettings#userSettings
            h2 Hello, #{name}
            nav
                ul
                    li
                        a(href='/profile') Profile
                    li
                        a(href='/subscription') Subscription
                    li
                        form(action='/sign-out', method='post')
                            input(type='submit', value='Sign-Out')
        link(href="https://fonts.googleapis.com/css?family=Raleway", rel="stylesheet")
        block css
        script.
            const userSettingsToggledEvent = new CustomEvent('onUserSettingsToggled');
            document.getElementById('userSettingsToggle').addEventListener('click', function() {
                let settings = document.getElementById('userSettings');
                if(settings.classList.contains('pop-up')) {
                    settings.classList.remove('pop-up');
                } else {
                    settings.classList.add('pop-up');
                }
                document.getElementById('main').dispatchEvent(userSettingsToggledEvent)
            });
        block scripts
```

Abbildung 5.5: Pug-Code, welcher websiteübergreifend als Theme fungiert. (Eigene Darstellung)

und ausgeführt. Entwickler können an bestehenden Systemen mit eigenen Komponenten andocken. Bspw. ist es möglich React-Komponenten zu entwickeln und diese an GoldenLayout anzuhängen, um das User Interface zu erweitern. Zusätzlich besteht die Möglichkeit die 3D-Interaktivität mit Klassen zu erweitern, welche an Cesium anknüpfen. Die entsprechenden Schnittstellen werden von der Applikation automatisch bereitgestellt. Zur Verbesserung der Intuitivität ist es zusätzlich im Rahmen weiterer Untersuchungen möglich die Entwicklung durch einen entsprechenden Editor zu ergänzen, der direkt in der Visualisierungsplattform bereitgestellt wird. Im Rahmen dieser Arbeit wird ein solcher Code-Editor jedoch nicht bereitgestellt.

5.4 Zusammenfassung

In diesem Kapitel wurden ausgewählte Aspekte der Implementierung des Konzepts beschrieben. Es wurde dabei primär auf das Zusammenspiel zwischen den Server- und Clientkomponenten eingegangen. Wichtige Details der Implementierung stellt die Ressourcenanforderung und -speicherung dar, welche sowohl vom Client als auch Server ausgeführt werden kann. Die gesamte Kommunikation über das AWS-Netzwerk erfolgt jedoch nur dann erfolgreich, wenn der Request dazu authorisiert ist. Die Authentifizierung wird durch eine nicht-relationale Datenbank festgestellt, welche alle Daten eines Nutzers und dessen Projekte speichert. Projektspezifische Daten werden in AWS gespeichert. Durch die asynchronen Features von ECMAScript können Ladevorgänge innerhalb der Szene durchgeführt werden, ohne dass Nutzer gestört werden. Zusätzlich können detaillierte 3D-Modelle durch das von Khronos im Juni 2017 veröffentliche glTF-Format 2.0 effizient geladen werden [51]. Die hochauflösende Grafik von modernen Grafikengines wird durch glTF 2.0 ebenfalls erreicht, da das sogenannte Physically-Based-Rendering unterstützt wird. Physikalisch korrekte Lichtberechnungen werden jedoch nur dann aktiviert, wenn das System leistungsstark genug ist - mobile Geräte besitzen daher ebenfalls die Möglichkeit automatisch auf eine geringere Auflösung zu wechseln.

Kapitel 6

Ergebnisse der Arbeit

Im folgenden Kapitel werden die Ergebnisse der Arbeit festgehalten. Zu Beginn wird ein Anwendungsszenario beschrieben, welches als Ausgangspunkt der Analyse eingesetzt wird. Das Anwendungsszenario basiert dabei auf der Implementierung, welche in Kapitel 5 beschrieben wurde. Als anschließende Analyse wird das Szenario mit Unity 2017 verglichen. Die in der Einleitung gestellte Aufgabenstellung wird anhand der analysierten Ergebnissen ausgewertet.

6.1 Dargestelltes Anwendungsszenario

Das visualisierte Anwendungsszenario (s. Abb. 6.1) hat zur Aufgabe die Umsetzung der gestellten Ziele der Arbeit aus Kapitel 1.2 zu überprüfen. Die Ziele umfassten die Identifikation von Ansätzen zur Realisierung einer webbasierten und mobilen Plattform zur Bereitstellung von hochauflösenden Szenarien zur Planung von Infrastrukturprojekten. Dabei ist, neben der intuitiven Darstellung komplexer Datenmengen mit Bezug auf den in den Grundlagen dargestellten Visualisierungszielen, die Einhaltung der gestellten Anforderungen (s. Tabellen 4.1 und 4.2) aus der Konzeption entscheidend. Das Anwendungsszenario besitzt folgende wesentliche Kernelemente zur Überprüfung der Anforderungen des Konzepts:

1. Visualisierung eines Visual Globes und 2D-Kartenmaterial von OpenStreetMap
2. Darstellung von georeferenzierten Terraindaten inklusive digitalen Orthophotos und Höhenmodellen.
3. Visualisierung der Stadtmitte Berlins.
4. Überlagerung von Informationslayern zur beispielhaften Darstellung von infrastrukturellen Daten.

Die primäre Visualisierungskomponente ist in diesem Szenario die Darstellung von Gebäuden, Flüssen und Straßen von Berlin. Der georeferenzier-

te Datensatz stammt dabei ebenfalls, wie das Kartenmaterial, von OpenStreetMap und wurde als *.osm* Datei extrahiert. Die Extraktion übernahm die 3D-Software *Blender*. Um die Gebäude in der konzipierten Plattform anzuzeigen, bestanden mehrere Möglichkeiten. Zum einen konnten die Modelle als GeoJSON- oder KML-Layer exportiert werden. Zum anderen bestand die Möglichkeit der Konvertierung zum effizienten Webformat glTF. Da GeoJSON- und KML-Layer als ergänzende Visualisierungselemente geplant sind, und das Format glTF zur Anzeige von großen Polygonmengen geeignet ist, stellt das Anwendungsszenario Berlin als glTF Modell dar. Zur Konvertierung von Daten des osm-Formats wurde der Collada Exporter von Blender zum Dateiformat Collada genutzt. Daraufhin wurde das Format Collada zu glTF mithilfe eines von der Khronos Group bereitgestellten Konvertierers collada2gltf transformiert. Im Szenario ist daraufhin nur noch eine Referenzierung des Ursprungspunktes mit den Berliner Koordinaten notwendig gewesen, um das Modell korrekt anzuzeigen (Latitude: 52.520645, Longitude: 13.409779). Das gesamte Modell von Berlin entält 867tausend Vertices, 461tausend Polygone und 1.2 Millionen Dreiecke. Die Darstellung findet anhand simpler Blinn-Phong-Pixel-Shader statt. Das Anwendungsszenario enthält außerdem folgende Informationslayer, die als GeoJSON-Layer eingebunden wurden:

1. Darstellung der Landesgrenzen des Landes Sachsen-Anhalt und dessen Landkreis Börde.
2. Darstellung von Gewerbeflächen im Landkreis Börde, sowie Grenzen und *Interesting Places* von Schalkau.
3. Beispielhafte Darstellung einer animierten Tour.

6.2 Testdurchführung

Der an dem Anwendungsszenario durchgeführte Test bildet Systemdaten, wie ein Wasserfalldiagramm, Downloadgröße und Bildfrequenz ab. Als Vergleichsplattformen wird *Unity* in der Version 2017.1.1 genutzt. Die konzipierte Plattform ist nicht uneingeschränkt vergleichbar, da die diese eine Art-'3D-Engine im Web' darstellt, während in Unity fertige Szenen zu WebGL konvertiert werden. Da das konzipierte System eine neuartige Plattform darstellt, existieren keine weiteren direkten Vergleichsmöglichkeiten. Das bedeutet, dass Unity Optimierungen an der Szene und den enthaltenden Modellen vornehmen kann. In den Tests sollte die konzipierte Plattform in der Bildfrequenz daher hinter der 3D-Engine Unity liegen. Die Tests finden anhand unterschiedlicher Systeme statt, um ein breites Spektrum an Hardware einzubeziehen. Darunter befinden sich das Lumia 950 mit Windows 10 Mobile und dem Browser Edge, ein PC mit Intel Core i5 4690k, Geforce GTX 1070 Grafikkarte und Chrome, sowie das Android Tablet Acer Predator 8 GT-810 ebenfalls mit Chrome. Zusätzlich zum genannten Vergleich wird ein

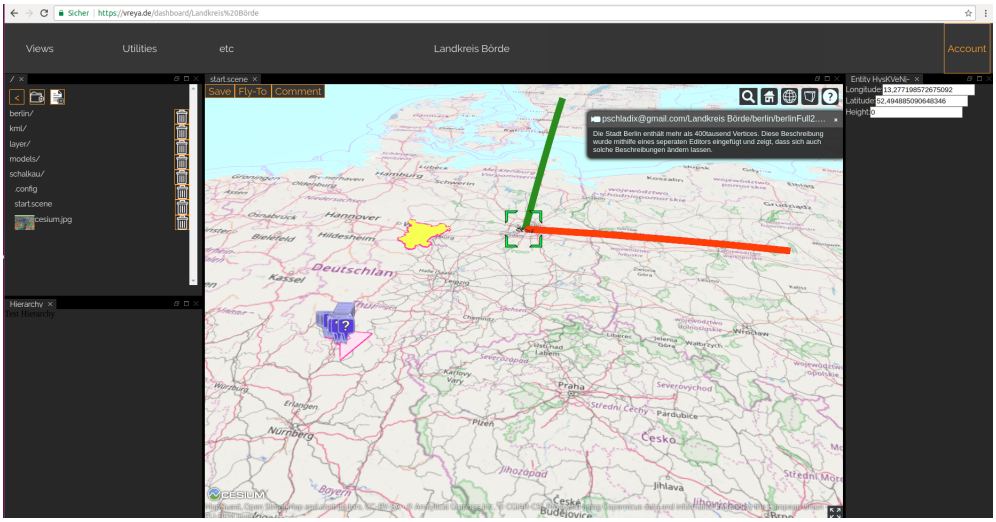


Abbildung 6.1: Anwendungsszenario zur Kontrolle der Ziele der Arbeit. Im Bild ist die Verwendung von mehreren Informationsquellen erkennbar. Die Informationsquellen sind: Landkreis Börde, Gewerbeflächen, sowie Schalkau als GeoJSON-Layer und Berlin als glTF-Modell

Optimierungsscan durchgeführt, inwieweit die Realisierung optimiert werden kann. Dieser Scan wird mithilfe des Webservice https://gtmetrix.com/ durchgeführt.
Aufgrund der Tatsache, dass das konzipierte System eine Webapplikation darstellt, sind Sicherheitstests notwendig, um die Stabilität des Systems zu gewährleisten. Im Konzept sind durch Verwendung eines Reverse-Proxy, Content-Delivery-Networks und SSL-Verschlüsselung grundlegende Sicherheitskonzepte vorhanden. Die Sicherheit der Plandaten, Projektdateien und Nutzeraccounts besitzt eine hohe Priorität. Zum Test der Sicherheit werden sogenannte Pentests und Schwachstellenscans eingesetzt [88]. Die Vorbereitung und Durchführung eines Pentests bedeutet einen hohen zeitlichen Umfang. Um die grundlegende Sicherheit des Systems zu testen, wird zur Auswertung ein automatisierter Schwachstellenscan durchgeführt. Schwachstellenscans werden von zahlreichen Webservices angeboten. In diesem Fall übernimmt der Webservice unter der Adresse https://pentest-tools.com den Schwachstellen-Scan und automatisierten Pentest. Zusätzlich wird über das Werkzeug 'SQL Inject Me' https://addons.mozilla.org/de/firefox/addon/sql-inject-me/ die Sicherheit gegenüber dem bekannten Angriff der SQL-Injection getestet. Der SQL-Injection-Test wird anhand der Login-Form durchgeführt.

6.3 Auswertung

Die vorgestellten Tests werden in diesem Abschnitt ausgewertet. Die Testergebnisse des Plattformvergleichs sind in der Tabelle 6.1 zusammengefasst. Festzustellen ist, dass sich das Szenario in beiden Plattformen visualisieren lässt. Die Perfomance ist in der Webvariante von Unity zum konzipierten System im Durchschnitt 3.6 Bilder pro Sekunde schneller. Die gesamte Downloadgröße des Szenarios beträgt in der Unity-Engine 82.4 MB, während es in der konzipierten Plattform 79.7 MB sind. Gleichzeitig versendet Unity keine Cache-Control-Header, welche die Lebenszeit der Dateien festhält. Die konzipierte Plattform ist jedoch in der Lage Cache-Control-Header zu versenden, sodass bei einem erneuten Laden Modelle nicht erneut heruntergeladen werden müssen. Dadurch verbessert sich die Downloadgeschwindigkeit in der neuen Plattform, wenn Modelle mehr als einmal geladen werden müssen. Eine Optimierung für mobile Hardware fehlt in der vorgestellten Realisierung. Es ist erkennbar, dass CSS-Dateien und Rendering-Verfahren für Smartphones und Tablets angepasst werden müssen. In weiteren Forschungsarbeiten ist zusätzlich eine mobile Optimierung durch neue Technologien, wie WebAssembly oder GraphQL zu untersuchen.
In einem weiteren Test, der mithilfe von https://gtmetrix.com/ durchgeführt wurde, sollte der Optimierungsbedarf der Webapplikation getestet werden. Diese Applikation zeigt erste Stellen zur Optimierung der Webapplikation an. Das Resultat des Reports ist im Anhang siehe Abb. D eingefügt. Wichtige Optimierungsmöglichkeiten sind die Komprimierung von Bilddateien und HTML. Bisher werden ausschließlich die Javascript und CSS Dateien komprimiert.
Im vorherigen Abschnitt wurden zusätzliche Tests beschrieben, um die konzeptionierte Plattform hinsichtlich der Sicherheit zu überprüfen. Für einen ersten Sicherheitsstest wurde das Werkzeug https://pentest-tools.com verwendet. Diese Applikation testet auf populäre Schwachstellen, wie XSS- und SQL-Injection, Server Konfigurationen, Cookies, verwendete Header sowie SSL Zertifikate. Im Rahmen der Testapplikation sind insgesamt 26 Sicherheitstests durchgeführt wurden. Diese Tests brachten **vier** Sicherheitslücken hervor, von denen drei eine leichte Gefahr darstellen. Eine vollständige Übersicht über den Testreport ist ebenfalls im Anhang siehe Abbildung D eingefügt. Die größte Sicherheitslücke stellt das Fehlen des sogenannten E-Tag-Headers dar, welche folgende Beschreibung besitzt:

> ETag (für entity tag, etwa Entitätmarke) ist ein im HTTP 1.1 eingeführtes Header-Feld. Es dient zur Bestimmung von Änderungen an der angeforderten Ressource und wird hauptsächlich zum Caching, also der Vermeidung redundanter Datenübertragungen, verwendet. [82]

Eine Möglichkeit zur Behebung der Sicherheitslücke ist das Setzen des ETag-

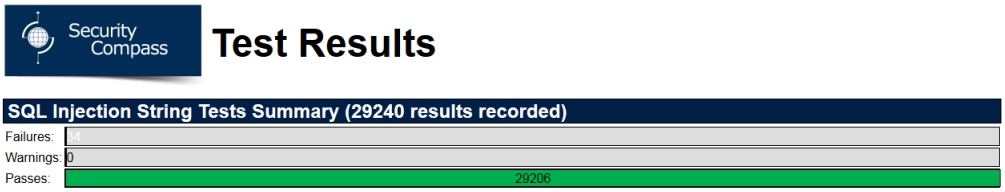


Abbildung 6.2: Übersicht über die Resultate von 29240 durchgeführten SQL-Angriffen. 29206 Angriffe wurden erfolgreich bestanden. 34 Angriffe lösten Fehler aus. Kein Angriff führte zu einer Warnung bzw. Sicherheitslücke.

Headers für jeden HTTP-Request einer Datei. Eine weitere Sicherheitslücke befindet sich im Fehlen der Security-Header X-Frame-Options, X-XSS-Protection, Strict-Transport-Security und X-Content-Type-Options. Um diese Fehler zu beheben, ist innerhalb des Konzepts die Konfiguration von NGINX (s. C.1) so anzupassen, dass die genannten fehlenden Header ergänzt werden. Die durchgeführten Sicherheitstests wurden mit einem weiteren Werkzeug 'SQL Inject Me' ergänzt, welches die Plattform konkret auf SQL-Angriffe überprüft. Die Durchführung des Tests (s. Abb. 6.2) anhand der Login- und Sign-Up Formen zeigt, dass insgesamt **29240** SQL-Angriffe durchgeführt wurden. Von diesen haben 29206 Angriffe erfolgreich den Test bestanden. Zusätzlich lösten 34 Tests Fehler in der Webapplikation hervor. Alle 34 Fehler mündeten in der Ausgabe 'Server Status Code: 302 Found'. Kein SQL-Angriff hatte nach diesem Test Erfolg.
Ein vollständiger Penetrations- und Sicherheitstest erfordert einen großen Testumfang. Manuelle Tests bezüglich des Editors und der Validierung der Eingabefelder stellen eine weitere Schnittstelle für potenzielle Sicherheitslücken dar. Da der benötigte Umfang für weitere Tests im Rahmen dieser Arbeit nicht gegeben ist, ist eine sicherheitsbezogene Evaluation in einer seperaten Arbeit durchzuführen. Zusammenfassend lassen sich die verwendeten Systemtests als positive Tendenz zu einem stabilen und performanten System deuten.

System	**Browser**	**Unity 2017.1.1**	**Konzept**
Lumia 950	Edge	7	4
Samsung Galaxy Tab A 2016	Chrome	16	12
Core i5 4690k, Geforce GTX 1070	Chrome	60	60
Core i5 4690k, Geforce GTX 1070	Firefox	58	57
Intel Xeon E5-1650, Quadro K4200	Chrome	49	39

Intel Xeon E5-1650, Quadro K4200	Firefox	41	37

Tabelle 6.1: Erreichte durchschnittliche Bildfrequenz in Frames per Second des konzeptionierten Systems im Vergleich zur Engine Unity.

6.4 Schlussfolgerung

In diesem Kapitel wurden die Ergebnisse der Arbeit ausgewertet. Hier zu wurde ein Anwendungsszenario in der konzeptionierten Plattform erstellt, welches als Grundlage der Systemtests verwendet wurde. Das Anwendungsszenario stellt die Stadt Berlin, sowie weitere Geoinformationslayer, visuell dar. Anhand einer Gegenüberstellung zur optimierten 3D-Engine Unity konnte gezeigt werden, dass die konzeptionierte Plattform Bildfrequenzen liefert, die für eine weitere Forschungsarbeit akzeptabel sind. Vorallem mobile Geräte (Smartphones) sind zu optimieren. Zusätzlich zu diesem Vergleich wurden weitere Tests durchgeführt, um die Sicherheit von Plänen und Dokumenten der Infrastrukturprojekte zu untersuchen. Die verwendeten automatisierten Schwachstellentests offenbarten wenige Sicherheitslücken, weshalb auf eine tendenziell sichere Plattform zu schließen ist. Die Sicherheit ist jedoch anhand einer vollständigen Evaluation im Rahmen einer zusätzlichen Arbeit zu zeigen.
Das Ziel der Identifikation von Ansätzen zur Realisierung einer webbasierten und mobilen Plattform zur Bereitstellung von hochauflösenden Szenarien zur Planung von Infrastrukturprojekten kann anhand der Testergebnisse als erfolgreich bewertet werden. Die Plattform stellt dreidimensionale Daten innerhalb eines Visual Globes dar. Der Visual Globe rendert zusätzlich georeferenziertes Kartenmaterial. 3D-Modelle können um zusätzliche Informationsschichten in den Formaten GeoJSON oder KML erweitert werden. Szenarien lassen sich beliebig im Browser anpassen, über mobile Geräte anzeigen und für andere Benutzer freigegeben. Projektdateien werden sicher auf Cloud-Servern gespeichert und von einer NGINX/NodeJS-Applikation verwaltet.

Kapitel 7

Zusammenfassung

Die infrastrukturelle Planung und Entwicklung ist in vielerlei Hinsicht ein schwer fassbares Themenfeld. Diese Arbeit beschäftigte sich daher zu Beginn mit dem Begriff der Infrastrukturentwicklung und erörterte anhand verwandter Arbeiten den aktuellen Stand der Technik. Es existieren Bestrebungen seitens etablierter Software-Lösungen, wie Autodesk Infraworks360 [33], bestehende Werkzeuge auf mobile Betriebssysteme zu überführen. Zugleich wird mit zunehmender Forschung des Themas virtueller Städte im Web auch die Lücke zu webbasierten Geoinformationssystemen untersucht [8]. Bei der Feststellung ergab sich jedoch, dass digitale Lösungen und Forschungsarbeiten die neuen Technologien des Web nicht hinreichend nutzen. Um eine Verbindung zwischen der Infrastrukturentwicklung und den neuen Technologien des Web zu schaffen, wurde daher eine neuartige Plattform konzipiert.
In der Planung von Infrastrukturprojekten ist eine Vielzahl von Stakeholdern beteiligt. Projektverantwortliche sind dabei Bund, Länder, Städte und private Anleger. Fachplaner konzipieren Entwürfe und unabhängige Gutachter überprüfen die angefertigen Pläne. In gesonderten gesetzlich angeordneten Planfeststellungsverfahren wird über die Zulässigkeit des Projektvorhabens entschieden [67, 89]. In den Planfeststellungsverfahren ist zusätzlich die öffentliche Auslegung gesetzlich festgeschrieben, wodurch betroffene Bürger anhand der öffentlichen Pläne Einwende einreichen können. Die Perspektiven der genannten Stakeholder ist, wie eben geschildert, in der zu konzipierenden Plattform zu integrieren. Das Konzept schlägt daher eine webbasierte Lösung vor. Administratoren und Verwalter können Projekte erstellen, zusätzliches Personal, wie Fachplaner und Entwickler, als Bearbeiter in das Projekt einladen und Gutachtern Leserechte gewähren. Die Öffentlichkeit wird einbezogen, sobald die Administratoren das Projekt für alle freigeben. Dies gewährt allen Besuchern die Möglichkeit das Projekt zu begutachten und Kommentare abzugeben. Das Konzept schlägt zusätzlich ein Diskussionsforum vor, um soziale Medien zu bündeln.

Das konzipierte System an sich verbindet die Grafik-API WebGL 2.0 mit einem Content-Delivery-Network, um eine hochleistungsfähige Serverinfrastruktur herzustellen, die hochauflösende Planungsdaten für eine Vielzahl von Nutzern verwaltet. Konkret eingesetzt wurde CesiumJS zur grafischen Darstellung, sowie den Amazon Web Services als Serverinfrastruktur. Die Authentifizierung und Projektverwaltung wird von einer NodeJS-Applikation serverseitig gesteuert. Die konzipierte Serverinfrastruktur erlaubt eine Skalierung auf beliebig viele Server durch Reverse-Proxys, sodass eine hohe Anzahl von Anfragen bearbeitet werden kann. Dahingehend lässt sich diese Architektur auch zur zukünftigen Bürgerbeteiligung an der Infrastrukturplanung verwenden. Das Ziel ist folglich eine Digitalisierung der bisher analog durchgeführten Planfeststellungsverfahren. Die Ergebnisse zeigen, dass die konzipierte Architektur eine performante und effiziente Darstellung im Web ermöglicht und zugleich durch neue Plandaten, 3D-Modelle und Informationsschichten erweitert werden kann.
Zusammenfassend lässt sich das Ziel der Konzeptionierung einer mobilen Plattform zur digitalen Infrastrukturplanung im Web als erfolgreich bewerten. Für die zukünftige Forschung lässt sich das Konzept zusätzlich als erste Grundlage für die kollaborative Infrastrukturplanung im Web einsetzen. Folglich sind neue Forschungslücken entstanden, die durch weitere Arbeiten zu untersuchen sind. Zum einen ist die Untersuchung innovativer Interaktionstechniken wichtig. Mobile und webbasierte Systeme müssen gleichermaßen durch Touch, wie auch durch Maus und Tastatur bedienbar sein. Desweiteren fehlen geeignete Mittel, um bestehende Projekte anderer Plattformen (Unity, Fraunhofer VRAuthor) gezielt in das neue System zu überführen. Daher sind neue Algorithmen zur Konvertierung oder Einbettung dieser Projektszenarien zu entwickeln. Zum anderen stellt sich die Sicherheit der konzipierten Plattform als Herausforderung dar. Für die Sicherheit der Planungsdaten gegen digitale Fremdzugriffe ist daher eine weitere Analyse notwendig.

Anhang A

Zusätzliche Informationen zu den Grundlagen

V1	V2	Projekt-Phase	Primär-verantwortung	Sekundär-verantwortung	Inhalt der Phase
Ia	Ia	Politisches Programm	Politik (überreg.)	Politik (reg.)	Verkehrspolitische Ziele der "Projektidee"
Ib	Ib	Machbarkeitsstudie	MBVI, LMV, etc.	Planer	Nachwes der Machbarkeit, erste Kostenabschätzung
Ic	Ic	Übergeordnete Ast in BVWP, LP, RP	Bund, Länder, Region	Bund, Länder, Region	Realisierungsabsicht, pol. Wille, Aufnahme in langfristige Haushaltsplanungen
IIa	IIa	Bauherren-Organisation A	Bauherr	Bauherr	Bildung einer Bauherren-Organisation für die Planungsphase
IIb	IIb	Aufgabenstellung I	Bauherr A	Bauherr A	Grobe Aufgabenstellung, Wille des AG
IIc	IIc	Vorplanung VP	Bauherr A	Planer	grobe Planung in Varianten

IId	IId	Aufgaben-stellung 2	Bauherr A	Bauherr A	Überarbeitete und end-gültige AST aus Basis der VP
IIe	IIe	Finanzierung I	Öff. Hand (s. Ie)	Bauherr A	Absicherung der Finanzierung, Einstellung in den Haushalt
IIf	IIf	Entwurfs-planung	Bauherr A	Planer	techn. Planung, Nach-weis der Ausführbarkeit, belastbare Kos-ten (ggf. parallel zu IIg)
IIg	IIg	Genehmigungs-planung	Bauherr, Genehmigung-sbehörde	Planer	Erstellung Unterlagen, Durchführung Planrechts-verfahren, Planfeststel-lungsbeschluss
IIh	IIh	Finanzierung 2	Öff. Hand (s. Ic)	Bauherr A	Freigabe des Projektes bzgl. Finanzierung
IIi	IVc	Ausführungs-planung	Bauherr A, V2: AN	Planer	Detailplanung, Werkplanung
IIj	IVd	Prüfung der AP	Bauherr A, V2: AN	Prüf.ing.	Prüfung der AP
III	III	Ausschreibung	Bauherr A	Planer	Erstellung der AU-Unterlagen, Ausschreibung, Vergabe
IVa	IVa	Bauherren-Organisation B	Bauherr B	Bauherr B	Bildung einer Bauherren-Organisation für die Bauphase
IVb	IVb	Bauvorbereitung	AN (Firma, AR-GE)	Bauherr B	Organisation, Ablauf-planung, Arbeitsvor-bereitung
IVc	IVe	Baudurchführung	AN (Fa, ARGE)	Bauherr B	Realisierung des Bau-vorhabens

IVd	IVf	Bauüberwachung	Bauherr B	Ing. büro	Überwachung der vertrags- und qualitsgerechten Erstellung des Bauwerkes
IVe	IVg	Abnahme 1	Bauherr B	AN (Fa, ARGE)	Abnahme der Leistung, Überprüfung auf Voll-ständigkeit (Vertrag) und Funktionsfähigkeit
IVf	IVh	Mängelbeseitigung	AN (Fa, ARGE)	Bauherr B	Beseitigung von Mängeln aus der 1. Abnahme
IVg	IVi	Abnahme 2	Bauherr B	AN (Fa, Arge)	Abnahme nach Mängelbeseitigung
Va	Va	Beginn Gewährleistung	AN (Fa, Arge)	Bauherr B	Gewährleistung, Umkehr der Beweislast
IVh	IVj	Inbetriebnahme	Bauherr B	AN (Fa, Arge)	Inbetrieb-nahme des Bauwerks
IVi	IVk	Schlussrechnung	AN (Fa, Arge)	Bauherr B	Schlussrechnung nach kpl. Fertigstellung
IVj	IVl	Schlusszahlung	Bauherr B	AN (Fa, Arge)	Anerkennung der vollständigen Leistung, Vergütungsanspruch
IVk	IVm	P-Abschluss	Bauherr B	Planer, AN (Fa, Arge)	Auswertung des Pro-jektes, Lessons-Learned, Best Practices

Vb	Vb	Nutzung	Nutzer (Bauherr C)	AN (Fa, Arge)	Nutzung der Anlage, Mängel-beseitigung nach Bedarf

Tabelle A.1: Teilaufgaben bei der Projektabwicklung [23, S. 34-35]

	Autoren 1)	Sektoren der Infrastruktur 2): wirtschaftl. 3)						soziale							institutionelle		
		Verkehr	Nachrichten	Energie	Wasser	Kommunale Einr.	Umweltschutz	Bildung	Wissenschaft	Gesundheit	Sport u. Erholg.	Soziales	Kultur	(Wohnungsbau)	Recht u. Ordnung	Verwaltung	(Verteidigung)
1	Vito	x		x	x												
2	Hirschmann	x	/	x	/			/		/					/		
3	Schmölders	x	x	x	x												
4	Stopper	x	x	x	x												
5	Tinbergen	x		x				x									
6	Metha	x															
7	Cootner	x	x	x				/		/				/			
8	Klaassen	x	x	x													
9	Stohler	x		x	x			x	x	x	x					x	x
10	Vogel/Duelli	x				x		x		x	x	x	x				
11	Raumordnungs-G.	x	x	x	x			x	x	x		x	x		/	x	
12	Jochimsen	x	x	x	x		x	x	x	x		x				x	
13	Frey B.	x		x	x		x	x	x	x					x	x	x
14	Tjulpanow	x	x	x	x												
15	Köhler/Görlich	x	x	x	x			x		x	x	x	x	x			
16	Recktenwald	x						x	x	x					x		
17	Frey R.	x		x	x			x	x	x	x					x	
18	Tuchtfeld	x	x	x	x			x	x	x					x		x
19	Schröder	x			x	x		x	x	x			x				
20	Kurihara	x	x		x			x		x	x	x					
21	Borchardt	x		x	x			x		x							
22	Bönisch-	x	x	x	x	x		x	x	x		x		x	x	x	x
23	Katterla	x			x		x	x	x	x	x		x				
24	Orientierungsrahmen	x				x		x	x	x	x		x	x	x		
25	Peters	x	x	x	x		x	x	x	x	x		x			x	
26	Formanek/Helms	x				x		x	x	x	x	x			x	x	
27	Buhr	x	x	x	x		x	x	x	x	x			x		x	
28	Biehl u. a.	x		x				x	x	x				x			
29	Brösse	x	x	x	x			x	x	x	x		x		x	x	
30	Nieswandt	x		x	x	x	x	x	x	x	x	x	x	x	x		
31	Heckhausen	x			x	x		x	x	x	x	x	x	x			
32	Recker	x	x	x	x			x	x	x				x			
33	Karchin	x	x	x	x		/	x	x	x	x		x				
34	Ronge/Schmieg	x						x	x	x	x			x			
35	Simonis	x	x	x	x	x	x	x	x	x							
36	Hatzold	x	x	x	x			x	x	x	x	x				x	
	Zahl d. Nennungen	36	19	26	26	8	8	30	23	29	16	10	11	10	10	11	4

x Zuordnung / Zuordnung bei extensiver Definition

Abbildung A.1: Nennung von Infrastruktur-Sektoren in der Literatur [25, S.17]

Bedürfnis	**Infrastrukturoutput (Gut oder Dienst)**	**Zugehöriger Kapitalbestand (materielle Infrastruktur)**
Wasser	Trinkwasser, Wasser für industrielle Nutzung, Bewässerung, Wasser zur Erzeugung von Elektrizität	Reservoire, Kanäle, Wasserstraßen, Rohrleitungen, Bewässerungsanlagen
Wärme	Gas, Öl, Elektrizität, Kohle, Atomkraft	Bohrinseln, Rohrleitungen, Erzeugungsanlagen, Kohlegruben
Licht	Strom, Gas	Produktionsanlagen (Kraftwerke), Bohrtürme, Stromleitungen, Rohrleitungen
Gesundheit	medizinische Versorgung, Abfallbeseitigung, Abwasserentsorgung	Krankenhäuser, Müllhalden, Abwasseranlagen
Sicherheit	Gesetzgebung (Gesetze), Rechtsprechung, Stabilität des Geldwertes, Schutz vor Verbrechen, Verteidigung nach außen, militärische Güter	öffentliche Gebäude, Polizeistationen, militärische Anlagen
Information	Gebrauch des Telefons, Mobilfunks, Radios, Fernsehens, Internets, Zeitungen	Einrichtungen der Telekommunikation, Postämter, Produktionsbetriebe der Zeitungsverlage
Erziehung Bildung	Kinderbetreuung, Vorlesungen, Forschung, Buchausleihe	Kindergärten, Schulen, Universitäten, Forschungseinrichtungen, Bibliotheken
Mobilität	Nutzung der Straßen durch Autos, Busse, Lastwagen	Straßen, Schnellstraßen
	Nutzung der Schienen durch Züge	Schienen, Bahnhöfe
	Nutzung der Flugplätze durch Flugzeuge	Flugplätze
	Nutzung der Häfen durch Schiffe	Häfen
Schutz der Umwelt	saubere Luft, sauberes Wasser	Filter zur Luftreinigung, Wasserwerke

Tabelle A.2: Materielle Infrastrukturen zur Deckung der Erfordernisse menschlichen Lebens [69]

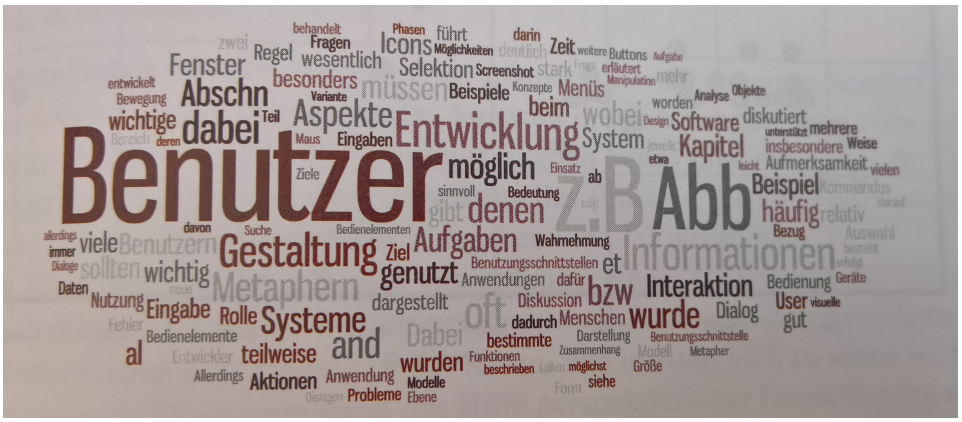


Abbildung A.2: Univariate Visualisierung einer WordC-loud zur Darstellung der häufigsten Wörter eines Buches. [20, S. 455]

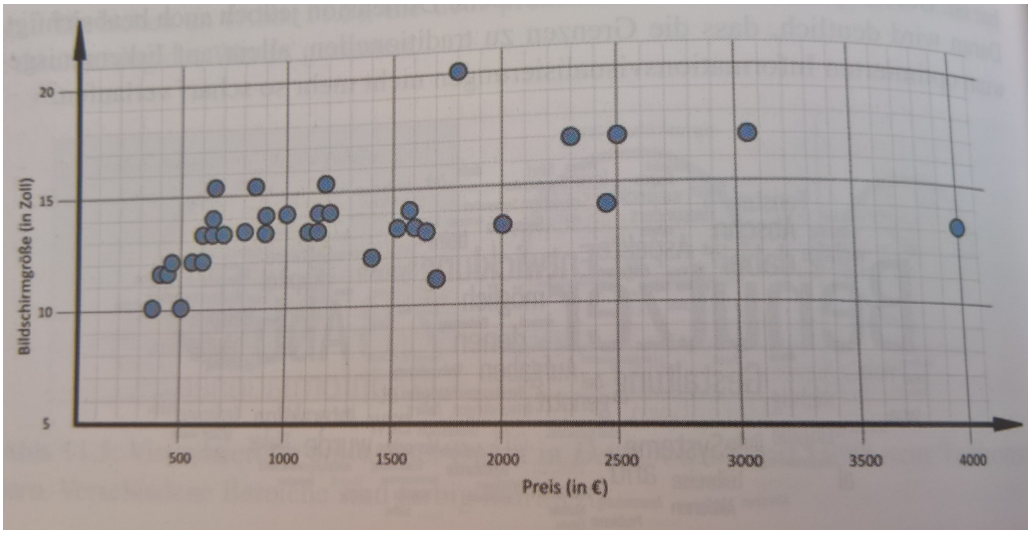


Abbildung A.3: Beispiel eines bivariaten Streudiagramms zum Vergleich von Notebooks anhand von Preisen und Bildschirmgrößen. [20, S. 456]

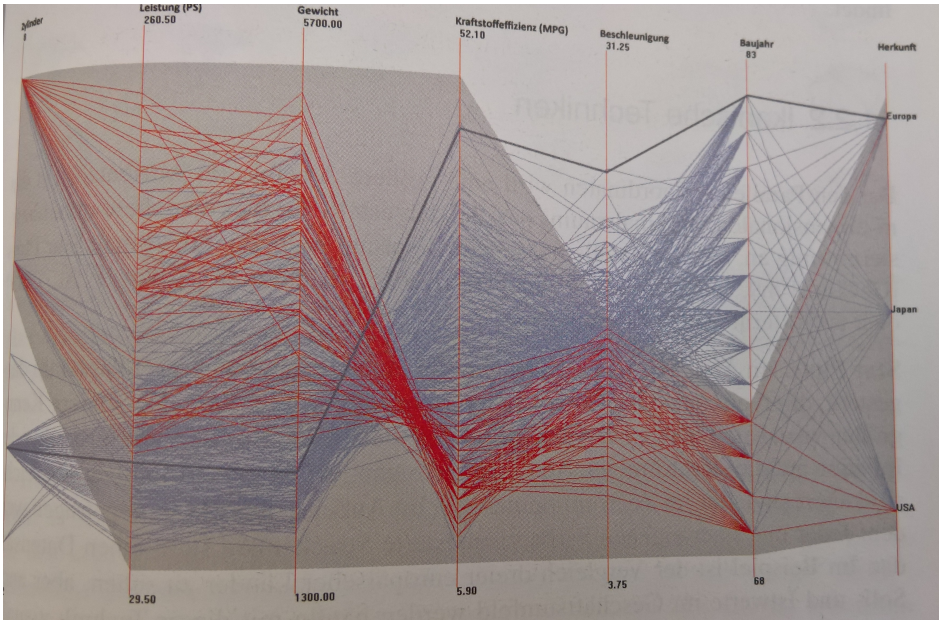


Abbildung A.4: Einsatz Paralleler Koordinaten am Beispiel eines Autodatensatzes mit sieben Attributen. [20, S. 461]

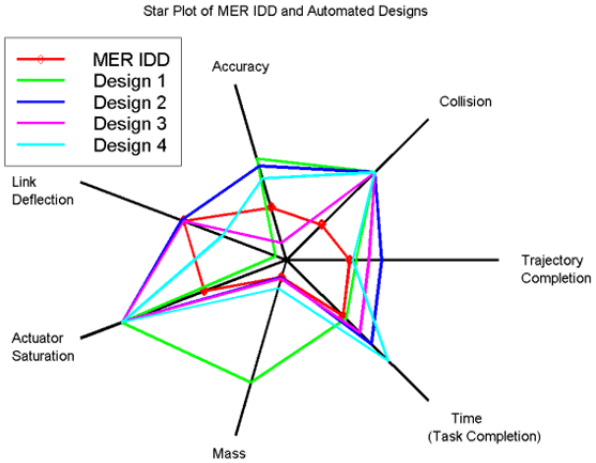


Abbildung A.5: Starplot basierend auf NASA-automatisierten Designs. [90]

Anhang B

Zusätzliche Informationen zum Konzept

```
1       // Constructor
2       function Foo(bar) {
3         // always initialize all instance properties
4         this.bar = bar;
5         this.baz = 'baz'; // default value
6       }
7       // class methods
8       Foo.prototype.fooBar = function() {
9
```

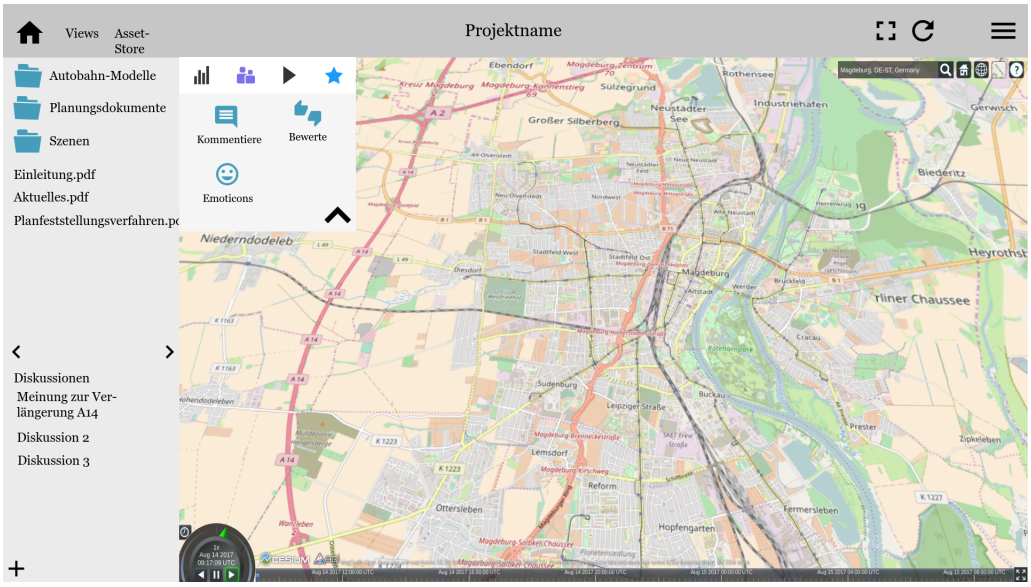
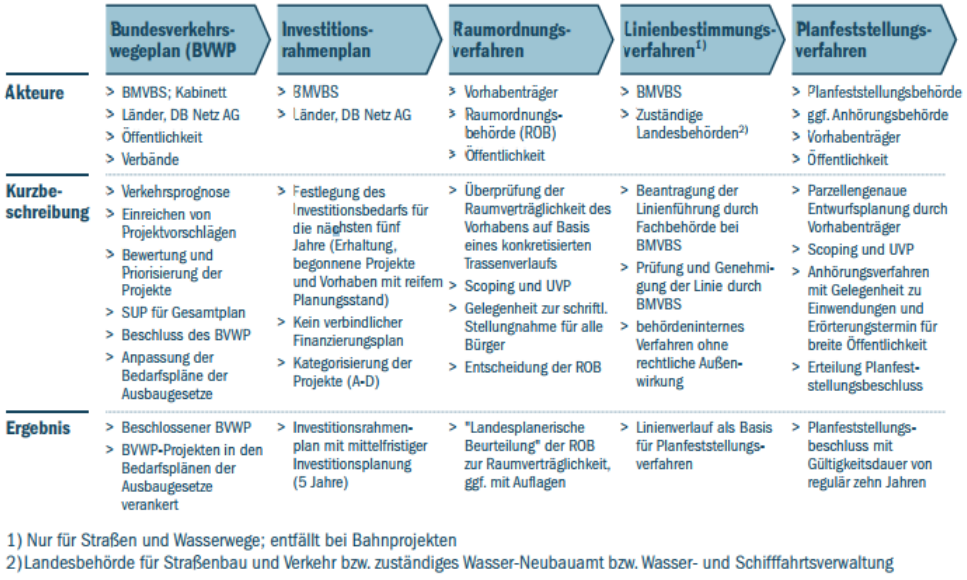


Abbildung B.1: Mockup der Projekt-Dashboard-UI mit Interaktionelementen. (Eigene Darstellung)



	Bundesverkehrswegeplan (BVWP	Investitionsrahmenplan	Raumordnungsverfahren	Linienbestimmungsverfahren[1]	Planfeststellungsverfahren
Akteure	> BMVBS; Kabinett > Länder, DB Netz AG > Öffentlichkeit > Verbände	> BMVBS > Länder, DB Netz AG	> Vorhabenträger > Raumordnungsbehörde (ROB) > Öffentlichkeit	> BMVBS > Zuständige Landesbehörden[2]	> Planfeststellungsbehörde > ggf. Anhörungsbehörde > Vorhabenträger > Öffentlichkeit
Kurzbeschreibung	> Verkehrsprognose > Einreichen von Projektvorschlägen > Bewertung und Priorisierung der Projekte > SUP für Gesamtplan > Beschluss des BVWP > Anpassung der Bedarfspläne der Ausbaugesetze	> Festlegung des Investitionsbedarfs für die nächsten fünf Jahre (Erhaltung, begonnene Projekte und Vorhaben mit reifem Planungsstand) > Kein verbindlicher Finanzierungsplan > Kategorisierung der Projekte (A-D)	> Überprüfung der Raumverträglichkeit des Vorhabens auf Basis eines konkretisierten Trassenverlaufs > Scoping und UVP > Gelegenheit zur schriftl. Stellungnahme für alle Bürger > Entscheidung der ROB	> Beantragung der Linienführung durch Fachbehörde bei BMVBS > Prüfung und Genehmigung der Linie durch BMVBS > behördeninternes Verfahren ohne rechtliche Außenwirkung	> Parzellengenaue Entwurfsplanung durch Vorhabenträger > Scoping und UVP > Anhörungsverfahren mit Gelegenheit zu Einwendungen und Erörterungstermin für breite Öffentlichkeit > Erteilung Planfeststellungsbeschluss
Ergebnis	> Beschlossener BVWP > BVWP-Projekten in den Bedarfsplänen der Ausbaugesetze verankert	> Investitionsrahmenplan mit mittelfristiger Investitionsplanung (5 Jahre)	> "Landesplanerische Beurteilung" der ROB zur Raumverträglichkeit, ggf. mit Auflagen	> Linienverlauf als Basis für Planfeststellungsverfahren	> Planfeststellungsbeschluss mit Gültigkeitsdauer von regulär zehn Jahren

1) Nur für Straßen und Wasserwege; entfällt bei Bahnprojekten
2) Landesbehörde für Straßenbau und Verkehr bzw. zuständiges Wasser-Neubauamt bzw. Wasser- und Schifffahrtsverwaltung

Abbildung B.2: figure

Überblick Planungs- und Genehmigungsprozess [4, S. 122].

```
10      };
11      // export the class
12      module.exports = Foo;
```

Listing B.1: Javascript nicht-minifiziert

```
1       function Foo(a){this.bar=a,this.baz='baz'}Foo.prototype.fooBar=
        function(){},module.exports=Foo;
```

Listing B.2: Javascript minifiziert

```
1  [1,2,3].map(n => n + 1);
2
3  async function asyncFunc1() {
4      const writer = openFile('someFile.txt');
5      writer.write('hello');
6      writer.write('world');
7      return writer.close();
8  }
9
10 async function foo() {
11     const [result1, result2] = await Promise.all([
12         asyncFunc1(),
13         asyncFunc1(),
14     ]);
15 }
```

Listing B.3: ES2017 Code

```
1  'use strict';
2
3  var _slicedToArray = function () { function sliceIterator(arr, i) { var _arr = [];
       var _n = true; var _d = false; var _e = undefined; try { for (var _i = arr
       [Symbol.iterator](), _s; !(_n = (_s = _i.next()).done); _n = true) { _arr.
       push(_s.value); if (i && _arr.length === i) break; } } catch (err) { _d =
       true; _e = err; } finally { try { if (!_n && _i["return"]) _i["return"](); }
       finally { if (_d) throw _e; } } return _arr; } return function (arr, i) { if (
       Array.isArray(arr)) { return arr; } else if (Symbol.iterator in Object(arr)) {
        return sliceIterator(arr, i); } else { throw new TypeError("Invalid
       attempt to destructure non-iterable instance"); } }; }();
4
5  var asyncFunc1 = function () {
6      var _ref = _asyncToGenerator( /*#__PURE__*/regeneratorRuntime.mark(
       function _callee() {
7          var writer;
8          return regeneratorRuntime.wrap(function _callee\$(_context) {
9              while (1) {
10                 switch (_context.prev = _context.next) {
11                     case 0:
12                         writer = openFile('someFile.txt');
13
14                         writer.write('hello');
15                         writer.write('world');
16                         return _context.abrupt('return', writer.close());
17
18                     case 4:
19                     case 'end':
20                         return _context.stop();
21                 }
22             }
23         }, _callee, this);
24     }));
25
26     return function asyncFunc1() {
27         return _ref.apply(this, arguments);
28     };
29 }();
30
31 var foo = function () {
32     var _ref2 = _asyncToGenerator( /*#__PURE__*/regeneratorRuntime.mark(
       function _callee2() {
33         var _ref3, _ref4, result1, result2;
34
35         return regeneratorRuntime.wrap(function _callee2\$(_context2) {
36             while (1) {
37                 switch (_context2.prev = _context2.next) {
38                     case 0:
39                         _context2.next = 2;
40                         return Promise.all([asyncFunc1(), asyncFunc1()]);
41
42                     case 2:
43                         _ref3 = _context2.sent;
```

```
44                              _ref4 = _slicedToArray(_ref3, 2);
45                              result1 = _ref4[0];
46                              result2 = _ref4[1];
47
48                          case 6:
49                          case 'end':
50                              return _context2.stop();
51                      }
52                  }
53              }, _callee2, this);
54          }));
55
56          return function foo() {
57              return _ref2.apply(this, arguments);
58          };
59      }();
60
61      function _asyncToGenerator(fn) { return function () { var gen = fn.apply(this,
            arguments); return new Promise(function (resolve, reject) { function step(
            key, arg) { try { var info = gen[key](arg); var value = info.value; } catch (
            error) { reject(error); return; } if (info.done) { resolve(value); } else {
            return Promise.resolve(value).then(function (value) { step("next", value); },
             function (err) { step("throw", err); }); } } return step("next"); }); }; }
62
63      [1, 2, 3].map(function (n) {
64          return n + 1;
65      });
```

Listing B.4: Konvertierter Javascript Code mithilfe von Babel s. B.3

VKE	Abschnitt	Status	Länge (km)
7	AK Schwerin2–AS Ludwigslust–AS Grabow3	fertiggestellt (Baubeginn November 2011; Freigabe am 21. Dezember 2015)	16,2
6	AS Grabow–AS Groß Warnow	im Bau (Baubeginn Juni 2015;[43] geplante Fertigstellung: Ende 2017)	9,8
5	AS Groß Warnow–AS Karstädt	fertiggestellt (Baubeginn Oktober 2011; Freigabe am 21. Dezember 2015)	12,6
4	AS Karstädt–AS Wittenberge	in der Planfeststellung (seit April 2009)	17,5
3.2b	AS Wittenberge–Landesgrenze BB/ST	in der Planfeststellung (seit Februar 2010)	2,6
3.1/3.2a	Landesgrenze BB/ST–AS Seehausen-Nord	im Planänderungsverfahren (seit Januar 2010)	8,8
2.2	südl. AS Seehausen-Nord–AS Osterburg	in der Planfeststellung (seit September 2014)	16,78
2.1	AS Osterburg–AS Stendal-Mitte	in der Planfeststellung (seit Juli 2011)	18,23
1.5	AS Stendal-Mitte–AS Stendal–AS Lüderitz	in der Planfeststellung (seit Januar 2011), seit Mai 2014 ruhend gestellt, eingestellt und neu in der Planfeststellung (seit April 2015)	12,89
1.4	AS Lüderitz–AS Tangerhütte	Baurecht besteht (seit 13. Dezember 2016)	14,83
1.3	AS Tangerhütte–nördl. AS Colbitz	in Bauvorbereitung (seit 20. Juli 2016)	8,51
1.2	nördl. AS Colbitz–AS Wolmirstedt	fertiggestellt (Baubeginn November 2011; Freigabe 29. Oktober 2014)	5,56
1.1	AS Wolmirstedt–AS Haldensleben–AS Dahlenwarsleben	in der Planfeststellung (seit April 2009)	11.5

Tabelle B.1: Darstellung der Streckenabschnitte der A14. [76]

Anhang C

Zusätzliche Informationen zur Realisierung

```
server \{
  listen 80;
  server_name domain.de;

    gzip on;
    gzip_comp_level 6;
    gzip_vary on;
    gzip_min_length  1000;
    gzip_proxied any;
    gzip_types text/plain text/html text/css application/json
    application/x-javascript text/xml application/xml application/xml+
    rss text/javascript;
    gzip_buffers 16 8k;

  return 301 https:\/\/\$host\$request_uri;
\}

server \{
  listen 443 ssl;
  server_name domain.de;

    ssl_certificate /etc/letsencrypt/live/domain.de/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/domain.de/privkey.pem;

    gzip on;
    gzip_comp_level 6;
    gzip_vary on;
    gzip_min_length  1000;
    gzip_proxied any;
    gzip_types text/plain text/html text/css application/json
    application/x-javascript text/xml application/xml application/xml+
    rss text/javascript;
    gzip_buffers 16 8k;

```

```
31      location ^~ /cesium/ {
32          alias /var/www/domain/node_modules/cesium/Build/Cesium/;
33      access_log off;
34      expires 10d;
35      }
36
37      location ^~ /static/ {
38      alias /var/www/domain/static/;
39      access_log off;
40      expires 10d;
41      }
42
43     location ~ /.well-known {
44              allow all;
45      }
46
47    location / {
48      proxy_http_version 1.1;
49      proxy_set_header Upgrade \$http_upgrade;
50      proxy_set_header Connection 'upgrade';
51      proxy_set_header Host \$host;
52      proxy_cache_bypass \$http_upgrade;
53
54      proxy_set_header X-Real-IP \$remote_addr;
55          proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
56      proxy_set_header X-Forwarded-Proto \$scheme;
57          proxy_set_header Host \$http_host;
58          proxy_set_header X-NginX-Proxy true;
59          proxy_read_timeout 5m;
60          proxy_connect_timeout 5m;
61          proxy_pass http://127.0.0.1:8000;
62      proxy_redirect off;
63    }
64  \}
```

Listing C.1: Verwendete NGINX-Konfiguration des Servers

```
1  {
2      "debug":false,
3      "imagerySplit":true,
4      "startPosition": {
5          "longitude": 13.409779, "latitude": 52.520645, "height": 5000, "
       flyDuration": 4
6      }
7      ,
8      "startOrientation": {
9          "heading": 0, "pitch": -1.5707963267948966, "roll": 0
10     }
11     ,
12     "models":[ {
13         "key":"pschladix@gmail.com/Landkreis Börde/models/Box.gltf",
14         "data": {
15             "name":"pschladix@gmail.com/Landkreis Börde/models/Box.gltf"
       ,
```

```
16              "id":"rJ2jgiXsb",
17              "position": {
18                  "x": 3999329.5969648506, "y": 777783.9619835519, "z":
        4891305.20644304
19              }
20              ,
21              "description":"",
22              "minimumPixelSize":50
23          }
24      }
25      ,
26      {
27          "key":"pschladix@gmail.com/Landkreis Börde/models/CesiumMan.gltf
        ",
28          "data": {
29              "name":"pschladix@gmail.com/Landkreis Börde/models/CesiumMan
        .gltf",
30              "id":"rkTilsQiZ",
31              "position": {
32                  "x": 3999329.5969648506, "y": 777783.9619835519, "z":
        4891305.20644304
33              }
34              ,
35              "description":"",
36              "minimumPixelSize":50
37          }
38      }
39      ,
40      {
41          "key":"pschladix@gmail.com/Landkreis Börde/berlin/berlinFull2.
        gltf",
42          "data": {
43              "name":"pschladix@gmail.com/Landkreis Börde/berlin/
        berlinFull2.gltf",
44              "id":"rkq2ejQs-",
45              "position": {
46                  "x": 3999329.5969648506, "y": 777783.9619835519, "z":
        4891305.20644304
47              }
48              ,
49              "description":"",
50              "minimumPixelSize":50
51          }
52      }
53      ],
54      "geojsonLayer":[ {
55          "key":"pschladix@gmail.com/Landkreis Börde/layer/boerde.GeoJson"
        ,
56          "data": {
57              "description":"",
58              "stroke": {
59                  "red": 1, "green": 0.4117647058823529, "blue":
        0.7058823529411765, "alpha": 1
60              }
```

```
61                ,
62                "fill": {
63                    "red": 1, "green": 0.7529411764705882, "blue":
      0.796078431372549, "alpha": 1
64                }
65                ,
66                "strokeWidth":3,
67                "markerSymbol":"?",
68                "clampToGround":false,
69                "extrudedHeight":0
70            }
71        }
72        ,
73        {
74            "key":"pschladix@gmail.com/Landkreis Börde/schalkau/schalkau.
      geojson",
75            "data": {
76                "description":"",
77                "stroke": {
78                    "red": 1, "green": 0.4117647058823529, "blue":
      0.7058823529411765, "alpha": 1
79                }
80                ,
81                "fill": {
82                    "red": 1, "green": 0.7529411764705882, "blue":
      0.796078431372549, "alpha": 1
83                }
84                ,
85                "strokeWidth":3,
86                "markerSymbol":"?",
87                "clampToGround":false,
88                "extrudedHeight":0
89            }
90        }
91        ],
92        "kml":[ {
93            "key":"pschladix@gmail.com/Landkreis Börde/kml/exported.kml",
94            "data": {
95                "clampToGround": false
96            }
97        }
98        ]
99    }
```

Listing C.2: Beispiel einer gespeicherten Szene im JSON-Format. Die Szene enthält zwei einfache Modelle, ein glTF-Modell von Berlin mit 461-Tausend Polygonen, zwei GeoJSON-Layer vom Landkreis Börde und Schalkau, sowie einen einfachen KML-Layer.

```
1  router.get('/', function(req, res) {
2      Project
3      .find({creator: req.user})
4      .select('name description')
5      .exec(function(err, projectList) {
```

```
6           if(err) {
7               console.log(err);
8               res.render('dashboard.pug', { pageTitle: 'Projects not found
        .. '});
9           } else {
10              res.render('dashboard.pug', {
11                  pageTitle: `Your Projects: \${req.user.email}`,
12                  name: req.user.prename,
13                  projectCreated: req.flash('projectCreated'),
14                  projectCreationFailed: req.flash('projectCreationFailed'
        ),
15                  projectList: projectList
16              });
17          }
18      })
19  })
```

Listing C.3: GET-Anforderung, welche alle Projekte eines Nutzers auflistet und an den Client sendet. Hier ist beispielhaft zu sehen, wie die Seite Dashboard mit den Informationen des Benutzeraccounts gefüllt wird.

```
1   // Registers a user by name, email and password (hashed with sha256)
2   router.post('/sign-up', function(req, res, next) {
3       console.log(req.body);
4       User.register(new User({
5           prename: req.body.prename,
6           surname: req.body.surname,
7           username: req.body.email,
8           email: req.body.email
9       }), req.body.password, function(err, user) {
10          if(err) {
11              console.log(err);
12              req.flash('error', 'A user with the given username is
        already registered or password was incorrect.');
13              res.redirect('/sign-up');
14              return;
15          }
16          var handler = passport.authenticate('local',
17          {
18              successRedirect: '/',
19              failureRedirect: '/sign-up',
20              failureFlash: true
21          });
22
23          handler(req, res, next);
24      })
25  })
```

Listing C.4: POST-Anforderung zur Registrierung eines Nutzers mithilfe des Vornamens, Nachnamens, Usernames und einer Email. Die Registrierung findet über die gewählte MongoDB-Datenbank statt.

```
1  // Returns the requested ressource directly by redirecting to aws
2  router.get('/ressource/path/*', (req, res, next) => { checkAuthorization
       (req.params[0], req, res, next); }, function(req, res) {
3      const s3 = new aws.S3({signatureVersion: 'v4'});
4      const bucketName = settings.bucketName;
5      const key = req.params[0];
6      s3.getSignedUrl('getObject', {
7          Bucket: bucketName,
8          Key: key,
9          Expires: 60
10     }, function(err, url) {
11         if(err) {
12             console.error(err);
13             return res.status(404).send('Object not found..');
14         }
15         res.redirect(url);
16     });
17 });
```

Listing C.5: GET-Route zur Anforderung einer Projektressource. Diese Funktion wird dabei nur ausgeführt, wenn der anfordernde Client authorisiert ist.

```
1  static async getList(key) {
2      // get aws folders
3      const xml = await Utility.makeRequest('GET', `/api/ressource/getList
       ?key=\${key}`);
4      // process data to return only a list of file/folder names
5      const parser = new DOMParser();
6      const parsed = parser.parseFromString(xml, 'text/xml');
7
8      // parse contents, keys and substring the filename from the uri
9      const files = _(parsed.getElementsByTagName('Contents'))
10     .chain()
11     .map(function(content) {
12         const complete_uri = \$(content).find('Key:eq(0):eq(0)').html();
13         const file = complete_uri.substr(complete_uri.lastIndexOf('/') +
        1);
14         return { fileName: file, key: complete_uri
15         };
16       })
17     .reject(function(fo) { return fo.fileName == ''; })
18     .sortBy(function(fo) { return Utility.isImage(fo.fileName) ? 1 : 0;
       })
19     .value();
20
21     // parse CommonPrefixes, Prefix and split the uri
22     const commonPrefixes = parsed.getElementsByTagName('CommonPrefixes')
       ;
23     let folders = [];
24     if(commonPrefixes.length != 0) {
25       folders = _.reduce(commonPrefixes, function(acc, commonPrefix) {
26         const f = _.map(commonPrefix.getElementsByTagName('Prefix'),
```

```
        function(prefix) {
27          const complete = prefix.innerHTML;
28          const split = complete.split('/');
29          const folderName = split[split.length - 2];
30          const folder = folderName + '/';
31          return { folderName: folder, key: complete }
32        });
33
34        return [...acc, ...f];
35      }, []);
36    }
37
38    return { files: files, folders: folders };
39 }
```

Listing C.6: Anforderung einer Liste von Dateien in einer AWS S3 Bucket. Rückgabewert ist eine XML-Datei, welche in einem Postprocessing Schritt geparst wird, um Dateien und Ordner zu extrahieren.

```
1  async loadScene(key) {
2      this.scene = { key: key, data: JSON.parse(await AWSUtility.get(key))
        };
3      const scene = this.scene.data;
4      const spl = key.split('/'); spl.splice(0, 2); const title = spl.join
       ('/');
5      this.props.glContainer.setTitle(title === '' ? '/' : title);
6      while(this.sceneContainer.firstChild)
7          this.sceneContainer.removeChild(this.sceneContainer.firstChild);
8      this.cesium = new Cesium.Viewer(this.sceneContainer, {
9          animation: false,
10         timeline: false,
11         baseLayerPicker: false,
12         terrainProvider : new Cesium.CesiumTerrainProvider({
13         url: 'assets.agi.com/stk-terrain/v1/tilesets/world/tiles',
14         requestWaterMask: true,
15         requestVertexNormals: true
16         }),
17         imageryProvider : Cesium.createOpenStreetMapImageryProvider({
18             url: 'https://a.tile.openstreetmap.org/'
19         }),
20         shadows: true,
21         terrainShadows: true,
22         projectionPicker: true
23     });
24     this.cesium.scene.globe.depthTestAgainstTerrain = true;
25     this.cesium.scene.globe.enableLighting = false;
26     this.cesium.clock.shouldAnimate = true;
27     this.cesium.camera.setView({
28         destination: Cartesian3.fromDegrees(startPosition.longitude,
       startPosition.latitude, startPosition.height),
29         orientation: {
30             heading: 0.0,
31             pitch: -Cesium.Math.PI_OVER_TWO,
```

```
32              roll: 0.0
33          }
34      });
35      // load models
36      if(scene.models) {
37          for(const model of scene.models) { this.loadModel(model); }
38      }
39      // load geojson
40      if(scene.geojsonLayer) {
41          for(const geoj of scene.geojsonLayer) {this.loadGeoJson(geoj);}
42      }
43  }
```

Listing C.7: Laden einer Szene in CesiumJS. Die Daten der Szene werden über die Serverarchitektur mit der eigenen Klasse AWSUtility abgefragt.

Anhang D

Zusätzliche Informationen zu den Ergebnissen

GTmetrix The web should be fast. Executive Summary

Performance Report for:
https://vreya.de/

Report generated: Tue, Sep 26, 2017, 12:56 AM -0700
Test Server Region: Vancouver, Canada
Using: Firefox (Desktop) 53.0, PageSpeed 1.15-gt1, YSlow 3.1.8

PageSpeed Score	YSlow Score	Fully Loaded Time	Total Page Size	Requests
A (100%)	A (96%)	4.0s	6.33MB	6

Top 4 Priority Issues

Specify a cache validator	A (98)	AVG SCORE: 93%	SERVER	HIGH
Optimize images	A (99)	AVG SCORE: 70%	IMAGES	HIGH
Minify CSS	A (99)	AVG SCORE: 95%	CSS	HIGH
Minify HTML	A (99)	AVG SCORE: 98%	CONTENT	LOW

How does this affect me?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, **Google has announced that they are using page speed in their ranking algorithm.**

About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google PageSpeed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

About the Developer

GT.net Performance Hosting

GTmetrix is developed by the good folks at **GT.net**, a Vancouver-based performance hosting company with over 21 years experience in web technology.

https://gt.net/

What do these grades mean?

This report is an analysis of your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The **grades you see represent** how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

What's in this report?

This report covers basic to technical analyses on your page. It is categorized under many headings:

- **Executive:** Overall score information and Priority Issues
- **History:** Graphed history of past performance
- **Waterfall:** Graph of your site's loading timeline
- **Technical:** In-depth PageSpeed & YSlow information

These will provide you with a snapshot of your performance.

GTmetrix

YSlow Recommendations

YSlow Recommendations

RECOMMENDATION	GRADE	RELATIVE	TYPE	PRIORITY
Use a Content Delivery Network (CDN)	C (70)	AVG SCORE: 16%	SERVER	MEDIUM
Add Expires headers	B (89)	AVG SCORE: 26%	SERVER	HIGH
Make fewer HTTP requests	A (100)	AVG SCORE: 33%	CONTENT	HIGH
Compress components with gzip	A (100)	AVG SCORE: 84%	SERVER	HIGH
Minify JavaScript and CSS	A (100)	AVG SCORE: 73%	CSS/JS	MEDIUM
Avoid URL redirects	A (100)	AVG SCORE: 87%	CONTENT	MEDIUM
Make AJAX cacheable	A (100)	AVG SCORE: 100%	JS	MEDIUM
Remove duplicate JavaScript and CSS	A (100)	AVG SCORE: 100%	CSS/JS	MEDIUM
Avoid AlphaImageLoader filter	A (100)	AVG SCORE: 99%	CSS	MEDIUM
Avoid HTTP 404 (Not Found) error	A (100)	AVG SCORE: 99%	CONTENT	MEDIUM
Reduce the number of DOM elements	A (100)	AVG SCORE: 93%	CONTENT	LOW
Use cookie-free domains	A (100)	AVG SCORE: 49%	COOKIE	LOW
Use GET for AJAX requests	A (100)	AVG SCORE: 100%	JS	LOW
Avoid CSS expressions	A (100)	AVG SCORE: 99%	CSS	LOW
Reduce DNS lookups	A (100)	AVG SCORE: 69%	CONTENT	LOW
Reduce cookie size	A (100)	AVG SCORE: 100%	COOKIE	LOW
Make favicon small and cacheable	A (100)	AVG SCORE: 100%	IMAGES	LOW
Configure entity tags (ETags)	A (100)	AVG SCORE: 88%	SERVER	LOW
Make JavaScript and CSS external	(n/a)		CSS/JS	MEDIUM

Pentest-Tools.com

Website Vulnerability Scanner Report

✔ **https://vreya.de**

Summary

Overall risk level:

Medium

Risk ratings:

Scan information:

Start time:	2017-09-26 11:01:52
Finish time:	2017-09-26 11:17:55
Scan duration:	963.0 seconds
Tests performed:	26/26
Scan status:	Finished

Findings

Other security issues found

Server leaks inodes via ETags, header found with file /, fields: 0xW/c10 0x8KEpDV4FmuyEngpx0YYi6abDJso

Details

Risk description:
These findings should be manually analyzed and it must be decided if they present a security risk or not.

Recommendation:
We recommend taking appropriate actions according to the results of the risk analysis performed.

Server software and technology found

Technology	unknown
Server	nginx 1.10.0
Operating system	unknown

Details

Risk description:
An attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:
We recommend you to eliminate the information which permit the identification of software platform, technology, server and operating system: HTTP server headers, meta information, etc.

More information about this issue:
https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002).

Missing HTTP security headers

HTTP Security Header	Header Role	Status
X-Frame-Options	Protects against Clickjacking attacks	Not set
X-XSS-Protection	Mitigates Cross-Site Scripting (XSS) attacks	Not set
Strict-Transport-Security	Protects against man-in-the-middle attacks	Not set
X-Content-Type-Options	Prevents possible phishing or XSS attacks	Not set

X-Content-Type-Options	Prevents possible phishing or XSS attacks	Not set

Details

Risk description:
Because the **X-Frame-Options** header is not sent by the server, an attacker could embed this website into an iframe of a third party website. By manipulating the display attributes of the iframe, the attacker could trick the user into performing mouse clicks in the application, thus performing activities without user's consent (ex: delete user, subscribe to newsletter, etc). This is called a Clickjacking attack and it is described in detail here:
https://www.owasp.org/index.php/Clickjacking

The **X-XSS-Protection** HTTP header instructs the browser to stop loading web pages when they detect reflected Cross-Site Scripting (XSS) attacks. Lack of this header exposes application users to XSS attacks in case the web application contains such vulnerability.

The HTTP **Strict-Transport-Security** header instructs the browser not to load the website via plain HTTP connection but always use HTTPS. Lack of this header exposes the application users to the risk of data theft or unauthorized modification in case the attacker implements a man-in-the-middle attack and intercepts the communication between the user and the server.

The HTTP **X-Content-Type-Options** header is addressed to Internet Explorer browser and prevents it from reinterpreting the content of a web page (MIME-sniffing) and thus overriding the value of the Content-Type header). Lack of this header could lead to attacks such as Cross-Site Scripting or phishing.

Recommendation:
We recommend you to add the X-Frame-Options HTTP response header to every page that you want to be protected against Clickjacking attacks.
More information about this issue:
https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet

We recommend setting the X-XSS-Protection header to "X-XSS-Protection: 1; mode=block".
More information about this issue:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

We recommend setting the Strict-Transport-Security header.
More information about this issue:
https://www.owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet

We recommend setting the X-Content-Type-Options header to "X-Content-Type-Options: nosniff".
More information about this issue:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

Interesting HTTP headers found

HTTP Header	Value	Remark
x-powered-by	Express	Interesting header
access-control-allow-origin	*	Interesting header
Content-Encoding	deflate	Server is vulnerable to the BREACH attack

Details

Risk description:
These HTTP headers usually expose information about the backend technology, software version or other sensitive information. This could be used by attackers to mount specific attacks against the server.

Recommendation:
We recommend you to analyze if the HTTP headers mentioned above are needed for the correct functionality of the web application. If not, you should reconfigure the web server not to send them anymore.

More information about this issue:
https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002).

No security issue found regarding HTTP cookies

Directory listing not found

No WAF/Load Balancer/Proxy was detected

Quellenverzeichnis

Literatur

[1] Uwe Altrock. *Mega-Projekte und Stadtentwicklung*. Altrock, 2003 (siehe S. 6, 7).

[2] BertelsmannStiftung. *Bürgerbeteiligung und Infrastrukturplanung. Management Report*. 2012. URL: https://www.bertelsmann-stiftung.de/fileadmin/files/user_upload/Buergerbeteiligung_und_Infrastrukturplanung.pdf (besucht am 26. 08. 2017) (siehe S. 42).

[3] Maria Brovelli u. a. „Land use and land cover maps of Europe: a Web-GIS platform“. In: XLI-B7 (Juni 2016), S. 913–917 (siehe S. 26, 35).

[4] Roland Berger Strategy Consultants Bundesverband der Deutschen Industrie e.V. *Best-Practices-Studie zur Verkehrsinfrastrukturplanung und -finanzierung in der EU*. 2013. URL: %5Curl%7Bhttp://bdi.eu/media/user_upload/20131024_BDI_Verkehrsinfrastruktur_Langfassung_gesamt.pdf%7D (besucht am 30. 06. 2017) (siehe S. 10, 40, 91).

[5] Muhammad Bute und M K. Ahmed. „Spatial Map Geographic Information System for Mobile Devices“. In: 31 (März 2017). Research Gate, S. 1–8 (siehe S. 27).

[6] Maria Francesca Costabile. *Human-Computer Interaction - INTERACT 2005. IFIP TC 13 International Conference, Rome, Italy, September 12-16, 2005, Proceedings*. Springer-Science & Business Media, 2005, 2005 (siehe S. 23).

[7] Petter Eide. *Qualication and Traceability of Requirements*. URL: http://www.idi.ntnu.no/grupper/su/fordypningsprosjekt-2005/eide-fordyp05.pdf (besucht am 16. 09. 2017) (siehe S. 42).

[8] J. Engel u. a. „An immersive visualization system for virtual 3D city models“. In: *2012 20th International Conference on Geoinformatics*. Juni 2012, S. 1–7 (siehe S. 26, 80).

[9] Runder Tisch GIS e.V. *3D-GIS und Energie*. Version 1.0. Runder Tisch GIS e.V., Geschäftsstelle am Lehrstuhl für Geoinformatik, 2015 (siehe S. 13, 14).

[10] VDI Gesellschaft Bauen und Gebäudetechnik. *Richtlinie VDI 7001. Kommunikation und Öffentlichkeitsbeteiligung bei Planung und Bau von Infrastrukturprojekten*. Berlin. Beuth Verlag (März 2014), 2014 (siehe S. 10–12).

[11] IFD-Allensbach. *Allensbacher Archiv, IFD Umfrage 10076*. s.a. http://www.ifd-allensbach.de/uploads/tx_reportsndocs/Oktober11_Europa.pdf. 2011. (Besucht am 29. 06. 2017) (siehe S. 9).

[12] Fraunhofer Institut IFF. *VRAuthor Software*. Magdeburg. URL: https://www.iff.fraunhofer.de/ (besucht am 31. 07. 2017) (siehe S. 39–41, 55).

[13] Sandvine Incorporated. *2016 Global Internet Phenomena*. 2016. URL: https://www.sandvine.com/downloads/general/global-internet-phenomena/2016/global-internet-phenomena-report-latin-america-and-north-america.pdf (besucht am 30. 08. 2017) (siehe S. 58).

[14] *Infrastruktur zwischen Standortvorteil und Investitionsbedarf*. Online erhältlich unter https://www.iwkoeln.de/_storage/asset/145746/storage/master/file/4408238/download/IW-Studie%20Infrastruktur.pdf. Feb. 2014. (Besucht am 14. 06. 2017) (siehe S. 1).

[15] Reimut Jochimsen. *Theorie der Infrastruktur. Grundlagen der marktwirtschaftlichen Entwicklung*. Mohr (Siebeck), 1966 (siehe S. 4).

[16] Konieczek-Woger und Köppel. *Für wen planen wir? Analyse und Vergleich zweier Planungskontroversen*. Universitätsverlag der TU Berlin, 2016 (siehe S. 6).

[17] Christine Kroemer. *Materielle Infrastruktur: Begriff, Formen und Funktionen*. Online erhältlich unter https://www.uni-trier.de/fileadmin/fb4/prof/VWL/SUR/Lehre/WS0607/SeminarInfrastruktur/privat/Leistungsnachweise/HA/Kroemer.pdf. Jan. 2007. (Besucht am 23. 06. 2017) (siehe S. 5).

[18] R. Miao, J. Song und Y. Zhu. „3D geographic scenes visualization based on WebGL“. In: *2017 6th International Conference on Agro-Geoinformatics*. Aug. 2017, S. 1–6 (siehe S. 26).

[19] Finian Mwalongo u. a. „State-of-the-Art Report in Web-based Visualization“. In: *Computer Graphics Forum* (2016) (siehe S. 32).

[20] Bernhard Preim. und R. Dachselt. *Interaktive Systeme. Band 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung*. Springer-Verlag Heidelberg 1999, 2010 (siehe S. 16–21, 88, 89).

[21] Bernhard Preim und R. Dachselt. *Interaktive Systeme. Band 2: User Interface Engineering, 3D-Interaktion, Natural User Interfaces*. Springer-Verlag Heidelberg 1999, 2015 (siehe S. 22).

[22] Lukas Riesen. *Infrastrukturinvestitionen für institutionelle Investoren*. Research Paper Nr. 4, erhältlich unter http://www.ppcmetrics.ch/files/publications/files/2014-12%2010%20Research%20Paper%20Infrastruktur.pdf. 2014. (Besucht am 29. 06. 2017) (siehe S. 9).

[23] Konrad Spang. *Projektmanagement von Verkehrsinfrastrukturprojekten (VDI-Buch)*. Springer Vieweg(18.März 2016), 2016 (siehe S. 7, 8, 85).

[24] Karl J. Thomé-Kozmiensky. *Immissionsschutz. Teil: Bd. 3., Aktuelle Entwicklungen im anlagenbezogenen Planungsprozess und Immissionsschutz*. Neuruppin:TK, 2012 (siehe S. 38).

[25] Hans. F Trunzer. *Infrastrukturinvestitionen und Wirtschaftswachstum*. Bad Honnef, Bock und Herchen, 1980 (siehe S. 86).

[26] Liqiang Zhang u. a. „Web-based visualization of large 3D urban building models“. In: *International Journal of Digital Earth* 7.1 (2014), S. 53–67. eprint: http://dx.doi.org/10.1080/17538947.2012.667159. URL: http://dx.doi.org/10.1080/17538947.2012.667159 (siehe S. 26).

Online-Quellen

[27] Agency9. *CityPlanner*. URL: https://cityplanneronline.com/ (besucht am 28. 06. 2017) (siehe S. 30, 31, 37).

[28] Akamai. *Akamai Customer Innovation*. URL: https://www.akamai.com/us/en/our-customers.jsp (besucht am 30. 08. 2017) (siehe S. 57).

[29] Amazon. *Amazon S3*. URL: https://aws.amazon.com/de/s3/details/ (besucht am 30. 08. 2017) (siehe S. 58).

[30] Amazon. *Amazon Web Services*. URL: https://aws.amazon.com/de/ (besucht am 30. 08. 2017) (siehe S. 57).

[31] ArcGIS. *Clean Streets*. URL: http://lahub.maps.arcgis.com/apps/MapJournal/index.html?appid=7279dc87ea9e416d9f90bf844505a54a (besucht am 04. 07. 2017) (siehe S. 29).

[32] Autodesk. *AutoCAD Civil 3D. Für eine bessere Tief- und Infrastrukturbauplanung*. URL: https://www.autodesk.de/products/autocad-civil-3d/overview (besucht am 17. 07. 2017) (siehe S. 28, 36).

[33] Autodesk. *Infraworks. Infrastructure design reimagined*. URL: https://www.autodesk.com/products/infraworks/overview (besucht am 17. 07. 2017) (siehe S. 27, 28, 36, 80).

[34] Autodesk. *Neu in AutoCAD Civil 3D | Features*. URL: https://www.autodesk.de/products/autocad-civil-3d/features (besucht am 17. 07. 2017) (siehe S. 28).

[35] Autodesk. *What's New In Infraworks | Features*. URL: https://www.autodesk.com/products/infraworks/features (besucht am 17. 07. 2017) (siehe S. 27).

[36] BabylonJS. *GitHub - BabylonJS: A Complete JavaScript framework for building 3D games with HTML5 and WebGL*. URL: https://github.com/BabylonJS/Babylon.js (besucht am 06. 07. 2017) (siehe S. 34).

[37] Prof. Dr. Frank Brettschneider. *VDI 7001: Kommunikation und Öffentlichkeitsbeteiligung bei Bau- und Infrastrukturprojekten*. URL: http://www.vdi-suedwest.de/fileadmin/user_upload/VDI_Stuttgart/AK_Vortragsflyer/AK_TGA/2015-02-09_Brettschneider.pdf (besucht am 30. 06. 2017) (siehe S. 9, 11).

[38] CesiumJS. *Cesium - WebGL Virtual Globe and Map Engine*. URL: https://cesiumjs.org/ (besucht am 06. 07. 2017) (siehe S. 34).

[39] CesiumJS. *New York City*. URL: https://cesiumjs.org/NewYork/ (besucht am 06. 07. 2017) (siehe S. 33).

[40] CodeRespawn. *Commits coderespawn/dock-spawn*. URL: https://github.com/coderespawn/dock-spawn/commits/master (besucht am 29. 08. 2017) (siehe S. 52).

[41] Open Geospatial Consortium. *Geography Markup Language*. URL: http://www.opengeospatial.org/standards/gml (besucht am 04. 07. 2017) (siehe S. 14).

[42] Open Geospatial Consortium. *Keyhole Markup Language*. URL: http://www.opengeospatial.org/standards/kml/ (besucht am 04. 07. 2017) (siehe S. 14).

[43] A Medium Corporation. *Netflix likes react*. URL: https://medium.com/netflix-techblog/netflix-likes-react-509675426db (besucht am 29. 08. 2017) (siehe S. 52).

[44] COWI. *Taking the faster road*. URL: https://damassets.autodesk.net/content/dam/autodesk/www/products/infraworks/fy18/case-studies/pdf/adsk_customer-story_cowi_en_no.pdf (besucht am 17. 07. 2017) (siehe S. 27).

[45] Facebook. *Showcase*. URL: https://facebook.github.io/react-native/showcase.html (besucht am 29. 08. 2017) (siehe S. 52).

[46] Internet Engineering Task Force. *RFC 7946, The GeoJSON Format*. URL: https://tools.ietf.org/html/rfc7946 (besucht am 04. 07. 2017) (siehe S. 14).

[47] Mozilla Foundation. *ECMAScript 2015*. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript/ECMAScript_2015_support_in_Mozilla (besucht am 16. 09. 2017) (siehe S. 52).

[48] Electron San Francisco. *Electron | Build cross platform desktop apps with Javascript, CSS and HTML*. URL: https://electron.atom.io/ (besucht am 06. 07. 2017) (siehe S. 32).

[49] Barthauer Software GmbH. *BARTHAUER: Softwarelösungen*. URL: http://www.barthauer.de/produkte/softwareloesungen.html (besucht am 05. 07. 2017) (siehe S. 30).

[50] Google. *Google Cloud Platform*. URL: https://cloud.google.com/customers/ (besucht am 30. 08. 2017) (siehe S. 57).

[51] Khronos Group. *glTF Spezifikation*. URL: https://github.com/KhronosGroup/glTF/blob/master/specification/2.0/README.md (besucht am 04. 07. 2017) (siehe S. 15, 73).

[52] Khronos Group. *WebGL - OpenGL ES for Web*. URL: https://www.khronos.org/webgl/ (besucht am 06. 07. 2017) (siehe S. 32).

[53] HafenCity.de. *HafenCity Hamburg Stand der Entwicklung*. URL: http://www.hafencity.com/de/ueberblick/hafencity-hamburg-stand-der-entwicklung.html (besucht am 28. 06. 2017) (siehe S. 7).

[54] Fraunhofer IGD. *About-instant3Dhub*. URL: http://instant3dhub.org/index.php/about-vaas/ (besucht am 04. 07. 2017) (siehe S. 31).

[55] The Imagineers. *Windplanner*. URL: https://windplanner.com/ (besucht am 28. 06. 2017) (siehe S. 30, 37).

[56] Epic Games Inc. *About Unreal Engine 4*. URL: https://www.unrealengine.com/features (besucht am 06. 07. 2017) (siehe S. 35).

[57] ESRI Inc. *ArcGIS*. s.a. https://www.youtube.com/watch?v=mLQN9AZREpM u. https://www.youtube.com/watch?v=tNLGZWRkrJs. URL: http://www.esri.com/arcgis/about-arcgis (besucht am 05. 07. 2017) (siehe S. 29).

[58] ESRI Inc. *Coastal Views*. URL: http://apps.esriuk.com/app/CoastalViews/2/wmt/view/49c7d86935944b2cbb616a35f3f29ed2/index.html (besucht am 04. 07. 2017) (siehe S. 29).

[59] ESRI Inc. *Submarine Cables*. URL: https://maps.esri.com/rc/cable/index.html (besucht am 04. 07. 2017) (siehe S. 29).

[60] Kangax. *ECMAScript 6 compatibility table*. URL: http://kangax.github.io/compat-table/es6/ (besucht am 29. 08. 2017) (siehe S. 50).

[61] Kripken. *GitHub - kripken/emscripten*. URL: https://github.com/kripken/emscripten (besucht am 06. 07. 2017) (siehe S. 35).

[62] Dipl.-Geogr. Christiane Martin (Leitung). *Urban Entertainment Center*. URL: http://www.spektrum.de/lexikon/geographie/urban-entertainment-center/8477 (besucht am 28.06.2017) (siehe S. 7).

[63] mrdoob. *GitHub - mrdoob/three.js*. URL: https://github.com/mrdoob/three.js/ (besucht am 06.07.2017) (siehe S. 34).

[64] Ryan Paul. *A behind-the-scenes look at LinkedIn's mobile engineering*. URL: https://arstechnica.com/information-technology/2012/10/a-behind-the-scenes-look-at-linkedins-mobile-engineering/2/ (besucht am 30.08.2017) (siehe S. 54).

[65] PlayCanvas. *GitHub - playcanvas/engine: JavaScript game engine built on WebGL and WebVR*. URL: https://github.com/playcanvas/engine (besucht am 06.07.2017) (siehe S. 34).

[66] Prof. Dr. Bernhard Preim. *Lecture: Three-dimensional & Advanced Interaction*. URL: http://isgwww.cs.uni-magdeburg.de/cas/teaching/TAI2017.php (besucht am 05.08.2017) (siehe S. 23).

[67] Landesverwaltungsamt Sachen-Anhalt. *Planfeststellung*. URL: https://lvwa.sachsen-anhalt.de/das-lvwa/wirtschaft-verkehr/planfeststellung/ (besucht am 15.07.2017) (siehe S. 7, 80).

[68] Landesverwaltungsamt Sachsen-Anhalt. *Rechtliche Zulassung von Bauvorhaben durch Planfeststellungsverfahren*. URL: https://lvwa.sachsen-anhalt.de/fileadmin/Bibliothek/Politik_und_Verwaltung/LVWA/LVwA/Dokumente/pressestelle/publikationen/flyer/planfeststellung.pdf (besucht am 15.07.2017) (siehe S. 11).

[69] Universität Siegen. *Zum Begriff der Infrastruktur*. URL: https://www.uni-siegen.de/infrastructure_research/infrastructure/ (besucht am 23.06.2017) (siehe S. 4–6, 87).

[70] Unity Technologies. *Unity - Products*. URL: https://unity3d.com/de/unity (besucht am 06.07.2017) (siehe S. 35).

[71] Unity. *Unity - Execution Order of Event Functions*. URL: https://docs.unity3d.com/Manual/ExecutionOrder.html (besucht am 07.10.2017) (siehe S. 47).

[72] Landesamt für Vermessung und Geoinformation Sachen Anhalt. *Landesluftbildsammlung*. URL: https://www.lvermgeo.sachsen-anhalt.de/de/leistungen/luftbildsammlung/main.htm (besucht am 04.07.2017) (siehe S. 15).

[73] Landesamt für Vermessung und Geoinformation Sachen-Anhalt. *DGM, DOP*. URL: https://www.lvermgeo.sachsen-anhalt.de/de/leistungen/landesaufnahme/dgm/atkis-dgm.htm (besucht am 04.07.2017) (siehe S. 15).

[74] WebCabin. *GitHub - WebCabin/wcDocker*. URL: https://github.com/WebCabin/wcDocker (besucht am 29. 08. 2017) (siehe S. 52).

[75] Wikipedia. *ACID*. URL: https://de.wikipedia.org/wiki/ACID (besucht am 30. 08. 2017) (siehe S. 55).

[76] Wikipedia. *Bundesautobahn 14*. URL: https://de.wikipedia.org/wiki/Bundesautobahn_14 (besucht am 31. 07. 2017) (siehe S. 40, 94).

[77] Wikipedia. *Comparison of web template engines*. URL: https://en.wikipedia.org/wiki/Comparison_of_web_template_engines (besucht am 29. 08. 2017) (siehe S. 50).

[78] Wikipedia. *ESRI*. URL: https://de.wikipedia.org/wiki/ESRI (besucht am 04. 07. 2017) (siehe S. 28).

[79] Wikipedia. *Flughafen Berlin Brandenburg*. URL: https://de.wikipedia.org/wiki/Flughafen_Berlin_Brandenburg (besucht am 29. 06. 2017) (siehe S. 9).

[80] Wikipedia. *Geoinformatik*. URL: https://de.wikipedia.org/wiki/Geoinformatik (besucht am 06. 10. 2017) (siehe S. 25).

[81] Wikipedia. *Geoinformationssysteme*. URL: https://de.wikipedia.org/wiki/Geoinformationssystem#Anwendungsgebiete (besucht am 03. 07. 2017) (siehe S. 13).

[82] Wikipedia. *HTTP ETag*. URL: https://de.wikipedia.org/wiki/HTTP_ETag (besucht am 26. 09. 2017) (siehe S. 77).

[83] Wikipedia. *Hydraulic Fracturing*. URL: %5Curl%7Bhttps://de.wikipedia.org/wiki/Hydraulic_Fracturing%7D (besucht am 30. 06. 2017) (siehe S. 11).

[84] Wikipedia. *Hypertext Transfer Protocol*. URL: https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol (besucht am 06. 10. 2017) (siehe S. 46).

[85] Wikipedia. *Infrastruktur*. URL: https://de.wikipedia.org/wiki/Infrastruktur (besucht am 23. 06. 2017) (siehe S. 4).

[86] Wikipedia. *Kognitive Karte*. URL: https://de.wikipedia.org/wiki/Kognitive_Karte (besucht am 05. 08. 2017) (siehe S. 23).

[87] Wikipedia. *Netzwerkdurchsetzungsgesetz*. URL: https://de.wikipedia.org/wiki/Netzwerkdurchsetzungsgesetz (besucht am 23. 08. 2017) (siehe S. 43).

[88] Wikipedia. *Penetrationstest (Informatik)*. URL: https://de.wikipedia.org/wiki/Penetrationstest_(Informatik) (besucht am 27. 09. 2017) (siehe S. 76).

[89] Wikipedia. *Planfeststellung*. URL: https://de.wikipedia.org/wiki/Planfeststellung (besucht am 01. 10. 2017) (siehe S. 80).

[90] Wikipedia. *Radar Chart*. URL: https://en.wikipedia.org/wiki/Radar_chart (besucht am 09. 07. 2017) (siehe S. 89).

[91] Wikipedia. *React*. URL: https://de.wikipedia.org/wiki/React (besucht am 29. 08. 2017) (siehe S. 52).

[92] Wikipedia. *Reverse Proxy*. URL: https://de.wikipedia.org/wiki/Reverse_Proxy (besucht am 28. 08. 2017) (siehe S. 54).

[93] Wikipedia. *Web 2.0*. URL: https://de.wikipedia.org/wiki/Web_2.0 (besucht am 14. 06. 2017) (siehe S. 2).

[94] WRLD. *WRLD Unity SDK - Asset Store*. URL: https://www.assetstore.unity3d.com/en/?utm_source=Unity&utm_medium=Asset%20Store#!/content/86284 (besucht am 06. 07. 2017) (siehe S. 33, 35).