

TU Dresden

Fakultät Forst-, Geo- und Hydrowissenschaften  
Institut für Kartographie

# Konzeptionierung und Realisierung einer neuen Visualisierung der Ergebnisse des Monitors der Siedlungs- und Freiraumentwicklung

**Masterarbeit**

**zur Erlangung des akademischen Grades  
Master of Science**

**Autor:** Loren Mucha  
MatNr. 3695824

**Tag der Einreichung:** 28. September 2017

**1. Betreuer:** Prof. Dr.-Ing. Dirk Burghardt  
**2. Betreuer:** Dr. Gotthard Meinel





## Leibniz-Institut für ökologische Raumentwicklung

Dr. Gotthard Meinel

Leiter des Forschungsbereichs  
Monitor der Siedlungs-  
und Freiraumentwicklung

Weberplatz 1  
01217 Dresden

Tel. +49 351 4679-254  
Fax +49 351 4679-212

mail: g.meinel@ioer.de  
<http://www.ioer.de>

Dresden, 01.April 2017

### Arbeitsthema:

### Konzeptionierung und Realisierung einer neuen Visualisierung der Ergebnisse des Monitors der Siedlungs- und Freiraumentwicklung

*Hintergrund der Fragestellung:* Der Monitor der Siedlungs- und Freiraumentwicklung (IÖR-Monitor) informiert seit 2010 über die Flächennutzungsentwicklung und in diesem Zusammenhang stehende Themen in Form von Indikatorkarten, Wertetabellen, Statistiken und Diagrammen im Internet. Der Monitor erfreut sich zunehmender Nutzung. Allerdings ist die Implementierung nicht mehr auf dem neuesten Stand und das derzeitige Design kann nicht mehr alle Anforderungen erfüllen, wie beispielsweise optimierte Darstellungen auf mobilen Endgeräten.

*Zielsetzung:* Ziel der Arbeit ist die Konzeptionierung, Implementierung und Testung eines neuen Kartenviewers im IÖR-Monitor auf Grundlage aktueller WebMapping-Technologien (u.a. Leaflet, JSON) sowie die Schaffung neuer Visualisierungsoptionen, um die Informationsfülle noch einfacher und strukturierter, endgerätenabhängig und hochperformant zu visualisieren.

*Arbeitsprogramm:* Konzeption zur Transformation von Inhalten und Funktionalitäten des Monitor des Siedlungs- und Freiraumentwicklung (IÖR-Monitor) in eine neue WebMapping-Oberfläche. Entwurf und Abstimmung der Programmoberfläche sowie der Karten- und Tabellenvisualisierung. Programmierung, Testung und Optimierung des neuen Kartenviewers einschließlich Tabellenfunktionen.

*Voraussetzungen:* Sehr gute Kenntnisse in der Web-Programmierung und in Kartensvisualisierung, sehr gute Kenntnisse von Datenbank-Technologien

**Zeitraum der Bearbeitung**  
05/2017 - 10/2017

### Betreuer

Dr. G. Meinel, IÖR (E-Mail: [G.Meinel@ioer.de](mailto:G.Meinel@ioer.de), Tel. 0351/4679254)

Leibniz-Institut für ökologische  
Raumentwicklung e.V. (IÖR)

Direktor:  
Prof. Dr. Dr. h.c. Bernhard Müller

Vereinsregister des  
Amtsgericht Dresden  
VR 1357

USt-IdNr. DE161411236  
USt-Nr. 203/140/03013

Bankverbindung:  
Deutsche Bank PGK AG  
BLZ: 870 700 24  
Kto.-Nr.: 507 1717 00

IBAN:  
DE21 8707 0024 0507 1717 00  
BIC: DEUTDEDDB870

## **Selbstständigkeitserklärung**

Hiermit versichere ich, dass ich die von mir vorgelegte Arbeit selbstständig und ohne unzulässige Hilfe Dritter verfasst habe. Es wurden keine anderen als die in der Arbeit angegebenen Hilfsmittel und Quellen benutzt. Die wörtlichen und sinngemäß übernommenen Zitate habe ich als solche kenntlich gemacht. Während der Anfertigung dieser Arbeit wurde ich nur von folgenden Personen unterstützt:

- Prof. Dr. Dirk Burghardt
- Dr. Gotthard Meinel

Weitere Personen waren an der geistigen Herstellung der vorliegenden Arbeit nicht beteiligt. Mir ist bekannt, dass die Nichteinhaltung dieser Erklärung zum nachträglichen Entzug des Hochschulabschlusses führen kann.

Dresden, 28. September 2017

Loren Mucha

## Kurzfassung

Der Flächenverbrauch stellt in Deutschland ein zunehmendes Problem dar. Täglich werden über 66 ha natürlicher Fläche durch Straßen oder Siedlungen versiegelt. In Anbetracht dieser Entwicklung steuert die Bundesregierung im Rahmen der Nationalen Nachhaltigkeitsstrategie auf das Ziel hin, bis zum Jahr 2020 die Neuinanspruchnahme von Flächen für Siedlungen und Verkehr auf 30 Hektar pro Tag zu verringern [Wilke, 2017].

Aus diesem Grund rückt das Flächenmonitoring für eine Beurteilung der Flächennutzung weiter in den Vordergrund. Hier leistet das Leibniz-Institut für ökologische Raumentwicklung mit dem Monitor der Siedlungs- und Freiraumentwicklung (IÖR-Monitor) einen wichtigen Beitrag, um den Zustand und die Flächenentwicklung in ganz Deutschland bewerten zu können.

Auf der Basis von Indikatoren in hoher räumlicher Auflösung für das terrestrische Staatsgebiet der Bundesrepublik Deutschland, werden diese in thematischen Karten veranschaulicht. Durch die Visualisierung der Indikatoren kann eine Bereitstellung von Informationen zum Zustand und zur Entwicklung des nur begrenzt verfügbaren und schutzbedürftigen Gutes Fläche bzw. Boden erfolgen.

Der Nutzer hat innerhalb des Dienstes zwei Kartenviewer zur Auswahl. Im Vektorviewer werden die Karten auf der Basis von Vektordaten mit unterschiedlicher räumlicher Ausdehnung visualisiert. Der Rasterviewer bietet basierend auf Rasterkarten eine mögliche Auflösung von bis zu 100m Rasterweite an.

Neben der kartographischen Visualisierung stellt der IÖR-Monitor Tabellen und Entwicklungsgraphen zu allen Indikatoren bereit.

Die Implementierungsbasis dieser Dienstleistung entsprach zum Zeitpunkt der Entwicklung den modernsten Web-Technologien. Seit dem offiziellen Start im Jahre 2010, erfolgten innerhalb der Anwendung keine großen Weiterentwicklungen. Durch die rasante Entwicklung des WWW und verändertes Nutzerverhalten, kann der IÖR-Monitor in der Generation des Web 2.0 nicht mehr alle Anforderungen an eine moderne WebMapping-Anwendung erfüllen. Einen treibenden Faktor stellt vor allem der schnelle Siegeszug von Smartphones dar und damit eine neue Generation von Geräten, die bei der Entwicklung berücksichtigt werden muss.

Mit AJAX lassen sich Anwendungen im Internet realisieren, welche in ihrem funktionalen Umfang und Reaktionsgeschwindigkeiten Desktop-Anwendungen sehr nahe kommen. Diese Schlüsseltechnologie des Web 2.0 stellt auch einen der Gründe für den schnellen Erfolg von Web-Anwendungen in unserer heutigen Zeit dar.

Von dieser Grundlage ausgehend, entstand in dieser Arbeit eine neue Visualisierung der Ergebnisse des IÖR-Monitors. Nach der Einführung in die Grundlagen moderner WebMapping-Technologien, werden durch eine Anforderungsanalyse die Entwicklungspotentiale der vorgenannten Implementierung validiert. Aufbauend auf die formulierte Anforderungsliste erfolgte die Konzeptionierung des User Interfaces und der Funktionalitäten. Im Anschluss erfolgt die Implementierung mit einer vollständigen Realisierung des Konzeptes. Durch einen Performance- und Nutzertest werden die Ergebnisse dieser Arbeit abschließend evaluiert.

## **Abstract**

The use of land is a growing problem in Germany. Over 66 hectares of natural land are being sealed daily by roads or settlements. In view of this development, the Federal Government is pursuing the goal of reducing the new use of land for settlements and traffic to 30 hectares per day by the year 2020 as part of the National Sustainability Strategy [Wilke, 2017].

In order to make the results of the area monitoring available to a broad public, the *Leibniz-Institut für ökologische Raumentwicklung* developed the *Monitor der Siedlungs- und Freiraumentwicklung* (IÖR-Monitor). On the base of indicators in high spatial resolution for the terrestrial territory of the Federal Republic of Germany, these will be vividly visualized in thematic maps. The user has the choice between an overview and detail viewer for this selection. In addition to cartographic visualization, the IÖR-Monitor offers tables and development graphs for all indicators.

The implementation base of this service was then developed under current web-technologies and has not been further developed since that date. The fast development of the WWW and changed user behavior, the IÖR-Monitor can not meet all requirements for a modern webmapping application. A driving factor is the fast success of smartphones and thus a new generation of devices which have to be taken into account during development.

With AJAX applications can be realized on the Internet, which come very close to desktop applications in their functional scope and response speed. This key technology of Web 2.0 also represents one of the reasons for the rapid success of webmapping in our time.

Proceeding from this foundation, a new edition of the IÖR-Monitor was developed in this thesis. After the introduction of the necessary basics, the validation of possible development potentials starts with a requirement analysis. Based on the formulated list of requirements, a conception phase begun where the user interfaces and functionalities arise. Subsequently, the implementation of the whole concept takes place with one complete implementation. With a performance and utility test, the results of this work has been evaluated.

## Inhaltsverzeichnis

|  |           |
|--|-----------|
| <b>Abbildungsverzeichnis</b>   | <b>10</b> |
| <b>Tabellenverzeichnis</b>   | <b>11</b> |
| <b>Abkürzungsverzeichnis</b>   | <b>12</b> |
| <b>1. Einleitung</b>   | <b>13</b> |
| 1.1. Motivation . . . . .  | 13        |
| 1.2. Ziel der Arbeit . . . . .   | 13        |
| 1.3. Aufbau der Arbeit . . . . .   | 14        |
| <b>2. Flächenmonitoring</b>  | <b>15</b> |
| 2.1. Definition . . . . .  | 15        |
| 2.2. Bedeutung und Ziel . . . . .  | 15        |
| 2.3. Monitor der Siedlungs- und Freiraumentwicklung (IÖR-Monitor) . . . . .      | 16        |
| <b>3. WebMapping</b>   | <b>17</b> |
| 3.1. Begriffe . . . . .  | 17        |
| 3.2. Karten im Internet . . . . .  | 18        |
| 3.3. Web 2.0 und WebMapping 2.0 . . . . .  | 18        |
| 3.4. Client-/Server-Modell . . . . .   | 21        |
| 3.4.1. Client . . . . .  | 21        |
| 3.4.2. Server . . . . .  | 22        |
| 3.5. WebMapping am Beispiel des IÖR-Monitors . . . . .                           | 23        |
| 3.5.1. Vektorviewer . . . . .  | 24        |
| 3.5.2. Rasterviewer . . . . .  | 24        |
| <b>4. Technologien des WebMapping 2.0</b>  | <b>26</b> |
| 4.1. Open Source Software . . . . .  | 26        |
| 4.2. Asynchronous JavaScript and XML (AJAX) . . . . .                            | 27        |
| 4.3. JSON . . . . .  | 28        |
| 4.4. Responsive Web Design . . . . .   | 28        |
| 4.4.1. Flexible Grids . . . . .  | 29        |
| 4.4.2. Media Queries . . . . .   | 29        |
| <b>5. Anforderungsanalyse</b>  | <b>31</b> |
| 5.1. Benutzeranalyse . . . . .   | 31        |
| 5.1.1. Methodik . . . . .  | 31        |
| 5.1.2. Ergebnisse . . . . .  | 32        |
| 5.2. Rahmenbedingungen . . . . .   | 33        |
| 5.3. Bestandsanalyse vorhandener Web Mapping Anwendungen . . . . .               | 34        |
| 5.3.1. Methodik . . . . .  | 35        |
| 5.3.2. Auswahl der zu analysierenden Anwendungen . . . . .                       | 36        |
| 5.3.3. Ergebnis . . . . .  | 36        |
| 5.3.3.1. Benutzeroberfläche . . . . .  | 37        |
| 5.3.3.2. Kartengestaltungs- und Analysemöglichkeiten . . . . .                   | 39        |
| 5.3.3.3. Kartenveröffentlichung und -weiterverwendung . . . . .                  | 41        |
| 5.3.3.4. Technische Aspekte . . . . .  | 41        |
| 5.3.4. Zusammenfassung . . . . .   | 41        |
| 5.4. Vergleich der Ergebnisse mit der Implementierung des IÖR-Monitors . . . . . | 43        |
| 5.5. Ableitung von Entwicklungspotentialen . . . . .                             | 45        |

|           |  |           |
|-----------|--|-----------|
| 5.6.      | Analyse von geeigneten Open Source Kartenframeworks . . . . .    | 47        |
| 5.6.1.    | Leaflet . . . . .  | 47        |
| 5.6.2.    | OpenLayers . . . . .   | 47        |
| 5.6.3.    | Vergleich . . . . .  | 48        |
| 5.7.      | Fazit . . . . .  | 49        |
| <b>6.</b> | <b>Konzeption und Design</b>                                     | <b>50</b> |
| 6.1.      | Benutzeroberfläche . . . . .                                     | 50        |
| 6.1.1.    | Navigation . . . . .   | 51        |
| 6.1.2.    | Kartenviewer . . . . .   | 52        |
| 6.1.3.    | MultiView . . . . .  | 53        |
| 6.1.4.    | Responsive Layout . . . . .                                      | 55        |
| 6.2.      | Funktionale Beschreibungen der interaktiven Funktionen . . . . . | 56        |
| 6.2.1.    | Interaktionen zwischen Karten- und Tabellenansicht . . . . .     | 56        |
| 6.2.2.    | Vergleichsmodus . . . . .  | 57        |
| 6.2.3.    | Suchen nach Indikatoren und Orten . . . . .                      | 58        |
| 6.2.4.    | Grundlegende Interaktionen . . . . .                             | 59        |
| 6.2.4.1.  | Ebenenübersicht . . . . .  | 60        |
| 6.2.4.2.  | Identify . . . . .   | 61        |
| 6.2.4.3.  | Änderung der Interpolationsmethode . . . . .                     | 61        |
| 6.2.4.4.  | Zoom . . . . .   | 62        |
| 6.2.4.5.  | Weiterführende Informationen . . . . .                           | 63        |
| 6.2.5.    | Kartengestaltung . . . . .                                       | 63        |
| 6.2.6.    | Räumliche Gliederung . . . . .                                   | 65        |
| 6.2.7.    | Werkzeuge . . . . .  | 66        |
| 6.2.8.    | Feedback und Nutzerunterstützung . . . . .                       | 67        |
| 6.3.      | Fazit . . . . .  | 68        |
| <b>7.</b> | <b>Implementierung</b>   | <b>69</b> |
| 7.1.      | Technologieauswahl . . . . .                                     | 69        |
| 7.1.1.    | jQuery . . . . .   | 69        |
| 7.1.2.    | Bootstrap . . . . .  | 70        |
| 7.1.3.    | Semantic UI . . . . .  | 70        |
| 7.2.      | Umsetzung des Konzeptes . . . . .                                | 70        |
| 7.2.1.    | Interface . . . . .  | 70        |
| 7.2.2.    | Interaktiver Gebietsviewer . . . . .                             | 72        |
| 7.2.2.1.  | Generierung der Gebietskarte . . . . .                           | 72        |
| 7.2.2.2.  | Generierung der Tabelle . . . . .                                | 73        |
| 7.2.2.3.  | Optimierung der Visualisierung . . . . .                         | 74        |
| 7.2.2.4.  | MultiView . . . . .  | 74        |
| 7.2.3.    | Rasterviewer mit Vergleichsmodus . . . . .                       | 75        |
| 7.2.4.    | Suchfunktion . . . . .   | 75        |
| 7.2.5.    | Grundlegende Interaktionen . . . . .                             | 77        |
| 7.2.5.1.  | Interaktionsbuttons . . . . .                                    | 77        |
| 7.2.5.2.  | Identify . . . . .   | 78        |
| 7.2.5.3.  | Zeit Slider . . . . .  | 79        |
| 7.2.5.4.  | Glätten der Karte . . . . .                                      | 79        |
| 7.2.6.    | Kartengestaltung . . . . .                                       | 80        |
| 7.2.7.    | Räumliche Gliederung . . . . .                                   | 80        |
| 7.2.8.    | Werkzeuge . . . . .  | 81        |
| 7.2.9.    | Feedback und Nutzerunterstützung . . . . .                       | 82        |
| 7.3.      | Fazit . . . . .  | 82        |

|  |            |
|--|------------|
| <b>8. Evaluierung</b>                        | <b>83</b>  |
| 8.1. Performance Test . . . . .              | 83         |
| 8.2. Nutzertests . . . . .                   | 85         |
| 8.2.1. Ziele und Art . . . . .               | 85         |
| 8.2.2. Aufbau . . . . .                      | 85         |
| 8.2.3. Durchführung . . . . .                | 87         |
| 8.3. Ergebnisse . . . . .                    | 89         |
| 8.3.1. Aufgaben . . . . .                    | 89         |
| 8.3.2. Gesamteindruck . . . . .              | 92         |
| 8.4. Fazit . . . . .                         | 94         |
| <b>9. Zusammenfassung und Ausblick</b>       | <b>95</b>  |
| 9.1. Zusammenfassung . . . . .               | 95         |
| 9.2. Ausblick . . . . .                      | 97         |
| 9.2.1. Optimierung der Performance . . . . . | 97         |
| 9.2.2. Funktionale Erweiterungen . . . . .   | 97         |
| <b>Literaturverzeichnis</b>                  | <b>99</b>  |
| <b>Anhang</b>                                | <b>108</b> |
| <b>A. Anhang</b>                             | <b>108</b> |
| A.1. Screenshots der Anwendung . . . . .     | 108        |
| A.2. Fragebogen des Nutzertests . . . . .    | 110        |

## Abbildungsverzeichnis

|     |  |    |
|-----|--|----|
| 1.  | Client-Server-Modell . . . . .   | 21 |
| 2.  | Oberflächen des Monitors der Siedlungs- und Freiraumentwicklung . . . . .                            | 24 |
| 3.  | Der Prozessfluss einer Ajax Anwendung . . . . .  | 28 |
| 4.  | Responsive Webdesign am Beispiel . . . . .   | 30 |
| 5.  | Zusammensetzung der befragten Probanden . . . . .  | 31 |
| 6.  | Architektur der serverseitigen Software Plattform . . . . .  | 34 |
| 7.  | Vertikaler Slider und Accordion . . . . .  | 37 |
| 8.  | Responsive-Layout der untersuchten Anwendungen innerhalb der Benutzeranalyse                         | 38 |
| 9.  | Vergleichsmodus in der Anwendung Geo Admin CH . . . . .  | 40 |
| 10. | Vergleich des Interfaces zwischen IÖR-Monitor und des BayernAtlas . . . . .                          | 43 |
| 11. | Entwurf der Benutzeroberfläche . . . . .   | 50 |
| 12. | Zeitslider mit unterstützender Beschriftung . . . . .  | 53 |
| 13. | Konzept der Multi View Ansicht ohne Header . . . . .   | 54 |
| 14. | Erster Entwurf des MultiView für mobile Enderäte . . . . .   | 55 |
| 15. | Konzept des Vergleichsmodus . . . . .  | 57 |
| 16. | Aktivitätsdiagramm der Suche . . . . .   | 59 |
| 17. | Aktivitätsdiagramm der Interaktionsbuttons auf der Karte . . . . .                                   | 60 |
| 18. | Interpolationsvergleich der Rasterkarten . . . . .   | 62 |
| 19. | Informations-Fenster . . . . .   | 63 |
| 20. | Konzept des Menüs der Kartengestaltung . . . . .   | 64 |
| 21. | Konzept des Menüs der räumlichen Gliederung . . . . .  | 65 |
| 22. | Konzept der Werkzeugwahl . . . . .   | 66 |
| 23. | Konzept der Fehlermeldungen für die Anwendung . . . . .  | 68 |
| 24. | Responsive Layout des IÖR-Monitors . . . . .   | 71 |
| 25. | Ablauf der Kartenerstellung . . . . .  | 72 |
| 26. | Ablauf der Tabellenerstellung . . . . .  | 73 |
| 27. | Bildschirmausschnitt der implementierten Suchfunktion nach Indikatoren mit dem Monkeypatch . . . . . | 76 |
| 28. | Anfrage-URL an den GeoNames Webservice . . . . .   | 76 |
| 29. | Identify Funktion . . . . .  | 78 |
| 30. | Screenshots der implementierten Werkzeuge . . . . .  | 81 |
| 31. | Tour durch die Website . . . . .   | 82 |
| 32. | Vergleich der Ladezeiten für die Visualisierung der Gemeinden . . . . .                              | 83 |
| 33. | Vergleich der Ladezeiten der administrativen Gebietseinheiten . . . . .                              | 84 |
| 34. | Vorkenntnisse der Probanden mit dem IÖR-Monitor . . . . .  | 88 |
| 35. | Auswertung von Aufgabe 1 . . . . .   | 89 |
| 36. | Auswertung von Aufgabe 2 . . . . .   | 90 |
| 37. | Auswertung von Aufgabe 3 . . . . .   | 91 |
| 38. | Auswertung von Aufgabe 4 . . . . .   | 92 |
| 39. | Auswertung des Gesamteindruckes . . . . .  | 93 |
| 40. | 3D-Visualisierung mit Rasterkarte . . . . .  | 98 |

## Tabellenverzeichnis

|    |   |    |
|----|---|----|
| 1. | Vergleich von WebMapping 1.0 und WebMapping 2.0-Anwendungen . . . . . | 20 |
| 2. | Benötigte WMS-Parameter eines GetMap-Requests . . . . .               | 23 |
| 3. | Vergleich der beiden Kartenviewer des IÖR-Monitors . . . . .          | 25 |
| 4. | Zusammengefasste Ergebnisse der Bestandsanalyse . . . . .             | 42 |
| 5. | Gegenüberstellung der Grundkarten . . . . .                           | 61 |
| 6. | Aufgaben des Nutzertests und deren Ziel . . . . .                     | 86 |

## Abkürzungsverzeichnis

|        |   |
|--------|---|
| AGS    | Amtlicher Gemeindeschüssel                                |
| AJAX   | Asynchronous JavaScript and XML                           |
| API    | Application Programming Interface                         |
| ATKIS  | Amtlich Topographisch-Kartographisches Informationssystem |
| BfN    | Bundesamt für Naturschutz                                 |
| BKG    | Bundesamt für Kartographie und Geodäsie                   |
| CSS    | Cascading Style Sheets                                    |
| DBMS   | Datenbankmanagementsystem                                 |
| ESRI   | Environmental Systems Research Institute                  |
| EU     | Europäische Union   |
| GIF    | Graphics Interchange Format                               |
| GIS    | Geoinformationssystem                                     |
| GML    | Geography Markup Language                                 |
| GPL    | GNU General Public License                                |
| GUI    | Graphical User Interface                                  |
| GWT    | Google Web Toolkit  |
| HCI    | Mensch Computer Interaktion                               |
| HTML   | Hypertext Markup Language                                 |
| HTTP   | Hypertext Transfer Protocol                               |
| IDE    | Integrated Development Environment                        |
| IÖR    | Leibniz-Institut für ökologische Raumentwicklung          |
| JPG    | Joint Photographic Experts Group                          |
| JS     | JavaScript  |
| JSON   | JavaScript Object Notation                                |
| KML    | Keyhole Markup Language                                   |
| LBM-DE | Landbedeckungsmodell für Deutschland                      |
| OGC    | Open Geospatial Consortium                                |
| PHP    | PHP Hypertext Preprocessor                                |
| REST   | Representational State Transfer                           |
| RIA    | Rich Internet Application                                 |
| SQL    | Structured Query Language                                 |
| SVG    | Scalable Vector Graphic                                   |
| UI     | User Interface  |
| URL    | Uniform Resource Locator                                  |
| WCS    | Web Coverage Service                                      |
| WebGIS | Webbasiertes Geografisches Informationssystem             |
| WFS    | Web Feature Service                                       |
| WMS    | Web Map Service   |
| WPS    | Web Processing Service                                    |
| WWW    | World Wide Web  |
| XML    | Extensible Markup Language                                |

## 1. Einleitung

### 1.1. Motivation

In Deutschland werden stetig neue Flächen für Arbeiten, Wohnen und Mobilität verbraucht. Täglich werden über 66 Hektar Land neu erschlossen, versiegelt und verbaut [Wilke, 2017]. Längst ist die Fläche ein knappes Gut, um dessen Nutzung unterschiedlichste Interessen miteinander konkurrieren. Um einen Ausgleich der Nutzungsansprüche zu gewährleisten, muss die Flächennutzung überwacht und geplant werden, damit eine nachhaltige Entwicklung ermöglicht wird. Häufig passiert dies über die amtliche Statistik. Allerdings sind bloße Zahlen mit schwachem räumlichem Bezug oft nicht aussagekräftig. Der Raumbezug spielt beim Flächennmonitoring eine entscheidende Rolle, da er Aussagen über räumliche Verteilungen und Tendenzen zulässt.

Mit dem Monitor der Siedlungs- und Freiraumentwicklung (IÖR-Monitor) entwickelt das Leibniz-Institut für ökologische Raumentwicklung Indikatoren, mithilfe derer die Nutzung sowie der Wandel der Flächennutzung beschrieben und analysiert werden können. Die Ergebnisse dieses Prozesses werden durch den IÖR-Monitor anhand thematischer Karten visualisiert und im Internet für jedermann verfügbar gemacht. Der Nutzer kann durch Interaktionen mit der Anwendung Karten erstellen, welche eine schnelle Gewinnung von Informationen zur Flächennutzung und -entwicklung ermöglichen. Die zur Verfügung gestellten Informationen dienen als Grundlage für Bewertungen und sind insbesondere für die Flächenhaushaltspolitik und die Raum- und Fachplanungen auf allen Ebenen von Bedeutung.

Im Zuge des Web 2.0 veränderte sich die Nutzung und Wahrnehmung des Internets. Durch eine rasante Verbreitung von mobilen Endgeräten und dem stetigen Entwicklungsprozess neuer Webtechnologien, können Internetanwendungen entstehen, welche der Usability von Desktop-Anwendungen sehr nahe kommen und eine Geräteunabhängigkeit realisieren.

Die aktuelle Implementierungsgrundlage des IÖR-Monitors wird dieser neuen Entwicklung und den damit verbundenen Anforderungen nicht mehr gerecht, womit Chancen und Möglichkeiten einer umfassenderen, interaktiven Visualisierung der Monitoringergebnisse über das WWW nicht genutzt werden. Mit der Reimplementierung durch modernste Webtechnologien kann der Grat möglicher Interaktionen, Darstellungen und Analysemöglichkeiten auf ein neues Level gehoben werden. Dies erlaubt eine Ergebnisvisualisierung in einer neuen Art und Weise. Wodurch Synergieeffekte auftreten, welche den Nutzerkreis erweitern, ohne dass besondere Technik oder Software durch den Nutzer vorausgesetzt werden.

### 1.2. Ziel der Arbeit

Basierend auf modernen Webtechnologien soll in dieser Arbeit eine Anwendung entstehen, welche den Anforderungen des Web 2.0 gerecht wird und durch eine Verbesserung der interaktiven Möglichkeiten zu einem verbesserten Verständnis der Ergebnisse des Flächenmonitoring führt. Mit der Anwendung soll es dem Nutzer möglich sein, auch auf einem mobilen Endgerät Interaktionen ohne Einschränkungen durchzuführen. Um dieses Ziel zu erreichen muss das

Layout und der technische Hintergrund grundlegend überarbeitet werden. Der IÖR-Monitor in seiner jetzigen Form basiert auf zwei unabhängig voneinander existierenden Kartenviewern, welche den Nutzer unterschiedliche Darstellungsformen der Indikatoren bieten. Hier liegt der Hauptschwerpunkt dieser Arbeit, einer Zusammenführung und ein Ausbau der interaktiven Möglichkeiten der zwei Kartenviewer, unter dem Dach einer Anwendung.

### **1.3. Aufbau der Arbeit**

Die vorliegende Arbeit gliedert sich in einen theoretischen Teil und der Beschreibung der praktischen Umsetzung, mit anschließender Evaluierung der Ergebnissen.

Im zweiten Kapitel wird der IÖR-Monitor kurz vorgestellt und die Wichtigkeit eines flächendeckenden Monitorings der Siedlungs- und Freiraumfläche erörtert. In den zwei darauf folgenden Kapiteln erfolgt eine Einführung in die theoretischen Grundlagen, welche für das Verständnis notwendig sind. Im fünften Kapitel wird eine Anforderungsanalyse durchgeführt und die wichtigsten Funktionalitäten der neuen Anwendung herausgearbeitet. Um dieses Ziel zu erreichen werden unterschiedliche Untersuchungen durchgeführt, welche die Filterung von möglichen Entwicklungspotentialen unterstützen.

Aufbauend auf diesen Erkenntnissen beginnt die anschließende Konzeptionierungs- und Implementierungsphase. In diesen Kapiteln entsteht das konkrete Konzept der neuen Anwendung mit einer anschließenden Realisierung.

Die abschließende Evaluierung bewertet die Ergebnisse der Implementierung und versucht mögliche Schwachstellen zu identifizieren. Mit einem Performance- und Nutzertest fand diese Validierung statt. Kapitel zehn fasst die Ergebnisse zusammen und schließt mit einem Ausblick ab.

## 2. Flächenmonitoring

### 2.1. Definition

Der Begriff Monitoring (von engl. to monitor = überwachen, kontrollieren) ist ein weitreichender Begriff, welcher in verschiedenen Bereichen eingesetzt wird. Beispiele finden sich in Naturwissenschaft, Politik, Medien oder der Informatik. Eine mögliche Definition nach Birkmann [2005] ist:

*„Monitoring ist eine laufende Beobachtung, Überwachung und Kontrolle von Vorgängen oder Prozessen innerhalb eines Systems mit einem kontinuierlichen Sammeln, Auswerten, Interpretieren und Bereitstellen von relevanten Daten, Indikatoren oder Ereignissen als Grundlage für Entscheidungen.“*

Als System wird innerhalb des Flächenmonitorings die quantitative und qualitative Entwicklung der Flächenbedeckung und Flächennutzung gesehen. Um den Begriff der Flächennutzung von der Flächenbedeckung zu differenzieren, findet sich im Glossar des IÖR-Monitors folgende Definition:

*„Unter Flächennutzung bzw. Landnutzung (land use) ist die Art der Inanspruchnahme von Teilen der festen Erdoberfläche durch den Menschen zu verstehen. Im Unterschied zur Bodenbedeckung (land cover) spielt hierbei der Nutzungsaspekt die entscheidende Rolle. Der Begriff Flächennutzung fokussiert stärker auf die bauliche Prägung, der Begriff Landnutzung mehr auf das unbebaute Gelände. Der verfügbare Raum kann gleichzeitig mehreren Nutzungsansprüchen unterliegen. Hauptflächennutzungsarten sind Siedlungs- und Verkehrsfläche, Landwirtschaftsfläche, Waldfläche und Gewässerfläche, die jeweils weiter untergliedert werden können.“* [aus IÖR-Monitor, 2014]

Neben der Flächennutzung spielt auch die Flächenbedeckung eine wichtige Rolle. Dieser Begriff steht für die physische und biologische Bedeckung der Erdoberfläche, einschließlich künstlicher Flächen, landwirtschaftlicher Flächen, Wäldern, natürlicher Gebiete und Wasserkörpern [INSPIRE, 2017]. Die Ergebnisse des Flächenmonitoring werden je nach Fragestellung zu verdichteten Informationen in Form von Indikatoren aufbereitet. Hierbei stellen sie die „*Messgrößen zur Bewertung eines Ist-Zustandes und Grundlage für die Verfolgung und Kontrolle von Entwicklungen und Veränderungen*“ [Kriese u. Bohnsack, 2006] dar.

### 2.2. Bedeutung und Ziel

Auch in der Bundesrepublik Deutschland stellt der Boden ein knappes Gut dar, wodurch zunehmend Flächenkonkurrenzen entstehen. Angesichts dieser Problematik und ambitionierter Flächensparziele in Politik, Raumplanung und Umweltschutz gewinnt das Flächenmonitoring zunehmend an Bedeutung [Meinel, 2013].

In Anbetracht der demographischen Entwicklung der Bevölkerung strebt die Bundesregierung im Rahmen der Nationalen Nachhaltigkeitsstrategie an, bis zum Jahr 2020 die Neuinanspruchnahme von Flächen auf 30 Hektar pro Tag zu senken [vgl. BMUB, 2017]. Um dieses

Ziel zu erreichen wurde das „*30-ha-Ziel*“ formuliert, mit dessen Hilfe die auf Dauer nicht vertretbare Neuinanspruchnahme und Zerstörung von fruchtbaren Böden reduziert werden soll. Um Rücksicht auf künftige Generationen und die wachsende Weltbevölkerung zu nehmen, ist der anhaltende Flächenverbrauch mit seinen negativen Folgen unverantwortlich [UBA, 2014]. Das Jahr 2015 wurde von der UNO zum internationalen Jahr des Bodens erklärt, was zeigt wie ernst die Thematik von den Entscheidungsträgern der Politik genommen wird.

Als Ziel wird grundsätzlich eine räumlich hochauflösende Ermittlung der realen Flächenbedeckung und Flächennutzung angestrebt. Durch die Berechnung geeigneter Indikatoren kann ein Nachweis dauerhafter (auch lokaler) Veränderungen der Flächennutzung und -bedeckung erreicht werden [vgl. Meinel u. Schumacher, 2009, S.54].

Ein Flächenmonitoring stellt, wie jedes Abbild der realen Welt, auch eine Rückschau dar. Basierend auf diesem Fakt ist eine hohe Aktualität der verwendeten Geodaten wichtig, da die berechneten Indikatoren als Entscheidungsgrundlage für Behörden und Politik dienen oder aber auch zur Information der Öffentlichkeit [BfN, 2010; Meinel, 2013].

Ein wichtiger Punkt stellt eine genaue Visualisierung der Indikatoren anhand von thematischen Karten dar. Mit deren Hilfe Aussagen über die Flächennutzung, über die Änderungen der Nutzungen und über den Zustand der Umwelt ermöglicht werden. Diese bilden die Grundlage für die umweltverträgliche, ressourcenschonende und nachhaltige Nutzung der Erdoberfläche [Kestler, 2017].

### **2.3. Monitor der Siedlungs- und Freiraumentwicklung (IÖR-Monitor)**

Die Umsetzung der nationalen Nachhaltigkeitsstrategie rückt das Thema Fläche in den Vordergrund. Um einen Beitrag über die notwendigen Anstrengungen für eine nachhaltigen Flächennutzung zu geben, hat das *Leibniz-Institut für ökologische Raumentwicklung Dresden* den „Monitor der Siedlungs- und Freiraumentwicklung“ ins Leben gerufen [Meinel u. a., 2014]. Mit einer fast flächendeckenden Erfassung von Geobasisdaten in Deutschland, ist eine geeignete Grundlage für die Flächenauswertung verfügbar. Diese Daten können mit statistischen oder anderen Fachdaten verknüpft werden. Um die Datenmenge der flächendeckenden Daten zu bewältigen, müssen leistungsfähige Verfahren der Geoinformatik eingesetzt werden. Der Fokus der Verarbeitung liegt dabei auf der automatisierten Aufbereitung, Auswertung und Berechnung von Daten zur Ableitung von Indikatoren für die Messung der Flächennutzung und Flächenbedeckung [vgl. Meinel u. Siedentop, 2007]. Die Umsetzung erfolgte unter Berücksichtigung der europäischen INSPIRE-Richtlinie. Als flächendeckend verfügbare Eingangsdaten dienen dem IÖR-Monitor das ATKIS Basis-DLM, das Landbedeckungsmodell für Deutschland (LBM-DE), die DTK25 und amtliche statistische Daten. Nach der Datenaufbereitung, erfolgt eine Indikatorberechnung, Speicherung und Analyse der Ergebnisse. Als finales Ergebnis werden die berechneten Indikatoren in der Anwendung präsentiert und bereitgestellt [IÖR, 2008].

Der IÖR-Monitor findet national eine große Beachtung und wurde im 3. Geo-Fortschrittsbericht der Bundesregierung als „exzellentes Beispiel für Open Government“ beschrieben [vgl. Bundesministerium des Innern, 2012, S.58-59].

### 3. WebMapping

Die vorrangige Aufgabe des WebMapping stellt die Visualisierung von Geoinformationen im Internet dar. Durch den Einsatz einer Anwendung mit dieser Basis können die Ergebnisse des Flächenmonitoring einer breiten Öffentlichkeit zugänglich gemacht werden. Dieses Kapitel soll die Grundlagen des WebMapping vermitteln und am Ende einen Einblick in die Visualisierung der Indikatoren für die Flächennutzung und -bedeckung geben, welche durch das Leibniz-Institut für ökologische Raumentwicklung mit dem IÖR-Monitor realisiert werden.

#### 3.1. Begriffe

Für die elektronische Kartendarstellung im Internet gibt es mittlerweile zahlreiche Bezeichnungen. Frank Dickmann bildete hier eine Unterteilung in zwei Begriffe: WebMapping und WebGIS.

WebMapping leitet sich aus dem Begriff „*Web-Map*“ ab und bezieht sich stärker auf den Herstellungsprozess der Karten. Er umfasst neben der Visualisierung der Karte im Internet auch Manipulationen der Ansicht. Beispielhafte Interaktionen hierfür sind das Zoomen, Verschieben sowie das Ein- und Ausblenden einzelner Ebenen (Layer). Diese Funktionen gelten in der Regel noch nicht als »GIS adäquat« [vgl. Dickmann, 2004, S.24].

Auf der anderen Seite steht das WebGIS. Dieser Begriff wird verwendet, wenn der Nutzer über die WebMapping-Funktionalitäten hinaus Sachdaten selbständig ändern und weiterführende analyseorientierte GIS-Operationen durchführen kann. Dies bedeutet vor allem, dass der Nutzer in die Lage versetzt wird, einen direkten Zugriff auf raumbezogene Daten zu bekommen, mit der Freiheit anhand dieser eine Informationsaufbereitung durchzuführen [Storch, 1999]. Im Zuge einer rasanten Entwicklung von Webtechnologien, findet zunehmend eine Verschmelzung der beiden Begrifflichkeiten statt. Durch die Ausstattung von Bildschirmkarten mit eigenen Funktionalitäten, welche auf dem Wachstum neuer Technologien fußen wird die Trennlinie immer diffuser und eine klare Abgrenzung ist schwer möglich. Beispielsweise ermöglicht AJAX (Kapitel 4.2) eine Sachdatenabfrage innerhalb von Webkarten, was früher den GIS Funktionalitäten zugeschrieben wurde.

Eine einheitliche und klare Abgrenzung zwischen WebMapping und WebGIS existiert nicht. Beispielsweise werden vielfach grundlegende GIS-Analyse-Funktionalitäten, wie die Sachdatenabfrage, ebenfalls dem WebMapping zugeordnet [vgl. Dickmann, 2004, S.24]. Aufgrund der schwierigen begrifflichen Unterscheidung nehmen KORDUAN und ZEHNER keine Einteilung vor, sondern führen die allgemeine Bezeichnung „*Internet-GIS*“ ein, der alle Anwendungen mit Internettechnologien und räumlichen Informationen einschließt [Korduan u. Zehner, 2008].

In dieser Arbeit wird der Begriff *WebMapping* im weiteren Sinne nach Frank Dickmann verwendet und schließt dabei analytische GIS-Funktionalitäten mit ein.

### 3.2. Karten im Internet

Ein sehr effektiver Weg, um eine Information in Form einer Karte zu visualisieren, ist es sie über eine Website zur Verfügung zu stellen. WebMapping Anwendungen lassen sich nach dem Interaktionsgrad in *statische* und *interaktive* Karten unterteilen [Mitchell, 2005].

Bei den sogenannten *static maps* oder *view only maps* handelt es sich um eine reine Kartenansichten, welche vorgefertigt wurden und sich unmittelbar als einzelne Grafiken in Webseiten integrieren lassen. Mit Beginn der WebMapping-Entwicklung wurden Papierkarten gescannt und lediglich in Form von Bitmaps als *statische* Karten im Internet visualisiert. Es entstanden umfangreiche Online-Sammlungen von statischen Karten. Durch das Fehlen von Interaktionen werden nur geringe Erfahrungen im Web Design benötigt um diese in einer Website aufzunehmen, da eine einfache Integration als Bilddatei erfolgt [Mitchell, 2005; Dickmann, 2004].

Durch die *interaktiven* Karten bekommt der Nutzer die Möglichkeit, mit der Karte zu interagieren und diese an seine Bedürfnisse anzupassen oder weiterführende Informationen zu erhalten. Beispielsweise kann der Nutzer die Karte verschieben, hineinzoomen, weitere Kartenebenen aktivieren oder eine Abfrage zu einem Kartenobjekt stellen. Der Kartenausschnitt wird dazu aktualisiert und neue Informationen werden dargestellt [Dickmann, 2004].

Eine klare Grenze zwischen *statischen* und *interaktiven* Kartendarstellungen ist nicht leicht zu definieren. Interagiert der Nutzer mit der Karte, indem er sie verschiebt (*panning*) bzw. die Zoomstufe verändert (*zooming*) und dabei keine neuen Informationen dargestellt werden, handelt es sich immer noch um eine statische Karte [Dickmann, 2004]. Mit diesen Interaktionen wird dem Nutzer kein »Mehrwert« geliefert, da beispielsweise durch das *panning* die Kartenansicht verschoben wird und keine neuen Informationen hinzugefügt werden.

Mitchell [2005] fasst die interaktiven Karten unter dem Begriff WebMaps zusammen, welcher auch in dieser Arbeit eingesetzt wird. Der Aufbau von WebMaps folgt dem Ebenen- bzw. Layerprinzip. Dieses bildet einzelne Darstellungsschichten ab [Koch, 2009].

### 3.3. Web 2.0 und WebMapping 2.0

Der Begriff des Web 2.0 geht aus einem Brainstorming zwischen *Tim O'Reilly* und *MediaLive International* hervor, um die Änderung der Nutzung des World Wide Web zu bezeichnen und wurde zum Synonym für die rasante Entwicklung des Internets [vgl. O'Reilly, 2017]. Zu Beginn der Internetentwicklung bestand die Aufgabe von Websites darin Informationen zu verbreiten, welche von einem Webmaster erstellt wurden. In der Generation 2.0 transformieren sie sich vielmehr zu einer Plattform, auf welche mit allen Endgeräten zugegriffen werden kann. Oftmals wird das Web 2.0 nur mit Asynchronous Java Script and XML (AJAX) in Verbindung gebracht, im schlimmsten Fall wird es allein darauf reduziert. Tatsächlich steckt hinter dem Web 2.0 vielmehr die Idee, das Verhalten der Nutzer zu ändern.

Der Nutzer wird nicht mehr als reiner Konsument von Daten, sondern als Teil einer Gemeinschaft angesehen, die Daten über das Internet austauscht. Das Web 2.0 definiert sich somit als

eine Ansammlung von Prinzipien und Praktiken. Folgend sind die elementaren Eigenschaften nach O'Reilly [2017] aufgeführt.

### **Das Web als Plattform**

Das Web wird im Web 2.0-Konzept als eine Plattform anstatt des lokalen Rechners dargestellt. Das bedeutet, dass auf die Anwendungen von jedem Endgerät das mit dem Internet verbunden ist und über einen Web Browser verfügt, zugegriffen werden kann. Hierdurch wird eine Installation der Anwendungen auf dem Endgerät unnötig [vgl. Alby, 2008, S.136].

### **Software-Lebenszyklen**

Da Software im Web 2.0 als Dienst und nicht als Produkt ausgeliefert wird, greifen die typischen Lebenszyklen von Softwareprodukten nicht. Web 2.0-Anwendungen werden nicht als Produkt ausgeliefert, sondern stellen vielmehr einen Dienst dar, welcher ständig weiterentwickelt wird. Sobald die Entwicklung neuer Funktionalität abgeschlossen ist, steht sie umgehend dem Nutzern zur Verfügung. Gleichwohl können Funktionen ohne Nutzerakzeptanz wieder entfernt werden, welche live darauf untersucht wurden [vgl. Karla, 2007, S.20].

### **Leichtgewichtige Programmiermodelle**

Ein weiteres Merkmal von Anwendungen dieser Generation ist die lose Kopplung verteilter Systeme, in welchem Daten möglichst einfach ausgetauscht werden können, indem leichtgewichtige Programmiermodelle wie Web-Services via Simple Object Access Protocol (SOAP) und Remote Procedure Calls (RPC), eingesetzt werden. Durch diese neue technische Grundlage können Dienste entstehen, die sich einfach aus bestehenden Diensten zusammenbauen lassen und deren Daten einen besonderen Wert haben [Volker u. a., 2007].

### **Geräteunabhängigkeit**

Das Web 2.0 besteht aus verschiedenen Diensten von völlig unterschiedlichen Rechnern. Solche Anwendungen sind nicht länger auf PC-Plattformen beschränkt. Dadurch besteht keine Notwendigkeit, ein spezielles Betriebssystem oder spezielle Software zur Nutzung von Diensten einzusetzen, die von Web 2.0-Anwendungen zur Verfügung gestellt werden.

### **Benutzerführung**

Durch die rasante technische Entwicklung wird die Benutzerführung in Anwendungen des Web 2.0 grundlegend verbessert. Durch AJAX (vgl. Kapitel 4.2) entstanden Benutzerschnittstellen, die sich in der Bedienung kaum von Desktop Anwendungen unterscheiden, jedoch in Web-Browsern ausgeführt werden können.

Gartner [2009] setzt den Begriff „*WebMapping 2.0*“ für Web 2.0-Anwendungen ein, welche einen räumlichen Bezug aufweisen. In der ersten Generation musste der Nutzer noch Kartenanwendungen als fertiges Produkte betrachten, wohingegen das Web 2.0 ihn zum „*Kartenproduzenten*“ erhebt.

Durch die konsequente Weiter- und Neuentwicklung von Technologien und damit immer besseren Interaktionsmöglichkeiten und Ansprechgeschwindigkeiten, stieg auch die Motivation von Entwicklern, geografische Informationen zu verwenden und damit zu experimentieren. Auch der Fakt, dass viele APIs frei nutzbar sind, beschleunigte den Übergang zum WebMapping 2.0 und ist einer der Hauptgründe für die konsequente Verwendung von sogenannten Open Source-Technologien in Web-2.0-Anwendungen [Eszter u. a., 2010].

Da Web 2.0 und die meisten Anwendungen auf Open Source-Software basieren und durch offene Schnittstellen (APIs) definiert sind, können die Angebote und Daten miteinander mittels so genannter „*mashups*“ kombiniert werden.

„*Mashup*“ steht dabei für eine Web-Anwendung, die Daten aus mehr als einer Datenquelle in einer integrierten Anwendung kombiniert [Maness, 2006]. Anwendungen dieses Typs besitzen drei architektonische Teile: den Datenanbieter, die Webapplikation und den Web Browser des Nutzers. Beispielsweise agiert Google als Datenanbieter für viele Google Map „*mashups*“. Dabei stellt Google referenzierte Karten und Satellitenbilder, eine standardisierte Nutzerschnittstelle sowie ein API zur Verfügung [vgl. Gibson u. Earle, 2006]. Web-Anwendungen verwenden die Google Map API, um Daten mit den Google-Karten zu verlinken, eigene Schnittstellen zu entwickeln oder anderen Nutzern zu ermöglichen, die verbundenen Daten zu verwenden [vgl. Gartner, 2009, S.282]. Dies soll beispielhaft die Möglichkeiten von Dienstkombinationen in einer Anwendung aufzeigen, welche durch den Einsatz leichtgewichtiger Programmiermodelle erreicht wird.

Zusammenfassend steht WebMapping der Generation 2.0 nicht zwingend für die Entwicklung grundsätzlich neuer Funktionalitäten zur kartografischen Informationsvermittlung, sondern vielmehr für dynamische und interaktive Benutzeroberflächen, welche einfach und intuitiv zu bedienenden sind [vgl. Haklay, 2010]. In der folgenden Tabelle 1 werden die wichtigsten Unterschiede zwischen den Generationen festgehalten.

| <b>WebMapping 1.0</b>                      | <b>WebMapping 2.0</b>  |
|--|--|
| herstellerzentriert                        | nutzerzentriert  |
| nur Lesen                                  | Lesen und Erstellen  |
| statische Karten                           | interaktive Karten   |
| vordefinierte Inhalte und Kartengestaltung | personalisierte Karten (eigene Inhalte und Kartengestaltung) |
| →Nutzer als aktiver Konsument              | →Nutzer als aktiver Produzent und Konsument                  |

Tabelle 1: Vergleich von WebMapping 1.0 und WebMapping 2.0-Anwendungen [abgeändert nach Maguire, 2007]

Durch die unaufhaltsame Entwicklung von Technologien ist das Web 2.0 genau genommen schon am Ende seiner zeitlichen Definition angekommen und das WebMapping steht kurz vor der Ära 3.0. Ein wichtiger Grund dieser Entwicklung ist die zunehmende digitale Mobilisierung der Gesellschaft. Im Jahr 2017 sind in Deutschland 20,2 Millionen Smartphones verkauft worden [vgl. statista, 2017a] und die Verkaufszahlen von PCs sind seit 2011 auf dem Rückmarsch, die Absatzzahlen von Tablets steigen seit 2010 stetig an. Aus diesem Grund werden zunehmend bestehende Web 2.0-Angebote für die Nutzung auf mobilen Endgeräten

erweitert. Mit der zunehmenden Bedeutung von Datensammeldiensten mit einer zentralen Vernetzung wird das Web der Zukunft auch das semantische Web genannt. Als Synergieeffekt entsteht eine Flut von Daten (Big Data), welche die zielgerichtete Suche und Verwendung von Informationen erschwert. Um dies zu verhindern muss eine logische und korrekte Verknüpfung erfolgen, um so richtige Schlüsse ziehen zu können. [vgl. Stuhec-Meglic, 2016]

### 3.4. Client-/Server-Modell

Die Grundlage von Web-Anwendungen bildet das Client-Server-Prinzip, welches als Standardkonzept für die Verteilung von Aufgaben innerhalb eines Netzwerkes gesehen werden kann. Ein Programm (Browser) fordert einen speziellen Dienst von einem anderen Programm (auf einem Server) an. Das angesprochene Programm auf dem Server bearbeitet die Anfrage und sendet, falls notwendig, ein Ergebnis zurück an den Clienten. Die Kommunikation geht dabei immer vom Clienten (Browser) aus, der eine Anfrage (request) an den Server stellt, während der Server passiv auf Anfragen wartet, diese bearbeitet und eine Antwort (reply) zurücksendet [Chantelau u. Brothuhn, 2010]. Abbildung 1 dokumentiert das Modell-Prinzip für das WebMapping .

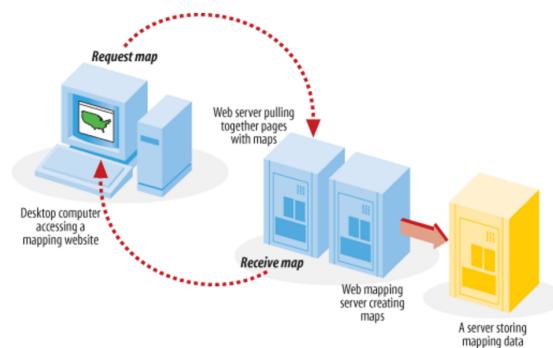


Abbildung 1: Client-Server-Modell, der Server wartet passiv auf Anfragen und sendet daraufhin Antworten zurück. Quelle: [Mitchell, 2005]

Hier fordert der Client eine Karte vom Server an. Der Server leitet die Kartenanfrage an den Mapserver weiter. Dieser wertet die Anfrage aus, greift auf die benötigten Geodaten zurück und generiert daraus eine entsprechende Grafik. Die fertige Karte wird nun an den Webserver gesendet, wo sie in eine HTML-Seite integriert und zurück an den Client-Browser geschickt wird. [Klauer, 2002; Mitchell, 2005]

#### 3.4.1. Client

Um eine WebMapping-Anwendung zu nutzen wird clientseitig lediglich eine Webbrowservor ausgesetzt. Dieser Browser stellt das Anwendungsprogramm dar, mit welchen die implementierten Funktionalitäten der Website genutzt werden können, indem über das HTTP-Protokoll die Daten abgefragt und dargestellt werden [Mitchell, 2005]. Je nachdem ob eine statische oder interaktive Karte implementiert ist, muss der Client entsprechend viel Arbeit verrichten. Handelt es sich um eine statische Karte, liegt die Aufgabe des Clienten (Browser) darin, eine Karte auf dem Bildschirm anzuzeigen. In der Informatik wird hier von einem „Thin-Client“

gesprochen, da die Darstellung nicht viele Ressourcen benötigt. Ist der Hauptteil der Anwendungslogik clientenseitig implementiert wird in der Fachsprache auch von einem „*Fat-Client*“ gesprochen. Dieser „*schwergewichtige Client*“ hat den großen Nachteil, das auf der Clientenseite Anwendunglogik installiert sein muss, was häufig bei Webbrowsern nicht der Fall ist. Gerade bei umfangreichen interaktiven WebMapping-Anwendungen, wird ein „*Smart-Client*“ verwendet. Bei diesem Clienten werden die benötigten Programmteile vor oder während der Benutzung der Anwendung über das Netzwerk automatisch auf den Client übertragen und installiert. Dadurch werden Updates oder Installationen von benötigten Anwendungsteilen vom Benutzer unbemerkt im Hintergrund durchgeführt. Als wichtiger Vertreter ist hier AJAX (siehe Kapitel 4.2) zu nennen, wobei das benötigte AJAX Framework erst bei Benutzung der Anwendung auf den Clienten übertragen wird [Chantelau u. Brothuhn, 2010]. Durch die Verwendung von „*Smart-Clienten*“ in der Verbindung mit AJAX, ist auch von „*Rich Internet Applications*“ die Rede [Mintert, 2009].

### 3.4.2. Server

Auf der Serverseite werden die Anfragen des Clienten angenommen und abgearbeitet, das Ergebnis wird an den jeweiligen Clienten zurückgesendet. Neben einem Webserver (z.B. Apache<sup>1</sup>), ist ein Kartenserver häufiger Bestandteil einer WebMapping-Anwendung. Beispiele für einen Open Source Kartenserver sind der *MapServer*<sup>2</sup> und *GeoServer*<sup>3</sup>. Diese bieten verschiedene standardisierte Webdienste des Open Geospatial Consortium (OGC) zum Austausch von Geodaten an. Ein sehr verbreiteter und in vielen WebMapping-Anwendungen verwendeter Dienst stellt der *WMS* dar. Ein *Web Map Service* (WMS) beschreibt eine HTTP-Schnittstelle, über die georeferenzierte Karten bereit gestellt werden können. Ein *WMS* bietet dabei unterschiedliche Operationen an. Mit der *GetCapabilities*-Anfrage werden Metadaten eines Dienstes angefordert, die dessen Inhalte und akzeptierte Anfrage-Parameter beschreiben, wie bspw. Informationen zu verfügbaren Layern, Projektionssystemen und dem Koordinatenausschnitt [vgl. Mitchell, 2005, S.240],[vgl. Korduan u. Zehner, 2008, S.179]. Als Antwort auf eine *GetCapabilities*-Anfrage wird ein XML-Dokument zurückgeliefert. Eine weitere wichtige Anfrage stellt die eigentliche Kartenanfrage dar. Diese wird über eine *GetMap*-Anfrage realisiert. Anhand der übergebenen Parameter liefert der Kartenserver dynamisch die gewünschte Kartengrafik. In der folgenden Tabelle sind die wichtigsten Parameter *GetMap*-Operation abgebildet. Diese Parameter werden heute meist über *AJAX* an den Server gesendet.

---

<sup>1</sup><https://httpd.apache.org/>

<sup>2</sup><http://mapserver.org/>

<sup>3</sup><http://geoserver.org>

| Parameter         | Beschreibung   |
|-------------------|--|
| SERVICE=WMS       | Art des Dienstes   |
| REQUEST=GetMap    | Karte anfordern  |
| VERSION=1.3.0     | Angabe der WMS-Version, die verwendet werden soll  |
| SRS=EPSG:4326     | Projektion des Kartenbildes über Angabe eines EPSG-Codes   |
| FORMAT=image/jpeg | Ausgabeformat  |
| LAYERS=S12RG      | Layernamen, die über eine kommasseparierte Liste angegeben werden  |
| WIDTH=800         | Breite des Kartenbildes in px  |
| HEIGHT=600        | Höhe des Kartenbildes in px  |
| BBOX=10,0,50,40   | Bounding Box-Koordinaten (links, unten, rechts, oben) in der Einheit des Referenzsystems   |
| STYLE=            | Falls Styles unterstützt werden, können die Layer in unterschiedlicher Darstellung angefordert werden. Ohne Angabe wird der Default-Style verwendet. |

Tabelle 2: Benötigte WMS-Parameter eines GetMap-Requests [Mitchell, 2005, S.239-240], leicht modifiziert.

Mit der *GetFeatureInfo*-Operation können Sachinformationen zu einzelnen Geometrien einer Karte angefragt werden [vgl. Mitchell, 2005, S.239],[vgl. Korduan u. Zehner, 2008, S.179]. Weitere verbreitete OGC-Dienste sind z.B. der Web Feature Service (WFS) und Web Coverage Service (WCS).

### 3.5. WebMapping am Beispiel des IÖR-Monitors

Die Konzeptionierung und Implementierung eines neuen Kartenviewers ist der Hauptbestandteil dieser Arbeit, welche auf die aktuelle Grundlage des IÖR-Monitors aufbaut. In diesem Kapitel soll die Anwendung vorgestellt werden.

Der Monitor der Siedlungs- und Freiraumentwicklung ging am 17.06.2010 unter der URL [www.ioer-monitor.de](http://www.ioer-monitor.de) ans Netz. Er visualisiert anhand von digitalen Karten- und Tabellen-diensten die am *Leibniz-Institut für ökologische Raumentwicklung* berechneten und aufbereiteten Indikatoren für das Flächenmonitoring. Zur Visualisierung der Indikatorwerte wurden zwei Viewer entwickelt, ein Vektor- und Rasterviewer. Neben der Kartendarstellung kann sich der Nutzer weitergehende Informationen und Erklärungen auch in Form von Kennblättern, statistischen Auswertungen, Glossars usw. anzeigen lassen. In Abbildung 2 sind die beiden entwickelten Oberflächen abgebildet, welche zum Zeitpunkt dieser Arbeit für den Nutzer verfügbar waren.

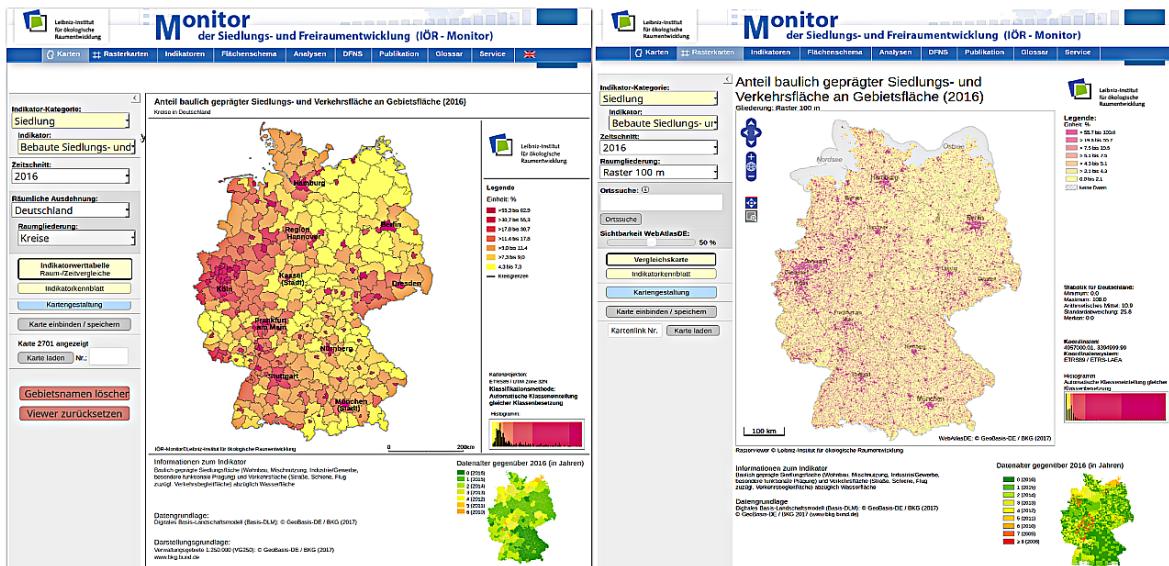


Abbildung 2: Die beiden Oberflächen des Monitors der Siedlungs- und Freiraumentwicklung (von links: Vektor- und Rasterviewer). Quelle: [Leibniz-Institut für ökologische Raumentwicklung, 2017a,b]

### 3.5.1. Vektorviewer

Eine Umfangreiche Dokumentierung der konzipierten und umgesetzten Anwendung wurde in Meinel [2009] durchgeführt. In diesem Kapitel sollen nur die wichtigsten Eigenschaften genannt werden.

Der Vektor- wurde als Übersichts-Viewer entwickelt, welcher einen schnellen Überblick zur räumlichen Verteilung eines Indikators in Deutschland bietet. Nach der entsprechenden Selektion des Zeitschnitts, der gewünschten räumlichen Ausdehnung und der Raumgliederung visualisiert er die Darstellung der Indikatorwerte in einem festen, der jeweiligen räumlichen Ausdehnung angepassten Maßstab. Hierbei können die Indikatoren bis hin zur Stadtteilebene abgefragt werden. Der Übersichtsviewer basiert auf der Scalable Vector Graphic (SVG)-Technologie. SVG ist der empfohlene Standard des W3C<sup>4</sup> zur Beschreibung von 2D Vektorgraphiken im XML-Format. Standard-Interaktionen mit der Karte, wie das Zooming und Panning werden in diesem Viewer nicht unterstützt. Die Kartengenerierung in Form einer Choroplethenkarte erfolgt mittels der Programmiersprache PHP und anhand einer MySQL- und PostgreSQL-Datenbank. Ergänzend zur Kartenansicht stellt der Vektorviewer einen Tabellendienst zur Verfügung. Dieser gibt eine komplette Übersicht der ausgewählten Indikatoren in der jeweiligen Gebietseinheit. Er ermöglicht den Vergleich zwischen zwei Indikatoren oder verschiedenen Zeitschnitten sowie die Darstellung eines Entwicklungsgraphen.

### 3.5.2. Rasterviewer

Mit ihrer Masterarbeit realisierte Christiane Köhler [2012] den ersten interaktiven Rasterviewer, welcher sich stark an der Designvorlage des Vektorviewers orientierte.

Der Rasterviewer dient innerhalb des IÖR-Monitors als Detailviewer. Anders als der Vektorviewer findet hier keine räumliche Unterteilung in administrative Einheiten statt, sondern

<sup>4</sup><http://www.w3.org/>

in Rasterzellen. Durch diese Darstellungsart wird die Bereitstellung von kleinräumigen Informationen unterhalb der Gemeindeebene ermöglicht. Zur Visualisierung der Rasterkarten werden die Rasterweiten: 100m, 200m, 500m, 1000m, 5000m und 10000m angeboten. Um die Rasterkarten darzustellen wird die Grafik durch den *MapServer* mit Datenbankanbindung generiert und an den Clienten gesendet. Als Technologie zur Visualisierung der Rasterkarten wird die JavaScript-Bibliothek *OpenLayers* eingesetzt. Mit Hilfe dieser Bibliothek konnte eine interaktive Gestaltung des Viewers erfolgen. Das Verständnis der Indikatorwerte wird durch die optionale Unterlegung des topographischen BKG-Kartendienstes *WebAtlasDE*<sup>5</sup> unterstützt. Der Detailviewer stellt einen wichtigen Baustein dar und erweitert das Angebot für die Datenexploration durch den Nutzer.

In der folgenden Tabelle 3 werden die beiden implementierten Viewer gegenübergestellt.

|   | <b>Vektorviewer</b>   | <b>Rasterviewer</b>                                   |
|---|---|---|
| Verwendung                                  | Übersichtsinformationen                                     | Detailinformationen                                   |
| Gebietsauwahl                               | statistische Auswahl<br>vorgegebener Gebietseinheiten       | frei wählbarer Ort über<br>Suchfeld                   |
| Räumliche Gliederung                        | Bund, Länder,<br>Raumordnungsregionen, Kreise,<br>Gemeinden | Raster (100, 250, 500, 1000,<br>5000, 10000 m)        |
| Darstellungsmöglichkeiten und<br>Funktionen | Choroplethenkarte, statistische<br>Parameter, Tabelle       | Choroplethenkarte,<br>WMS-Dienste, Pan, Zoom,<br>Info |
| Technische Grundlagen                       | SVG und XML   | Kartenserver: MapServer,<br>Client: Open-Layers       |
| Datenbank                                   | Indikatoren: MySQL, Gebietsgeometrien: PostgreSQL           |   |

Tabelle 3: Vergleich der beiden Kartenviewer des IÖR-Monitors [modifiziert Förster, 2011, S.120]

Neben den Viewern können ausgewählte Indikatoren anhand von OGC-Schnittstellen in das eigene Desktop-GIS eingebunden werden. Für die administrativen Gebietseinheiten stehen WFS- und auf der Rasterseite WCS- und WMS- Dienste bereit. Die Visualisierung der Indikatoren wird durch umfangreiches Informationsmaterial ergänzt. Für jeden Indikator wird die Berechnungsmethodik und die Bedeutung in einem Kennblatt dargestellt. Wiederkehrende, zentrale Begriffe werden in einem Glossar erklärt.

<sup>5</sup>[www.geodatenzentrum.de/geodaten/gdz?l=webatlasde](http://www.geodatenzentrum.de/geodaten/gdz?l=webatlasde)

## 4. Technologien des WebMapping 2.0

In diesem Kapitel werden die wichtigsten Technologien vorgestellt, mit deren Hilfe sich eine Anwendung erstellen lässt, welche den Anforderungen des Web 2.0 gerecht wird.

### 4.1. Open Source Software

Nach Gartner [2009] zeichnen sich WebMapping-Anwendung der Generation 2.0 durch den konsequenten Einsatz von Open Source Technologien aus. Mit diesem Kapitel sollen die Grundlagen dieser Softwarerichtung gelegt werden, welche zum weiteren Verständnis dieser Arbeit notwendig sind.

Der Begriff bedeutet vor allem, dass der Entwickler, welcher Software unter dieser Bezeichnung anbietet, nicht nur Einblicke in den Quellcode, sondern ebenso Veränderungen daran erlaubt [Haar u. Keck, 2003]. Die Open Source Initiative (OSI) hat diesen Begriff definiert und setzt sich dafür ein, Wissen aufzubauen und das Bewusstsein für nicht-proprietäre Software zu fördern. Zehn Kriterien muss eine Lizenz einhalten, um als Open Source zu gelten. Die wichtigsten sollen folgende genannt werden.

Neben dem Offenlegen des Quellcodes, darf die Software uneingeschränkt weitergegeben (verkauft oder verschenkt) werden, jedoch ist die Forderung nach Lizenzgebühren untersagt. Um seine Software mit dieser Bezeichnung zu führen, darf der Quellcode nicht verschleiert werden, d.h. dieser muss in seiner ursprünglichen Form einzusehen sein. Wird ein Produkt, welches anhand dieser Software implementiert wurde, weitergegeben muss dieses die gleichen Lizenzbedingungen wie das Original aufweisen. Verändert der Entwickler während des Kompilierens durch weitere Daten den Quellcode, kann es untersagt werden die Software weiterzugeben, es darf aber erfolgen. Dazu kann die Pflicht auferlegt werden, der neuen Software einen anderen Namen oder eine neue Versionsnummer zu geben. Personengruppen oder einzelne Personen dürfen anhand dieses Modells nicht diskriminiert und Entwicklern darf nicht vorgeschrieben werden für welchen Zweck er seine Entwicklung nutzen soll. Zudem ist es Pflicht die Rechte einer Software bei der Weitergabe mitzuführen. Ist es notwendig eine zusätzliche Lizenz zu erwerben, handelt es sich nicht um Open Source Software. Wird die Software innerhalb einer Zusammenstellung ausgeliefert, dürfen die Rechte nicht eingeschränkt werden. [Open Source Initiative, 2007]

Ähnlich zu proprietärer Software ist der Entwickler\Autor einer Open Source Software deren Urheber und behält das Copyright an seiner Schöpfung. Eine entscheidende Rolle für die Nutzungsrechte spielt der von Richard Stallmann geprägte Begriff *Copyleft*. Wird eine Software (mit oder ohne Änderungen) weitergegeben, greifen die *Copyleft*, welche in vier Kategorien unterteilt werden.

Die erste Kategorie ist das starke *Copyleft*. Hier verpflichtet sich der Lizenznehmer bei Weitergabe einer veränderten GNU GPL Software diese wieder komplett unter der gleichen Lizenz mit Quelltext zur Verfügung zu stellen. Ein Beispiel dafür stellt die in 70 % der Open Source verwendete Lizenz GNU General Public License (GPL) dar.

Mit der Kategorie, dem schwache *Copyleft* wird der Entwickler verpflichtet bei Weitergabe ei-

ner veränderten GNU GPL Software den komplett gleichen Quelltext unter der selben Lizenz zur Verfügung zu stellen. Die letzte Art stelle die *kein Copyleft* dar. Hier kann die Software ohne Quelltext und Freiheiten weitergegeben und in proprietärer Software verwendet werden.

Software mit mehreren unterschiedlichen Lizenzen aus Kategorie 1 bis 3 fallen unter den Schutz der stärksten Lizenz. [Kleijn, 2016]

Das GNU-Projekt beschreibt *Copyleft* im Vergleich zum Copyright sehr anschaulich [Kallas, 2001]:

*„Proprietäre Software-Entwickler verwenden das Copyright, um den Benutzern ihre Freiheit zu nehmen; wir verwenden es, um ihnen ihre Freiheit zu garantieren. Deshalb haben wir den Namen umgedreht und aus dem 'Copyright' das 'Copyleft' gemacht.“*

## 4.2. Asynchronous JavaScript and XML (AJAX)

Mit der Verwendung von asynchronem JavaScript und XML (AJAX) konnten die Interaktionsmöglichkeiten des Nutzers mit der WebMapping-Anwendung auf ein neues Level gehoben werden. Der Begriff an sich wurde ursprünglich im Jahr 2005 von dem richtungweisenden Artikel *A New Approach to Web Applications* von Jesse James Garrett geprägt [GARRETT, 2005]. In WebMapping-Anwendungen der ersten Generation führte jede Interaktion mit der Karte zu einem neuen Laden der Website und damit zu Wartezeiten für den Nutzer. Mit AJAX funktionierende Anwendungen können Inhalte austauschen, ohne dass ein Neuladen der Website notwendig wird. Ein Beispiel aus der Internetkartographie wären Nutzerinteraktionen mit der Karte via zooming oder panning. Hierbei ermöglicht AJAX ein Anpassen der Kartenansicht im Hintergrund ohne ein Neuladen der Anwendung zu erzwingen.

Diese Technologie beschränkt die Kommunikation zwischen Clienten und Server auf die tatsächlich zu verändernden Elemente einer Seite. Mit AJAX können Anfragen an einen Server gesendet, Daten erhalten und angezeigt werden, ohne auf eine Nutzerinteraktion zu warten [Olsen, 2007]. Durch den Mix an Technologien, welche durch AJAX vereint werden, können WebMapping-Anwendungen entstehen, die ein Ansprechverhalten von Desktop-Anwendungen besitzen, ohne dass eine Anwendung auf dem Endgerät installiert werden muss. AJAX hat damit entscheidend zu der wachsenden Popularität von WebMapping-Anwendungen beigetragen.

Mit der Einführung einer JavaScript Engine zwischen dem Benutzerinterface und dem Server, wird das klassische „Anfrage/Antwort“ Interaktionsmuster aufgebrochen. Es kann jetzt gezielt auf Benutzereingaben reagiert und Inhalte der angezeigten Webseite dynamisch angepasst werden. Während die Seite beim Benutzer angezeigt wird, besteht die Möglichkeit neue Daten dynamisch zu laden und in die Seite einzufügen, ohne dass ein Neuladen der Seite erzwungen wird. Die AJAX Engine erlaubt den Benutzern eine asynchrone Kommunikation zwischen dem Server und der Anwendung (siehe Abbildung 3) [GARRETT, 2005].

Wechselt der Kartennutzer seine angezeigte Karte durch eine Interaktion, wird keine neue Anfrage an den Server gestartet, darum kümmert sich die AJAX Engine. Diese startet mittels XML die Serveranfragen asynchron, ohne die Interaktion des Nutzers mit der Anwendung zu beeinflussen. Wird eine neue Karte vom Server empfangen, kann ein Austausch erfolgen ohne dass die Website neu geladen werden muss.

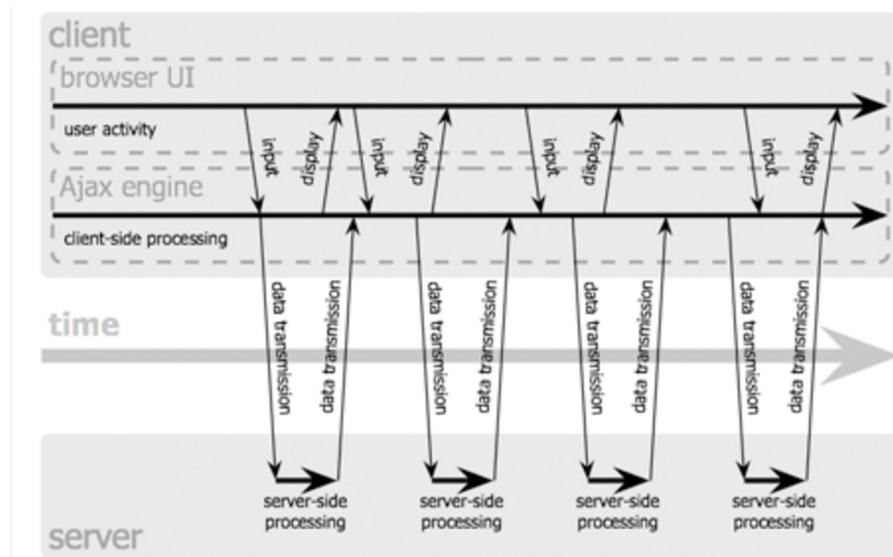


Abbildung 3: Der Prozessfluss einer Ajax Anwendung. Quelle: [GARRETT, 2005]

### 4.3. JSON

Um durch AJAX effektiv mit dem Server zu kommunizieren, haben sich in der letzten Zeit verschiedene Datenformate bewährt. Für den Austausch von reinen Daten bietet sich das XML Format an. Nicht umsonst steht das X in AJAX in Verbindung mit diesem Datenformat. Jedoch ist das Übergeben komplexer Objekte mit XML sehr umständlich [Blankerts, 2006], wie sie beispielsweise im WebMapping vorkommen.

Hier bietet die JavaScript Object Notation (JSON) einen performanten Lösungsansatz zur Übertragung von Objekten, welches von allen Browsern direkt unterstützt wird. Bei JSON handelt es sich um ein schlankes, textbasiertes Datenaustauschformat. Es wurde von Douglas Crockford mit dem Ziel entwickelt, ein Datenformat zu schaffen, welches von Mensch und Maschine leicht lesbar ist. JSON ist zum einen gekennzeichnet durch die Zuweisung von Namen/Werte Paaren und zum anderen durch eine geordnete Liste. Es werden die Datentypen *object*, *array*, *number*, *string*, *true*, *false* oder *null* unterstützt. [Ecma International, 2013]

Um Geometrien zwischen dem Clienten und Server auszutauschen, wurde GeoJSON entwickelt. Dieses Format erweitert JSON um Geometrien, Objekte (engl. features) oder Objektsammlungen (engl. feature collection). Das GeoJSON Format unterstützt die Geometrietypen *Point*, *LineString*, *Polygon*, *MultiPoint*, *MultiLineString* und *MultiPolygon* sowie Kombinationen dieser mit einer *GeometrieCollection* [Butler, 2008].

### 4.4. Responsive Web Design

Die zunehmende Verbreitung von mobilen Endgeräten und die Forderung des Web 2.0 nach einer Geräteunabhängigkeit zwingt die Entwickler von Web-Anwendung sich auf verschiedene Benutzertypen und die immer unterschiedlicheren Bildschirmgrößen zur Darstellung ihrer Inhalte auf eine flexible Art einzustellen. Hier bietet Responsive Webdesign eine Lösungsmöglichkeit an. Dieses Konzept wurde 2010 zum ersten Mal in einem Online-Beitrag von Ethan Macotte verwendet. Hier thematisiert er eine technische Version von Websites, die sich an

jeden Gerätetyp automatisch anpassen [Macotte, 2010]. Um eine WebMapping-Anwendung zu erstellen, die sich auf unterschiedlichen Geräten optimal darstellen lässt, gibt es verschiedene Wege.

Eine Möglichkeit ist das Erstellen separater Webseiten, die auf jeden Gerätetyp spezialisiert sind. Zum Beispiel gibt es eine Webseite für PCs, eine für Tablets und eine für Smartphones. Da es sich jeweils um eigene Website handelt, ist auch der Entwicklungs- und Pflegeaufwand höher.

Eine sinnvollere Alternative stellt die Möglichkeit dar, eine Anwendung zu erstellen, deren Layout sich auf unterschiedliche Geräte flexibel anpasst und damit auch eine Kompatibilität mit zukünftigen Websites gewährleistet [Mohorovičić, 2013]. Responsive Webdesign ist für diesen Kontext ein ausgezeichneter Lösungsansatz. Eine Website, die auf diesem Design beruht, besitzt nur einen HTML-Code und der Inhalt passt sich durch die Möglichkeiten von modernen Webtechnologien u.a. CSS3 oder JavaScript den unterschiedlichen Geräten an.

Zur Realisierung werden *flexible Grids*, *flexible Medientypen* und *Media Queries* benötigt, die es erlauben Webseiten, unabhängig der Bildschirmgröße eines Geräts, anzupassen.

#### 4.4.1. Flexible Grids

*Flexible Grids* bestehen aus GUI-Elementen, die mit relativen Maßen (z. B. in Prozent, statt mit festen Pixelangaben) eingebunden werden, je nachdem wie viele und welche Elemente auf der Webseite existieren [Moore, 2014]. Durch die Verwendung von *Flexible Grids* wird eine einfache Größenveränderung möglich gemacht, die sich nach der Anzeigemöglichkeit, wie der Bildschirmgröße, richtet [vgl. Mohorovičić, 2013, S. 1207].

#### 4.4.2. Media Queries

Um dem Browser mitzuteilen, bei welcher Bildschirmgröße er welches Layout verwenden soll, werden sogenannte *Media Querries* verwendet. Durch sie ist es möglich eine genauere, endgerätespezifische Beschreibung für die unterschiedlichen Styles abzulegen. Damit wird die Option geschaffen, unter einem HTML-Dokument mehrere Layouts zu entwickeln und einzusetzen [vgl. Verclas u. Linnhoff-Popien, 2012, S.154]. Über diese Anweisungen können unterschiedliche Möglichkeiten von Bedingungen über den Media Type- sowohl die Hardwarecharakteristika (z.B. den Bildschirm) als auch die Softwarecharakteristika (z.B. den Browser inkl. seiner Einstellungen)-eines Gerätes identifiziert und vergeben werden. Die CSS3-Spezifikationen werden von allen aktuellen Browsern unterstützt und bilden somit eine gute Grundlage für das Responsive Webdesign [vgl. Mohorovičić, 2013, S. 1208]. Durch den Fakt, dass bei Smartphones die Breite geringer ist als die Höhe, wird die width-Funktion am häufigsten eingesetzt. Ein Code-Beispiel für dieses *Media Query* lautet wie folgt:

```
1 @media only screen and (max-width : 550px) { .container{width:50%;}}
```

Mit dieser Abfrage lässt sich in wenigen Codezeilen die Über- oder Unterschreitung einer bestimmten Auflösung prüfen. Ist eine Breite von 550px erreicht, ändert sich das Layout des angesteuerten Elementes. In diesem Beispiel wird die Breite auf 50% erhöht.

Mit Hilfe des Responsive Design kann sich eine WebMapping-Anwendung an die Bedürf-

nisse und das Gerät dynamisch anpassen. Mit dieser Technologie bleibt der Kerninhalt einer Anwendung für den Nutzer erhalten und die Nutzung aller angebotenen Funktionen werden nicht verhindert, wenn der Nutzer unterschiedliche Endgeräte verwendet, da sich sowohl das Layout der Webseite als auch die Medienelemente in ihrer Darstellungsform und Größe verändern [vgl. Kollmann u. Michaelis, 2015, S.408].

Mit der folgenden Abbildung 4 wird das Konzept noch einmal veranschaulicht dargestellt.

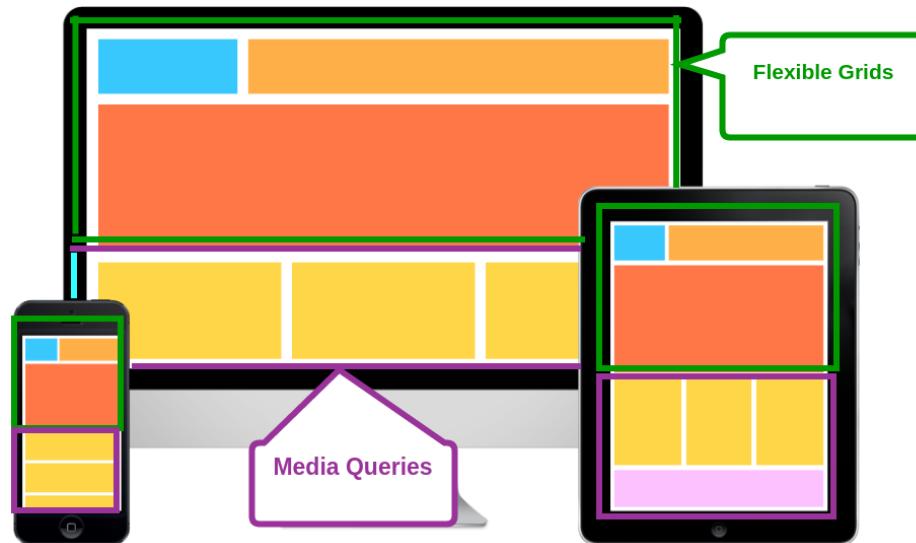


Abbildung 4: Beispielwebsite mit Responsive Webdesign. Quelle: [Caldwell, 2013]

Die oberen Bereiche der Anwendung passen sich durch *Flexible Grids* dynamisch an die Breite des Wiedergabemediums an. Muss das Layout des Elementes grundlegend verändert werden, wird dies über *Media Queries* realisiert. In den lila eingefärbten Bereichen der Abbildung ist zu sehen, dass sich bei einer bestimmten Breite die Elementanordnung ändert. Kommt es zu einer Registrierung des festgelegten Schwellwertes, wird die vertikale- in eine horizontale Struktur transformiert, was in der Abbildung auf dem Smartphone eintritt.

## 5. Anforderungsanalyse

Die durchgeführten Phasen in diesem Kapitel umfassen die Analyse von Aufgaben, Benutzern und des Kontextes einschließlich der Rahmenbedingungen, mit dem Ziel Anforderungen an die Entwicklung abzuleiten. Diese Analysen sind dabei gezielt darauf gerichtet, die Anforderungsdefinition an eine neue Anwendung zu unterstützen.

### 5.1. Benutzeranalyse

Um ein grundsätzliches Verständnis der Aufgabe zu entwickeln, ist es wichtig, potentielle Benutzer kennenzulernen und zu verstehen. Hier liegt die Kernaufgabe dieser durchgeführten Analyse. Die Gewinnung eines vertieften Verständnisses von Benutzern und ihren Fähigkeiten, Qualifikationen, Rollen und Vorstellungen [Preim u. Dachselt, 2015]. Diese Form der Untersuchung wurde von Daniela Förtsch [2017] im Rahmen ihrer beruflichen Tätigkeit am Leibniz-Institut für ökologische Raumentwicklung durchgeführt. Die Ergebnisse ihrer Arbeit wurden genutzt, um die Definitionen von Anforderungen an die zu konzipierende Anwendung zu bereichern.

#### 5.1.1. Methodik

Daniela Förtsch [2017] führte 50 persönliche und telefonische Interviews durch. Hierbei dokumentiert Abbildung 5 die Zusammensetzung und Verteilung der Zielgruppen und zeigt das potentielle Nutzer während der Studie vor allem im Wirtschaftssektor gesehen wurden.



Abbildung 5: Zusammensetzung der befragten Probanden. Quelle: [Förtsch, 2017]

Das Interview stützte sich dabei auf vier Kernfragen, welche wichtige Erkenntnisse für die Weiterentwicklung des IÖR-Monitors generieren sollten. Die folgenden vier Kernfragen wurden aus den Aufzeichnungen der Studie abgeleitet bzw. zusammengefasst:

1. *Welchen Hintergrund besaßen die Befragten, bei der Verwendung von Geodaten des IÖR-Monitors?*

Diese Frage zielt darauf ab, herauszufinden für welche Aspekte der Nutzung die Anwendung eingesetzt wird.

2. *Welche Erfahrung wurden mit der Anwendung gemacht ?*

Auf der Grundlage dieser Fragestellung sollten die Nutzererfahrungen herausgefunden werden, um mögliche Entwicklungspotentiale zu erkennen.

3. *Welche Anregungen können für die Weiterentwicklung des Monitors gegeben werden?*

Diese Fragestellung baut auf der zweiten Frage auf, dient der Filterung von gewünschten Funktionen und soll auch Anregungen für potentielle Änderungen der Oberflächengestaltung geben.

4. *Wie wird das Anwendungspotential der Anwendung eingeschätzt?*

Mit dieser Frage soll ein Feedback der Befragten erhoben werden, inwieweit die Anwendung mit ihren Diensten und Analysemöglichkeiten aktuelle Problematiken und Fragestellungen innerhalb des jeweiligen Interessengebietes unterstützt.

### 5.1.2. Ergebnisse

Die gesammelten Ergebnisse wurden in einer Excel-Tabelle dokumentiert und stellten einen wichtigen Baustein für die Konzeption der zu planenden Neuauflage des IÖR-Monitor dar. In diesem Abschnitt werden die wichtigsten Ergebnisse zusammenfassend dargestellt. Dabei konnten aus der Erhebung von Daniela Förtsch [2017] zwei Kategorien gebildet werden. Diese betreffen die Oberfläche und den funktionalen Umfang der Anwendung. Folgend sollen diese beiden Kategorien mit den gefilterten Ergebnissen vorgestellt werden.

#### Oberfläche

Alle Nutzer empfanden die Trennung zwischen Raster und Vektorviewer als störend und empfahlen eine Zusammenführung. Ein weiterer Punkt betraf die Kategorisierung der Indikatoren, welche als unverständlich dargestellt und als Anregung eine Vereinfachung empfohlen wurde. Besonders die Präsentation der Interaktionsmöglichkeiten und deren Verteilung hat viele Nutzer „erschlagen“ und sollte nach Anregungen vereinfacht dargestellt werden.

Zusammenfassend fand immer das Schlagwort einer einfacheren und intuitiveren Bedienung seine Verwendung, welche einen verständlicheren Zugang zu den einzelnen Funktionalitäten ermöglichen soll.

#### Funktionen

Ein großer Teil der Benutzer wünscht sich eine höhere Interaktivität mit den Vektorkarten und einer damit verbundenen Umstellung zu einer stärkeren Verankerung von Interaktionsoptionen mit der Karte. Nutzer aus dem Planungsbereich favorisierten die Möglichkeit, indivi-

duelle Gebietskulissen erstellen zu können. Hier wird vor allem die Chance gesehen, separierte Auswertungen aus einem selbst zusammengestellten Gebiet zu bekommen.

Ein weiterer Punkt betraf die Rasterkarten. Hier fand eine Anregung von Nutzern aus dem wissenschaftlichen Sektor statt, welche eine farbliche Anpassungsperspektive empfahlen. Als Grund wurde die Notwendigkeit bei Veröffentlichungen in Schwarz/Weiß genannt.

Mit den zwei Kategorien wurden die wichtigsten Erkenntnisse der Benutzeranalyse zusammengetragen, welche für die Weiterentwicklung ein hohes Maß an Berücksichtigung finden muss. Betrachtet man die genannten funktionalen Verbesserungswünsche, wird deutlich dass viele Nutzer in der Erhöhung der Interaktivität Optimierungen nannten. Vor allem im Vektorviewer konnte ein hohes Maß an Entwicklungspotential erkannt werden, da die momentane Implementierung nur eine sehr eingeschränkte Interaktion zulässt. Die gegebenen Antworten spiegeln auch das Verhalten der überwiegend beruflichen Nutzer wieder. Diese Gruppe besucht WebMapping-Anwendung meist, um im Kontext ihrer Aufgabe Analysen durchzuführen, wo ein hohes Maß an Interaktionsmöglichkeiten gefragt ist und die Ergebnisse übersichtlich dargestellt werden müssen. Eine moderne und leicht verständliche intuitive Oberfläche bietet für diese Nutzergruppe sicherlich einen guten Ausgangspunkt, um die anstehenden analytischen Aufgaben erfolgreich durchzuführen.

## 5.2. Rahmenbedingungen

In diesem Kapitel werden die Einflüsse und Vorgaben identifiziert, die als Rahmenbedingungen zu beachten sind. Durch vorangegangene Entwicklungen des IÖR-Monitors entstand bereits eine funktionierende serverseitige Software-Plattform, welche in dieser Arbeit nicht verändert werden soll. Diese konkreten Merkmale ordnen [vgl. Preim u. Dachselt, 2015, S.95] den direkten Vorgaben zu, welche folgend erörtert werden.

Um die Rasterkarten dem Clienten zur Verfügung zu stellen, wird ein *MapServer*<sup>6</sup> verwendet. Dieser erstellt die angeforderten Karten-Grafiken und sendet diese in Form von Web Map Services (siehe Kapitel 3.4.2) über den Apache-Webserver zurück. Die Geometrien, welche für die Generierung von Gebietseinheiten innerhalb der Vektoransicht benötigt werden, liegen in einer *PostgreSQL*<sup>7</sup>-Datenbank. In dieser Datenbank werden auch die Geometrien der Zusatzlayer gespeichert, welche in der Raster- und Vektoransicht als zusätzliche Layer geladen werden können. Die Indikatoren und deren Werte sind in einer *MySQL*<sup>8</sup>-Datenbank gespeichert. Für jedes Jahr, liegt eine entsprechende Tabelle vor, in welcher alle verfügbaren Indikatoren und deren Werteausprägungen vorhanden sind. Auch die Freigabe, Farbliche Gestaltung, zusätzliche Inhalte, Fehlercodes und Hinweistexte der Indikatoren sind in entsprechenden Tabellen abgelegt.

Für die serverseitige Programmierung steht *PHP*<sup>9</sup> in der Version 5 zur Verfügung. Diese Software-Plattform bildet eine funktionierende Infrastruktur für die Bereitstellung der ver-

---

<sup>6</sup><http://mapserver.org/>

<sup>7</sup><https://www.postgresql.org/>

<sup>8</sup><https://www.mysql.com>

<sup>9</sup><http://php.net>

fügbaren Inhalte auf dem Apache-Webserver, welche vom Clienten angefragt werden können. Die gegebene Architektur ist in Abbildung 6 dargestellt.

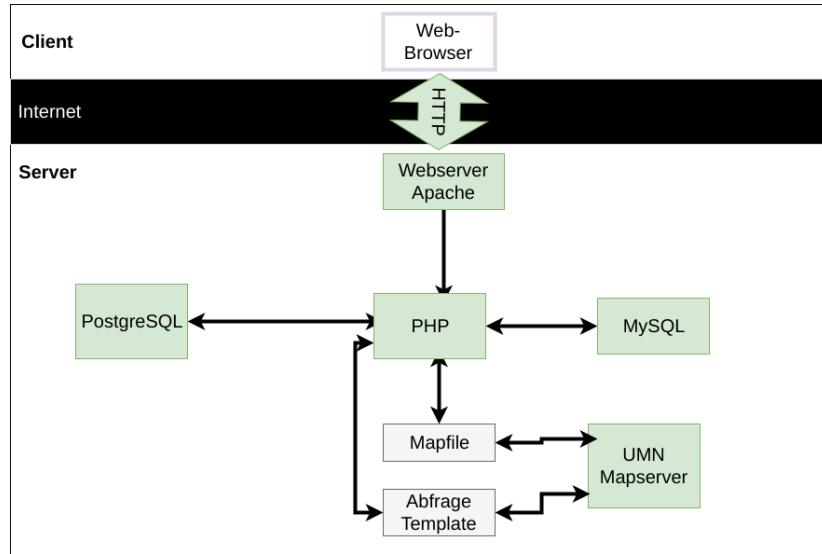


Abbildung 6: Architektur der Serverseitigen Software Plattform

Die Generierung und Bereitstellung der Rasterkarten, durch die Zusammenarbeit von *PHP* und *Mapserver* wird in dieser Arbeit nicht verändert und angepasst eingesetzt. Ein *PHP* Script generiert das entsprechende *Mapfile*, welches als zentrales Element für den *MapServer* gesehen werden kann. Der Hauptzweck dieser Datei ist, die Layer und deren Datenquelle festzulegen sowie die Darstellung zu definieren (z.B. Farbe, Symbole, Labels). Dieses *Mapfile* wird von dem Clienten über eine HTTP-Schnittstelle aufgerufen und die Rasterkarte mit dem verwendeten Kartenframework visualisiert.

Neben den direkten Vorgaben formuliert das Leibniz-Institut für ökologische Raumentwicklung Anforderungen an die Gestaltung, welche nach Preim u. Dachselt [2015] den unscharfen Vorgaben zugeordnet werden.

Diese betreffen vor allem die Benutzerschnittstelle, also den clientseitigen Teil der Anwendung. Eine Forderung stellt sich an die farbliche Gestaltung der Oberfläche, welche die beiden Farbtöne des Institut-Logos als Hauptfarben vorsieht. Eine weitere Vorgabe stellt die Existenz eines Headers mit festen Inhaltspunkten dar. Dieser muss das Logo des Institutes beinhalten und auch die Punkte: Impressum, Datenschutz und Copyright. Alle Elemente sollten eine Weiterleitung auf die entsprechenden Inhalt der Instituts-Website besitzen und in einem neuen Tab geöffnet werden.

### 5.3. Bestandsanalyse vorhandener Web Mapping Anwendungen

Bei dieser Analyse sollen vor allem Momentaufnahmen von ausgewählten WebMapping 2.0 Anwendungen gemacht werden. Die Untersuchungsergebnissen dienen durch eine Gegenüberstellung mit der aktuellen Implementierung des IÖR-Monitors, der Ableitung möglicher Entwicklungspotenziale der zu konzipierenden Anwendung und damit einer Formulierung von Anforderungen. Dabei geht es um eine qualitative Betrachtung von WebMapping-Anwendungen im Web 2.0 und den Vergleich dieser anhand einer Kriterienliste.

### 5.3.1. Methodik

Am Beginn fand eine Festlegung der Kriterien statt, welche eine Grundlage für die darauf folgende Analyse bildeten. Dabei lag der Fokus auf ausgewählten, typischen WebMapping 2.0-Anwendungen, die über eine grafische Benutzeroberfläche verfügen und direkt im Webbrower ohne eigene Programmierung oder Softwareinstallation nutzbar sind. Alle Anwendungen wurden mit einem Smartphone und einem Desktop-PC auf die festgelegten Kriterien untersucht. Diese sind in der folgenden Liste dokumentiert:

#### 1. Benutzeroberfläche

Ein wichtiger Bestandteil des Web 2.0 und in diesem Zuge auch des WebMapping 2.0 stellt die Geräteunabhängigkeit dar [Gartner, 2009]. Diese Forderung stellt sich vor allem an den Aufbau des Interfaces der Anwendung, es muss die Möglichkeit schaffen auf unterschiedliche Darstellungsanforderungen reagieren zu können.

Mit diesem Kriterium soll diese Fähigkeit näher untersucht werden und deren Realisierung. Der IÖR-Monitor zeichnet sich durch einen großen Funktions- und Inhaltsumfang aus. Um mögliche Lösungsansätze zu finden ist es wichtig, eine analytische Betrachtung des Navigationsaufbaus durchzuführen, mit welchem der Nutzer interagiert.

#### 2. Kartengestaltungs- und Analysemöglichkeiten durch kartographische Interaktion

Interaktivität ist mit der wachsenden Beliebtheit von WebMapping-Anwendungen und der rasanten Entwicklung der Webtechnologien heute zu einem fundamentalen Bestandteil in der Entwicklung und Gebrauchstauglichkeit von Kartenanwendungen geworden [vgl. Skupin u. Hagelman, 2005, S.160]. Standard Interaktionen, wie das Verschieben durch Panning oder das Hineinzoomen in die Karte durch das Mausrad, ist in heutigen WebMapping-Anwendungen nicht mehr wegzudenken. Diese Formen der Interaktion unterstützen die Orientierung und Navigation in der digitalen Karte, welche gesammelt und dokumentiert werden sollen.

Das Ziel dieser Fragestellung geht über die klassische Interaktionen hinaus und versucht mögliche nutzergesteuerte Anpassungen der digitalen Karten herauszufiltern und diese zu analysieren.

Viele Nutzer kennen diverse Analysewerkzeuge, wie beispielsweise eine Messfunktion von klassischen Desktop-GIS Anwendungen. In Zeiten einer zunehmenden Annäherung zwischen Desktop und Web-Anwendungen durch den Einsatz von AJAX [vgl. Dickmann, 2004, S.24], können umfangreichere kartographische Analysewerkzeuge innerhalb einer Web-Anwendung umgesetzt werden. Hier soll die Untersuchung aufzeigen, welche Optionen der Nutzer für erweiterte Analysen zur Verfügung hat und wie Informationen aus der digitalen Karte gewonnen werden können.

#### 3. Kartenveröffentlichung und -weiterverwendung

Viele Nutzer möchten die selbst erstellten Kartenansichten gerne teilen oder dauerhaft erhalten, weshalb es wichtig ist diese Möglichkeit für eine WebMapping 2.0 Anwendung in Betracht zu ziehen. Goodchild [2007] begründet die Notwendigkeit dieser Funktionalität mit der Ursache das de facto fast jeder Mensch ein lokaler Experte in seiner vertrauten Umgebung ist. Mancher will seine Erfahrungen weitergeben und teilen [vgl.

Goodchild, 2007, S.24-32]. Diese Möglichkeit selbst erstellte Inhalte mit anderen Nutzern auszutauschen, wird auch als ein Markenzeichen des Web 2.0 gesehen, welches in die Untersuchung mit einbezogen werden muss [Vgl. Blumauer u. Pellegrini, 2009, S.108]. Auch während der Benutzeranalyse (siehe Kapitel 5.1) durch Daniela Förtsch [2017] wurde diese Option als eine sehr wichtiges Funktion genannt, welche implementiert sein sollte.

#### 4. Technische Aspekte

Dieser zusätzliche Aspekt geht auf die Performance der Kartendarstellungen ein und versucht damit festzuhalten, wie flüssig die Interaktionen erfolgen können.

##### 5.3.2. Auswahl der zu analysierenden Anwendungen

Inzwischen existieren eine Reihe von WebMapping-Anwendungen, in welcher der Nutzer direkt im Browser mit der Karte interagieren kann. Bei dem Thema Karten im Internet denken viele Nutzer sofort an Google Maps. Diese Anwendung ist geräteunabhängig gestaltet und kann mit einem ansprechenden Aufbau und Bedienung punkten, besitzt jedoch im Zusammenhang mit der interaktiven Kartengestaltung nur begrenzte Möglichkeiten die Karte nach nutzerseitigen Vorstellungen zu gestalten. Für die Bestandsanalyse im Rahmen dieser Arbeit wurde eine Beschränkung auf Anwendungen unternommen, welche innerhalb ihres Datenbestandes und Nutzerkreises dem IÖR-Monitor nahekommen.

Ein weiterer Punkt ist die anvisierte Unabhängigkeit von Plattform und Endgeräten, welche von den geplanten Anwendungen realisiert sein sollte. Die Auswahl fiel dabei auf vier kommunale Angebote, mit einem breit gefächerten Angebot an Geodaten, womit eine Vergleichbarkeit zu der geplanten Anwendung geschaffen werden kann. Alle Anwendungen werden von dem jeweiligen Landesbetriebs für Geoinformation und Vermessung bereitgestellt. Sie stellen dem Nutzer Geobasisdaten und Luftbilder als Hintergrundkarten zur Verfügung. Zusätzlich sind umfangreiche Analysetools und thematische Karten vorhanden, in welchen auch Ergebnisse für das Flächenmonitoring des jeweiligen Anbieters visualisiert werden können.

Die untersuchten Anwendungen sind:

- ***Geo-Online*<sup>10</sup>**
- ***Geo Viewer Sachsen*<sup>11</sup>**
- ***BayernAtlas*<sup>12</sup>**
- ***Geo Admin CH*<sup>13</sup>**

##### 5.3.3. Ergebnis

Im folgenden Kapitel sollen die Ergebnisse der Studie diskutiert werden. Hierbei werden wissenschaftliche Erkenntnisse und Untersuchungen mit einbezogen.

<sup>10</sup><https://www.geoportal-hamburg.de/Geoportal/geo-online>

<sup>11</sup><https://geoviewer.sachsen.de/>

<sup>12</sup><https://geoportal.bayern.de/bayernatlas/>

<sup>13</sup><https://map.geo.admin.ch/>

### 5.3.3.1. Benutzeroberfläche

In allen Anwendungen fiel besonders auf, dass der Nutzer immer im aktuellen Fenster gehalten wurde und nur selten eine Weiterleitung erfolgte. Für externe Inhalte wie beispielsweise die Hilfe leiteten der *BayernAtlas* und *Geo Admin CH* den Nutzer in ein neues Tab, wo die Inhalte erschienen. Der Aufbau und auch die Navigation wiesen in den untersuchten Anwendungen eine ähnliche Gestaltung auf und konnten in einem Schema zusammengefasst werden. Alle Anwendungen platzierten als zentrales Element die Karte in ihrer Ansicht, welche die komplette Breite des Displays nutzt. Dieses Layout entspricht auch den besten Ergebnissen einer Evaluation in Form einer Usability Studie, in welcher verschiedene Ansätze für die Layoutgestaltung von WebMapping-Anwendungen getestet wurden und eine Konzentration des Layouts in zwei Teile die besten Resultate erzielten: einen Kartenviewer und Navigationsmenü [Manlai u. a., 2007]. Erweiterte Navigationsinhalte der Anwendungen konnten durch eine an-/abschaltbare Organisation eine bessere Übersicht erreichen und setzen damit konsequent die Forderung von Manula Hofman um, sich wiederholende Elemente ein- und ausblendbar zu gestalten [vgl. Hoffmann, 2013, S. 60].

Für Selektion und Auswahl von Kartenelementen setzen die Anwendungen ganz unterschiedliche gestalterische Möglichkeiten ein. Bis auf den *Geo Viewer Sachsen*, wird ein sogenanntes Accordion verwendet [vgl. Hahn, 2017, S.716]. Bei diesem GUI-Element werden die möglichen Inhalte einer Kategorie versteckt und bei der Wahl angezeigt. Hierbei wurden die einzelnen Accordion-Listen untereinander angeordnet und nach einem Klick- oder Touch Event einzeln angezeigt. Der *Geo Viewer Sachsen* geht einen anderen Weg und setzt den sogenannten Slider ein. Hier werden ähnlich des Accordions nicht alle Inhalte angezeigt, sondern nach und nach verfügbar gemacht. In der folgenden Abbildung 7 werden diese Implementierungsmöglichkeiten für die Auswahl anhand von zwei Screenshots graphisch dargestellt.

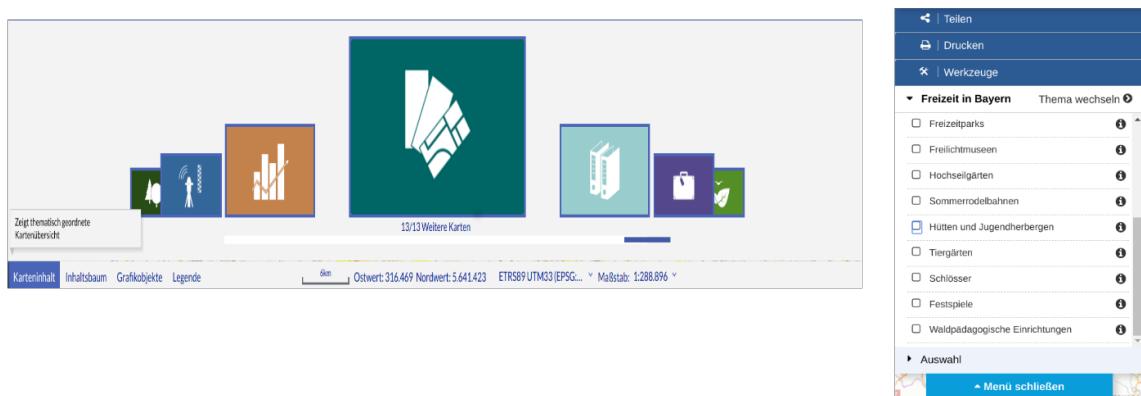


Abbildung 7: Vertikaler Slider des *Geo Viewer Sachsen* im Vergleich zum Accordion des *BayernAtlas* zur Kartenauswahl, Quelle: [Staatsbetrieb Geobasisinformation und Vermessung Sachsen, 2017; Bayerisches Staatsministerium der Finanzen, für Landesentwicklung und Heimat, 2017]

Die Navigationsleiste wurde in den Anwendungen entweder unterhalb oder oberhalb der Karte angeordnet. Die wichtigsten Funktionen zur kartographischen-Interaktion fanden ihre Platzierung in Form von Buttons auf der Karte, was eine schnelle Erreichbarkeit ermöglicht. Die Anordnung dieser Elemente fand ausnahmslos am rechten Rand des Bildschirmes statt. Diese räumliche Nähe der Elemente suggeriert Nutzern einen thematischen oder kausalen

Zusammenhang [Norman, 1988], mit den Interaktionsmöglichkeiten der Karte. Der *BayernAtlas* und auch der *Geo Viewer Sachsen* setzten diese Regel außer Kraft und fügten einen „*Impressum-Button*“ zwischen die kartographischen Interaktionselemente. Für diese Funktion ist laut Hoffmann [2013] eher der Fußbereich oder Kopfbereich einer Anwendung vorgesehen. Um dem Nutzer die Suche nach Orten und Karteninhalten zu erleichtern, verwenden alle Anwendungen eine Suchfunktionalität in Form einer Suchleiste. Besonders die Autovervollständigung, welche die Suche unterstützt, führt zu einem positiven Anstieg der Usability für den Nutzer. Da sie in vielen Fällen das Finden von bestimmten Inhalten vereinfacht [vgl. Fähnrich u. Franczyk, 2010, S. 1080]. Dafür werden beim Starten einer Sucheingabe, passende Inhalte in Form einer Liste eingeblendet, welche durch ihre Auswahl in der Karte visualisiert werden können. Beim Aufrufen der Anwendungen mit einem Smartphone, passte sich die Platzierung der Navigationsleiste entsprechend an. Auf Abbildung 8 werden die Transformationen des Layouts an einen schmalen Bildschirm dargestellt.

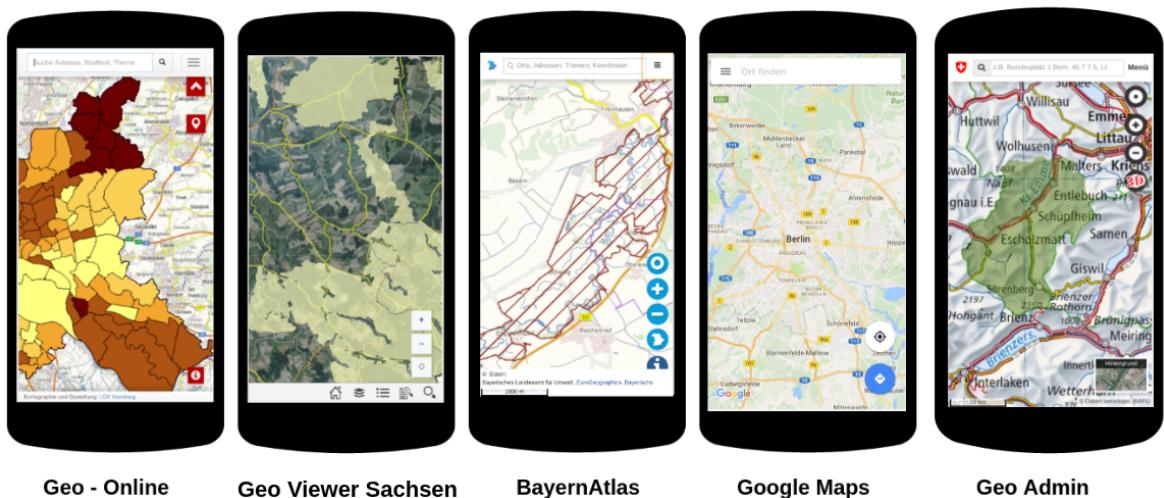


Abbildung 8: Responsive-Layout der untersuchten Anwendungen innerhalb der Benutzeranalyse, Quelle: [Landesbetrieb Geoinformation und Vermessung. Kartographie und Gestaltung: LGV Hamburg, 2017; Staatsbetrieb Geobasisinformation und Vermessung Sachsen, 2017; Bayerisches Staatsministerium der Finanzen, für Landesentwicklung und Heimat, 2017; Google, 2017; Schweizerische Eidgenossenschaft, 2017]

Auf der Abbildung ist gut zu sehen, dass alle Anwendungen bis auf den *Geo Viewer Sachsen*, die Suchleiste als zentrales Element innerhalb der Navigationsleiste platzieren. Im überwiegenden Teil der Anwendungen wird ein Responsive Webdesign für die Realisierung der Responsivität verwendet, lediglich im *Geo Viewer Sachsen* wird der Nutzer beim Seitenaufruf auf eine mobil angepasste Seite weitergeleitet. Dieses Vorgehen wird im Webdesign eher als ineffizienter Ansatz gesehen (siehe Kapitel 4.4) [vgl. Knoll u. Niederl, 2015, S. 104]. Die Navigationsleiste wird in allen mobilen Layouts durch einen Button aktiviert, wodurch sich das Navigationsmenü ausfährt, welches den ganzen Bildschirm nutzt, um auf die eingeschränkte Breite bei einem Smartphone zu reagieren. Die Bedienung bleibt dadurch weiterhin übersichtlich und Interaktionen durch Touch können problemlos erfolgen, da die GUI-Elemente sich in ihrer Dimensionierung dementsprechend anpassen.

Fortschrittsanzeigen, d.h. Informationen darüber ob die Anwendung noch arbeitet [Ormerod

u. Lansdale, 1994], wurden bei keiner der untersuchten Anwendungen implementiert. Gerade bei ladeintensiven Karten, wie z.B. der Geologischen Karte, erhält der Nutzer keine Rückmeldung zum Zustand des Ladeprozesses. Die Karte wird nach der erfolgreichen Abfrage vom Kartenserver, dem Viewer hinzugefügt. Meist erfolgt jedoch eine Antwort des Servers nicht synchron und der Nutzer wird in diesem Fall nicht ausreichend informiert.

Eine „Hilfe“-Funktionalität war in allen Anwendungen vorhanden. Deren Gestaltung kam jedoch keine große Bedeutung zu und hatte eher einen sehr unübersichtlichen Charakter. Im *Geo Viewer Sachsen* fand die Umsetzung dieser Funktion nicht statt und in den anderen drei Anwendungen nur durch eine lange Liste mit Informationen zu den einzelnen Inhalten. Bis auf den *BayernAtlas* wurden dem Benutzer in allen untersuchten Anwendungen Hilfen in Form von Mouseover Texten angeboten, was eine schnelle Hilfe darstellt.

### 5.3.3.2. Kartengestaltungs- und Analysemöglichkeiten

Alle vier Anwendungen boten dem Nutzer die Möglichkeit an, seine Grundkarte zu wechseln. Dabei kombinierten sie Dienste aus verschiedenen Quellen, wofür eine „Mashup“ Lösung den Einsatz fand. Zur Auswahl wurde dem Nutzer immer die Kategorien: Luftbild, Schwarz\Weißkarte und Basiskarte angeboten. Hierbei ließen sich bei allen Anwendungen die Grundkarten deaktivieren. In der Anwendung *Geo Admin CH* ist erweiternd sogar die Möglichkeit implementiert, den Karten-Viewer in einen dreidimensionalen Modus zu versetzen. Neben dem Austausch der Grundkarte, sind nur begrenzte Veränderungsmöglichkeiten des Aussehens der angezeigten Karte implementiert. Beispielsweise kann die Transparenz der visualisierten thematischen Karte verändert werden. In allen Anwendungen besteht die Möglichkeit anhand von einem Zeichenwerkzeug Vektorgrafiken zu erstellen und damit die digitale Karte zu überlagern, die farbliche Gestaltung ist hierbei variabel. Um punktuelle Informationen aus der durch den Nutzer selektierten Karte zu gewinnen, wurde in allen Anwendungen eine Sachdatenabfrage implementiert. Hierbei wird durch Klick oder Touchevent auf die gewünschte Position der Pixelwert für die gewählte Karte angezeigt. Die Anwendung *Geo-Online* geht jedoch in diesem Fall über eine Koordinatenabfrage nicht hinaus und verliert dadurch eine wichtige Funktion für den Nutzer.

Innerhalb der Vielfalt von Analysewerkzeugen unterschieden sich die Anwendungen nicht sehr stark voneinander. Standardmäßig ist ein Messinstrument zur Strecken und Flächenmessung vorhanden. Eine überaus nützliche Funktion der Analyse, konnte durch den Vergleichsmodus innerhalb des *BayernAtlas* und *Geo Admin CH* realisiert werden. Hier besteht die Möglichkeit zwei selbst gewählte Karten miteinander vergleichen zu können, indem ein vertikaler Slider den kartographischen Darstellungsbereich trennt. Der Nutzer kann eine Karte jeweils für den linken und rechten Teil wählen, wodurch Zusammenhänge analysiert werden können. Abbildung 9 dokumentiert den Vergleichsmodus. Zur Veranschaulichung der Funktionalität wurde für die linke Karte der Indikator des Holzvorrates mit den Erdbebenzonen im rechten Teil gegenübergestellt.

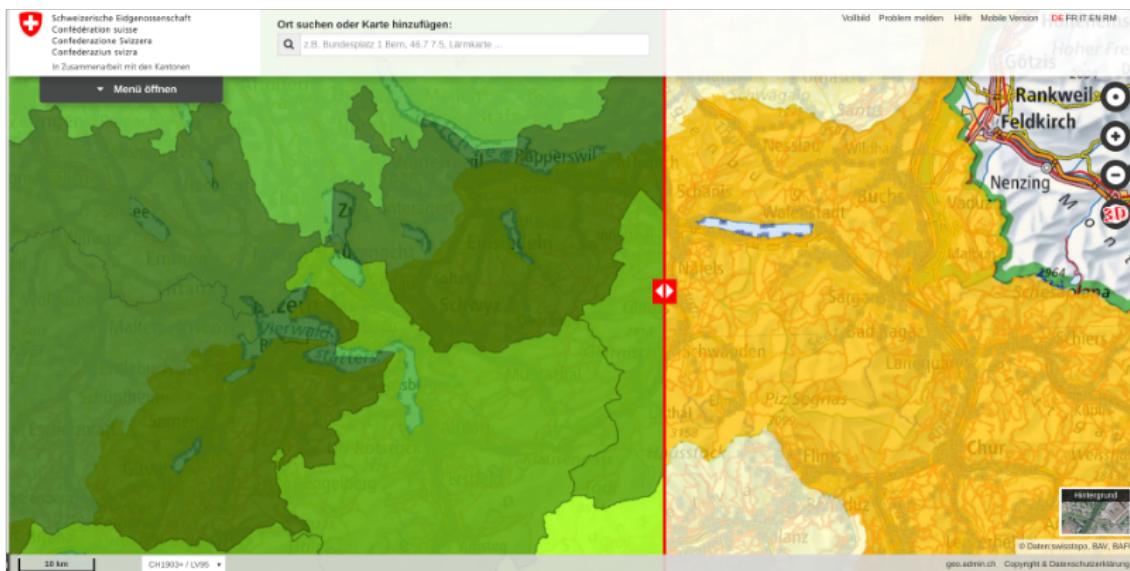


Abbildung 9: Vergleichsmodus in der Anwendung Geo Admin CH. Quelle: [Schweizerische Eidgenossenschaft, 2017]

Für den *Geo Viewer Sachsen* muss angemerkt werden, dass in der mobilen Variante keinerlei Werkzeuge zur Verfügung stehen, wohingegen die anderen Anwendungen dem Nutzer den vollen Funktionsumfang anbieten. Auffallend im *BayernAtlas* war das Fehlen der Legende, obwohl sie ein wichtiges kartographisches Element ist.

Die räumliche Navigation in der Karte geschieht in allen drei Anwendungen durch *panning* auf dem Bildschirm via Maus oder Touch. Aus älteren Generationen der WebMapping-Anwendungen sind noch die Navigationspfeile zur räumlichen Orientierung bekannt, mit diesen konnte sich der Nutzer durch Interaktion anhand der Himmelsrichtungen bewegen, welche in allen Anwendungen fehlten. Mit dieser ausgelassenen Implementierung suggerieren die Entwickler die Überflüssigkeit dieser Interaktionselemente und zeigen, dass sich die direkte Orientierung auf dem Bildschirm durchgesetzt hat. Um in die Karte herein- oder herauszuzoomen wurden entsprechende Interaktions-Buttons vorgesehen. Funktional finden diese vor allem bei der Nutzung von Zeigegeräten ohne Mausrad ihre Verwendung. Andererseits kann die Anpassung der Zoom-Stufe via Touch oder Mausrad erfolgen. Auf dem Smartphone blieben alle Navigationsbuttons erhalten.

Das zur Verfügung stellen von Zoom-Buttons im mobilen Layout wird bei einer Usability Studie durch die Universität Yunlin, Taiwan in Frage gestellt. Hier konnten die Autoren beweisen, dass Nutzer durch die Vertrautheit mit dem Touchdisplay, ausschließlich Touch-Interaktionen zur räumlichen Navigation innerhalb der Karte nutzten [Manlai u. a., 2007]. Die prominente Anwendung Google Maps setzte diese Erkenntnisse bereits um und verwendet keine Zoom Buttons auf ihrer Oberfläche.

Eine Übersichtskarte stellt in allen Anwendungen den globalen Kontext zu der Kartenposition des Nutzers mit der gesamten räumlichen Ausdehnung her. Dafür wird der aktuell visualisierte Bereich in der Gesamtkarte markiert, in allen Fällen durch eine farblich hervorgehobene Bounding Box. Per Klick oder Drag & Drop Interaktion mit dem markierten Bereich der Übersichtskarte, kann die Position des Darstellungsbereiches verschoben werden.

Um den Nutzerstandort zu bestimmen, ist aus vielen mobilen Anwendungen der Ortungbutton bekannt. Erlaubt der Nutzer dem Browser seine Position zu bestimmen, ermittelt eine Funktion aus der nutzerseitigen IP-Adresse diesen. Im *Geo Viewer Sachsen* steht diese Funktionalität nur im mobilen Layout zur Verfügung, in der Desktop-Version bleibt sie deaktiviert. Die anderen Anwendungen führen keine Unterscheidung durch und stellen die Ortung für alle Endgeräte zur Verfügung.

### 5.3.3.3. Kartenveröffentlichung und -weiterverwendung

Um eine digitale Karte zu veröffentlichen stellen alle untersuchten Anwendungen verschiedene Funktionen zur Verfügung. Eine Option bestand darin, selbst erstellte Karten als PDF drucken und auf der Festplatte des Nutzers zu speichern. Eine weitere Möglichkeit stellt die Generierung eines Links dar, womit Karten über soziale Netzwerke geteilt und hierdurch auch längerfristig gespeichert werden können. Diese Funktion ist in allen Anwendungen enthalten. Um mit eigenen Daten zu arbeiten, stellten alle Anwendungen eine Importfunktion zur Verfügung, wodurch vorgegebene Datenformate im Kartenviewer visualisiert werden können. Für die Erhaltung von gezeichneten und importierten Vektorkarten, wurde ebenfalls eine Exportfunktion integriert, hierbei variierte das Angebot der verfügbaren Ausgabeformate. Der *BayernAtlas* und *Geo Admin CH* boten erweitert die Möglichkeit an, eigens erstellten Karten in andere Web-Anwendungen zu integrieren. Hier sieht Sabine Henning u. Vogler [2011] einen wichtigen Vorteil von WebMapping-Anwendungen in der Gegenüberstellung mit digitalen Globen. Für diese Exportfunktionen bieten sich verschiedene Verfahren wie *widget-* oder *object* embedding an [vgl. Henning u. Vogler, 2011, S. 92]. Beide Anwendungen generierten einen *Link* welcher als *Object* integriert werden kann, ohne das eine Zeile Code notwendig ist. Auch in diesem Analysepunkt schränkt der *Geo Viewer Sachsen* seine Nutzer auf dem Smartphone ein und bietet keinerlei Möglichkeiten, die angezeigte Karte zu teilen oder einen Export durchzuführen.

### 5.3.3.4. Technische Aspekte

Karten werden durch die Hilfe von AJAX schnell im Hintergrund geladen, ohne ein Neuladen der gesamten Seite zu fordern. Im *Geo Viewer Sachsen* wird in der mobilen Anwendung nach jeder Kartenwahl ein Neuladen der Anwendung erzwungen, was in Zeiten moderner Web Technologien unnötig ist und die Nutzerfreundlichkeit stark beeinträchtigt. Alle Anwendungen setzen auf moderne Kartenframeworks, welche den Clienten stärker in die Verteilung der Visualisierungslasten einbeziehen.

### 5.3.4. Zusammenfassung

Die Untersuchung demonstrierten deutlich wie eine Anwendung im Web 2.0 realisiert werden kann. Mit der Hilfe von AJAX wurden Funktionalitäten umgesetzt, welche denen von Desktop-GIS sehr nahe kommen. Die Interaktion war sehr flüssig und zeigte die vielfältigen funktionalen Interaktions-Möglichkeiten moderner WebMapping-Anwendungen. Plewe [2007] ordnet Anwendungen welche ein dynamisches, stufenloses Zoomen und Browsen in Satelliten- und Kartendaten ermöglichen, der „vierten Generation von WebMaps“ zu. Diese Zuordnung kann für alle untersuchten Anwendungen getroffen werden. Auch die Forderung des Web

2.0 nach einer Geräteunabhängigkeit konnte bei allen analysierten Anwendungen erfolgreich umgesetzt werden. Hier spielte vor allem die Responsivität der Benutzeroberflächengestaltung eine wichtige Rolle, welche sich an die Display Parameter anpasst. Die Anwendungen demonstrierten in der Analyse deutlich, dass auch mobiles WebMapping mit einem hohen Interaktionsgrad möglich ist.

Während der Recherche nach geeigneten Anwendungen wurde festgestellt, dass eine Unterstützung von mobilen Endgeräten in sehr wenigen WebMapping-Anwendungen ähnlicher Natur eine Realisierung fand. Durch die rasante Entwicklung des Smartphone Marktes ist in diesem Bereich zukunftsorientiert sicherlich ein weiterer Entwicklungsschub zu erwarten.

|   | <b>Geo Viewer Sachsen</b>  | <b>Bayern-Atlas</b>  | <b>Geo Admin CH</b>                       | <b>Geo Online</b>                    |
|---|--|--|---|--------------------------------------|
| <b>Benutzeroberfläche</b>                           |  |  |   |                                      |
| Responsivität                                       | Weiterleitung in eine neue Oberfläche, mit eingeschränkten Funktionsumfang | Responsive Webdesign, mit vollem Funktionsumfang           |   |                                      |
| Menü  | Slider   | Accordion im Dropdown-Menü                                 |   |                                      |
| <b>Kartengestaltungs- und Analysemöglichkeiten</b>  |  |  |   |                                      |
| Werkzeuge   | Zeichenwerkzeug, Messen, Sachdatenabfrage                                  | Zeichenwerkzeug, Messen, Vergleichsmodus, Sachdatenabfrage | Flurstückssuche, Messen, Zeichenwerkzeuge |                                      |
| Räumliche Navigation                                | Zooming, Panning, Übersichtskarte  |  |   |                                      |
| Karten-gestaltung                                   | Wechsel der Hintergrundkarte   |  | Wechsel der Hintergrundkarte, 3D Modus    | Wechsel der Hintergrundkarte         |
| <b>Kartenveröffentlichung und -weiterverwendung</b> |  |  |   |                                      |
| Import  | nur mit Registrierung  | KML\GPX, WMS   | KML\GPX, WMS                              | KML                                  |
| Export  | PDF, GIF, JPG, PNG, KML, GPX, Link   | KML, GPX, PDF, Link, URL-Objekt                            | KML, GPX, PDF, Link, URL-Objekt           | PDF, KML                             |
| <b>Technische Aspekte</b>                           |  |  |   |                                      |
| Karten-framework                                    | ArcGIS API mit DOJO  | OpenLayers 3   |   |                                      |
| Performance   | Desktop sehr flüssig, mobil ausreichend                                    | sehr gut   |   |                                      |
| <b>Hilfe</b>  |  |  |   |                                      |
| Hilfe   | nein   | ja, durch externe Seite                                    | ja, durch externe Seite                   | ja, durch interne Funktionsübersicht |

Tabelle 4: Zusammengefasste Ergebnisse der Bestandsanalyse

## 5.4. Vergleich der Ergebnisse mit der Implementierung des IÖR-Monitors

Um die generierte Resultate der Bestandsanalyse auf die vorliegende Version des IÖR-Monitors zu übertragen und damit erste Ansatzpunkte für Entwicklungspotenziale zu identifizieren, findet in diesem Kapitel eine Gegenüberstellung statt. Hier werden nach den gebildeten Kriterien der vorangegangenen Auswertung, die zusammengefassten Ergebnisse aus der Bestandsanalyse mit der aktuellen Implementierung des IÖR-Monitors verglichen.

### Benutzeroberfläche

Mit Abbildung 10 werden die Interfaces des *BayernAtlas* als Vertreter für ein modernes WebMapping-Layout und des IÖR-Monitors gegenübergestellt.

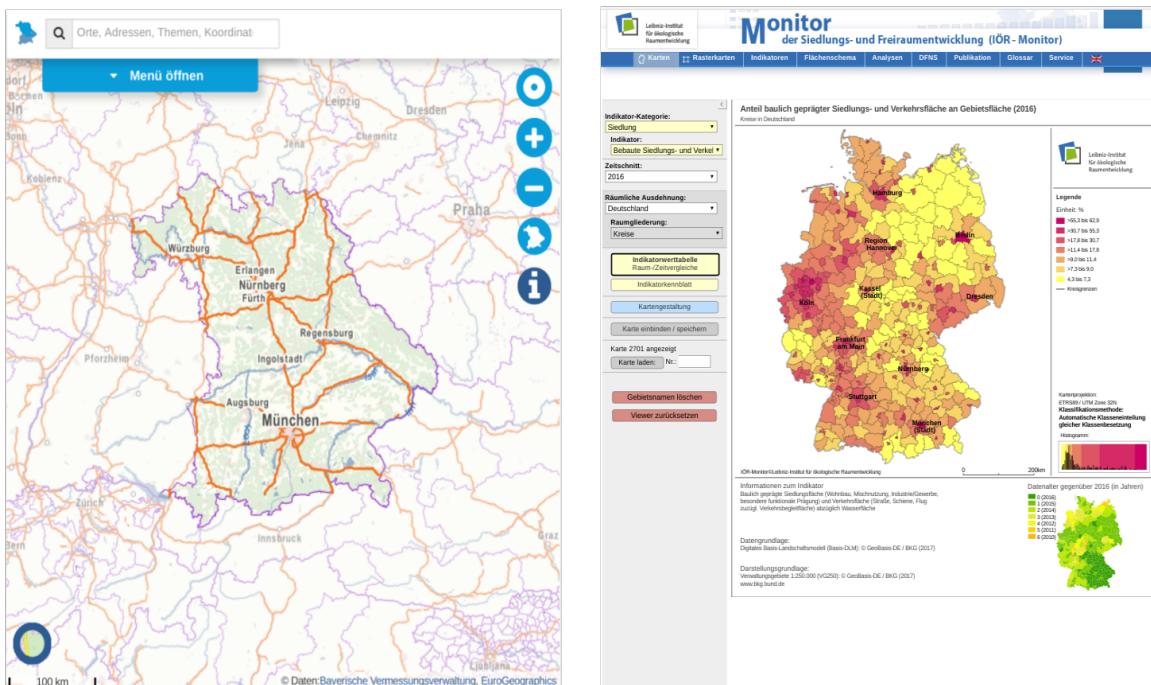


Abbildung 10: Vergleich des Interfaces zwischen IÖR-Monitor (rechts) und BayernAtlas (links). Quelle: [Bayerisches Staatsministerium der Finanzen, für Landesentwicklung und Heimat, 2017; Leibniz-Institut für ökologische Raumentwicklung, 2017b]

Anhand der Abbildung wird deutlich, wie in modernen Layouts die komplette Breite der Oberfläche für die Karte genutzt wird. Der IÖR-Monitor verwendet nur ein Drittel der Oberfläche für das wichtigste Element einer WebMapping-Anwendung, die Karte. Innerhalb der untersuchten Anwendungen wird versucht, den Nutzer größtenteils in der aktuellen Ansicht zu halten und Weiterleitungen zu vermeiden. Der IÖR-Monitor geht hier andere Wege und leitet den Nutzer für Einstellungen und externe Inhalte in ein neues Fenster. Schon im Jahre 1999 setzte Jakob Nielsen diese Art der Nutzerführung auf den zweiten Platz seiner „*The Top 10 Web Design Mistakes of 1999*“ [Nielsen, 1999] und sollte unbedingt vermieden werden. Beispielsweise wird die Tabellenansicht der Indikatorenwerte des gewählten Indikators, in einem neuen Fenster geöffnet. Hier prangert Nielsen an, dass die Website dem Nutzer diktiert, wie sich der Desktop zu verhalten hat. Die Anwendungen aus der Benutzeranalyse haben gut gezeigt, wie durch interne Fenster Weiterleitungen vermieden werden können.

Betrachtet man Abbildung 10 genauer, sind die gegensätzlichen Designansätze der Navigation ersichtlich. Auf der linken Seite steht die Anwendung aus der Bestandsanalyse, welche eine Navigation besitzen, in der sich nicht benötigte GUI-Elemente abschalten lassen und auf der rechten Seite der IÖR-Monitor, wo dem Nutzer viele Möglichkeiten sofort dargestellt werden und nur bedingt die Möglichkeit der Ein- und Ausblendung Verwendung findet. Lawrence u. Tavakol [2007] warnen im Falle des gewählten Designs im IÖR-Monitor vor einer Überfrachtung der Inhalt. Sie favorisieren eine, wie auch von [Hoffmann, 2013] geforderte Reduktion der Inhalte durch Nutzer gesteuerte Ein- und Ausblendefunktionalitäten. Die Anwendungen aus der Bestandsanalyse setzen ausnahmslos auf ein sogenanntes „*Clean Webdesign*“. Ein Layout das durch intuitive interaktive GUI-Elemente in der Navigation die Inhalte auf das Wesentliche reduziert und eine hohe Navigationstiefe besitzt [vgl. Post, 2017].

Eine Responsivität ist im IÖR-Monitor nicht vorhanden, wodurch die Forderung an eine WebMapping 2.0 Anwendung fehlt, geräteunabhängig zu sein.

Das wichtige Feature einer Suchfunktion integriert der IÖR-Monitor nur für Orte und bietet damit keine Möglichkeit nach Indikatoren zu recherchieren. Eine unterstützende Autovervollständigung fehlt bei der Suche, wohingegen diese Funktionalität bei den Anwendungen der Bestandsanalyse einen hohen Mehrwert zeigte.

Ein Plus des IÖR-Monitor ist die Integration von Zustandsanzeigen in die Kartenansicht, welche dem Nutzer das Laden der Seite signalisieren. Hier unterscheidet sich die Anwendung des Leibniz-Institutes von den Ergebnissen der Bestandsanalyse, wo dieses Feature nur vereinzelt vorhanden war.

### Kartengestaltungs- und Analysemöglichkeiten

Der Detailviewer des IÖR-Monitor bietet dem Nutzer, ähnlich der verglichenen Anwendungen, gute Interaktions- jedoch geringe Gestaltungsmöglichkeiten der Karte an. Die Farbwahl des visualisierten Indikators ist als gegeben zu betrachten, jedoch kann die Sichtbarkeit über einen Tranzparenzregler eingestellt werden. Mit dem verwendeten OpenLayers 2 Kartenframework wurde die räumliche Standard-Navigation in der Karte durch *panning* und *zooming* realisiert.

Gegensätzlich verhält sich der Übersichtsviewer des IÖR-Monitors. In diesem Teil wurde auf die SVG-Technologie gesetzt, welche eine Ausstattung mit sehr geringen kartographischen Interaktionsmöglichkeiten fand. Ein Hineinzoomen oder Verschieben der Kartensicht mit einem Zeigegerät oder dem Finger ist nicht möglich. Der Nutzer kann den Darstellungsbereich mit der Auswahl von administrativen Gebietseinheiten beeinflussen. Für die Selektion von möglichen Darstellungsbereichen, steht im linken Navigationsbereich ein Dropdown-Menü bereit. Interaktionsmöglichkeiten zwischen Karten- und Tabellenansicht können nur durch wechselseitiges neu laden genutzt werden. Beim Überqueren einer SVG-Gebietseinheit mit der Maus (Mouseover) werden dem Nutzer Informationen zum Indikatorwert und zur Aktualität der Datengrundlage geliefert. Analysewerkzeuge, wie beispielsweise das Messen von Strecken und Flächen, sind nicht implementiert. Eine Grundkarte ist nur im Detailviewer vorhanden, welche jedoch nicht ausgetauscht werden kann.

### Kartenveröffentlichung und -weiterverwendung

Um eine erstellte Karte zu teilen, bietet der IÖR-Monitor einen ähnlichen Funktionsumfang wie die untersuchten Anwendungen. Es können Karten gespeichert und gedruckt werden. Des Weiteren ist auch ein Export in verschiedene Bildformate möglich. Ein Import von nutzereigenen Daten ist nicht implementiert.

### Technische Aspekte

Bei der Verteilung der Anwendungslogik unterscheiden sich die Viewer des IÖR-Monitors. Die Vektorkarten basieren zum größten Teil auf einem serverseitigen Teil, wohingegen die Rasterkarten clientenseitig realisiert wurde. Die Interaktion mit der Rasterkarte gestaltet sich nicht so flüssig wie bei den untersuchten Anwendungen. Bei dem Versuch alle Gemeinden als Vektorgrafik zu laden, kam es wiederholt zu Abstürzen der Seite.

## 5.5. Ableitung von Entwicklungspotentialen

Im Zusammenspiel mit der vorangegangenen Analyseauswertung werden in diesem Kapitel mögliche Entwicklungspotentiale des IÖR-Monitors erörtert und dokumentiert indem Anforderungen an die zu entwickelnde Anwendung abgeleitet werden. Diese Kriterien basieren auf den Ergebnissen der vorangegangenen Analysen und sollen die Basis der folgenden Konzeptionierungsphase sein.

### Bedienung und Nutzerunterstützung

- Viele Probanden aus der Benutzeranalyse empfanden die Trennung zwischen Raster- und Vektorviewer als sehr unverständlich, demzufolge eine **Fusion der beiden Darstellungsarten** in der neuen Anwendung realisiert werden sollte.
- Die Forderung von O'Reilly nach einer Endgeräteunabhängigkeit ist ein wichtiges Kriterium, an eine WebMapping 2.0 Anwendung [vgl. O'Reilly, 2017]. Weshalb die neue Anwendung **Responsive** zu realisieren ist.
- Die untersuchten Anwendungen haben sehr gut demonstriert, wie eine **übersichtliche Navigation** implementiert werden kann. Durch den Entwurf und der Realisierung eines neuen Layouts soll der Nutzer schneller unterscheiden können, welche Informationen der Anwendung relevant bzw. irrelevant für ihn sind.
- Durch die Forderung von Nielsen [1999], nach einer maximalen **Minimierung von Weiterleitungen** in Web-Anwendungen, muss dies auch in der zu entwickelnden Anwendung berücksichtigt werden.
- Ein wichtiges Ergebnis der Benutzeranalyse stellte die Forderung nach einem schnelleren Zugriff auf die gewünschten Indikatorkarten dar. Für diese Aufgabe boten die untersuchten Anwendungen eine **Suchfunktion mit Auto vervollständigung** an. Dieses Feature kennen viele Nutzer aus der Verwendung von Suchmaschinen im Internet. Für

die Suchfunktionalität muss in der neuen Oberfläche ein zentraler Platz reserviert werden, welcher dem Nutzer sofort auffällt. Mit der Verbindung von Indikatoren mit räumlichen Gebietseinheiten (z.B. Kreise) muss die Suche mit Autovervollständigung auch auf Ortsnamen ausgeweitet und implementiert werden.

### Kartengestaltungs- und Analysemöglichkeiten

- Mit dem gesetzten Kriterium, die Weiterleitungen innerhalb der Anwendung zu minimieren, muss eine **interaktive Vereinigung der Tabellenansicht und Vektorkarte** erfolgen. Im jetzigen Zustand wird die Tabelle in einem neuem Tab geöffnet und dem Nutzer keine Optionen zur wechselseitigen Interaktion zwischen den Ansichten geben, ohne ein neu laden durchzuführen. Mit einer Integration der Tabellen- in die Kartenansicht, soll der räumliche Bezug zu den tabellarischen Werten hergestellt und durch Einführung von Interaktionen die Analysemöglichkeiten erhöht werden.
- Viele potentielle Nutzer nannten in der Benutzeranalyse eine vereinfachte **Erstellung von eigenen Gebietseinheiten** als wichtiges zukünftiges Tool. Diese Möglichkeit darf in der neuen Anwendung nicht fehlen. Die Einführung dieser Funktionalität muss auch auf den untersten administrativen Gebietsebenen (z.B. Gemeinden) möglich sein und deren Abfragen mit der Tabellenansicht synchronisiert werden.
- Die fehlenden intuitiven räumlichen **Navigationsmöglichkeiten**, durch *zooming* und *panning* im Übersichtsviewer, müssen im Rahmen des Redesigns implementiert werden. Für die geplante Kartenausdehnung auf den gesamten Bildschirm sind diese Standardinteraktionen unablässig, um sich in der Karte zu orientieren [vgl. Harrower u. Sheesley, 2005, S.79].
- Die analysierten Anwendungen der Bestandsanalyse boten dem Nutzer diverse **kartographische Werkzeuge** an, welche auch in der neuen Anwendung vorkommen sollen und somit die Interaktionsmöglichkeiten nutzerseitig steigern. Für die vorliegende Arbeit wird der Angebotsumfang auf übliche Funktionen begrenzt. In der Konzeptionsphase wird eine Auswahl umzusetzender Werkzeuge stattfinden.
- Ein wichtiger Bestandteil von WebMapping 2.0 ist die **Anpassung der angezeigten Karteninhalte** an eigene Vorstellungen. In diesem Punkt soll eine Veränderungsoption für die Farbdarstellung der Rasterkarten möglich sein, welche in der Benutzeranalyse als Anmerkung geäußert wurde. Erweiternd bietet ein Wechsel der Hintergrundkarten eine variable Hervorhebung der visualisierten Indikatorwerte. Zur Realisierung bietet sich die Einbindung von unterschiedlichen API's externer Anbieter an.
- Durch die Existenz der Indikatoren in verschiedenen Zeitschnitten wäre eine Möglichkeit für die **Vergleichende-Analyse** wünschenswert. Hier zeigte der vom *BayernAtlas* und *Geo Admin CH* implementierte Vergleichsslider (Kapitel 5.3.3.2), eine Möglichkeit für die zeitliche Gegenüberstellung, indem der Nutzer für den linken und rechten Teil der Karte unterschiedliche Indikatoren wählen\visualisieren kann. Unterschiede zwischen den Pixelwerten der Visualisierungen werden durch das horizontale Bewegen des Sliders sichtbar, indem je nach Richtung Bereiche ein- oder ausgelendet werden.

- Der **Import von nutzereigenen Datenquellen** stellt eine Möglichkeit dar, eigene Daten mit denen des IÖR-Monitor zu kombinieren. Für diese Funktion sollen die Standardformate: GeoJSON, KML und GPX unterstützt werden.

## Technische Aspekte

- Die Geschwindigkeit der Indikatorvisualisierung ist im Übersichtsviewer auf der Gemeindeebene nicht performant. Hier muss eine Möglichkeit gefunden werden, den **Visualisierungsvorgang zu beschleunigen** und eine flüssige Darstellung während der Karteninteraktion zu erreichen. Die Grundlage bildet hier die Wahl eines geeigneten Kartenframeworks im folgenden Kapitel.

## 5.6. Analyse von geeigneten Open Source Kartenframeworks

Um die Wichtigkeit eines Kartenframeworks zu verdeutlichen, verwenden Michael Moore & Andy Walz einen interessanten Vergleich, indem sie die Kartenanwendung als Auto darstellen und das Framework als den Motor. Das Framework ist verantwortlich für das Laden und Darstellen der Karten und die Handhabung von Nutzerinteraktionen [Moore u. Walz, 2016]. Durch die rasante Entwicklung von Web Technologien stehen dem Entwickler zahlreiche Lösungsmöglichkeiten zur Verfügung. Neben den kostenpflichtigen Optionen wie beispielsweise die *ArcGIS API*<sup>14</sup> des Branchenprimus *Esri* existieren auch mehrere leistungsstarke Open Source Lösungen. In dieser Arbeit soll eine Open Source Lösung eingesetzt werden, wodurch die Wahl eines geeigneten Frameworks auf die zwei bekanntesten Vertreter begrenzt wurde. Folgend werden die beiden in Frage kommenden Frameworks hinsichtlich ihrer Relevanz und Einsetzbarkeit für die geplante Anwendung untersucht.

### 5.6.1. Leaflet

Mit der von Vladimir Agafonkin entwickelten Open Source JavaScript-Framework können kreative und interaktive Karten erstellt werden. Neben den älteren WebGIS-Frameworks stellt Leaflet<sup>15</sup> einen „Newcomer“ dar. Vladimir Agafonkin entwickelte Leaflet 2010 bei der Firma *CoudMade*<sup>16</sup> und 2011 wurde die erste Version veröffentlicht [Derrough, 2013]. Die Entwicklergemeinschaft um das Framework herum hat in den letzten Jahren zahlreiche Open Source Plug-ins entwickelt, welche auf der Website des Projektes<sup>17</sup> gelistet sind. Die aktuelle Version ist Leaflet 1.2.0. welche auch in diesem Vergleich verwendet wurde.

### 5.6.2. OpenLayers

OpenLayers ist ein Open Source Projekt, welches von einer großen Entwicklungsgemeinschaft getragen wird. Der Entwicklungsprozess wird von einem derzeit achtköpfigen Projektkomitee überwacht. Das Framework versucht alle erforderlichen Dinge, die ein Entwickler für die Erstellung einer eigenen WebMapping-Anwendung benötigt, zur Verfügung zu stellen. Das

<sup>14</sup><https://developers.arcgis.com/javascript/>

<sup>15</sup><http://leafletjs.com>

<sup>16</sup><http://cloudmade.com/>

<sup>17</sup><http://leafletjs.com/plugins.html>

heißt, es werden nicht nur GIS verwandte Konzepte wie Karten, Ebenen oder Standardformate bereitgestellt, sondern auch manipulative Dokumentenelemente, um auch asynchronen Anforderungen gerecht zu werden [Perez, 2012]. Die Grundfunktionen orientieren sich an den Funktionen der bekannten Google Maps Anwendungen und übertreffen diese in bestimmten Bereichen. OpenLayers basiert auf *Google Closure Tools*<sup>18</sup>, wobei es sich um umfangreiche JavaScript Werkzeuge handelt, welche eine hohe Codequalität sichern. Aktuell ist OpenLayers in der Version 4.3.1 zu bekommen.

### 5.6.3. Vergleich

Leaflet und auch OpenLayers stellen zwei leistungsstarke Frameworks für das WebMapping dar, welche jedoch Unterschiede aufweisen. Leaflet besitzt eine einfache und leicht zu verste hende API. OpenLayers stellt dem Entwickler eine sehr große und komplexe API zur Verfügung, mit der sich mehr Funktionen realisieren lassen können als mit Leaflet [ThinkGeo, 2015]. Dieser Unterschied wird vor allem bei einem Vergleich des JavaScripts deutlich, mit welchem die jeweilige Bibliothek eingebunden wird. Leaflet benötigt 6563 und OpenLayers 218.455 Codezeilen [Black Duck Software, 2017].

Der Einstieg wird dem Entwickler in Leaflet durch eine umfangreiche und verständliche Dokumentationen mit vielen Beispielen, leicht gemacht. Während der Recherche dieser Arbeit traten in OpenLayers viele Tutorials auf, welche nicht funktionierten und weiterführende Fehler-Analysen erforderten. In Leaflet konnten alle Tutorials ohne Probleme ausgeführt werden, besaßen jedoch nur einen Umfang von 5 Beispielanwendungen, wohingegen OpenLayers 125 anbietet.

Viele Benutzer des IÖR-Monitors kommen aus der Verwaltung, wo der *Internet Explorers 8* häufig eingesetzt wird [Förtsch, 2017], welchen Leaflet problemlos unterstützt. OpenLayers in seiner aktuellen Version hat den Support für diesen Browser eingestellt [Sanchez, 2015]. Die Entwicklung von Leaflet begann mit dem Hintergedanken, auch von Beginn an mobiles Mapping zu unterstützen, wofür interaktive Steuerelemente integriert wurden, die speziell nach dem Post-WIMP (Windows, Icons, Menüs, Zeiger) Schnittstellen konzipiert wurden [Vgl. Muehlenhaus, 2014, S.225]. Auf dieser Grundlage kann eine exzellente Unterstützung mobiler Endgeräte erfolgen. Um konsistenten Code zu generieren, bietet OpenLayers mit dem integrierten *Google Closure Tools* eine gute Grundlage, welche das notwendige JavaScript Dokument automatisch erstellt. Auch im Funktionsumfang steht Leaflet hinter OpenLayers, da hier viele Funktionen nur über Plugins eingebunden werden können, welche je nach Version unterschiedlich unterstützt werden [Sanchez, 2015].

Durch die vorhandene Verbreitung des *Internet Explorers 8* bei potentiellen Nutzern des IÖR-Monitors muss dieser von dem zu verwendenden Framework unterstützt werden. Durch die begrenzte Bearbeitungszeit dieser Arbeit sollte auch die Einarbeitungszeit in einen verträglichen Rahmen fallen. Mit der geplante Geräteunabhängigkeit der Anwendung, muss die zu verwendende Bibliothek auch ein mobiles Layout unterstützen. Leaflet bietet durch eine gute Dokumentation und einfach gehaltene API einen schnellen Einstieg. Des Weiteren

<sup>18</sup><https://developers.google.com/closure>

vereint das Framework eine exzellente Unterstützung mobiler Endgeräte und des *Internet Explorers 8*. Diese Gründe führten dazu Leaflet in dieser Arbeit einzusetzen.

### 5.7. Fazit

Mit der Filterung und Formulierung der Ansprüche an die neue Anwendung des IÖR-Monitors konnte in diesem Kapitel ein wichtiger Grundstein für die folgende Konzeptionsphase gelegt werden. Die Bestandsanalyse zeigte mögliche Lösungsansätze, um eine WebMapping-Anwendung zu realisieren, welche den Ansprüchen des Web 2.0 gerecht wird. Anhand der generierten Ergebnisse und einer Analyse des aktuellen IÖR-Monitors formten sich die formulierten Entwicklungspotentiale ab, welche anschließend erörtert wurden.

Anhand der Rahmenbedingungen konnte ermittelt werden, dass bereits eine leistungsfähige Serverinfrastruktur zur Speicherung und Abfrage der notwendigen Informationen vorhanden ist. Diese wird im weiteren Teil der Arbeit angepasst eingesetzt. Leaflet bildet in der neuen Anwendung den Motor als Kartenframework, um eine performante und ansprechende Visualisierung der Indikatoren auf jedem Endgerät zu ermöglichen.

## 6. Konzeption und Design

Nachdem in der Anforderungsanalyse die Entwicklungspotentiale des IÖR-Monitors gefiltert wurden, wird auf dieser Basis in diesem Kapitel entwickelt, was genau dem Nutzer später wie präsentiert werden soll. Dies betrifft vor allem die Komplexität und den Umfang der Funktionen sowie Aufgaben und Inhalte. Auf der Basis gegebener Rahmenbedingungen wird ein Design entwickelt und die einzelnen geplanten Funktionen erläutert.

### 6.1. Benutzeroberfläche

Die Benutzeroberfläche ist die eigentliche Schnittstelle zwischen einer Software und deren Benutzenden. Im Designprozess ist es deshalb von enormer Wichtigkeit, der Gestaltung genügend Gewicht zu geben.

Das folgende Kapitel geht auf die Elemente der Benutzeroberfläche und deren Gestaltung ein. In dem Gestaltungsprozess sollen die Inhalte des IÖR-Monitors in einer neuen Oberfläche integriert werden. Hierbei ist es wichtig, die in Kapitel 5.5 gefundenen Entwicklungspotentiale hinsichtlich der Benutzeroberfläche zu fokussieren. Bei der Gestaltung wird ein möglichst schlichtes und einfach zu bedienendes Design angestrebt. Das Layout in Abbildung 11 besteht je nach Darstellungsart aus mehreren Bereichen, welche je nach Funktionalität hinzugefügt oder ausgeblendet werden.

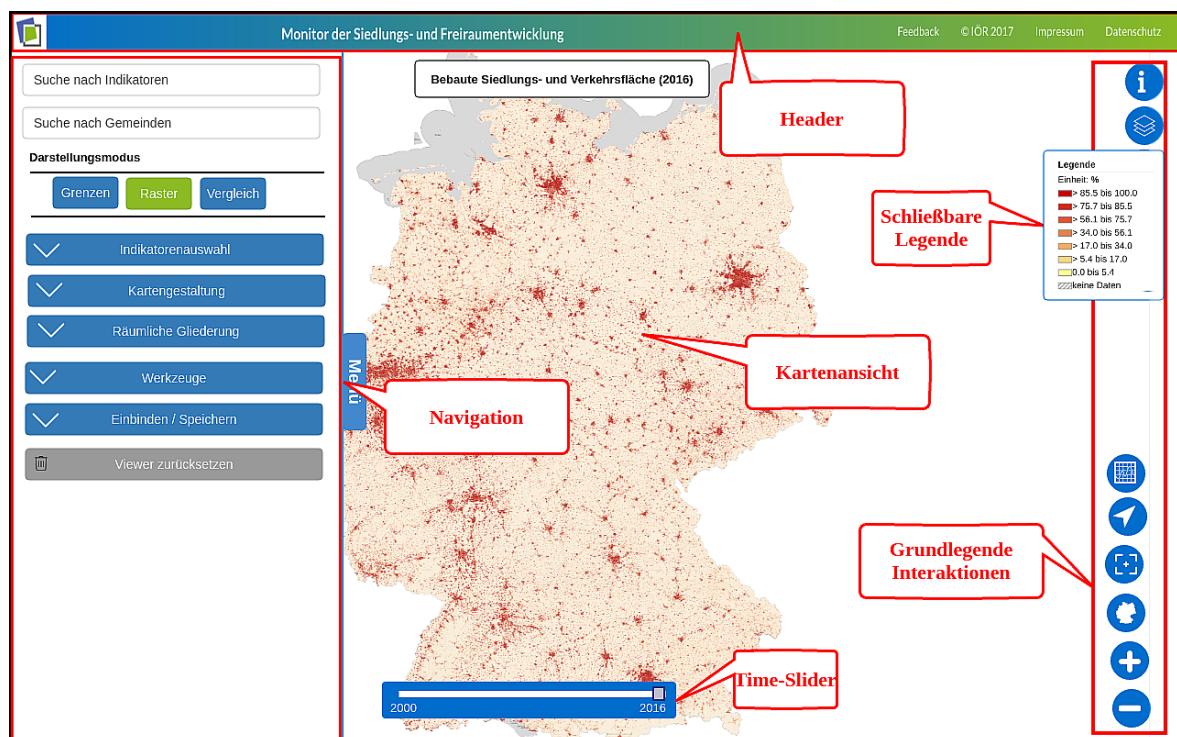


Abbildung 11: Entwurf der Benutzeroberfläche

Im Designprozess war es sehr wichtig eine intuitive Benutzbarkeit durch den Nutzer zu erreichen. Die Oberfläche wurde aus diesem Grund übersichtlich strukturiert und bewusst einfach gestaltet. Die Startseite gliedert sich in ein ein- und ausfahrbares Navigationsmenü, einen Header und Kartenansicht. Die Gestaltung entspricht der Designrichtung des *Clean-Webdesign*, welche die angezeigten GUI-Elemente nur auf das Nötigste reduziert, aber eine

hohe Navigationstiefe besitzt [vgl. Post, 2017].

Das Navigationsmenü und auch die Karte werden bei allen Ansichten beibehalten. Der konzeptionelle Entwurf sieht dabei vor, die Weiterleitungen so minimal wie möglich zu halten und damit die von [Nielsen, 1999] gesetzte Forderung umzusetzen. Vor allem die Regel „weniger ist mehr“ für den Informationsgehalt, als auch Wahlmöglichkeiten bezüglich Funktionalitäten und Interaktionsmechanismen, spielte bei der Konzeption eine große Rolle.

Um einen Entwurf aller Layouts zu erstellen, wurde das Open Source Produkt *Pencil*<sup>19</sup> verwendet. Mit dieser Software kann ein authentisches Modell der Oberfläche erstellt werden.

Der farbliche Gestaltung wurden durch die Vorgaben des Instituts Grenzen gesetzt. Die Farbtöne Blau und Grün bildeten hierbei die Grundfarben. Auf grelle Farben und unnötige Gestaltungselemente wird verzichtet, so dass ein schlichtes Design entsteht. In ihrer Checkliste an Farben für Websites fordert Hoffmann [2013] die Kolorierung eines Projektes auf ein Minimum zu begrenzen, weshalb in dieser Arbeit der Hauptanteil der Farbgebung dem „IÖR“ Blauton zu kam. Der Grund dieser Wahl als Grundfarbe kommt der Wirkung dieses Farbtons zu, welcher subtile Vertrauen, Zuverlässigkeit und Stabilität vermittelt [vgl. Meidl, 2014, S.22]. Die Farbe Grün wurde verwendet, um die Aktivität verschiedener GUI-Elemente zu visualisieren. Kommt es beispielsweise zu einer Aktivierung des Button für die Glättung der Karte (siehe Kapitel 6.2.4), wird die Hintergrundfarbe des aktvierten Elementes auf Grün gesetzt und somit den Nutzer signalisiert, dass eine Funktionalität aktiv ist.

Der Grünton wurde nicht ohne Grund gewählt, da im Webdesign verschiedene Studien gezeigt haben, das mit dieser Farbe gestaltete Elemente mehr Aufmerksamkeit bekommen [Hahn, 2017, S.405]. In den folgenden Kapiteln werden anhand des Konzeptes die wichtigsten Be standteile der WebMapping-Anwendung und deren zugrundeliegenden Designentscheidungen näher beleuchtet.

### 6.1.1. Navigation

*„Die Abbildung der Struktur als anklickbare Menüpunkte auf der Internetseite wird als Navigation bezeichnet. Sie ist das zentrale Element, da nur mit ihrer Hilfe die Inhalte erschlossen werden können.“* [aus Martin u. Richter, 2008, S. 13].

Dieses Zitat dokumentiert die zentrale Bedeutung der Navigation einer Anwendung und dementsprechend viel Aufmerksamkeit muss sie innerhalb des Gestaltungsprozesses bekommen. Das Navigationsdesign wurde aus dem von Ben Shneiderman entwickelten Prinzipien für das Design graphischer Benutzeroberflächen abgeleitet. Dabei legte er die Weise fest, wie ein Nutzer mit der Oberfläche interagiert, indem er mehrere Teile bildete: Übersicht, Zoom und Filter, Details auf Nachfrage [Shneiderman, 1996]. Der Benutzer erhält einen Überblick (*overview*) durch die Gesamtansicht des Navigationsmenüs. Die Auswahl von interessanten Bereichen (*zoom*) wird durch das Dropdown-Menü möglich. Filtern kann er gewünschte Bereiche durch integrierte Auswahlmechanismen, wie beispielsweise die Wahl des gewünschten Indikators. Mit der Selektion wird die Funktion letztendlich ausgeführt.

Zu Beginn des Designprozesses bestand die Frage, ob ein vertikales oder horizontales Menü besser geeignet ist, um die Inhalte der zu konzipierenden Website aufzunehmen. Für die

<sup>19</sup><https://pencil.evolus.vn>

Entwicklung des Konzepts wurde eine Studie von [Schnürer u. a., 2015] zur Inspiration herangezogen. Diese Studie untersuchte vier unterschiedliche GUI-Designs einer WebMapping Anwendung. Hierbei wurden fünf verschiedene „Mock-up“ Designs erstellt und evaluiert. Um die Usability zu messen, mussten die Probanden diverse Aufgaben an den Prototypen durchführen, als Indikatoren dienten dabei die Anzahl der Mausklicks und die Zeit, welche für das Lösen der Aufgabe notwendig war. Ein Layout dieser Studie wurde nach dem Vorbild von Google Maps erstellt, mit einem seitlich ausfahrbaren Navigationsmenü. Ein anderes GUI-Design basierte auf einer ähnlichen Gestaltung wie die Anwendungen aus der Bestandsanalyse (siehe Kapitel 5.3), mit einer Navigationsleiste, welche sich von oben nach unten dynamisch ausfährt und den Inhalt freigibt.

Googles Ansatz erzielte in der Studie die besten Ergebnisse, gefolgt von dem Layout der Bestandsanalyse-Anwendungen. Auch Martin u. Richter [2008] empfehlen eine vertikal angeordnete Navigation, um umfangreiche Inhalte darzustellen. Das Menü sollte laut „*Usability-Guru*“ Jacob Nielsen immer links platziert werden. Dies begründet er mit der Ursache, dass eine linksbündig platzierte Navigationsleiste leichter für das menschliche (westeuropäische) Auge zu scannen ist. Auf diese Weise kann in einer Linie nach unten geblickt werden, während es sich bei einer anderen Formatierung immer wieder neu orientieren muss [Nielsen, 2008]. Diese Gründe führten zu der Entscheidung die Navigation linksbündig und dynamisch von links einblendbar zu gestalten. Die Inhalte der bestehenden Website sollten dabei nicht verändert werden, weshalb sie kategorisiert und nach der Forderung von Hoffmann [2013] ein- und ausblendbar gestaltet wurden.

Anhand eines Accordion fand eine Realisierung dieser Forderungen statt, welches die Kategorien in Listen organisiert. Per Klick auf den Titel wird der Inhalt eingeblendet. Die Inhalte der anderen Einträge bleiben versteckt. Um den Nutzer diese Möglichkeit zu verdeutlichen, wurden die Buttons durch Pfeile ergänzt, welche bei verborgenem Inhalt nach unten zeigen und bei ausgefahrenem nach oben. Mit dem Einsatz des dynamisch ausfahrbaren Menüs sind die Navigationsmöglichkeiten übersichtlich dargestellt, auf welche schnell zugegriffen werden kann. Durch das Ein- und Ausblenden der Navigation, wird der Karte mehr Platz gegeben und die Übersichtlichkeit erhöht.

Der inhaltliche und funktionale Aufbau der Navigation bleibt bei allen Darstellungsarten (Raster, Gebiete) gleich, jedoch müssen Adaptionen der Menüinhalte erfolgen. Auf diese dynamischen Anpassungen wird im folgenden Kapitel 6.2 genauer eingegangen.

### 6.1.2. Kartenviewer

Um einen schnellen Zugriff auf die wichtigsten Interaktionen mit der Karte zu ermöglichen, wurden dafür Buttons vorgesehen. Damit diese bei aufgeklappter Navigationsleiste weiterhin sichtbar sind, fand eine vertikale Anordnung am rechten Rand statt. Durch die Lage der Buttons auf der Karte soll das Gestaltungsgesetz der Nähe ausgenutzt werden, welches besagt: das eine räumliche Nähe von Elementen dazu führt, dass Informationen als zusammengehörig wahrgenommen werden [vgl. Preim u. Dachselt, 2010, S.56].

Das Design der Buttons wurde stark an das von Google entwickelte „*Material Design*“<sup>20</sup> angelehnt. Dieses setzt Ränder und Flächen als bekannte visuelle Strukturen voraus, die den

<sup>20</sup><https://material.io>

Nutzer das Design auf physischer Ebene und auch auf der Sinnebene direkt verstehen lässt [Google, 2017]. Dieses Design der Buttons und auch des Zeit-Sliders soll dem Nutzer sofort erkennen lassen, mit welchen Bereichen er interagieren kann. Um den Inhalt der Funktionalität zu deklarieren, wurden aussagekräftige Icons verwendet, welche eine Selbstbeschreibungsfähigkeit besitzen sollen. Hierfür kamen Standard Icons zum Einsatz, welche kosten- und lizenzfrei von der Website *FLATICON*<sup>21</sup> bezogen wurden. Falls jedoch die Bedeutung der Icons unklar ist, unterstützen den Nutzer „Mouse-Over Texte“, welche die Funktionalität beschreiben. Der Vorteil dieser Vorgehensweise ist die Einsparung von Platz und die Erreichung einer Sprachunabhängigkeit, als wenn anstatt der Icons Texte verwendet werden [vgl. Stephan, 2010, S.363]. Um dem Nutzer in Kenntnis zu setzen, welchen Indikator er gerade visualisiert, wurde mittig im oberen Teil der Karte eine Überschrift gesetzt, welche den Indikator und räumliche Gliederung anzeigt.

Für einen intuitiven Wechsel der Zeitschnitte ist ein interaktiver Zeitlider in den unteren Bereich der Karte integriert. Mit diesem kann der Nutzer die zeitliche Dimension festlegen. Auch die bekannte Anwendung *Google Earth* <sup>22</sup> verwendet einen horizontal ausgerichteten Zeitslider, um eine Wahl des Zeitschnittes zu ermöglichen. In mehreren Usability Studien wurde die verbesserte Usability [vgl. Harrower, 2001] dieses Ansatzes bestätigt. Diese Erkenntnis führte zu der Entscheidung, dass das Dropdown-Menü der alten Anwendung durch den Slider ersetzt wird.

Um den Nutzer bei der Einstellung des gewünschten Zeitschnittes zu unterstützen, fordern Preim u. Dachselt [2010] eine verständliche Form des Vorgangs. Das bedeutet, dass der Benutzer zumindest grob abschätzen kann, was er bei einer bestimmten Eingabe zu erwarten hat. Für diesen Zweck wurden die Minimal- und Maximal-Werte der möglichen Zeitschnitte als Beschriftung ergänzt und eine Einblendung des gewählten Jahres über dem Slider integriert. Die folgende Abbildung 12 dokumentiert die Unterstützungen des Nutzers während der Slide-Interaktion.

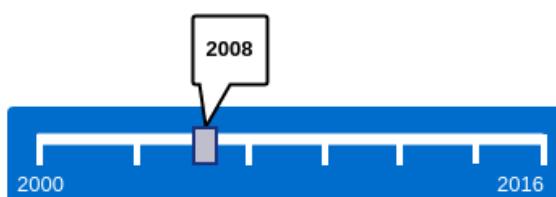


Abbildung 12: Zeitslider mit unterstützender Beschriftung

### 6.1.3. MultiView

Die Tabellenansicht wurde in der vorangegangenen Anwendung in einem neuen Tab geöffnet, wodurch der Nutzer in Hinsicht auf die Usability eingeschränkt wurde, da er zwischen den einzelnen Tabs hin und her schalten musste. Durch den starken Zusammenhang der Daten bietet sich eine Kombination der Ansichten in einem MultiView an. Diese Systeme kommen bereits seit langem auf Desktop-PC's zum Einsatz, zum Beispiel als Werkzeuge zur Dateiverwaltung, wo in verschiedenen Sichten die Laufwerke, Ordner und Dateien dargestellt werden. Ein Vor-

<sup>21</sup><https://www.flaticon.com/>

<sup>22</sup><https://www.google.com/intl/de/earth/>

teil dieses Systems ist die kognitive-Entlastung des menschlichen Arbeitsspeichers, weil die einzelnen Darstellungen weniger komplex sind und die Querbezüge zwischen zu kombinierenden Informationsobjekten nicht symbolisch oder grafisch repräsentiert werden, sondern interaktiv [Baldonado u. a., 2000]. In dieser Arbeit wird der MultiView durch eine horizontale Teilung des Bildschirmes erreicht. Bei dem ersten Start der Anwendung werden 70 % der Breite an die Kartenansicht vergeben und der restliche Anteil der Tabellendarstellung zugewiesen. Durch die Integration eines horizontalen Sliders besitzt der Nutzer die Möglichkeit, eine Veränderung der horizontalen Dimensionierung der einzelnen Ansichten nach seinen Vorstellungen vorzunehmen. Möchte der Nutzer die gesamte Karte zur Verfügung haben, ist die Tabellensicht abschaltbar gestaltet und kann bei Bedarf jederzeit eingeblendet werden. Für diese Funktionalität ist im rechten oberen Bereich ein Button integriert. Als Icon kommt ein Kreuz zum Einsatz, welches im Bereich des GUI-Designs als Standard Element zum Schließen von Ansichten steht. Ist die Tabellenansicht durch den Nutzer deaktiviert, wird ein Button in der Kartenansicht eingeblendet, welcher die Tabellenansicht wieder aktiviert. Mit diesen beiden Interaktionsbuttons wird dem Nutzer ein schneller Zugriff auf die Funktion zur Ein- und Ausblendung der Tabelle ermöglicht.

Innerhalb des Rasterviewers sind die Werte auf einzelne Pixelwerte bezogen, aus diesem Grund kann auf eine Tabellenansicht verzichtet werden. Die folgende Abbildung 13 stellt das graphische Konzept der Visualisierungstechnik dar, welches auch die in Kapitel 6.2.1 beschriebene Synchroniserungs- und hervorhebende Interaktion durch „Mouse Over“ darstellt.

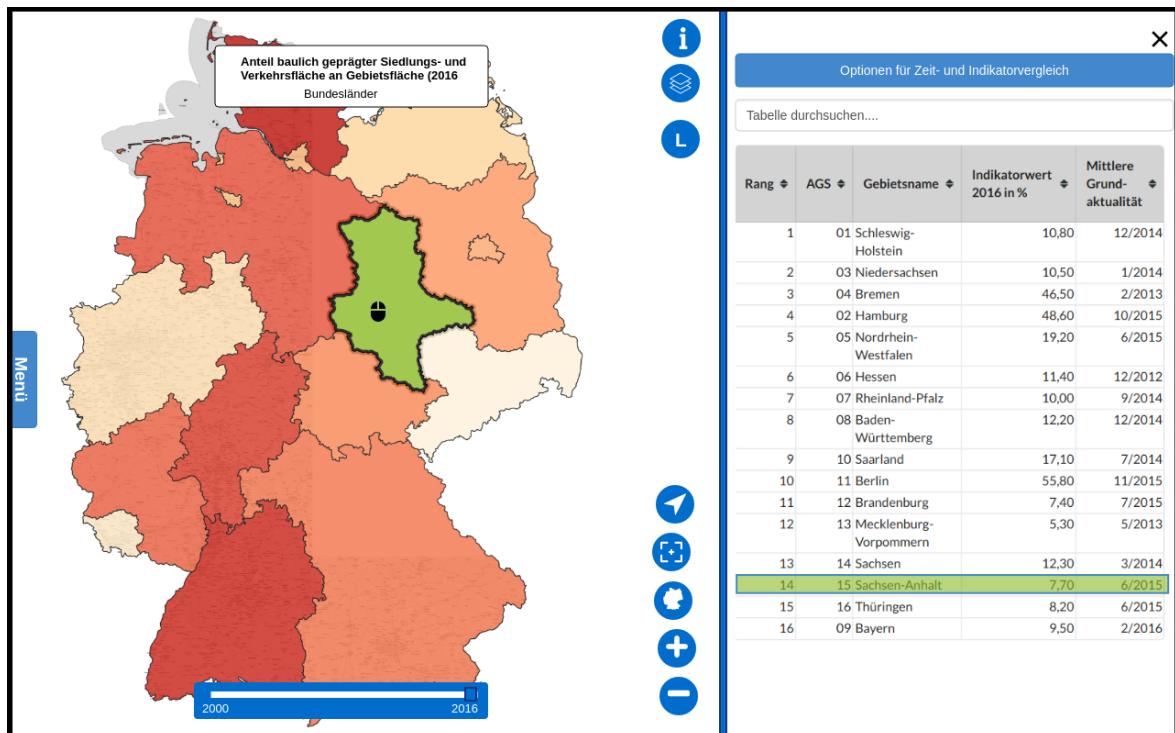


Abbildung 13: Geplante Multi View Ansicht ohne Header

Damit bei sehr langen Tabellen der Tabellenkopf sichtbar bleibt, wird dieser angeheftet und verschwindet damit nicht aus dem Sichtbereich des Nutzer, wenn er durch die Tabellenansicht scrollt. Hierbei bleibt dem Nutzer die Orientierung in der Tabelle erhalten und er braucht nicht zwischen dem Tabellenkopf und Tabelleninhalt hin- und herscrollen. Diese Praxis wurde

in mehreren UI-Studien als eine wichtige Funktionalität ermittelt, welche die Nutzererfahrung mit der Anwendung erhöht [Mun, 2014].

Für zukünftige Tabelleninteraktionen und deren Elemente wurde ein Drop-Down Menü in den oberen Bereich integriert, welches bei Aktivierung die Inhalte und deren Möglichkeiten freigibt. Auf diese erweiterten Funktionen zur Tabelleninteraktion wird in dieser Arbeit jedoch nicht eingegangen. Es wurde lediglich der oben beschriebene Platzhalter vorgesehen, welcher die zukünftigen Optionen aufnimmt.

#### 6.1.4. Responsive Layout

Durch die zunehmende Verbreitung von mobilen Endgeräten, müssen sich auch WebMapping-Anwendungen auf verschiedene Bildschirmgrößen zur Darstellung ihrer Inhalte auf eine flexible Art und Weise einzustellen.

Eine gute Lösungsansatz zur Überwindung dieser Herausforderung stellte das Responsive Webdesign dar (vgl. Kapitel 4.4). Mit diesem Ansatz und dessen Möglichkeiten wird die neue Anwendung auf mobile Endgeräte optimiert.

Eine erste Anpassung des Layouts muss im MultiView erfolgen, da für diesen Ansatz die zur Verfügung stehende Bildschirmbreite von Desktop-PC genutzt wird, welche durch die verdrehte Dimensionierung auf einem mobilen Endgerät nicht eingesetzt werden kann. Um diese Problematik zu lösen wird dem Nutzer ein Button angeboten, um zwischen der Tabellen- und Kartenansicht zu wechseln. Der MultiView wird in diesem Fall nicht verwendet, indem die zwei Ansichten separaten Fenstern zugeordnet werden.

Ein früherer Entwurf (Abbildung 14) mit einem horizontal angepassten Slider, welcher die Höhe des mobilen Enderätes nutzt, wurde wieder verworfen, da hierdurch der Karte zu wenig Platz gegeben werden kann. Im mobilen Layout wird die Tabellenansicht umstrukturiert, um sich an die gegebenen Bedingungen anzupassen. Eine Tabelle Responsive zu machen ist laut [Hahn, 2017, S.520] „*kein Vergnügen*“. Um dies doch zu realisieren wurde ein Ansatz von Sebastian Maier verwendet, in dem die Spalteninhalte untereinander dargestellt werden [Maier, 2010]. Für die Traversierung von der Tabellen- in die Kartenansicht, wird ein Button mittig am rechten Rand vorgesehen.

Dieser hebt sich durch eine Überlagerung der Tabelle ab und suggeriert dem Nutzer, dass er diesen nutzen kann, um zurück in die Karte zu gelangen. Durch die störende Überlagerung des Buttons beim Interagieren mit der Tabelle wird dieser beim Start einer Interaktion ausgeblendet und nach deren Ende wieder angezeigt. Der Navigationsbereich erhält ausgefahren 90% der Bildschirmbreite, um die Inhalte besser darzustellen.

In der Kartenansicht wurde eine Reduktion von nicht relevanten Interaktionsbuttons durchgeführt. Dieses Vorgehen beruht sich auf die Ergebnisse der in der Bestandsanalyse vorgestellten

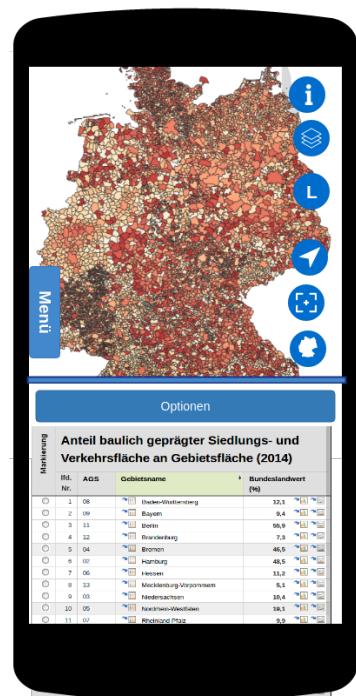


Abbildung 14: Erster Entwurf

Usability Studie, welche erfolgreich evaluierte, das Zoom-Buttons auf Touch-Displays redundant sind [Manlai u. a., 2007]. Der Branchenprimus Google setzt diese Erkenntnisse in seiner Anwendung *Google Maps*<sup>23</sup> ein, indem die Zoom-Buttons auf einem mobilen Endgerät nicht vorhanden sind. Auch Muehlenhaus [2014] stellt in seinem Buch die Verwendung von Zoom-Buttons infrage, hierbei führt er eine ähnliche Argumentation an. Diese Praxis wird auch in der neuen Oberfläche angewendet, um den Platz auf dem schmäleren Display optimal zu nutzen.

Der Header-Bereich bietet mit seinen Menüpunkten zu wenig Platz, um diese auf einem mobilen Endgerät vollständig darzustellen. Als Lösung werden die einzelnen Menüpunkte in einer Liste versteckt, welche bei Bedarf sichtbar geschaltet werden kann. Für diese Interaktion wird am rechten Header-Rand ein Button integriert, welcher bei seiner Aktivierung die inkludierten Inhalte horizontal einblendet.

## 6.2. Funktionale Beschreibungen der interaktiven Funktionen

Durch die wachsende Beliebtheit von WebMapping-Anwendungen und der rasanten Entwicklung von Web-Technologien, ist die Interaktivität zu einem fundamentalen Bestandteil in der Entwicklung und Gebrauchstauglichkeit von Kartenanwendungen geworden [vgl. Skupin u. Hagelman, 2005, S.160]. Auf Desktop-Anwendungen fungiert die Maus als indirektes Zeigegerät, auf Laptops kommen noch Touchpads und Trackpoints hinzu. Mit der wachsenden Menge an mobilen Endgeräten stieg auch die Anzahl, von auf eine direkte Touch Eingabe optimierten Applikationen. Dies kann als eine der direktesten Formen des Zeigens gesehen werden, da mit dem Finger unmittelbar auf den Bildschirm gezeigt wird [Preim u. Dachselt, 2015].

Das folgende Kapitel geht auf die wichtigsten Funktionen ein, die in der neuen Anwendung verfügbar sein sollen und welche anhand der Anforderungsanalyse (siehe Kapitel 5) gefiltert wurden. In alle erläuterten Funktionen soll die Geräteunabhängigkeit einbezogen werden und aus diesem Grund mögliche Interaktionen durch Touch oder indirekte Zeigegeräte Berücksichtigung finden. In diesem Kapitel werden nicht alle verfügbaren Funktionen erörtert, sondern nur die wichtigsten, welche sich von der vorangegangenen Entwicklung abheben und die neue Anwendung bereichern. Viele Funktionalitäten des IÖR-Monitors müssen nicht neu programmiert werden, sondern sind mit individuellen Anpassungen zu realisieren, weshalb sie in dieser Arbeit nicht weiter Betrachtung finden.

### 6.2.1. Interaktionen zwischen Karten- und Tabellenansicht

Durch den Einsatz einer MultiView-Benutzeroberfläche, müssen zur Herstellung eines Kontextes zwischen den Ansichten, interaktive Funktionen für den Nutzer bereitstehen.

Eine erste wichtige Funktionalität stellt die Synchronisierung zwischen Tabellen- und Kartenansicht dar. Hierbei bildet die Visualisierungstechnik des „Focus-and-Context“ eine sehr gute Realisierungsmöglichkeit. Diese Technik hebt die im Fokus liegenden Teile der Visualisierung hervor und blendet gegebenenfalls deren Details ein, ohne aber den Gesamtkontext

---

<sup>23</sup><https://maps.google.de/>

zu vernachlässigen [Schumann u. Kreuseler, 2003]. Selektiert der Nutzer in der Kartenansicht durch Hover ein entsprechendes Gebiet, wird dieses in der Tabellenansicht hervorgehoben. Invers funktioniert diese Funktion auch mit der Tabellenansicht, indem die selektierte Zeile des Gebietes in der Karte markiert wird. Dieser Zusammenhang zwischen der räumlichen Lage des Gebietes und dem dazugehöriger Tabellenwert, wird mit einer farblichen Betonung verdeutlicht. Hierdurch hebt sich das Paar von den anderen Daten ab und rückt diese in den Fokus des Nutzers. Der Grünton für die Rückmeldung von Funktionen (siehe Kapitel 6.1) wird auch für diese Hervorhebung verwendet.

Auf mobilen Endgeräten steht der Hover-Event nicht zur Verfügung, da Touch Geräte diesen nicht unterstützen. Hier hat der Benutzer keinen Zeiger, mit dem er auf der Anzeigefläche umherfahren könnte und es zur Auslösung des Hover kommt [vgl. Schlegel, 2013, S.350]. Hier übernimmt das Click- bzw. Touch- die Aufgaben des Hover-Events, welcher mit dem kartographischen Identify aus Kapitel 6.2.4 ausgelöst wird.

### 6.2.2. Vergleichsmodus

Ein Grundproblem der Visualisierung von raum-zeitlichen Daten ist es, einzelne Zustände sowie die Art und den Umfang der Veränderung einzufangen [Kraak u. MacEacren, 1994]. Hier können Werkzeuge dazu beitragen, dass Nutzer die Zusammenhänge oder Differenzen zwischen räumlichen und zeitlichen Ausprägungen schneller erfassen [Koussoulakou u. Kraak, 1992]. Um dem Nutzer ein Analysetool für den zeitlichen Vergleich von Indikatoren anzubieten, wird ein Vergleichsmodus in die Anwendung integriert. In der folgenden Abbildung 15 ist der geplante Aufbau der Funktion abgebildet.

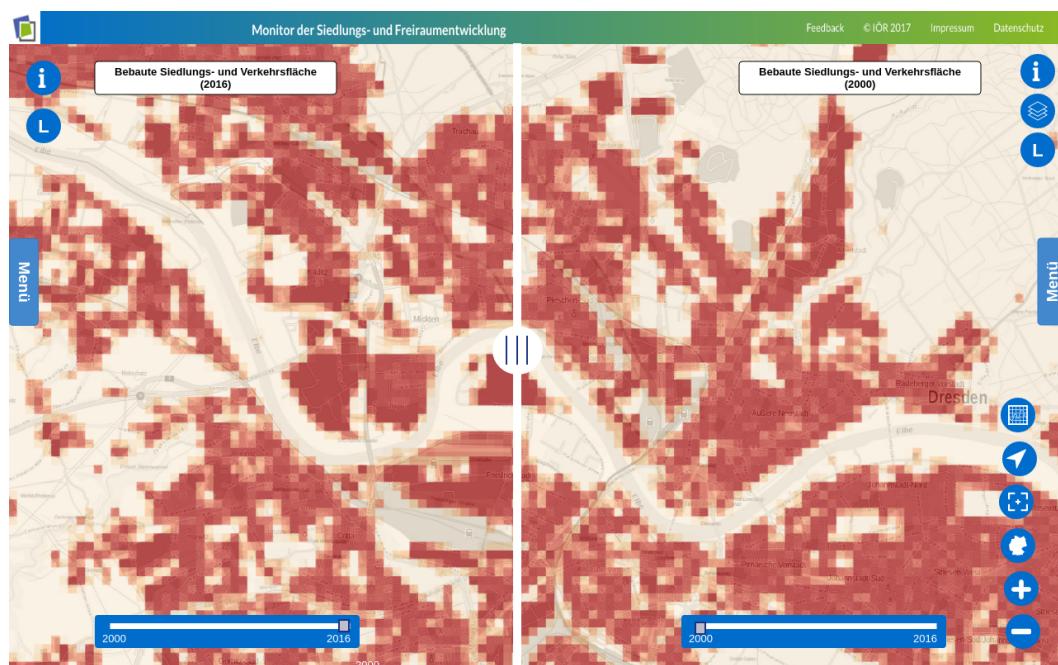


Abbildung 15: Konzept des Vergleichsmodus

Zur Realisierung wird die standardmäßige Kartenansicht um einen mittig platzierten Slider erweitert und ein rechtes Navigationsmenü hinzugefügt. Die linke Kartenansicht erhält eigene Legende- und Informations-Buttons des dargestellten Indikators. Der Slider teilt die Karten-

ansicht in zwei unabhängige Karten. Durch seine vertikale Beweglichkeit können Unterschiede analysiert werden, indem im jeweiligen Darstellungsbereich der Kartenansicht, Pixel ein- oder ausgeblendet werden.

Die beiden Navigationsmenüs dienen zur Kartenmanipulation der entsprechenden Seite. Mit Hilfe des linken Menüs kann die linksseitige Karte verändert werden und mit dem rechten die rechtsseitige Karte. Durch die beidseitigen Zeit-Slider erfolgt eine zeitliche Anpassung der jeweiligen Indikatorkarte.

Die zwei konzipierten horizontalen Zeit-Slider innerhalb des Vergleichsmodus, können durch ihren hohen Bedarf an Bildschirmbreite im mobilen Design nicht übernommen werden. Im Responsive Layout werden sie gedreht und dem Nutzer als vertikale Version angeboten. Der große Vorteil dieser Vorgehensweise ist die effektivste Ausnutzung der Bildschirmmaße auf einem mobilen Endgerät.

Um effizient den gewünschten Indikator der entsprechenden Seite hinzuzufügen, wird eine Indikator-Suche in jedes der beiden Navigationsmenüs integriert.

### 6.2.3. Suchen nach Indikatoren und Orten

Viele Nutzer wünschten sich innerhalb der Benutzeranalyse einen vereinfachten Zugriff auf einzelnen Indikatoren. Hier bietet die Möglichkeit der Suche einen guten Ansatzpunkt, um schnell den gewünschten Indikator zu finden und in der Karte darzustellen. Für die ortsbezogene Informationsfilterung muss weiterführend auch eine Suchfunktionalität diese Aufgabe erfüllen.

Um dafür einen effektiven Weg zu finden, gibt es zahlreiche Theorien und Modelle. Beispiele hierfür sind der Prozess der *Informationssuche* [Marchionini, 1997] oder das *5-Phase Framework* [Shneiderman u. Plaisant, 2010]. In allen Modellen wird die Suche als ein iterativer Prozess dargestellt, was bedeutet, dass die Suchanfrage nach und nach durch die Bewertung der erhaltenen Ergebnisse verfeinert wird, bis die relevante Information gefunden ist [Gundelsweiler u. Reiterer, 2008].

Zur Umsetzung der Indikatorsuche steht dem Nutzer ein Suchfeld zur Verfügung, welches das erste Element der Navigation darstellt. Wird mit der Eingabe begonnen, blendet die Autovervollständigungsfunktion passende Treffer ein. Dabei signalisiert ein Wechsel der Rahmenfarbe des Eingabefeldes, dass eine Funktionalität aktiviert wurde. Hierbei werden die Treffer passend zu der Indikatorkategorie in einer Liste gruppiert. Ein großer Vorteil dieser Funktionalität stellt die Vermeidung von Eingabefehlern dar, da bereits nach wenigen Eingaben passende Treffer eingebettet werden und eine Selektion durch den Nutzer erfolgen kann [Henzen u. Bernhard, 2013]. Um den Findungsprozess zu unterstützen, werden die passenden Treffer in der Ergebnisliste farblich hervorgehoben.

Während der Eingabe wird das Einblenden der Ergebnisse iterativ angepasst, da eine erhöhte Anzahl an vorgegebenen Schriftzeichen, die möglichen Ergebnisse eingrenzt. Hat der Nutzer den passenden Indikator gefunden, kann er diesen durch Anklicken in der Karte visualisieren. In der folgenden Abbildung 16 wird der Vorgang als Aktivitätsdiagramm dargestellt.

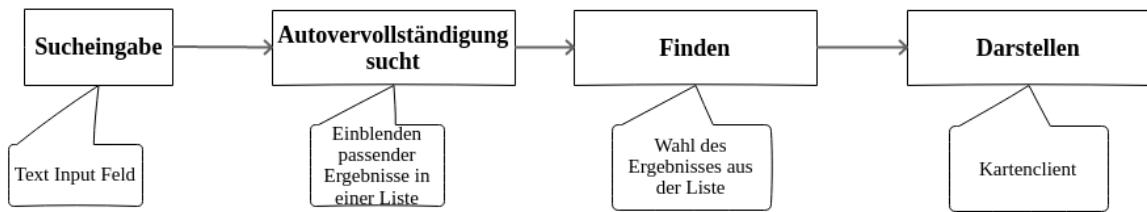


Abbildung 16: Aktivitätsdiagramm der Suche

Analog und nach dem gleichen Muster der Indikatorenrecherche, funktioniert die Ortssuche. Hier werden die Ergebnisse jedoch nicht gruppiert, sondern anhand passender Treffer eingebettet.

Visualisiert der Nutzer sehr viele Gebietseinheiten innerhalb der Kartenansicht des Gebietsmodus erscheinen synchron viele Einträge in der Tabellenansicht. Um nach dem „*Information-Seeking Mantra*“ [Shneiderman, 1996] eine bestmögliche Übersicht zu bekommen, bietet sich innerhalb der Tabellenansicht auch der Einsatz einer Suchfunktionalität an. Für diesen Zweck wurde nach dem Vorbild der Ortssuche ein Suchfeld über die Tabellenansicht integriert. Startet der Nutzer mit seiner Eingabe, wird der Inhalt der Tabellenansicht geleert und durch passende Suchergebnisse gefüllt. Dieser Vorgang geschieht iterativ während des Tippvorgangs und bietet damit die Vorteile der Autovervollständigung an.

Die Gestaltung der Oberflächenelemente und der Benutzerführung wurde von der marktführenden Suchmaschine „Google“<sup>24</sup> abgeleitet. Mit einem Marktanteil von 95% in Deutschland [statista, 2017b] kann davon ausgegangen werden, dass dieses Such-Schema den meisten Nutzern vertraut ist.

#### 6.2.4. Grundlegende Interaktionen

Um grundlegende Interaktionen mit der Karte schneller erreichbar zu gestalten, werden dafür Buttons in der Kartenansicht integriert. Mit diesen Buttons sollen dem Nutzer in erster Linie „ansichtsverändernde“ Interaktionen zu Verfügung gestellt werden, um die Defizite des Bildschirms auszugleichen. Mit Hilfe dieser Funktionen können nicht oder schlecht sichtbare Kartenbereiche durch nutzerdefinierte Aktionen sichtbar gemacht werden.

Diese einfachen Interaktionsmöglichkeiten zur Steuerung von Kartenansichten sind in den letzten Jahren für viele Nutzer zur Selbstverständlichkeit geworden.

Die Buttons beinhalten jedoch nicht nur Interaktionen zur Manipulationen der Kartenansicht, sondern verfügen auch über informative Inhalte zum visualisierten Indikator. In der folgenden Abbildung 17 sind diese „Schnellzugriffe“, welche durch die Buttons ausgelöst werden als Aktivitätsdiagramm dargestellt. Es zeigt, welche Aktion durch einen Klick oder ein Touch-Event ausgelöst wird. In den folgenden Abschnitten werden diese Funktionen kurz vorgestellt.

<sup>24</sup><https://www.google.de>

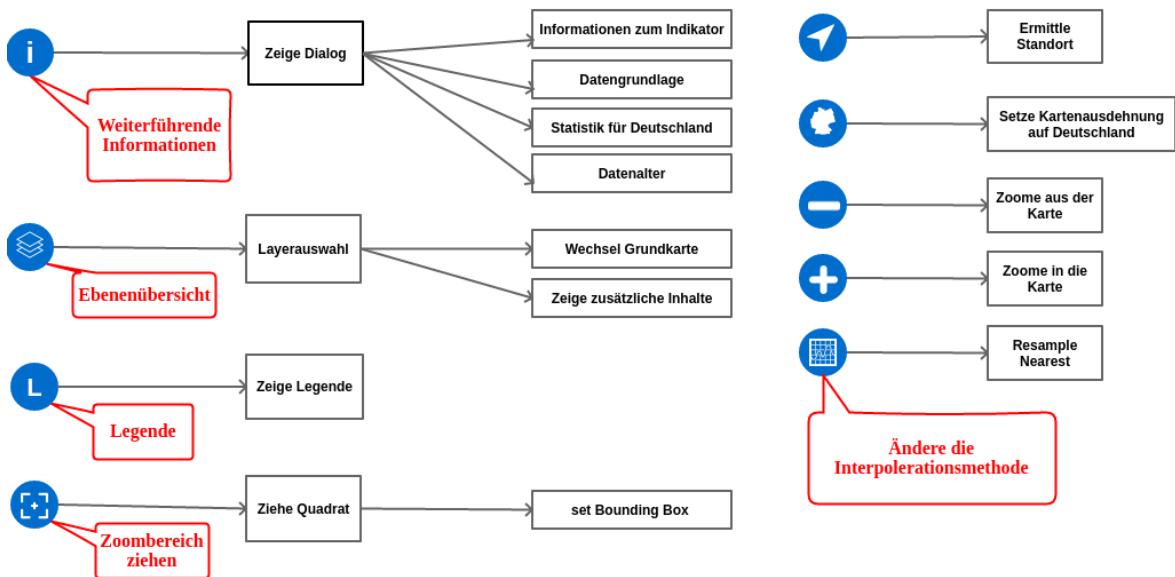


Abbildung 17: Aktivitätsdiagramm der Interaktionsbuttons auf der Karte

#### 6.2.4.1. Ebenenübersicht

Eine wichtige Neuerung gegenüber des alten IÖR-Monitors stellt die Möglichkeit dar, zwischen den Grundkarten zu variieren. Diese Funktion ermöglicht es dem Nutzer, die Indikatkarten mit unterschiedlichen Grundkarten zu visualisieren. Hiermit wird eine Option eingeführt, die Indikatoren unterschiedlich hervorheben zu können. Angelehnt an die Möglichkeiten der untersuchten Anwendungen innerhalb der Bestandsanalyse, hat der Nutzer vier Optionen zur Wahl. In Tabelle 5 werden die verwendeten Grundkarten vorgestellt.

Durch die farbliche Gestaltung der Indikatkarten müssen alle Grundkarten ohne eine Abbildung von Farbe dargestellt werden, um eine bestmögliche Sichtbarkeit dieser zu gewährleisten. Möchte der Nutzer keine Hintergrundkarte, steht ihm auch diese Möglichkeit zur Verfügung. Eine Besonderheit stellt sicherlich die Option dar, eine TopPlus Hintergrundkarte anzubieten, was durch die freundliche Unterstützung des *Bundesamt für Kartographie und Geodäsie (BKG)*<sup>25</sup> realisiert werden konnte. Hier hat das BKG ein Verfahren entwickelt, „*das auf Basis vorhandener Geodaten hochaktuelle Präsentationsgraphiken unterschiedlicher Maßstäbe vollautomatisch erzeugt*“ [BKG, 2017].

<sup>25</sup><https://www.bkg.bund.de>

|                      | <b>OpenStreetMap</b>  | <b>Satellit</b>  | <b>TopPlus</b>   |
|----------------------|---|--|--|
| <b>Daten-zugriff</b> | Javascript API  | Javascript API   | Javascript API   |
| <b>Anbieter</b>      | OpenStreetMap Foundation  | ESRI Inc.  | Bundesamt für Kartographie und Geodäsie  |
| <b>Zugang</b>        | Öffentlich  | Öffentlich   | Nur Behörden   |
| <b>Vorteil</b>       | Meist sehr aktuelle Karten,<br>Oft hohe Qualität/<br>Genauigkeit/Dichte,<br>kostenfrei  | aktuelle und<br>hochwertige<br>Bilder,<br>wolkenfreie und<br>teilweise scharfe<br>Luftbilder | Einheitliche grafische<br>Ausgestaltung, individuelle<br>Konfigurierbarkeit auf Grund<br>eines sehr umfangreichen und<br>differenzierten Layersets,<br>höchste Detailgenauigkeit |
| <b>Nachteil</b>      | Gewöhnungsbedürftige, z.T.<br>abstrakte Darstellung,<br>Detaildichte hängt sehr von<br>der Region ab, Genauigkeit<br>geringer als bei Karten der<br>Landesvermessungssämler | z.T. noch erhöhte<br>Ladezeiten durch<br>Beta-Status   | nur für Behörden zugänglich,<br>Aktualisierung noch nicht<br>geklärt, fehlende<br>Dokumentation zur<br>Strukturierung und<br>Konfiguration des<br>umfangreichen Layersets        |
| <b>Beispiel</b>      |    |            |   |

Tabelle 5: Gegenüberstellung der Grundkarten, Quellen: [vgl. Klöckner; van de Sand, 2017; Schneider, 2010]

Für eine verbesserte Kartengestaltung werden in die Layerauswahl separat auswählbare Zusatzebenen hinzugefügt: das Autobahnnetz, das Fernbahnnetz der DB und die Hauptfließgewässer.

#### 6.2.4.2. Identify

In der Vektoransicht können punktuelle Informationen durch die synchronisierte Ansicht zwischen Tabellen- und Kartenansicht abgerufen werden, jedoch fehlt diese Option in der Rasteransicht. Hier kommt eine sogenannte geographische „Identify“ Funktion zum Einsatz [Mittlböck u. a., 2012]. Klickt der Nutzer auf eine gewünschte Position, werden die dazugehörigen Werte der Karte in Form eines Pop-Up angezeigt. Durch die abschaltbar gestaltete Tabellenansicht muss die „Identify“ Funktion auch in der Vektoransicht vorhanden sein. Diese muss genau dann aktiviert werden, wenn die Tabelle nicht mehr sichtbar ist, um auch hier Werte aus der Karte abfragen zu können.

Eine Besonderheit stellt der geplante Vergleichsmodus dar, wo zwei Rasterkarten, getrennt durch einen Slider, visualisiert werden. Hier muss die Identify Funktion prüfen, ob ein Klick auf die linke oder rechte Karte erfolgte und eine dementsprechende Ermittlung der Werte durchführen.

#### 6.2.4.3. Änderung der Interpolationsmethode

Standardmäßig interpoliert der *MapServer* die Rasterkarten mit der *Nearest-Neighbour* Methode, in welcher einer Zuweisung des nächsten Nachbarn erfolgt [Warnerdam, 2013]. Innerhalb von hohen Zoomstufen wird einer der Nachteile dieses Verfahrens sichtbar, welcher in

einer sehr *pixeligen* Darstellung niederschlägt.

Um dem Nutzer eine Glättung der Indikatorkarte zu ermöglichen, wird ein entsprechender Button als Schnellzugriff in den Rasterviewer integriert. Durch die Aktivierung dieses Buttons wird die Interpolation des *MapServer* auf eine *Bilineare Interpolation* umgestellt. Hier wird der neue Wert einer Zelle anhand eines gewichteten Entfernungs durchschnitts der vier nächstgelegenen Eingabezellmittelpunkte ermittelt [Warnerdam, 2013]. Diese Interpolationsmethode bewirkt eine Glättung der Karte. In Abbildung 18 sind die beiden möglichen Optionen für die Variierung der Interpolationsmethoden dargestellt. Die Screenshots stammen aus der Analysephase, in welcher die bestmögliche Parametrisierung gefiltert wurde, die Ergebnisse sind als *WMS* in der GIS-Software *QGIS*<sup>26</sup> abgebildet.

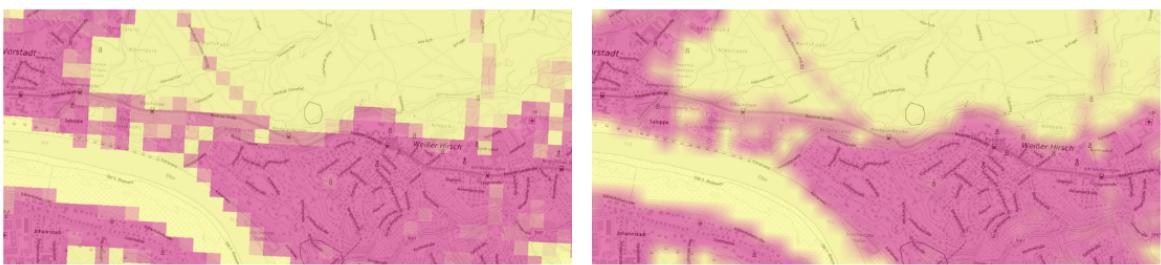


Abbildung 18: Interpolationsvergleich der Rasterkarten (links: **Nächster Nachbar**, rechts: **Bilinear**)

#### 6.2.4.4. Zoom

Die Zoom Interaktion stellt sicher eine der ersten und bekanntesten kartographischen Interaktionen dar. Sie wurde bereits in den 1993 entwickelten, ersten Map Viewer integriert, welcher vom Xerox's Palo Alto Research Center implementiert wurde [vgl. Plewe, 1997].

Durch Zoom Interaktionen kann der Nutzer den Maßstab der Karte verändern. Zoomt er in die Karte hinein, werden kleine Elemente der Karte sichtbar und im Gegenzug führt ein herauszoomen zu einer besseren Gesamtübersicht. Die Interaktionen können mit Hilfe der entsprechenden Buttons oder aber durch das Mausrad ausgeführt werden. Auf dem mobilen Endgerät werden diese Buttons nicht angeboten, da durch diverse Studien bewiesen wurde, dass Nutzer eher die Möglichkeiten der Touch Interaktion nutzen (siehe Kapitel 5.3.3.2) und dadurch diese Buttons redundant sind. Mittels des Pinching (Verkleinern) und Spreading (Vergrößern) durch Daumen und Zeigefinger wird das Heraus- und Hereinzoomen durch den Nutzer ausgelöst [vgl. Schlegel, 2013, S.22].

Eine weitere wichtige Funktion stellt der Button für die Definition eines nutzerspezifisch festgelegten sichtbaren Ausschnittes dar. Wird der Button aktiviert, besitzt der Nutzer die Möglichkeit ein Rechteck in der Karte zu ziehen, in welches der Darstellungsbereich der Kartenaussicht eingepasst wird.

<sup>26</sup><https://www.qgis.org>

#### 6.2.4.5. Weiterführende Informationen

Viele Informationen zum visualisierten Indikator sind nicht in der Karte dargestellt, damit die Übersichtlichkeit bewahrt wird. Für die Darstellung dieser wird die Möglichkeit genutzt, Informationen auf Verlangen des Benutzers zur Verfügung zu stellen.

In dieser Anwendung werden dafür zwei Buttons vorgesehen. Einer beinhaltet die Legende und der andere zusätzliche Inhalte der Indikatorkarte. Als wichtiger Bestandteil einer Karte wird die Legende nach jeder Selektion eines Indikators und der damit verbundenen Visualisierung eingeblendet und kann durch Klick-Interaktion wieder ausgeblendet werden. Der Button für die weiterführenden Informationen öffnet ein Dialog-Fenster, welches die Inhalte in einem Accordion organisiert. Der Nutzer kann die Position des Informations-Fensters durch *Drag&Drop* festlegen und bei Bedarf wieder schließen. Für die Positionierung innerhalb der Anwendung werden dabei keine Restriktionen gesetzt.

Folgende Inhalte nimmt das Dialog-Fenster auf:

- **Informationen zum Indikator:** Erläutert die Grundlagen des jeweiligen Indikators.
- **Datengrundlage:** Dokumentiert die zugrundeliegenden Geodaten, welche für die Berechnung des Indikators verwendet wurden.
- **Statistik für Deutschland:** Dieser Punkt gibt dem Nutzer einen Überblick über die Minimal- und Maximalwerte, arithmetisches Mittel, Standardabweichung und den Median. Weiterführend werden die Werte anschaulich in einem Histogramm visualisiert.
- **Datenalter:** Der Menüpunkt Datenalter beinhaltet eine Übersichtskarte, welche die Hauptkarte in ihrer ganzen geografischen Ausdehnung in einer navigierbaren, kleineren Referenzkarte darstellt und den aktuellen Hauptkartenausschnitt markiert. Sie dient dem Nutzer zur besseren Orientierung und Übersicht. Als Karte wird hier das Datenalter des gewählten Indikators visualisiert. Zoomt der Anwender in die Karte, passt sich der markierte Ausschnitt der Übersichtskarte an. In Abbildung 19 wird diese Übersichtskarte dargestellt, welche durch eine rot umrandete Bounding Box den aktuellen Sichtbereich des Nutzers darstellt. Die linke Legende zeigt welche Werte der Kolorierung zugewiesen wurden.

#### 6.2.5. Kartengestaltung

Mit dieser Interaktionsmöglichkeit soll der Nutzer in die Lage versetzt werden, die Karte nach seinen Vorstellungen zu gestalten. Vor allem Probanden der Benutzeranalyse, welche dem wissenschaftlichen Bereich entstammten, empfanden einen möglichen Wechsel der Farbgebung einer Indikatorkarte als wichtige zukünftige Option.

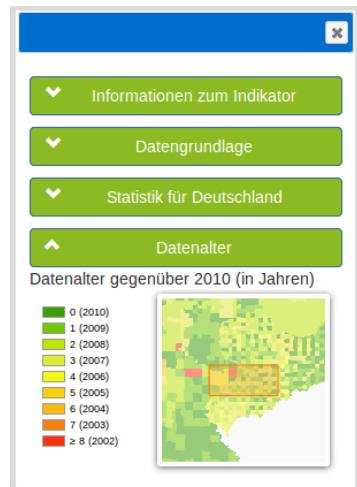


Abbildung 19: Informations-Fenster

Um diesen Vorgang so einfach wie möglich zu gestalten, wurde ein bereits vorhandenes Interaktionskonzept für die Auswahl der kategorisierten Farben verwendet. In den beiden dominierenden GIS-Softwareen *ArcGIS*<sup>27</sup> und *QGIS*<sup>28</sup> kann der Nutzer aus bereits vorgefertigten Farbschemas auswählen, welche in einem Dropdown angeboten werden. Dieses Schema wird übernommen, jedoch die Vielfalt der Wahlmöglichkeiten auf vier begrenzt, da dies für eine erste Testphase ausreicht.

Um eine möglichst eindeutige Zuordnung der Kolorierung zu erreichen, ist bei der Auswahl der Farbreihen auf die Unterscheidbarkeit von Nachbarfarben und die visuelle Gleichabständigkeit zu achten [Koch u. Bollmann, 2002].

Zur Auswahl werden dem Nutzer typische Farbreihen angeboten, welche aus einem Dropdown-Menü gewählt werden können (siehe Abbildung 20). Ausgewählt wurden die Farbreihen mit der Open Source Web-Anwendung *COLORBREWER*<sup>29</sup>, mit dessen Hilfe sich Farbreihen für den kartografischen Kontext entwickeln lassen.

Verändert der Nutzer die gewählte Klassenanzahl des visualisierten Indikators, passt sich auch die Farbreihe an diese Klassifizierung an. Wird eine Farbreihe ausgewählt, nimmt diese den vorhanden Platz im Drop-Down Header ein. Durch ein rechts platziertes Kreuz kann der Nutzer die gewählte Farbreihe entfernen und der Standard wird wiederhergestellt. Um die Sichtbarkeit der Hintergrundkarte zu steuern, ist ein Transparenz-Slider vorgesehen. Durch die Wahl der Klassifizierung wird der Nutzer in die Lage versetzt, eine Anpassung der dargestellten Werte in Wertegruppen zu verändern. Ändert der Nutzer die Klassenanzahl, sollte sich auch die Farbreihe automatisch an diese Wahl anpassen.

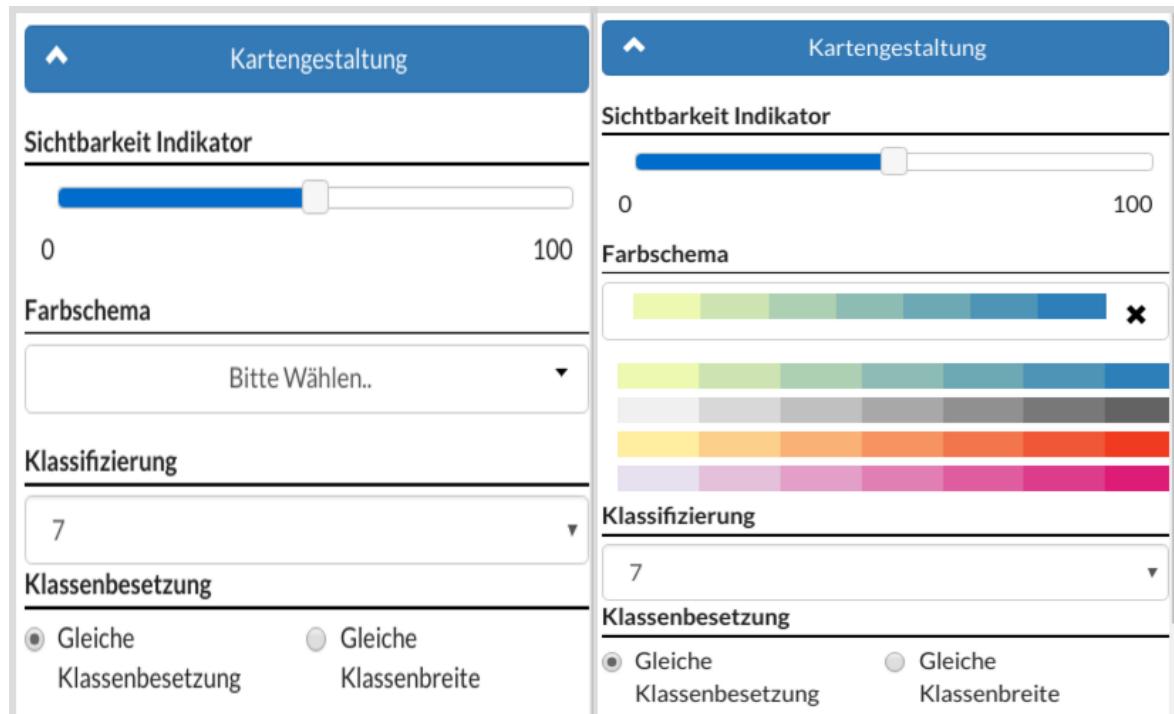


Abbildung 20: Konzept des Menüs der Kartengestaltung, mit einer gewählten Farbreihe (rechts)

<sup>27</sup><https://www.arcgis.com>

<sup>28</sup><https://www.qgis.org>

<sup>29</sup><http://colorbrewer2.org>

### 6.2.6. Räumliche Gliederung

Durch die unterschiedlichen Visualisierungstechniken innerhalb der beiden Darstellungsmodi muss auch die Anpassung der räumlichen Aufteilung variabel gestaltet werden. Die Vektoransicht basiert auf administrativen Gebietseinheiten und bietet damit einen gute Übersicht über variierende Werte. In der Rasteransicht basiert die räumliche Gliederung auf der Rasterweite, welche in unterschiedlichen Größen angeboten wird. Um eine bestmögliche Anpassung zu erreichen, passen sich die möglichen Interaktionen an den Darstellungsmodus an. Abbildung 21 illustriert diese Variation der Inhalte. In der Rasteransicht sind die möglichen räumlichen Anpassungen auf die Rasterweite begrenzt, welche durch einen Slider durchgeführt werden können. Je nach Indikator sind mögliche Einstellungen unter dem Slider abgebildet.

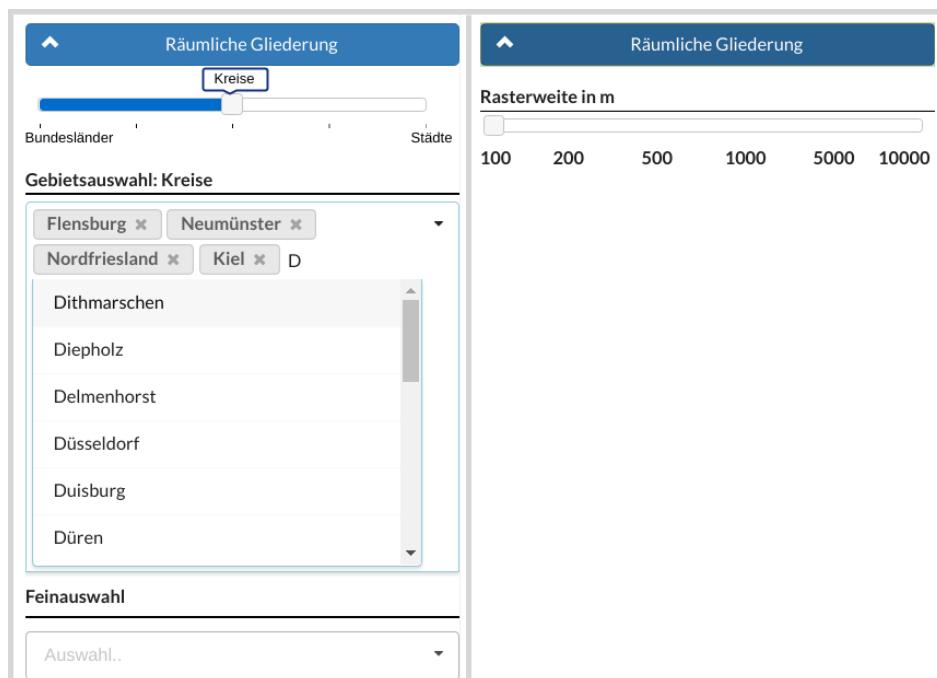


Abbildung 21: Konzept des Menüs der räumlichen Gliederung für die Raster- (links) und Vektoransicht (rechts)

Um dem Nutzer ein ähnliches Interaktionskonzept in der Vektoransicht zu bieten, wird auch hier ein Slider eingesetzt, durch welchen er die gewünschte administrative Gebietseinheit wählen kann. Dabei wird die Möglichkeit geschaffen, alle Gemeinden in Deutschland auszuwählen und dadurch auch in der Karte anzuzeigen. Nach jeder Wahl wird die gewählte Gebietseinheit für ganz Deutschland angezeigt. Um die Möglichkeit einer Individualisierung zu schaffen, wird die Ansicht um eine selektive Auswahl erweitert. Diese besteht aus einem Dropdown-Menü welches alle visualisierten Gebietseinheiten beinhaltet. Die Liste kann je nach gewählter Gliederung sehr groß sein, womit die Übersicht verloren geht. Hier empfiehlt sich die Integration einer Suchfunktionalität innerhalb des Drop-Down Menüs [Babich, 2017]. Selektiert der Nutzer eine Gebietseinheit, wird die Karte geleert und die Auswahl visualisiert. Hierbei passt sich der Sichtbereich automatisch an die Auswahl an. Um den Nutzer bei der Mehrfachauswahl zu unterstützen, wird für jede Selektion ein Tag gebildet und dem Kopfbereich des Menüs zugeordnet. Der Vorteil dieser Technik ist ein ständiger Überblick über die in der Kartenansicht visualisierten Gebiete. Möchte der Nutzer eine einzelne Gebietseinheit

entfernen, befindet sich für diese Interaktion ein Kreuz im Tag. Dadurch wird der Tag und das Gebiet aus der Visualisierung entfernt. Mit der Feingliederung kann der Nutzer die gebildeten Gebietseinheiten verfeinern, indem er beispielsweise die selektierten Kreise in Gemeinden unterteilt. Jede Veränderung der Gebietseinheit wird mit der Tabelle synchronisiert, womit der Nutzer einen ständigen Überblick besitzt. Verändert der Nutzer die räumliche Gliederung durch den Slider, wird das Drop-Down Menü geleert und mit den entsprechenden Inhalten gefüllt.

### 6.2.7. Werkzeuge

Das Vorhandensein von einfach zu bedienenden und intuitiven Werkzeugen bezeichnet [Haklay, 2010] als einen der Gründe für den Erfolg von heutigen WebMapping-Anwendungen. Wie vielseitig diese sein können, demonstrierten die Anwendungen in der Bestandsanalyse (siehe Kapitel 5.3.3). In dieser Arbeit soll die zu konzipierende Anwendung um eine Auswahl an Werkzeugen erweitert werden, welche in heutigen WebMapping-Anwendungen in die Kategorie Basiswerkzeuge fallen. Die geplanten Funktionen sind in der folgenden Abbildung 22 abgebildet.



Abbildung 22: Konzept der Werkzeugwahl

Das Fullscreen Werkzeug versetzt den Nutzer in die Lage den gesamten Bildschirm für die Anwendung zu nutzen. Wird die Funktionalität aktiviert, blendet der Browser seine Schaltfläche aus und der Header wird deaktiviert.

Mit dem Messwerkzeug können Distanzen und Flächen gemessen werden. Die Distanz- und Flächenmessung wird durch das Setzen von Messpunkten auf der Karte gestartet, indem ein Fenster die aktuell ermittelten Ergebnisse anzeigt. Mit der Lupenfunktion können Bereiche ausgewählt und herangezoomt werden. Diese Funktion soll dem Nutzer vor allem dann unterstützen, wenn er eine sehr hohe Transparenz der Indikatorkarte eingestellt hat und somit die Grundkarte nicht mehr sichtbar ist. Mit der Lupe kann er jetzt die Grundkarte herangezoomt darstellen.

Mit der Import Funktion erhält der Nutzer die Möglichkeit, eigene Dateien zu nutzen und in der Anwendung zu visualisieren. Unterstützt werden die Dateiformate: KML, JSON, GPX. Diese können mit der Hilfe eines Importfensters in die Anwendung geladen werden. Alle Werkzeuge geben dem Nutzer Feedback über ihre Funktionalität. Aktiviert der Nutzer ein Werkzeug durch den entsprechenden Button, wird dieser Grün hinterlegt.

### 6.2.8. Feedback und Nutzerunterstützung

Die Interaktion des Nutzers mit der Anwendung lässt sich als Kommunikation zwischen Mensch und Maschine verstehen. Menschen benötigen fortwährend Auskünfte über den Zustand des Gegenübers und des Kommunikationsprozesses. Während bei zwischenmenschlichen Kommunikationen diese Informationen of implizit (z.B. durch Mimik und Gestik) ersichtlich sind, müssen sie bei der Kommunikation mit der Anwendung in Form von Statusinformationen, Rückmeldungen und Fehlermeldungen explizit erstellt werden [vgl. Stephan, 2010, S.376]. Obwohl Web 2.0 Anwendungen in der Interaktion typischerweise schnell reagieren, kann es auch hier zu Verzögerungen kommen, etwa weil die Anwendung gerade umfangreiche Operationen durchführt, um neue Daten darstellen zu können [vgl. Blumauer u. Pellegrini, 2009, S.98]. Gerade bei dem Visualisierungsvorgang aller administrativen Gebietseinheiten kleiner der Raumordnungsregionen muss mit einer erhöhten Rechenzeit gerechnet werden. Um den Nutzer über den Zustand des Systems zu informieren, empfiehlt sich eine Ladeanzeige, welche sich nach dem erfolgreichen Abschluss des Prozesses automatisch schließt [Stephan, 2010].

Die Visualisierung aller Gemeinden stellt für den Clienten einen sehr rechenintensiven Vorgang dar, wo je nach Leistungsfähigkeit des Systems mit einer erhöhten Rechenzeit gerechnet werden muss. Wählt der Nutzer die gewünschte Gebietsgliederung, weiß ihn eine Meldung auf die Problematik hin. Innerhalb der Meldung besteht die Möglichkeit den Vorgang abzubrechen oder fortzuführen. In seinem Buch nennt Thissen [2001] diese Meldung als essentiell, da viele Nutzer eine Seite verlassen, wenn die Wartezeit auf eine Rückmeldung des Systems 10s übersteigt. Diese eingeblendete Nachricht integriert auch Hinweise an den Nutzer, wie die Ladezeit reduziert werden kann, indem beispielsweise eine Vorauswahl bestimmter Kreise stattfindet, welche darauffolgend in Gemeinden gegliedert werden.

Ein weiterer Punkt stellt die unterschiedliche Verfügbarkeit der Indikatoren für die Raster- und Gebietsansicht dar. Viele Indikatoren liegen nicht für beide Darstellungsarten vor, worüber der Nutzer informiert werden muss, wenn seine Interaktion einen Fehler wirft. Zum Beispiel: hat der Nutzer einen Indikator und Zeitschnitt in der Rasteransicht gewählt und möchte den gleichen Indikator in der Gebietsaufteilung der Kreise visualisieren, muss ihm eine Fehlermeldung signalisieren, dass die gewünschte Interaktion nicht möglich ist. Der Grund liegt meist in der fehlenden Existenz von Geodaten, für den gewünschten Indikator im System. In der folgenden Abbildung 23 werden drei ausgewählte Rückmeldungen des Systems dargestellt.



Abbildung 23: Konzept der Fehlermeldungen für die Anwendung: 1. Ladeanzeige; 2. Meldung für eine erhöhte Rechenzeit; 3. Ein Indikator ist im gewählten Zeitschnitt nicht vorhanden

Da viele Nutzer sich durch die Neuauflage des IÖR-Monitors auf eine neue Oberfläche einstellen müssen, wird diese Phase mit einer sogenannten *Tour* durch die Website unterstützt. Startet der Nutzer diese Funktion, wird er interaktiv durch die Website geführt, indem er sich mit einem Vor- und Zurück-Button durch die Inhalte und Funktionen der Anwendung bewegt. Ähnlich einer Museumsführung, werden nach einem vorher festgelegten Drehbuch Stopps eingelegt und die wichtigsten Inhalte erläutert. Um die Tour jederzeit zu beenden, wird ein entsprechender Button integriert.

Um die Weiterentwicklung der Anwendung zu unterstützen und Rückmeldung des Nutzers zu bekommen, wird eine Funktion integriert, mit welcher der Nutzer eine Nachricht an den Entwickler versenden kann. Aktiviert er das entsprechende Feld, wird ein Dialog-Fenster geöffnet, in welches er seine Nachricht eintragen und versenden kann.

### 6.3. Fazit

In diesem Kapitel wurden das Layout der neuen Anwendung konzipiert und alle grundlegenden Neuerungen durch theoretische Überlegungen festgehalten. Es wurden konzeptionelle Entscheidungen getroffen.

Wie kann eine Realisierung der Vereinigung von Übersichts- und Detailviewer anhand einer interaktiven und intuitiven Oberfläche erfolgen? Diese Frage war eine der wichtigsten Aufgaben, die in diesem Kapitel beantwortet werden sollte. Durch den Einsatz eines MultiView wurde eine Option der Realisierung gefiltert, welche mit Interaktionsmöglichkeiten angereichert werden muss, um dem Nutzer einen Mehrwert zu bieten. Die Basis des Konzeptes basierte dabei auf den in Kapitel 5.5 ermittelten Anforderungen an die neue Anwendung des IÖR-Monitors.

Mit diesem Kapitel wurde die folgende Implementierungsarbeit vorbereitet, welche das Konzept realisiert.

## 7. Implementierung

In diesem Kapitel sollen die konzipierten Inhalte umgesetzt werden. Der Hauptfokus des Implementierungsteils liegt dabei auf dem clientseitigen Teil der Anwendung, da das Leibniz-Institut bereits über eine vorhandene serverseitige Plattform verfügt, welche in dieser Arbeit eingesetzt wird.

### 7.1. Technologieauswahl

Für die clientseitige Anwendungslogik ist **Javascript**<sup>30</sup> unablässig. Mit Hilfe dieser 1995 von *Netscape*<sup>31</sup> entwickelten Skriptsprache können: Benutzerinteraktionen ausgewertet, Inhalte verändert bzw. nachgeladen oder neu generiert werden. Damit wird die Möglichkeit geschaffen, HTML und CSS zu erweitert [Koch, 2009]. Ein großer Nachteil dieser Scriptsprache sind die verschiedenen Schnittstellen unterschiedlicher Herkunft, die zum Teil alles andere als einfach und intuitiv sind. Gerade für das bei interaktiven Karten wichtige Event-Handling, besitzt JavaSript kein verlässliches Fundament. Hier helfen **Bibliotheken**, indem sie eine Abstraktionsschicht über JavaScript legen. Dabei ist es wichtig, dass die Bibliotheken aus Modulen bestehen, welche einzeln geladen werden können. Passiert dies nicht, läuft die Webanwendung Gefahr, an einem großen Anteil an Skripten, welche geladen aber nicht verwendet werden, überfrachtet zu werden [Mintert, 2009]. Folgend sollen die in dieser Arbeit verwendeten Open Source JavaScript Bibliotheken vorgestellt werden. Das ausgewählte Kartenframework Leaflet, welches in Kapitel 5.6 ausgewählt wurde, bildet dabei die Grundlage für der Indikatorvisualisierung.

#### 7.1.1. jQuery

Diese Open Source JavaScript-Bibliothek wurde von John Resig entwickelt und 2006 auf dem Barcamp in New York City veröffentlicht. JQuery besitzt eine sehr schlanke Syntax, mit der bereits schon mit wenigen Zeilen Code beeindruckende Ergebnisse generiert werden können [Vdovkin, 2012]. Die Bibliothek bietet sehr viele browserübergreifende Funktionen und beseitigt dabei die Nachteile von JavaScript. Unter dem Motto „*Write Less, Do More*“<sup>32</sup> können vielfältige Einsatzmöglichkeiten für die Realisierung genutzt werden:

- Elemente auf einer Website selektieren
- Selektierte Elemente manipulieren
- Hilfsfunktionalitäten, wie beispielsweise die *each*<sup>33</sup>-Schleife
- Erweitertes Event-System, um Interaktionen des Benutzers mit der Anwendung zu steuern
- Animationen und Effekte

---

<sup>30</sup><https://www.javascript.com>

<sup>31</sup><http://isp.netscape.com>

<sup>32</sup><https://jquery.com/>

<sup>33</sup><http://api.jquery.com/jquery.each/>

- AJAX Funktionalitäten, um Daten (z.B. Karte) mit dem Server auszutauschen, ohne ein komplettes Neuladen der Seite zu erzwingen (siehe Kapitel 4.2).

Die Bibliothek bietet dem Entwickler eine umfangreiche API Dokumentation<sup>34</sup>, womit der Einstieg vereinfacht wird.

### 7.1.2. Bootstrap

Bootstrap<sup>35</sup> ist ein Ergebnis der beiden Entwickler Mark Otto und Jacob Thornton. Die Bibliothek ist ursprünglich aus dem Vorhaben entstanden, die internen Analyse- und Verwaltungswerkzeuge von Twitter<sup>36</sup> weiterzuentwickeln. Boostrap wurde 2011 veröffentlicht und hat seit diesem Zeitpunkt stark an Popularität gewonnen. Neben einem Responsive Grid-System, bietet es Gestaltungsvorlagen für die vielfältigen UI Elemente (z.B. Buttons, Tabellen, Formulare) an. Optional stellt die Bibliothek auch JavaScript Erweiterungen zur Verfügung [Spurlock, 2013]. Gerade wenn die Anwendung mobil optimiert sein soll, nimmt die Bibliothek dem Entwickler, durch die dynamische Anpassung der UI Elemente an den Bildschirm, viel Arbeit ab.

### 7.1.3. Semantic UI

Semantic-UI wurde von dem New Yorker Webentwickler Jack Lukic ins Leben gerufen [Herrmann, 2013]. Diese Bibliothek bietet Gestaltungsvorlagen für über 50 UI-Elemente, welche auch eine Responsivität besitzen. Vor allem in der Gestaltung von Tabellen und Dropdown-Menüs liegen die Stärken dieser Bibliothek [Lukic, 2015].

## 7.2. Umsetzung des Konzeptes

In diesem Kapitel wird das Konzept des vorangegangenen Kapitels umgesetzt. Alle verwendeten Plugins basieren auf Open Source Lösungen, welche von der Website Github<sup>37</sup> bezogen wurden. Hierbei handelt es sich um einen Online-Dienst, auf dessen Servern Entwickler ihre Open Source Software der Öffentlichkeit zur Verfügung stellen können.

### 7.2.1. Interface

Die konzipierte Benutzeroberfläche wurde mit Hilfe eines HTML-Grundgerüstes und der entsprechenden Gestaltung durch CSS realisiert. Jede Strukturierung durch HTML fand in dafür vorgesehenen Dateien statt, zu diesem Zweck wurden für die Darstellungsmodi: Raster, Gebiete und Vergleich jeweils eine HTML-Datei angelegt.

---

<sup>34</sup><https://api.jquery.com/>

<sup>35</sup><http://getbootstrap.com>

<sup>36</sup><https://twitter.com>

<sup>37</sup><https://github.com>

Das Responsive Web Design wurde durch CSS-Links mit *Media Queries* realisiert, indem das Layout dynamisch an drei Bildschirmbreiten angepasst wird:

- $\geq 960\text{px}$  Laptop und Desktop
- $960\text{px} - 550\text{px}$  Tablet
- $\leq 550\text{px}$  Smartphone

Hier werden vor allem Adaptionen des Navigationslayouts vorgenommen. Die Grenzwerte richten sich nach häufig verwendeten Bildschirmauflösungen. Allerdings kann man mit der Bildschirmauflösung nicht bestimmen, um welchen Gerätetyp es sich handelt. So hat beispielsweise das iPhone 5 eine Auflösung von  $1136 \times 640$  Pixel und würde je nach Orientierung als Tablet eingestuft. Die Karte wurde anhand von *Fluid Grids* auf 100% gesetzt, womit sie auf allen Geräten die komplette Höhe und Breite der Anwendung belegt. Für die Grundgestaltung der Interaktionselemente kam die Bootstrap Bibliothek an die Farbgebung angepasst, zum Einsatz. Mit dieser Vorgehensweise wird die Responsivität der betroffenen Elemente durch die Bibliothek übernommen.

Die Gestaltung der Navigationsbreite an das Responsive Layout wurde in den jeweiligen *Media Queries* verankert. Die Tabelle und auch die Drop-Downs im Navigationsmenü fanden mit Semantic-UI ihre Realisierung, welche ähnlich wie Bootstrap eine automatische Anpassung der Elemente an die Bildschirmabmessung durchführt. Viele UI-Elemente werden jedoch über die JQuery Bibliothek dynamisch erstellt und verändert, dies beinhaltet auch die Anpassungen an das Responsive Layout. Diese dynamischen Anpassungen werden in den folgenden Kapiteln in die Dokumentation einbezogen. In der folgenden Abbildung 24 ist das finale Ergebnis des implementierten Responsive Design abgebildet. Hier ist gut zu erkennen, wie die Interaktions-Buttons für den Wechsel zwischen Tabellen- und Kartenansicht, den MultiView ersetzen. Auch die vertikale Rotation der Zeit-Slider des Vergleichsmodus, zur Perfektionierung der Bedienbarkeit wird ersichtlich.

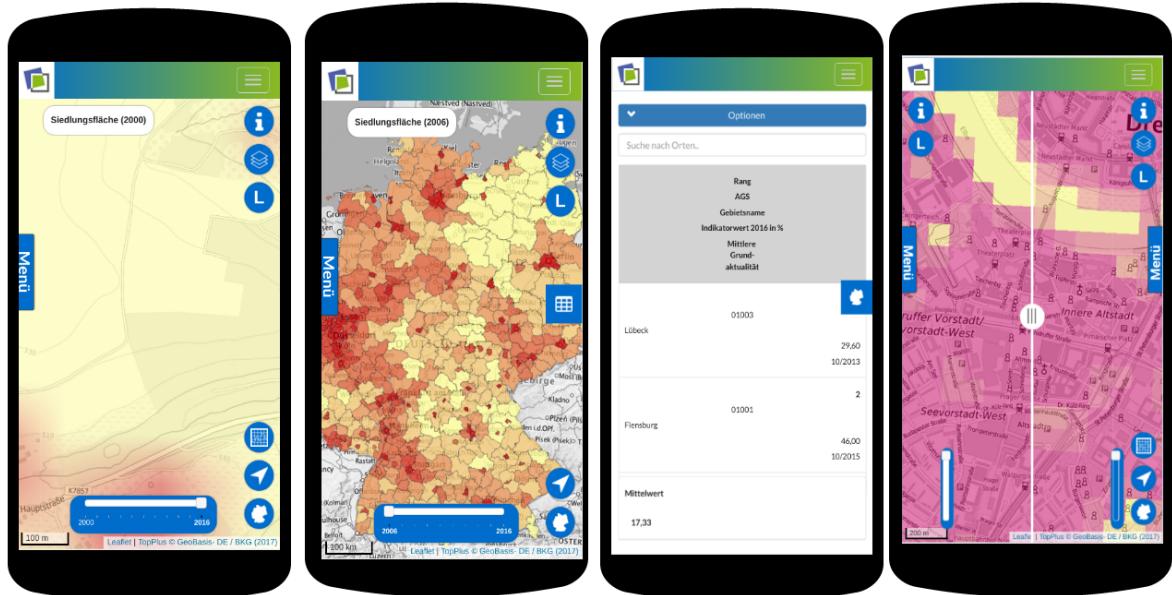


Abbildung 24: Responsive Layout des IÖR-Monitors von links: Raster-, Gebiets-, Tabellenansicht und der Vergleichsmodus

### 7.2.2. Interaktiver Gebietsviewer

In diesem Kapitel wird die Implementierung des Gebietsviewers mit dem konzipierten MultiView erläutert. Hierbei geht dieses Kapitel genauer auf die Generierung der thematischen Choroplethenkarte der Indikatoren ein. Weiterhin werden die implementierten Interaktionsmöglichkeiten zwischen der Tabellen- und Kartenansicht behandelt.

#### 7.2.2.1. Generierung der Gebietskarte

Der Ablauf der Generierung einer Gebietskarte wird in dem folgend abgebildeten Ablaufdiagramm 25 dargestellt.

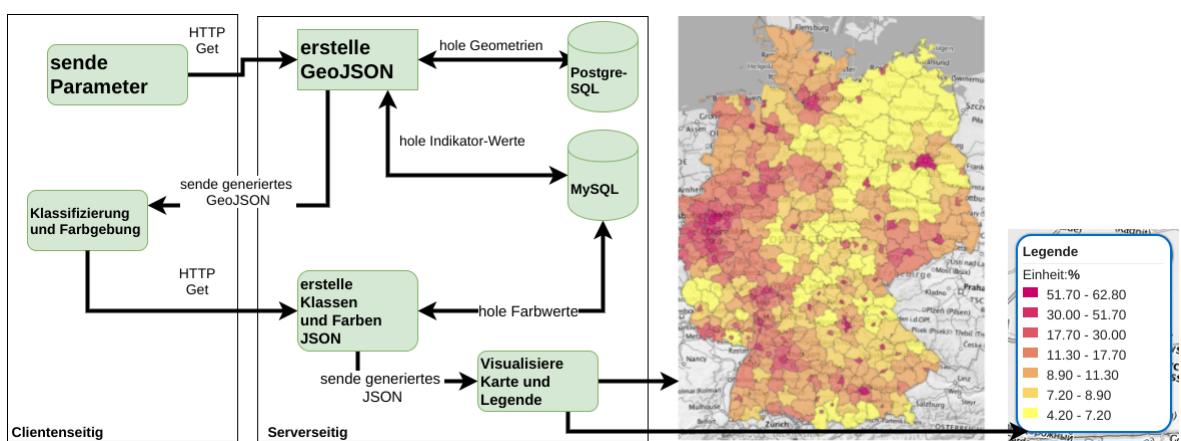


Abbildung 25: Ablauf der Kartenerstellung

Um die Vektordaten der administrativen Gebietseinheiten zu visualisieren, bietet Leaflet das GeoJSON-Format als Ersatz für die SVG-Technologie an.

Mit Hilfe der JQuery *GET* Methode kommt es via *AJAX* zu einer Übermittlung der benötigten Parameter an den Server. Als Parameter werden der gewählte Indikator, Zeit und die gewünschte administrative Gliederung an das entsprechende PHP-Script gesendet. Dieses holt anhand der gegebenen administrativen Gliederung die dazugehörige GeoJSON-Geometrien, welche in der PostgreSQL-Datenbank abgelegt sind. Innerhalb dieser Anfrage wird in der PostgreSQL die Geometriespalte der entsprechenden räumlichen Gliederung mithilfe der Funktion *ST\_AsGeoJSON* in das GeoJSON-Format überführt.

Um die entsprechenden Indikator-Werte mit den passenden Geometrien zusammenzulegen, wird anhand einer Schleife für jeden Amtlichen Gemeindeschlüssel der GeoJSON eine Abfrage an die MySQL-Datenbank gestellt, welche alle Indikator-Werte speichert. Die Ergebnisse werden im PHP JSON-kodiert und können zusammen mit der Geometriespalte im GeoJSON-Format an den Client übertragen werden.

Empfängt der Client ein Ergebnis, muss die GeoJSON für die farbliche Gestaltung klassifiziert werden. Hierfür wird eine neue JQuery *GET* Methode an den Server gestellt und das über HTTP-angesprochene PHP Skript berechnet die Klassen und holt die entsprechenden Farbwerte aus der MySQL-Datenbank. Das Ergebnis wird als JSON zurückgeschickt. Signalisiert *AJAX* das erfolgreiche Eintreffen, wird jeder Indikator-Wert der GeoJSON Datei anhand der empfangenen Klassen-Datei klassifiziert und dem entsprechenden Farbwert zugeordnet.

Auf dieser Grundlage erfolgt auch die Generierung der entsprechenden Legende.

Wurde die Karte erfolgreich dem Kartenobjekt hinzugefügt, beginnt die Visualisierung der Tabellenansicht, welche im folgenden Kapitel erläutert wird.

Für beide Abläufe spielen die zwei Datenformate GeoJSON und JSON eine große Rolle. Das GeoJSON-Format wird eingesetzt um Geometrien mit Werten zu verbinden, welche in der Kartenansicht dargestellt werden. Bei der Verwendung von Werten ohne Koordinaten stellt das JSON-Format einen simpleren Ansatz dar.

Ein Problematik der alten Anwendung entstand mit der Unterbrechung des Visualisierungsvorgangs durch den Nutzer, wenn er während der Kartengenerierung die Gebietseinheit wechselte. War dies der Fall, kam es zu einer Fehlermeldung und Abbruch des Vorgangs. In der neuen Anwendung wurden hierfür AJAX Abbruchbedingungen implementiert. Wählt der Nutzer eine neue Gebietsaufteilung, wird die Generierung der Gebietskarte unterbrochen und mit den neuen Parametern von vorn begonnen.

### 7.2.2.2. Generierung der Tabelle

Auf der Grundlage der generierten GeoJSON wird die dazugehörige Tabelle dynamisch erstellt. Wurde die Karte erfolgreich visualisiert, beginnt der Erstellungsvorgang der Tabelle mit den Indikator-Werten der kartographisch dargestellten Indikatoren. Hierfür werden alle AGS aus der GeoJSON mit einer Schleife gefiltert, in einem Array gespeichert und anschließend über HTTP-GET an den Server geschickt. Jetzt holt das angesprochene PHP-Skript die entsprechenden Werte aus der MySQL Datenbank und generiert die HTML-Datei, welche an den Clienten zurück gesendet wird. Ist der Vorgang erfolgreich verlaufen, übernimmt JQuery das Hinzufügen der HTML und veranlasst damit die Visualisierung der Tabelle.

Nach dem erfolgreichen Visualisieren wird der Mittelwert aus den dargestellten Indikator-Werten gebildet und es erfolgt eine Freigabe der Tabelle für nutzerseitige Interaktion. Diese werden während der Generierung geblockt, um den Rendervorgang nicht zu unterbrechen. Abbildung 26 bildet das dazugehörige Aktivitätsdiagramm ab.

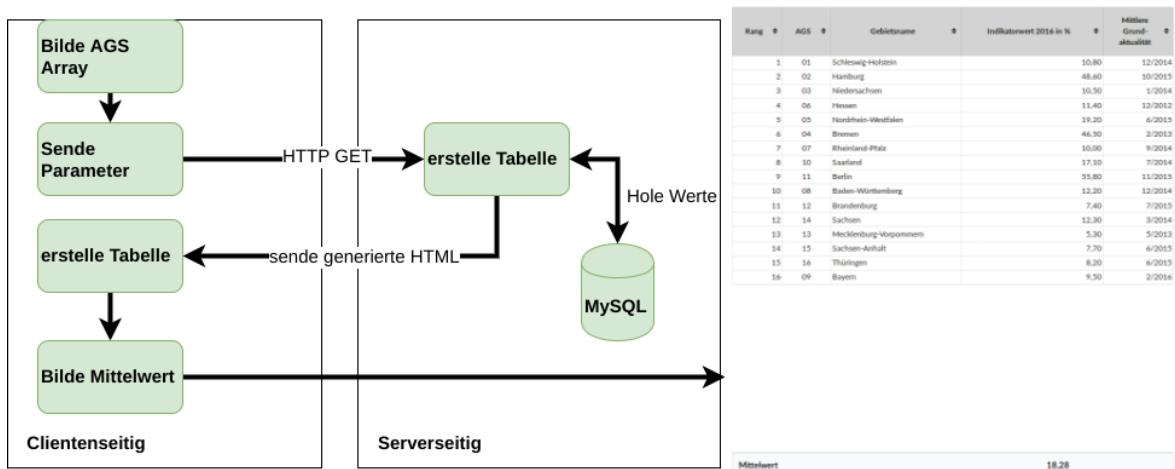


Abbildung 26: Ablauf der Tabellenerstellung, mit beispielhafter Parametrisierung

### 7.2.2.3. Optimierung der Visualisierung

Die Darstellung aller Gemeinden stellt den Clienten vor große Herausforderungen. Um den Prozess so effizient wie möglich zu gestalten, wurden verschiedene Vorgehensweisen getestet. Je nach gewähltem Zeitschnitt werden für die deutschlandweite Ausdehnung im Jahr 2016 über 11.092 Werte und Geometrien geladen und visualisiert. Einen ersten Ansatz für das parsen (analysieren) der generierten GeoJSON stellt JQuery mit der *Each*<sup>38</sup>-Funktion zur Verfügung. Dieser Ansatz benötigte jedoch für die Visualisierung über zwei Minuten. Giulio Bai [2009] empfiehlt auf seinen Blog für große Datenmengen die Bibliothek zu verlassen und die native JavaScript *For* Schleife einzusetzen. Mit diesem nativen Ansatz konnte die Generierung der Karte auf 30 Sekunden reduziert werden.

Ein weiteres Problem entstand bei dem Rendering der Tabelle. Ab einer Anzahl von 2000 Gemeinden kommt es bei den meisten Browsern zu einer Überlastung und unterschiedliche Fehlermeldungen signalisieren dem Nutzer eine Unterbrechung der Anwendung. Um dieses Problem zu lösen, wird das gebildete AGS-Array dynamisch aufgeteilt und jedes Teil Array an den Server geschickt. Wurde ein Teil erfolgreich der Tabelle hinzugefügt, startet die Funktion iterativ mit dem nächsten Teil. Um AJAX mitzuteilen, dass alle Gemeinden dargestellt wurden, findet zu Beginn eine Ermittlung der Gesamtanzahl der darzustellenden Gemeinden statt. Während des Visualisierungsvorgangs wird jede angezeigte Zeile gezählt. Stimmen diese mit der Gesamtanzahl überein, kann die Funktion beendet werden. Mit dieser stapelweisen Abarbeitung können alle Gemeinden in der Tabelle dargestellt werden, welche deutschlandweit existieren und das Speicherproblem tritt nicht mehr in Erscheinung.

### 7.2.2.4. MultiView

Mit der MultiView Ansicht des Gebietsviewers wird sicherlich eine der größten Neuerungen innerhalb des IÖR-Monitors realisiert. Um je nach Displaybreite die Tabellenansicht ein- oder auszublenden, bietet JQuery eine Funktionalität zur Ermittlung der Displaybreite und -höhe an. Ist die Bildschirmbreite kleiner als die Höhe, kann auf die Verwendung eines Smartphones geschlussfolgert werden. In diesem Fall wird die Tabelle ausgeblendet, anderenfalls kommt es zur Initialisierung des MultiView.

Zur Umsetzung dieses Systems wurde das von Jakub Jankiewicz entwickelte JQuery Plugin *JQuery Splitter* verwendet, welches als Open Source zur Verfügung steht [vgl. Jakub, 2016]. Dieses Plugin teilt vorher definierte Bereiche durch einen beweglichen Slider. Dadurch konnte die konzipierte Oberfläche umgesetzt werden, welche der Karten- und der Tabellenansicht jeweils einen eigenen Bereich zuweist.

Die Position des Sliders wird über einen *Flexible-Grid* dynamisch festgelegt, indem für die Karte eine Zuweisung von 6/10 der Bildschirmbreite erfolgt.

Mit einem Event Handler fand eine Realisierung der konzipierten Interaktionen zwischen der Tabelle und der Karte statt. Kommt es zu einem *Hover* Event wird die entsprechenden Position bestimmt, über welches Gebiet und damit AGS der Nutzer mit seinem Zeigegerät fuhr. Jetzt wird eine Funktion angestoßen, welche den Partner in der Tabelle herausgesucht. Das Ergebnispaar wird hervorgehoben, indem der Style geändert wird. Invers funktioniert das Event-Handling auch für die Tabelle.

---

<sup>38</sup><http://api.jquery.com/jquery.each/>

Um die Tabelle an- und auszuschalten bietet JQuery die Möglichkeit Elemente anzusteuern und diese ein- und auszublenden. Hierfür werden die Tabelle und der Slider auf *hidden* gesetzt und damit ausgeblendet. Zur Aktivierung dieses Modus wird ein Button für das Hinzufügen in der Kartenansicht eingeblendet. Mit dessen Wahl werden die versteckten Elemente wieder sichtbar geschaltet.

Das Ordnen der Tabellenwerte nach Minimal- und Maximalwerten konnte durch das von Christian Bach programmierte JQuery Plugin „*tablesorter*“ übernommen werden [vgl. Bach, 2014].

### 7.2.3. Rasterviewer mit Vergleichsmodus

Die vorangegangene Implementierung des Rasterviewers im IÖR-Monitor wurde über Open-Layers 2 implementiert. Durch die Entscheidung Leaflet zu verwenden, mussten die Funktionalitäten in das gewählte Kartenframework überführt werden. Die serverseitige Implementierung des Rasterviewers wurde dabei an das neue Kartenframework angepasst, jedoch von der Grundlogik nicht verändert. Je nachdem welche Auswahl der Nutzer getroffen hat, werden die eingestellten Parameter an den Server übergeben. Hier generiert ein PHP Skript das Mapfile, welches den MapServer anweist nach welchen Parametern er die Rasterkarte zur Verfügung stellen soll. Als Ergebnis wird eine *GetMap*-URL an den Clienten zurückgesendet, anhand derer Leaflet den WMS visualisieren kann.

Um den konzipierten Vergleichsmodus zu realisieren, fand des Open Source Plugin von Arthur Street seine Verwendung, dessen Sourcecode angepasst eingebunden wurde [vgl. Street, 2017]. Mit Hilfe dieses Plugins kann eine Teilung der Kartenansicht in zwei Teile erfolgen. Hierfür kommt ein vertikaler Slider zum Einsatz, welcher in beide Richtungen bewegt werden kann. Für die Auswahl der linken und rechten Indikatorkarte wurde beidseitig ein Navigationsmenü integriert. Trifft der Nutzer eine Wahl, wird die entsprechende Karte anhand der gesetzten Parameter vom Server angefragt und der empfangene WMS für die gewählte Seite visualisiert. Durch die zwei autarken Navigationselemente kann der Nutzer beliebige Indikatoren und Zeitschnitte gegenüberstellen und analysieren, indem er den Slider verschiebt.

### 7.2.4. Suchfunktion

Startet der Nutzer eine Indikatorenrecherche, registriert ein Event-Listener die Eingabe und führt die zuständige Funktion aus, welche nach jeder Eingabe den Inhalt des Suchfeldes an den Server sendet. Auf dem Server überprüft das angesprochene PHP-Skript, welche Indikatornamen aus der MySQL-Datenbank dem Suchtext entsprechen. Das Ergebnis wird zurück an den Clienten gesendet und die Indikatornamen in einer Liste eingeblendet. Dabei erfolgt eine Einordnung der Ergebnisse in das Kategoriensystem des Leibniz-Institutes.

Um die eingegebenen Buchstaben aus dem Suchfeld in den gefundenen Indikatornamen farblich hervorzuheben, wurde ein sogenannter *Monkeypatch* implementiert. Hierbei handelt es sich um einen Weg, das Verhalten der Methode zu erweitern ohne diese verändern zu müssen [Pangsakulyanont, 2015]. Iterativ wird nach jeder Eingabe die Funktion zur Darstellung der Ergebnisliste um eine Methode zur Hervorhebung der passenden Buchstaben erweitert. Das Ergebnis der Implementierung wird in Abbildung 27 dargestellt.

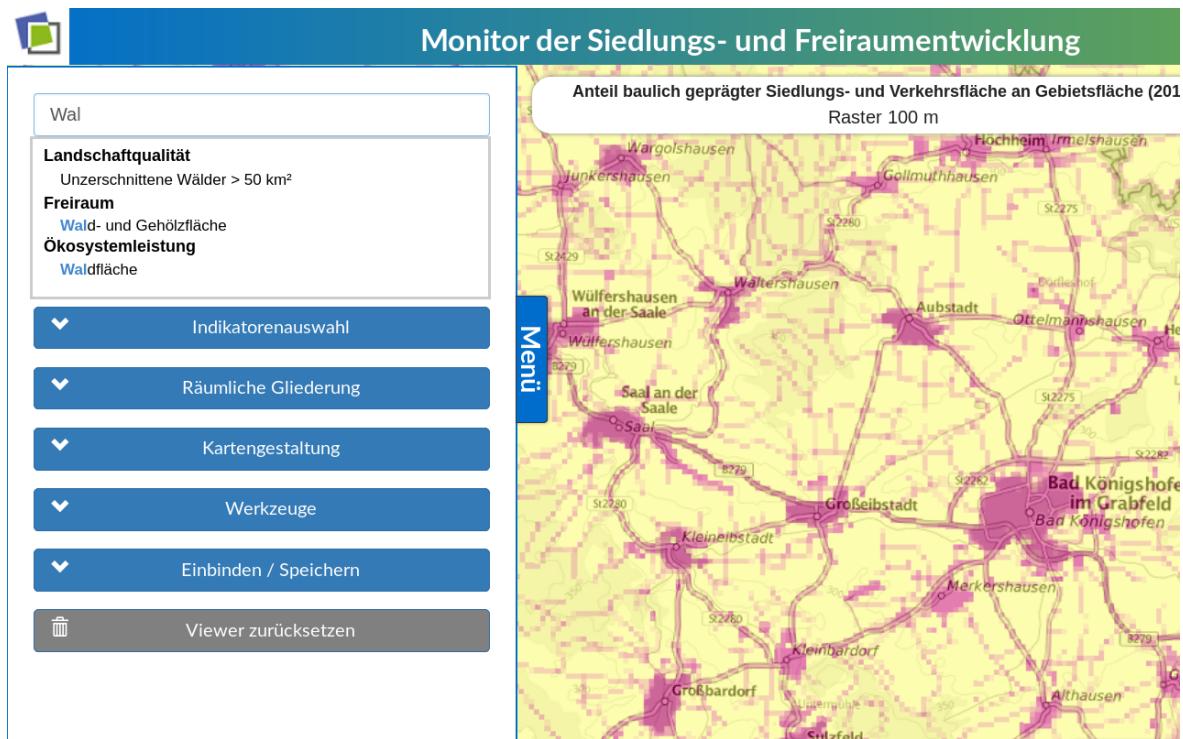


Abbildung 27: Bildschirmausschnitt der implementierten Suchfunktion nach Indikatoren mit dem *Monkeypatch*

Findet der Nutzer seinen gewünschten Indikator innerhalb der Liste, kann er diesen durch das Anklicken der Bezeichnung in der Karte visualisieren.

Basierend auf diesem Paradigma funktioniert auch die Ortssuche. Einen Unterschied stellt die Inanspruchnahme eines externen Services dar. Die Anfrage wird in diesem Fall an den Server von GeoNames<sup>39</sup> gesendet. Bei diesem Service handelt es sich um eine *freie* geographische Datenbank, die einen Suchdienst nach geographischen Orten als Webservice verfügbar macht. GeoNames aggregiert dabei Daten aus unterschiedlichsten freien und öffentlichen Datenquellen [McConnachie, 2007]. Nach jeder Sucheingabe des Nutzers wird der Inhalt des Suchfeldes an den Service geschickt. Die Anfrage URL ist in der folgenden Abbildung 28 beispielhaft für eine Sucheingabe dargestellt.



Abbildung 28: Anfrage-URL an den GeoNames Webservice

Der Suchstring ist die Eingabe des Nutzers in das Suchfeld, welcher an den Server übermittelt wird. Die anderen Parameter dienen der Sucheingrenzung, hierbei kann der Webservice

<sup>39</sup><http://www.geonames.org>

nur von registrierten Entwicklern kostenfrei genutzt werden, weshalb der Nutzernname mit angegeben wird. In dieser Anwendung erfolgt eine Anforderung einer *JSON* Datei angefordert, alternativ kann auch eine *XML* verwendet werden.

Der Datenaustausch erfolgte mit der JQuery-Bibliothek. Als Ergebnis sendet der Service alle passenden Ergebnisse, welche dem Suchstring entsprechen und in der Datenbank vorhanden sind, zurück. Zu Beginn der Sucheingabe kommt es vermehrt zu Duplikaten in der Ergebnisliste, diese werden anhand einer Schleife iteriert und eliminiert. Grund für die Entstehung der Dopplungen ist die Zwischenspeicherung der Ergebnisse innerhalb von GeoNames. Anschließend können die empfangenen Bezeichnungen gespeichert und dem Nutzer angezeigt werden. Nach jedem Eintippen eines Buchstabens in das Suchfeld erfolgt der Start einer neuen Anfrage, bis der Nutzer seinen gewünschten Ort aus der Liste gewählt hat oder keine Interaktion mit dem Suchfeld mehr erfolgt. Die Darstellung des Ortes in der Karte wird durch die mitgelieferten Koordinaten des Services möglich. Wählt der Nutzer ein Ergebnis, zeigt ihm ein Marker die räumliche Position an und ein Zoom auf diesen in der Karte erfolgt.

Die tabellarische Suchfunktion innerhalb der Tabellenansicht wurde durch eine Iteration der gesamten Tabelle gelöst. Beginnt der Nutzer einen Suchvorgang, werden alle passenden Einträge in der Tabelle herausgesucht. Sobald der erste Treffer erzielt ist, werden alle Tabellenelemente auf unsichtbar geschaltet und die passenden Zellen sortiert unter dem Tabellenkopf angezeigt.

### 7.2.5. Grundlegende Interaktionen

#### 7.2.5.1. Interaktionsbuttons

Die grundlegenden kartographischen Interaktionen sind als Buttons in die Karte integriert und mit einem Klick-Event versehen. Für die Implementierung der **Zoombuttons** konnten die bereits vorhandenen Funktionen von Leaflet genutzt werden. Mit einer Adaption an den jeweiligen Klick-Event kann der Nutzer die gewünschte Zoomstufe der Kartenansicht steuern. Um in einen entsprechenden Bereich zu zoomen, welchen der Nutzer mit dem Aufziehen eines Quadrates durch sein Zeigegerätes definierte, kam das Plugin *Leaflet .ZoomBox* zum Einsatz. Dieses kann als Open Source in die Anwendung integriert werden [Ward u. a., 2015].

Die durch den Nutzer austauschbaren **Hintergrundkarten** fanden mit der Leaflet Klasse *tileLayer* ihre Umsetzung, welche die Karten zur schnelleren Darstellung als Bildkacheln laden. Alle verwendeten Hintergrund-Rasterkarten werden in dieser Arbeit über die WMS-Schnittstelle des Anbieters abgefragt.

Nur die API des *BKG* für den *Topplus* Dienst bot eine Graustufen Version an, für den *OpenStreetMap*- und *Esri*-Dienst wurde ein Plugin von Ilya Zverev verwendet, welches die Farben der Karten durch ein „*Makeover*“ in eine Grauskala umwandelt [Zverev, 2017] und dadurch die konzipierte Farblosigkeit realisiert.

Die Generierung und Bereitstellung der **Zusatzlayer** fand nach dem Modell von Kapitel 7.2.2.1 statt, indem die Geometrien via HTTP und PHP aus der PostgreSQL abgefragt und als GeoJSON visualisiert werden.

Das durch den Nutzer verschiebbare Informationsfenster zu dem jeweils visualisierten Indikator wurde auf der Grundlage des JQuery Dialog<sup>40</sup> umgesetzt. Dieses nimmt die weiterführenden Inhalte auf, welche jeweils anhand der vom Nutzer gesetzten Parameter vom Server abgefragt werden.

Die Integration der Übersichtskarte in das Informationsfenster konnte durch eine angepasste Verwendung des Source Codes von Robert Nordan [2017] realisiert werden. Mit dieser Übersichtskarte wird das Datenalter der administrativen Gebietseinheiten oder der Rasterkarte dargestellt. Je nach gewähltem Modus wird entweder die entsprechende Raster- oder Vektor-karte angezeigt. Die zusätzlichen Informationen zu dem jeweiligen Indikator werden synchron zum Laden der Karte über eine Serverkommunikation aus der Datenbank geladen.

Für die **Legende** wurde ein selbst erstelltes Plugin implementiert, welches zu Beginn des Kartenladevorgangs eingeblendet wird. Durch Klick-Interaktionen mit dem Button kann der Nutzer die Legende schließen und auch wieder öffnen.

#### 7.2.5.2. Identify

Mit dieser Funktion wird eine punktuelle Sachdatenabfrage für die angezeigte Indikator-Karte implementiert. Klickt der Nutzer an eine gewünschte Position in der Karte, blendet ihm ein Pop-Up weiterführende Informationen zu dem gewählten Indikator an der räumlichen Selektion ein. Durch die Platzierung von Elementen mit einem Klick-Event auf der Karte muss in diesem Fall eine Funktion vorhanden sein, welche prüft, ob der Nutzer in die Karte geklickt hat oder in eines der interaktiven Elemente. Wäre diese Funktionalität nicht vorhanden, würde jedes Klick-Event innerhalb der Karte die *Identify* Funktion aufrufen. Demzufolge wird diese „*Prüf-Funktion*“ bei jedem Klick in den Kartenbereich als erstes ausgeführt. Registriert die Funktion keinen Event auf ein interaktives Element der Karte, kann auf einen Klick in die Karte geschlussfolgert und die *Identify* Funktion ausgelöst werden. Diese verwendet je nach Darstellungsmodi unterschiedliche Funktionen.

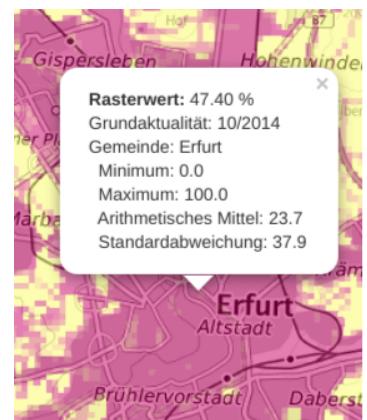


Abbildung 29: Identify Funktion

Im Gebietsviewer ist es möglich die gewünschten Informationen auf Grundlage der GeoJSON und Tabelle zu ermittelt. Klickt der Nutzer in die Karte, registriert das betroffene Gebiet den ausgelösten Event und die entsprechenden Informationen können gefiltert und im Pop-Up angezeigt werden.

Im Rasterviewer liegt jedoch keine Datei sondern eine Rastergrafik vor, anhand derer sich keine punktuellen Informationen filtern lassen. Diese Aufgabe muss an den MapServer übergeben werden, welcher durch die *GetFeatureInfo* Operation Rasterwertabfragen zu dem jeweiligen

<sup>40</sup><https://jqueryui.com/dialog>

WMS ermöglicht (vgl. Kapitel 3.4.2). Um dem Nutzer einen Bezug zu der angeklickten Koordinate zu geben, wird neben dem Rasterwert die entsprechende Gemeinde und deren Statistik angezeigt. Für die Umsetzung dieser Funktion, muss eine Ermittlung der Gemeinde anhand der Pixelkoordinate erfolgen. Diese Aufgabe wird in ein PHP-Skript auf dem Server ausgelagert, welches die Ermittlung anhand der übergebenen Koordinaten durchführt und darauf basierend die geforderten Statistiken berechnet. Als Ergebnis sendet die Funktion eine JSON-Datei zurück, aus welcher die Werte gefiltert und als Text dem Pop-Up hinzugefügt werden. Abbildung 29 stellt das Resultat eines nutzerseitigen Klicks in die Karte dar, auf welche das Pop-Up mit den geforderten Inhalten eingeblendet wird.

Im Vergleichsmodus muss vor Auslösung der Funktion noch eine Prüfung stattfinden, in welche Karte geklickt wurde, da hier zwei Karten durch einen Slider getrennt visualisiert werden. Die Unterscheidung erfolgt vom jeweiligen Abstand des registrierten Klick-Events zum Slider und führt zu einer Differenzierung der Klick-Position. Ausgehend von der Überlegung, dass der Slider durch seine Teilung der Karte als Mittelpunkt gesehen werden kann, wird die Zuordnung durchgeführt. Ist der ermittelte X-Wert kleiner als die Slider Position, kann eine Auslösung der *Identify*-Funktion für die linke Karte erfolgen. Alle X-Werte darüber lassen eine Schlussfolgerung auf die rechte Karte zu.

#### 7.2.5.3. Zeit Slider

Der konzipierte Slider für den Wechsel der Zeitschnitte wurde mit der JQuery UI<sup>41</sup> implementiert. In allen Gebietsmodi kommt die gleiche Methodik zum Einsatz, welche in diesem Kapitel erläutert wird. Wechselt der Nutzer einen Indikator, fordert eine Funktion alle verfügbaren Zeitschnitte vom Server an. Die Antwort erfolgt in Form einer JSON, in der alle Jahre als Array gespeichert sind. Diese werden ausgelesen und dem Slider als mögliche Auswahloptionen zugewiesen. Die Beschriftung übernimmt ein Plugin, welches von Simon Goellner [2016] als Open Source zur Verfügung gestellt wurde.

Empfängt der Client das JSON, werden alle Jahreszeiten gefiltert und sortiert. Die Minimal- bzw. Maximal-Werte kommen jeweils an den Beginn und das Ende des Sliders. Damit kann der Nutzer zu Beginn die mögliche Spanne für zeitliche Einstellungen erfassen. Trifft der Nutzer eine Wahl wird die entsprechende Methode ausgelöst, welche für die Kartengenerierung zuständig ist. Die Anpassung des Sliders an mobile Endgeräte wird durch *Flexible-Grids* im CSS durchgeführt. Die konzipierte horizontale Drehung des Sliders im mobilen Vergleichsmodus, übernimmt eine veränderte Parametrisierung in JQuery.

#### 7.2.5.4. Glätten der Karte

Dieser Interaktionsbutton wird dem Nutzer innerhalb der Rasteransicht angeboten, um die Interpolationsmethode des MapServers für die Rasterkarte zu variieren. Löst der Nutzer diese Funktion aus, kommt es zum Ansprechen eines PHP Skripts auf dem Server. Dieses ersetzt die standardmäßige Nearest-Neighbour- durch die Bilineare Interpolation. Hierfür wird ein neues Mapfile generiert, welches den Kartenserver anweist diese Interpolationsmethode zu verwenden. Nach der Fertigstellung des Prozesses erfolgt eine Mitteilung an den Clienten,

<sup>41</sup><https://jqueryui.com>

dass der Prozess beendet ist. Durch diese Bestätigung erfolgt ein anstoßen des neuen Ladens der Karte. Befindet sich der Nutzer im Vergleichsmodus, muss der Prozess für beide Karten durchgeführt werden. Ist die Karte geglättet, findet eine grüne Hinterlegung des entsprechenden Buttons statt, wodurch dem Nutzer signalisiert wird, dass er durch erneutes Anklicken wieder in die standardmäßige Nearest-Neighbour Interpolation zurückkehren kann.

### 7.2.6. Kartengestaltung

Der Transparenz-Slider wurde ähnlich des Zeit-Sliders umgesetzt. Für die Änderung der Transparenz stellt Leaflet eine entsprechende Funktionalität zur Verfügung. Verändert der Nutzer anhand des Sliders die Transparenz der visualisierten Indikatorkarte, werden die gesetzten Parameter an die Funktion übergeben.

Für die Anpassung der farblichen Gestaltung findet für jedes Farbschema eine Speicherung der Minimal- und Maximalwerte in einem Array statt. Diese beiden Werte werden an den Server geschickt, wo eine Durchführung der Farbreihengenerierung erfolgt. Die Antwort des Servers erfolgt als JSON Datei, anhand derer die HTML-Listen mit der farblichen Gestaltung erstellt und dem Dropdown-Menü hinzugefügt werden. Wählt der Nutzer ein Farbschema, wird eine Funktion aktiviert, welche die Farbgebung der Karte anhand der Wahl neu bestimmt und diese neu lädt.

### 7.2.7. Räumliche Gliederung

Der Slider für die Raumgliederung wurde nach der selben Grundlage wie der in Kapitel 7.2.5.3 erläuterte Zeit Slider implementiert, nur bilden hier die möglichen Parameter der Raumgliederung die Grundlage. Um die Slider-Optionen zu setzen, werden für jeden Indikator die verfügbaren Raumgliederungen vom Server abgefragt. Bewegt der Nutzer den Slider wird die Karte mit den gesetzten Parametern neu geladen, welche an die jeweilige Funktion zur Generierung der Karte übergeben werden.

Im Gebietsviewer kommt es, erweiternd zu dem Auswahl-Silder, noch zu einer Ergänzung durch ein Untermenü für die Zusammenstellung individueller Gebietseinheiten. Nach jeder Interaktion mit dem Slider erfolgt eine Anpassung der Optionen an die gesetzten Parameter. Das konzipierte Dropdown-Menü (siehe Kapitel 6.2.6) wird dabei durch die in der GeoJSON enthaltenen Gebiete gefüllt. Trifft der Nutzer eine Wahl wird die GeoJSON geleert und nur das gewünschte Gebiet in der Karte angezeigt. Das Anpassen der Bounding-Box übernimmt eine Leaflet Funktion. Jetzt kann der Nutzer beliebig viele Gebiete aus der Liste wählen und in der Karte anzeigen.

Nach jeder Selektion wird für die Nutzer-Wahl ein Tag erstellt und dem Dropdown-Menü Header hinzugefügt, womit ein schneller Überblick über die Auswahl möglich ist und eine Möglichkeit für erweiterte Interaktionen besteht.

Dabei werden die dargestellten Tags gefiltert und mit der GeoJSON synchronisiert. Diese Tags zeigen dem Nutzer an, welche Gebiete er der Karte hinzugefügt hat. Durch ein Kreuz kann er diese wieder aus der Karte entfernen. Um das geplante Layout der Tags umzusetzen (siehe Abbildung 21), bietet Semantic-UI eine Funktionalität, welche einen Handler für das

Hinzufügen und Entfernen beinhaltet. Anhand dieses Handlers kann die GeoJSON entsprechend der Nutzerwahl gestaltet werden. Da sich die Tabellenansicht aus in der GeoJSON bestehenden Gebieten zusammensetzt, erfolgt ein automatisierter Abgleich und der Nutzer bekommt die gewünschten Informationen zu den gewählten Gebieten in der Karte angezeigt.

### 7.2.8. Werkzeuge

Für die Umsetzung der Werkzeuge kamen größtenteils Plugins zum Einsatz, welche über *Github* bezogen und an die Anwendung angepasst wurden. Alle Werkzeuge können durch separate Buttons im Navigationsmenü ausgelöst und sollen folgend vorgestellt werden.

Um die konzipierte Messfunktion zu nutzen, wurde ein von Brandon Copeland [2016] programmiert Open Source Code verwendet. Anhand dieses Tools können Strecken und Flächen gemessen werden.

Das Werkzeug, welches in dieser Anwendung die Hintergrundkarte vergrößert, wird als Lupe bezeichnet und von Benjamin Becquet [2016] implementiert und als Open Source zur Verfügung gestellt.

Die Importfunktion, mit der eigene Vektorgeometrien in die Anwendung importiert werden können, stellt eine Arbeit von Simon Bats [2015] dar. Mit diesem Plugin können GeoJSON-, GPX- und KML-Dateien importiert und visualisiert werden.

Für den Fullscreen Modus wurde eine eigene Funktion erstellt. Diese versetzt den Browser in einen Fullscreen Modus und blendet dabei den Header aus. Durch Betätigen der Taste F11 oder erneutes Drücken des grün hinterlegten Buttons wird der Modus beendet.

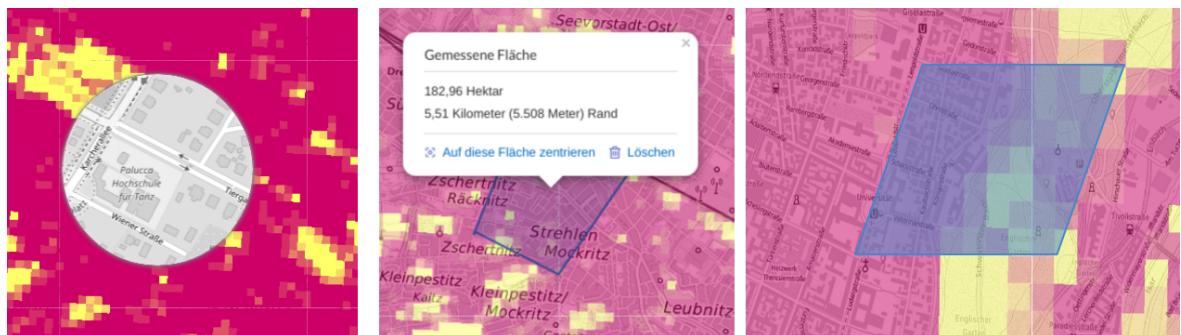


Abbildung 30: Screenshots der implementierten Werkzeuge von links: Lupe, Mess- und Importfunktion anhand einer KML

Der Code aller verwendeten Plugins musste zwingend an die Bedingungen und das Design der neuen Anwendung angepasst werden. Hierbei wurde beispielsweise die Lupe um ein Touch-Event erweitert, um diese auch auf mobilen Endgeräten nutzen zu können. Die Buttons der Messfunktion waren auf mobilen Endgeräten sehr schwer zu treffen, wenn der Nutzer nur einen Finger zu Interaktion besaß. Hier erfolgte eine Adaption in das Responsive Design.

### 7.2.9. Feedback und Nutzerunterstützung

Wechselt der Nutzer den Darstellungsmodi, wird der angezeigte Indikator und die Position der Karte beibehalten. Da jedoch nicht alle Indikatoren für jede Ansicht verfügbar sind, prüft eine Funktion, ob die Darstellung möglich ist. Wenn dies nicht der Fall ist, kommt eine Fehlermeldung mit dem Open Source Plugin von Krzysztof Śmiałek [2017] zum Einsatz. Diese Erweiterung wurde auch für alle anderen konzipierten Meldungen verwendet.

Die Prüfung erfolgt vor dem Wechsel der Ansichten, indem in der MySQL Datenbank geschaut wird, ob die gewünschte Daten des gewählten Indikators vorhanden sind. Findet die Funktion entsprechende Werte, kommt es zu einer positiven Antwort an den Clienten und die Umschaltung erfolgt. Falls eine Negativmeldung empfangen wird, erscheint die implementierte Fehlermeldung.

Die Ladeanzeige für den Visualisierungsprozess der Karte konnte mit *Leaflet.Spin* umgesetzt werden [Bats, 2016]. Diese startet bei jeder Serveranfrage und endet nach der erfolgreichen Tabellenvisualisierung. Jede Fehlermeldung unterbricht die Ladeanzeige.

Die konzipierte Tour durch die Website ist anhand des Open Source Plugin „*Bootstrap Tour*“ realisiert worden [Sossou, 2017]. Mit der Hauptaufgabe, die Neuorientierung des Nutzers in der Website zu erleichtern, kommt ihr eine wichtige Rolle zu. Auf Grundlage dieser Tour und dem Hinzufügen von eigenem Funktionen, wird der Nutzer nach einem konzipierten Drehbuch durch die Website geführt. Um nach jedem Schritt die entsprechenden Funktionen auszuführen, wurde eine Rückmeldung der Tour integriert. Sobald der Nutzer den Button für den nächsten Schritt klickt, werden die entsprechenden Funktionen ausgeführt.



Abbildung 31: Zwei beispielhafte Schritte bei der interaktiven Tour durch die Website

### 7.3. Fazit

In diesem Abschnitt fand eine erfolgreiche Umsetzung des in Kapitel 6 erstellten Konzeptes statt. Mit der erfolgten Implementierung konnten alle skizzierten funktionalen Anforderungen, sowie die Qualitätsanforderungen und Rahmenbedingungen erfüllt werden. Bei der Umsetzung wurde auf eine lose Kopplung geachtet, wodurch einzelne Komponenten ausgetauscht oder kombiniert werden können. Auf dieser Grundlage sind zukünftige Änderungen ohne große Umstrukturierungen der Anwendung möglich.

## 8. Evaluierung

Diese Kapitel widmet sich der Bewertung des IÖR-Monitors, indem die Qualität der implementierten Anwendung evaluiert wird. Mit dieser Untersuchung sollen Schwachstellen des Prototypen identifiziert und die vorangegangene Implementierung mit der aktuellen verglichen werden.

### 8.1. Performance Test

Geschwindigkeit und Verfügbarkeit aller Komponenten beeinflussen die Qualität einer Anwendung ebenso wie die Gestaltung. Aus diesem Grund ist es wichtig, auch diese Komponenten in einer Evaluierung näher zu betrachten [Krüger, 2011].

Mit der in diesem Abschnitt beschriebenen Evaluierung, sollen die Ladezeiten für administrative Gebietseinheiten innerhalb der beiden Anwendungen verglichen werden. Die Tests wurden auf einem *ASUS* Notebook mit einem *Intel i7* Prozessor der neuesten Generation durchgeführt. Als Browser diente die aktuellste Version von *Google Chrome*<sup>42</sup>, während des Tests lag eine Datenrate von 32,92 Mb/s an. Für den sogenannten Benchmark-Test fand eine Validierung der Berechnungszeit statt, welche die Anwendungen zur Generierung und Visualisierung der administrativen Gebietseinheiten benötigten. Da in der alten Anwendung die Tabelle nicht synchronisiert zu der Karte lädt, sondern dies erst durch den Nutzer angestoßen werden muss, wurden die beiden Ansichten getrennt gemessen und zusammenfassend addiert. Das Ergebnis der Untersuchung ist in der folgenden Abbildung 32 zusehen.

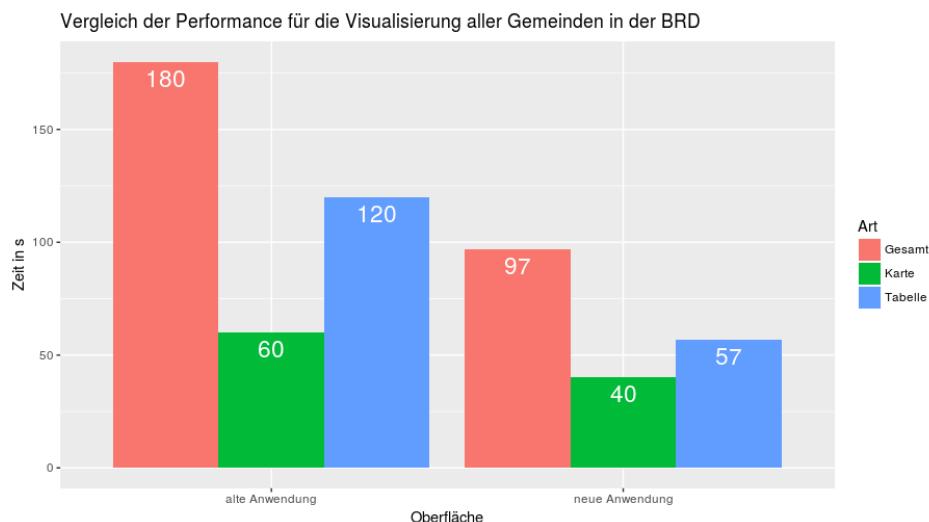


Abbildung 32: Vergleich der Ladezeiten für die Visualisierung der Gemeinden, in der neuen (NO) und alten Anwendung (AO)

Die erzielten Ergebnisse können jedoch nicht als Standard gesehen werden, da eine Erreichung der Werte stark von der technischer Ausstattung des Clienten, der Internetverbindung und serverseitiger Auslastung abhängen.

Betrachtet man die Bilanz, wird deutlich, dass vor allem mit dem neu implementierten Lösungsansatz für die Generierung einer Gebietskarte eine Steigerung erreicht wurde. Anhand

<sup>42</sup><https://www.google.de/chrome>

der Umstellung auf Leaflet mit einer GeoJSON-Datei als Ersatz für die SVG-Technologie, konnte eine vollständige Visualisierung in 97 Sekunden erfolgen, wohingegen die alte Anwendung eine Zeit von 180 Sekunden benötigte. Damit konnte die benötigte Dauer um die Hälfte gesenkt werden.

Aber in welcher Signifikanz liegt eine Steigerung der Performance vor? Diese Fragestellung lässt sich durch einen Wilcoxon-Test ermitteln. Dieser Test prüft, ob sich die Verteilungen der Grundgesamtheiten zweier Stichproben bezüglich ihrer Lage unterscheiden [Maike, 2015]. Mit Hilfe dieser Überprüfung konnte ein p Wert von 0,03 ermittelt werden, welcher einen sehr signifikanten Unterschied der vorliegenden Daten darstellt.

Eine weitere Schlussfolgerung aus den validierten Ergebnissen geht aus der Ladezeit hervor, welche der Nutzer abwarten muss, bis die gewünschte Visualisierung im Browser erscheint. Hier rechtfertigt die gemessene lange Ladezeit der Gemeinden den implementierten Hinweis durchaus, um den Nutzer zu informieren, dass für die vorliegende Visualisierungsarbeit überdurchschnittlich viel Zeit benötigt wird.

Der Test untersuchte als einen weiteren Punkt auch die Visualisierungsdauer der anderen Gebietseinheiten, deren Resultate in der folgenden Abbildung dargestellt sind. In diesem Testabschnitt erfolgte die Darstellung der Ergebnisse für Karten- und Tabellenansicht, als zusammengefasste Addition der ermittelten Werte.

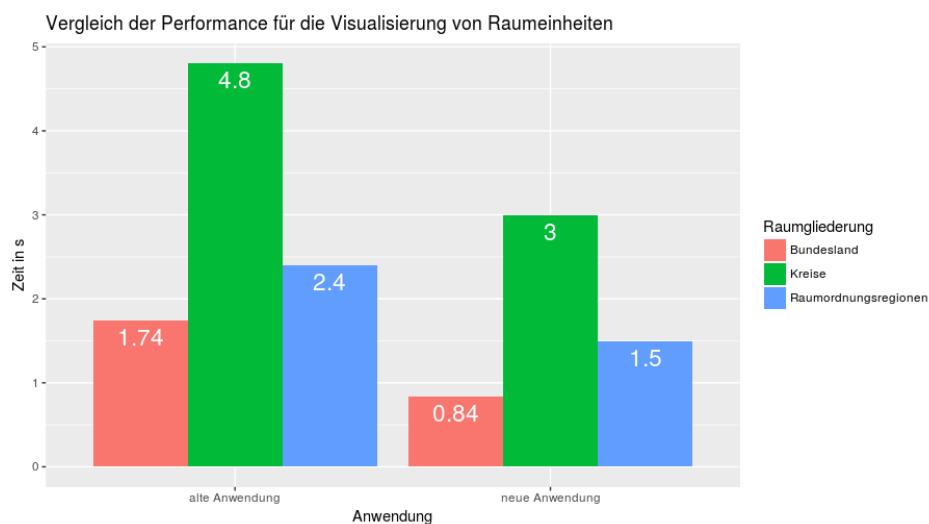


Abbildung 33: Vergleich der Ladezeiten administrativen Gebietseinheiten

Auch hier lagen Unterschiede in der Visualisierungsdauer vor, welche durch den Wilcoxon-Test als „sehr signifikant“ eingestuft werden konnten. Damit zeigt auch hier die neue Anwendung eine performantere Leistung, als die alte Anwendung .

## Zusammenfassung

Anhand dieses Tests konnte ermittelt werden, wie die Verwendung eines Smart Client und aktueller Technologien einen signifikanter Unterschied, in Bezug auf die Performance der Kartengenerierung erzielen. Dieses Ergebnis belegt, dass mit dem umgesetzten Layout und der damit verbunden Ausdehnung der Karte auf den gesamten Bildschirm, keine Geschwindi-

keitsbeeinträchtigungen einhergehen. Vielmehr konnte die Performance gesteigert werden und damit eine mögliche Erhöhung der Nutzererfahrung stattfinden, welche im folgenden Kapitel evaluiert wird. Hiermit demonstriert die implementierte Methodik die performanteste Zeit und bestätigte damit das gestellte Kriterium, die Generierung der Choroplethenkarte zu beschleunigen.

## 8.2. Nutzertests

Um das Verhalten des Prototyps im Hinblick auf die gestellten Anforderungen zu überprüfen, muss die in dieser Arbeit entstandene Anwendung einer Evaluierung unterzogen werden, die in diesem Kapitel beschrieben wird. Am Beginn findet eine Dokumentierung allgemeiner Punkte statt, wie etwa die Art der Evaluierung sowie Aufbau und Ziele. Anschließend erfolgt eine Diskussion der erzielten Resultate in einem gesonderten Bereich, dies geschieht anhand der gestellten Anforderungen.

### 8.2.1. Ziele und Art

Dieser Test hat das Ziel, die Qualität der neuen Anwendung für den IÖR-Monitor zu überprüfen und zu testen, wobei erste Schwachstellen identifiziert werden sollten. Neben der qualitativen Validierung soll auch ein Vergleich zwischen dieser Arbeit und der existierenden Implementierung erfolgen. Anhand der Gegenüberstellung fand eine Hinterfragung statt, ob die anfangs gestellten Anforderungen an die neue Anwendung erreicht werden konnten und demzufolge auch die anvisierte Steigerung der Usability erfolgt.

Teilaspekte, die in dieser Evaluierung im Vordergrund standen, waren unter anderem das Verständnis in Bezug auf den Aufbau der Anwendungen. Hier galt es zu erörtern, ob der Benutzer die einzelnen Darstellungsbereiche richtig erkennen und einordnen konnte.

Ein weiteres Kriterium befasste sich mit den angebotenen Interaktionsmöglichkeiten. Diese wurden angesichts ihres Verständnisses und intuitiv richtigen Einsatz validiert.

Das Nutzerverständnis hinsichtlich des Zweckes, den die unterschiedlichen Sichten erfüllen sollten, war ebenfalls Gegenstand der Evaluierung.

Durch diese Studie fand keine Überprüfung von Hypothesen statt, sondern vielmehr eine Erkundung, wie die Neuausrichtung der Oberfläche bei potentiellen Nutzern aufgenommen wird. Empirische Untersuchen mit dieser Erkundungsabsicht werden auch unter der Bezeichnung *pilot study* geführt [vgl. Hegner, 2003, S.46].

### 8.2.2. Aufbau

Für die Evaluierung kam ein Fragebogen (siehe Anhang A.2) zum Einsatz, der eine Basis für die Validierung beider Anwendungen bildete. Um den Fragebogen nach dem Untersuchungsdesign auszurichten, fand eine Unterteilung in drei Teile statt.

Der ersten Teil diente der Filterung des Kenntnisstandes mit WebMapping bzw. WebGIS-Anwendungen. Hier waren mögliche Antworten bereits vorgegeben und der Proband konnte skizzieren, in welchem Grad er den Aussagen zustimmte. Durch diese Fragestellung konnte

eine Einteilung der Probanden in Experten und erfahrene Nutzer erfolgen. Folgende Fragen sollten anhand einer Likert-Skala beantwortet werden:

1. Wie häufig verwenden Sie Web-GIS Anwendungen privat oder im Studium/Beruf?
2. Wie gut kommen Sie im Allgemeinen mit den bisher genutzten Web-GIS Anwendungen zurecht?
3. Wie eingehend kennen Sie den IÖR-Monitor?

Der zweite Teil bezog sich auf den Hauptteil der Evaluierung und beinhaltete die Durchführung von gegebenen Aufgaben, welche anhand von Interaktionen mit den beiden Anwendungen gelöst werden konnten.

Hier wurde versucht zu ermitteln, wie der Nutzer die Effizienz der Anwendungen bewertet, welche er für die jeweilige Aufgabe verwendete. Um diese zu messen, sollten die Probanden nach jeder Aufgabe eine Note zwischen eins und fünf vergeben. Vergab der Proband die Note eins, signalisierte er damit seine volle Zustimmung. Mit der Note fünf konnte er seine volle Ablehnung ausdrücken. Die Testaufgaben waren für beide Anwendungen identisch, um einen späteren Vergleich der Ergebnisse zu ermöglichen. An diesem Test sollten Experten und weniger erfahrene Nutzer teilnehmen, wodurch der Schwierigkeitsgrad der Aufgaben ein ausgewogenes Mittelmaß bekommen musste. Bei der Aufgabenformulierung wurde versucht reale, analytische Fragestellungen von potentiellen Nutzern zu formulieren, welche mit den beiden Anwendungen gelöst werden können. In der folgenden Tabelle 6 sind die Aufgaben, mit der dazugehörigen Zielstellung im Test, gelistet.

| <b>Formulierung der Aufgabe</b>  | <b>Zielstellung im Test</b>   |
|--|---|
| Wie hat sich Ihrer Meinung nach die Siedlungsfläche in Niederrossig entwickelt, betrachten Sie dafür die Zeitschnitte 2006 und 2016 ?  | Ziel dieser Aufgabe war die Evaluierung des implementierten Vergleichsmodus innerhalb der neuen Anwendung , verglichen mit den Möglichkeiten der alten Anwendung . Ein weiterer wichtiger Untersuchungspunkt stellte hier die neu implementierte Ortssuche dar. |
| In welchen Städten ist für das Jahr 2015 das Überschwemmungsgebiet am höchsten? Visualisieren Sie das Ergebnis in der Kartenansicht.   | In dieser Aufgabe fand ein Vergleich des MultiView mit dem implementierten Lösungsansatz der alte Anwendung statt.  |
| Sie möchten anhand der räumlichen Gliederung den Anteil an Landschaftsschutzgebieten für das Jahr 2012 in vier Kreisen vergleichen, folgende Kreise sollen analysiert werden: Sächsische Schweiz-Osterzgebirge, Garmisch-Partenkirchen, Bautzen, Kiel. Stellen Sie nur diese vier Kreise in der Tabellen- und Kartenansicht dar. Nach der Zusammenstellung merken Sie, das Kiel und Bautzen fälschlicherweise gewählt wurden, diese sollen jetzt wieder entfernt werden. | Es sollte festgestellt werden, ob die neu implementierte Funktionalität zur Zusammenstellung von Gebietseinheiten den Nutzer zu mehr Effizienz und Zufriedenheit verhilft.  |
| Ihnen gefällt die aktuelle Farbgebung des Indikators nicht, weshalb eine Grauwerts Skala verwendet werden soll. Nach der Interaktion merken Sie, das die Farbe doch nicht geeignet ist, weshalb die Karte in die ursprünglichen Farbgebung zurückgesetzt werden soll. Der Indikator und Zeitschnitt kann für diese Fragestellung nach Belieben gewählt werden.   | Der Hauptfokus dieser Aufgabe lag in der Überprüfung der neu implementierten Farbwahl. Wie empfinden die Nutzer die Art der Auswahl und wo liegen die Schwächen?  |

Tabelle 6: Aufgaben des Nutzertests und deren Ziel

Als abschließender Teil des Aufgabenblockes sollten die Probanden eine Tour durch die neue Anwendung unternehmen, indem die dafür vorgesehene Funktionalität genutzt wurde. Ziel dieses Teiles war neben der Suche nach Schwachstellen, die geplante Hilfe in ihrer Aufgabe als Stütze zur Neuorientierung zu evaluieren.

Im dritten Teil des Fragebogens hatten die Teilnehmer die Möglichkeit, den Gesamteindruck der Anwendungen zu bewerten. Dieser Teil baut auf die Eindrücke auf, welche durch die Interaktionen mit den Oberflächen gewonnen wurden.

Um die Bearbeitungszeit so gering wie möglich zu gestalten, integriert der Schlussteil bereits vorformulierte Fragen, welche anhand von fünf Bewertungsmöglichkeiten für jede Anwendung eingestuft werden konnten. Vorgegeben wurden dabei fünf Optionen beginnend bei sehr gut, über eine neutrale Bewertung, bis hin zu mangelhaft. Mit der Verwendung dieser möglichen Antwortgebung kann ein durch den Nutzer schnelles und effizientes Beantworten der Fragen erfolgen, ohne seine Zeit höher zu beanspruchen [vgl. Jotz, 2015]. Folgende Fragen wurden zur Erfassung des Gesamteindrucks formuliert:

8. Wie gut bist du insgesamt mit den Anwendungen zurechtgekommen ?
9. Bitte bewerte die Übersichtlichkeit der Oberflächen.
10. Wie gefiel dir die Reaktionsgeschwindigkeit der Anwendungen?
11. Wie verständlich waren Beschriftungen und Symbole in den Oberflächen ?

Um persönliches Feedback zu hinterlassen, fand eine Integration von zwei Textfeldern in den Fragebogen statt, welche die Probanden nutzen konnten um besonders positive oder negative Punkte textlich festzuhalten. In einer abschließenden Frage konnten die Probanden den Grad angeben, mit welchem sie die beiden Anwendungen weiterempfehlen würden. Hierfür fand nach dem Vorbild der vorangegangenen Fragen, eine Likert-Skala mit fünf Optionen ihren Einsatz.

Das gewählte Erhebungsdesign besitzt innerhalb der Aufgabenböcke den Nachteil, dass die Reihenfolge, in der die Teilexperimente durchgeführt werden, einen Störfaktor darstellt [vgl. Preim u. Dachselt, 2015, S.182]. Die dadurch auftretenden Lerneffekte können durch *Counterbalancing* eliminiert werden. Zur Realisierung dieses Prinzips wurden die Probanden in Gruppen aufgeteilt, welche die Aufgaben in unterschiedlicher Reihenfolge bearbeiteten. Um einen Spannungsbogen aufrecht zu erhalten, begann und endete der Test mit einer leichteren Aufgabe.

### 8.2.3. Durchführung

Bevor der eigentliche Test mit potentiellen Nutzern durchgeführt wird, empfiehlt Nielsen [1998] eine Evaluierung des Fragebogens und dazugehörigen Aufgaben. In diesem sogenannten *Pretest* kommt es zu einer empirischen Prüfung des Fragebogens.

In dieser Arbeit wurde der *Pretest* eingesetzt, um vor allem die Verständlichkeit der Aufgabenbeschreibungen zu validieren. Die Durchführung erfolgte mit einem Probanden, wobei verschiedene Unklarheiten an der Aufgabenstellung identifiziert und der konzipierte Fragebogen überarbeitet werden konnte. Der verbesserte Fragebogen stellte die finale Validierungs-

grundlage des eigentlichen Nutzertests dar.

Für den Test haben sich insgesamt neun Probanden bereit erklärt. Die Probandenzahl erscheint etwas gering für einen solchen Test, jedoch schreibt Nielsen [1999] diesbezüglich, dass bei Usability-Tests fünf Probanden ausreichen um ca. 75% der Fehler aufzudecken, jeder weitere findet sehr häufig bereits bekannte Fehler.

Mit einer durchschnittlichen Erfahrung von 4.3 Jahren (Standardabweichung: 1.87) mit Web-Mapping bzw. WebGIS Anwendungen, konnten alle Teilnehmer als erfahrende Nutzer eingestuft werden. In der folgenden Abbildung 34 sind die Vorkenntnisse der Probanden mit dem IÖR-Monitor abgebildet.

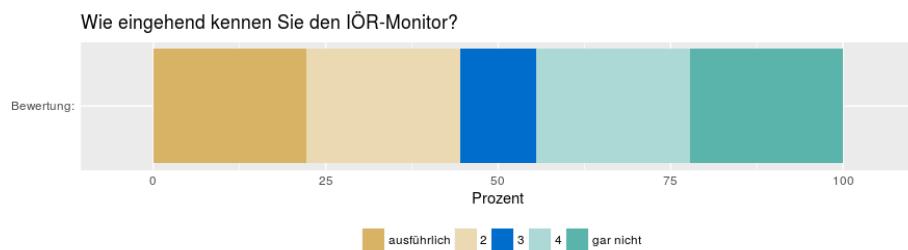


Abbildung 34: Vorkenntnisse der Probanden mit dem IÖR-Monitor

Die Tests fanden als qualitative Einzeltests in einer Liveumgebung statt. Für die Tests wurde ein Desktop-PC des Leibniz-Institutes verwendet, welcher einen 22 Zoll Monitor besaß. Als Betriebssystem kam Windows 7 zum Einsatz. Um die Anwendung aufzurufen wurde *Google Chrome* verwendet.

Nach Begrüßung der Probanden folgte ein Einleitungsgespräch. In diesem Gespräch wurden die Teilnehmer über das Ziel der Studie informiert und wichtige Fachbegriffe und Funktionen aufgeklärt. Durch das anschließende Vorlegen des Fragebogens und der Bitte den ersten Teil auszufüllen, startete der Hauptteil. Nach dem Fertigstellen dieses Abschnitts begannen die Aufgaben mit den beiden Anwendungen. Um den aktuellen Eindruck des Probanden festzuhalten, fand eine Befragung im direkten Anschluss der Aufgabenfertigstellung statt. Hier wurden die Teilnehmer gefragt, welche Bewertung sie für die Effizienz der jeweiligen Anwendung vergeben würden.

Hatten Probanden Probleme bei der Aufgabenbewältigung, wurden durch den Testleiter vorher definierte Hilfestellungen gegeben und diese in separaten Aufzeichnungen vermerkt. Um lautes Denken und Beobachtungen festzuhalten, war es wichtig diese während der probandenseitigen Interaktion schriftlich zu festzuhalten.

Im abschließenden Teil sollten die Teilnehmer ihren Gesamteindruck wiedergeben, welchen sie während der Aufgaben gewonnen hatten. Hierfür stand der Fragebogen bereit, in welchem die Beurteilung erfolgte.

### 8.3. Ergebnisse

In diesem Kapitel sollen die erzielten Ergebnisse diskutiert werden. Der Hauptaugenmerk wird dabei auf die durchgeführten Aufgaben und deren Bewertung durch die Probanden gelegt. Im letzten Teil wird der Gesamteindruck beider Anwendungen analysiert.

#### 8.3.1. Aufgaben

Die erste Aufgabe konnte von allen Probanden gut gelöst werden. Jedoch zeigte die Herangehensweise Unterschiede auf. Um den gewünschten Ort zu suchen, standen in beiden Anwendungen Suchfelder zur Verfügung, welche von allen Probanden erkannt wurden. Bei der alten Anwendung warteten fünf Nutzer bei der Suche auf das Einblenden von Ergebnissen, was sie aus bekannten Suchfunktionalitäten anderer Dienste ableiteten. Da diese Funktion nicht implementiert war, wurde der komplette Suchbegriff eingegeben und gefunden.

In der neuen Anwendung konnte die implementierte Auto vervollständigung alle Nutzer zu einer schnelleren Lösung der Aufgabe verhelfen.

Mit dem Fehlern eines Markers in der alten Anwendung welcher Nutzern das Suchergebnis räumlich verortet, veranlasste vier Probanden zu erweiterten Suchprozessen und zeigt die Berechtigung dieser Kenntlichmachung in der neuen Anwendung. Ein weiteres Problem trat in der *Monkey-Patching* Funktion der Neuimplementierung auf. Hatte der Proband den gewünschten Ort in der Auto vervollständigungsliste gefunden, bestand die erste Interaktion darin die Enter-Taste zu drücken, um das Ergebnis in der Karte darzustellen. Diese Option war jedoch nicht vorhanden und erforderte eine erneute Interaktion, indem das Ergebnis angeklickt wurde. Hier muss in Zukunft eine Möglichkeit dieser Tastenkombination implementiert werden.

Durch die Konzeptionierung und Implementierung des Vergleichsmodus, sollte eine Möglichkeit geschaffen werden, beliebige Rasterkarten gegenüberzustellen und damit eine räumliche Analyse durchzuführen. Diese Neuerung fand bei allen Probanden eine überaus positive Akzeptanz und konnte damit ihren Sinn und Zweck bestätigen. Um die gleiche Aufgabe in der aktuellen Implementierung des IÖR-Monitors zu lösen, mussten die Zeitschnitte herausgesucht und visuell geprüft werden, was viele Nutzer als sehr umständlich empfanden. Die Ergebnisse der Aufgabe in der folgenden Abbildung 35 heben deutlich hervor, wie die neue Anwendung eine durch die Probanden empfundene Steigerung gegenüber der vorangegangen Implementierung, zum Lösen der Aufgabe erreichen konnte.

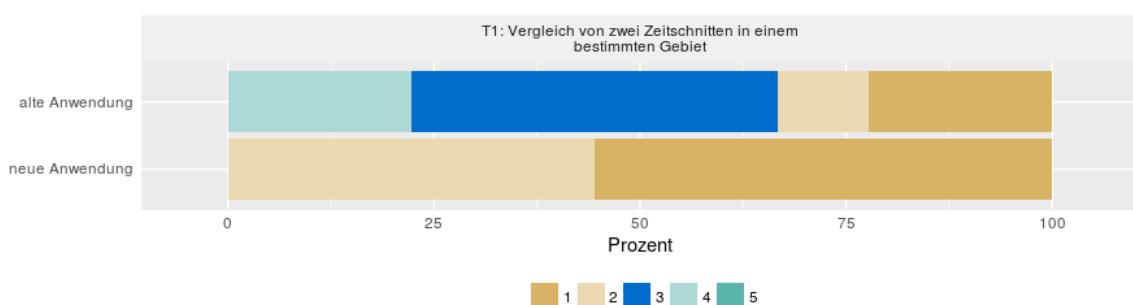


Abbildung 35: Auswertung von Aufgabe 1

In der zweiten Aufgabe zeigte sich deutlich wie der Implementierungsansatz des aktuellen IÖR-Monitors die essentielle Funktionalität der Tabellenansicht zu wenig hervorhebt und fehlende Interaktion zwischen den beiden Ansichten Probleme bereiteten.

Probanden mit wenigen Vorkenntnissen im Bezug auf den IÖR-Monitor, versuchten die Aufgabe anhand der Kartenansicht zu lösen. Nachdem kein Lösungsansatz gefunden wurde, brachte eine Hilfestellung den Erfolg und damit ein Auffinden der Tabellenansicht.

Um sich die Stadt mit der höchsten Indikatorausprägung in der Karte anzeigen zu lassen, steht eine entsprechende Markierungsoption in der alten Anwendung bereit. Diese führt jedoch nicht synchron zu einer Darstellung der entsprechenden Gebietseinheit in der Kartenansicht. Erst ein erneutes Laden der Karte mit F5 visualisiert die selektierten Gebiete. Zur Bewältigung dieser aneinander gereihten Interaktionen war kein Proband in der Lage, es mussten Hilfestellungen gegeben werden. Ist die Funktion jedoch bekannt, stellt sie keine Problematik mehr dar.

In der neuen Anwendung wurde die Aufgabe durch den MultiView von allen Teilnehmern schnell gelöst. Zwei Probanden empfanden die ständige *Mouseover*-Synchronisierung zwischen der Tabellen- und Kartenansicht als störend und wünschten sich eine Funktion, diese abzuschalten. Durch die aktuelle Implementierung kann dies nur über eine Deaktivierung der Tabellenansicht geschehen. Für diesen Fall sollte in zukünftigen Entwicklungsprozessen eine Möglichkeit integriert werden. Abbildung 36 dokumentiert, wie durch die Implementierung des MultiView ein Anstieg der positiven Beurteilung der Effizienz erreicht wurde, indem die Tabellensicht integriert und durch Interaktionsmöglichkeiten zwischen beiden Ansichten verbessert wurde.

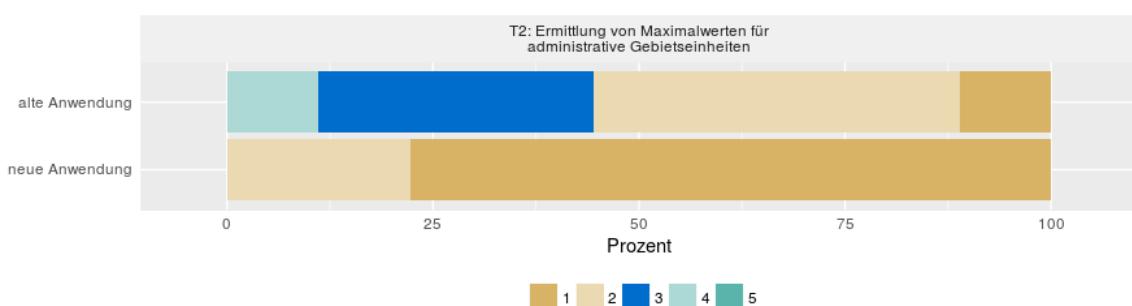


Abbildung 36: Auswertung von Aufgabe 2

Um sich in der dritten Aufgabe eine individuelle Gebietskulisse aus Kreisen zusammenzustellen, mussten die Probanden in beiden Anwendungen unterschiedliche Wege gehen. In der alten Anwendung passiert dies durch Drücken der *STRG* oder *Shift* Taste. Erfolgt die Wahl eines Kreises durch die implementierte Durchsuchungsfunktion des Dropdown-Menüs, wird in Kombination mit der Tastenkombination die Auswahl auf den selektierten Kreis forciert und die vorherige Wahl zurückgesetzt. Dieses Interaktionsschema führte dazu, dass sieben Probanden die Aufgabe nicht lösen konnten und Hilfestellungen bekamen, um sich die Kreise anzeigen zu lassen. Das Herausnehmen von einzelnen administrativen Gebieten aus der getroffenen Wahl, konnte von keinem Teilnehmer realisiert werden.

In der neuen Anwendung waren alle Probanden fähig, die Aufgabe ohne Probleme lösen und damit die Implementierungsansatz durch Tags (siehe Kapitel 6.2.6) als effizientere Lösung

zu bestätigen. Eine Schwachstelle der neuen Anwendung konnte bei vier Probanden beobachtet werden. Durch das zu Beginn eingefahrene Dropdown-Menü, welches die möglichen Kreise beinhaltet, war die Auswahlmöglichkeit nicht ganz eindeutig und ein Suchprozess begann. Alle Probanden waren aber in der Lage die Funktionalität eigenständig zu finden. Drei Probanden versuchten die Gebiete mit der Tabellenansicht zusammenzustellen, was jedoch nicht implementiert wurde. Hier musste eine Hilfestellung gegeben werden, um die Aufgabe zu lösen. Dieser Fakt löste die Überlegung aus, diese Funktion in den Weiterentwicklungsprozess zu integrieren und damit eine multiple Auswahlmöglichkeit einzuführen. Das Ergebnis der Auswertung aus Aufgabe drei ist in Abbildung 37 abgebildet und bestätigt den neuen Interaktionsansatz dieser Arbeit, bei dem die Tastenkombination durch Tags ersetzt wurde.

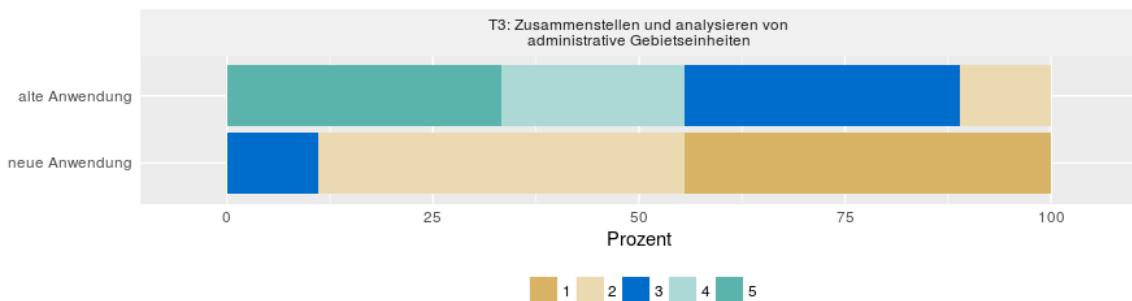


Abbildung 37: Auswertung von Aufgabe 3

Die letzte Aufgabe war als eine leichte Schlussaufgabe gedacht, welche von allen Probanden in beiden Anwendungen bewältigt wurde. In der alten Anwendung fand eine Weiterleitung der Probanden in ein neues Fenster statt, wo die Selektion der gewünschten Parameter möglich war. Für die neue Anwendung wurde die Farbwahl aus Interaktions-Schemas von modernen Desktop-GIS Anwendungen abgeleitet.

Um die Wahl zu bestätigen, mussten die Probanden in der alten Anwendung einen Zurück-Button verwenden, welcher von allen Teilnehmern erst nach einem Suchprozess gefunden wurde. Die erste Interaktion um zurück auf die Kartenansicht zu gelangen, war meist das Verwenden der Browser Funktion *Zurück*. Dies blieb aber ohne Ergebnis, da die Anwendung die Farbwahl erst nach dem Klicken des Buttons durchführt.

Die meisten Nutzer fühlten sich von der Vielfalt an Möglichkeiten in der alten Anwendung überfordert, empfanden aber die Möglichkeit eigene Farben durch Eingeben von Farbcodes zu erstellen sehr positiv. Diese Möglichkeit fehlt in der neuen Anwendung, da die Festlegung hier über vordefinierte Farbskalen erfolgt. Diese Funktionalität sollte in den Aufgabenkatalog für potenzielle Erweiterungen addiert werden. In der neuen Anwendung wurde die Funktion zur Farbwahl ohne Komplikationen von allen Teilnehmern der Studie gefunden und als effizient bewertet. Die folgende Abbild dokumentiert die Bewertungen der Probanden für Aufgabe vier.

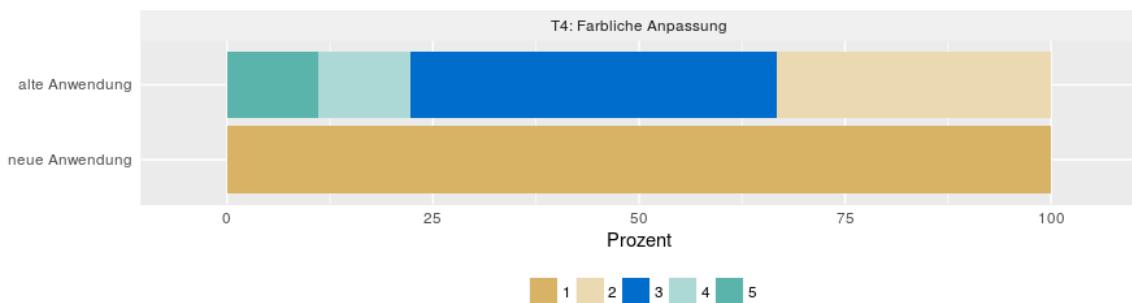


Abbildung 38: Auswertung von Aufgabe 4

In der letzten Aufgabe fand eine Evaluierung der implementierten Tour durch die Website statt. In diesem Teil des Tests durchliefen die Probanden diese Funktion, welche von allen ein sehr positiv Feedback fand. Das Fehlen einer Fortschrittsanzeige nannten drei Probanden als wichtiges Feature um zu sehen, wann die Tour endet. Alle Probanden sprachen sich dafür aus, anhand der Funktionalität einen schnelleren Einstieg in die Website zu finden. In der aktuellen Implementierung wird die Tour als eine Gesamttour durch alle Darstellungsarten durchlaufen, was drei Nutzer als negativ empfanden. Sie empfahlen für jeden Modus eine separate Tour zu realisieren, welche jeweils in die wichtigsten Funktionen der Ansicht einführt.

Für alle Aufgaben konnten allgemeine Beobachtungen gemacht werden, welche sich durch alle Aufgaben hindurch zogen. Eine davon wurde im Interaktionsschema zur Auswahl der Indikatoren sichtbar. Nutzer, welche bereits Erfahrung mit dem IÖR-Monitor besaßen, verwendeten zielstrebig das Dropdown-Menü der neuen Anwendung zur Auswahl der Indikatoren und ignorierten die neu implementierte Suchfunktion.

Probanden mit weniger Erfahrung nutzten sofort die Indikatorsuche und hatten demzufolge Probleme bei der Indikatorauswahl innerhalb der alten Anwendung da sie hier durch das Fehlen der Suche erst den gewünschten Indikator heraussuchen mussten.

In der neuen Anwendung hatten vier Probanden zu Beginn Probleme, das ausfahrbare Navigationsmenü zu entdecken, welches dann nach einer kurzen Orientierungszeit gefunden wurde.

### 8.3.2. Gesamteindruck

In diesem Teil des Fragebogens bestand für die Probanden die Möglichkeit, den Gesamteindruck der beiden Anwendungen zu bewerten, welchen sie während dem Lösungsprozess und den damit verbundenen Interaktionen gewonnen hatten. Betrachtet man die Auswertung der Fragen in Abbildung 39 wird deutlich, dass die neue Anwendung bei allen Probanden einen positiveren Eindruck hinterließ.

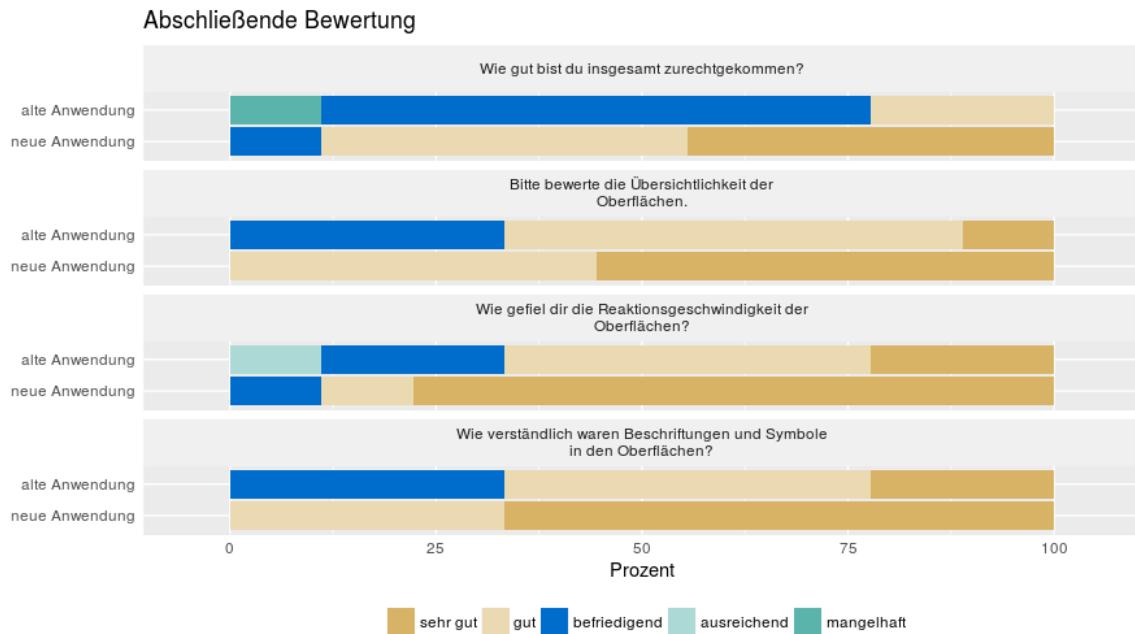


Abbildung 39: Auswertung des Gesamteindruckes

Vor allem die intuitive Gestaltung der Oberfläche und schnellere Reaktionsgeschwindigkeit waren einige der am meisten genannten zustimmenden Aspekte. Auch die Implementierung der Auto vervollständigung und die Zusammenführung der Karten- und Tabellenansicht empfanden alle Probanden als eine sehr wichtige Neuerung. Fünf Teilnehmer merkten an, dass bei einem erstmaligen Besuch die Navigationsleiste ausgefahren werden sollte, da diese sonst schwer ersichtlich ist.

In allen Anwendungen wurde der gebotene Funktionsumfang besonders hervorgehoben. Die Neu strukturierung des Navigationsmenüs stieß durchweg auf ein positives Feedback. Die Evaluierung zeigte hier eine bessere Orientierung der Nutzer, bei der Suche nach funktionalen Möglichkeiten.

Auf die Fragestellung, ob die Probanden die jeweilige Anwendung auch anderen empfehlen würden, lag der Durchschnitt in der neuen Anwendung bei einer Einstufung in „völlig“. Die alte Anwendung wurde durchschnittlich mit „neutral“ bewertet.

## Zusammenfassung

Vergleicht man die Kritikpunkte der Nutzertests mit den gefilterten Anmerkungen der Benutzeranalyse (siehe Kapitel 5.1), können Parallelen gezogen werden. Viele Probleme der alten Anwendung innerhalb dieser Evaluierung, wurden auch in der Benutzeranalyse genannt. Sehr erfreulich ist, dass die neue Anwendung diese Kritikpunkte erfolgreich beseitigen konnte, was durch die Ergebnisse der Evaluierung belegt wurde. In allen Fragen hat die in dieser Arbeit entstandene Implementierung einen überzeugenden Eindruck hinterlassen und konnte demonstrieren, dass ein Mehrwert für den IÖR-Monitor erreicht wurde. Dies liegt unter anderem in der konsequenten Umsetzung der zu Beginn der Realisierung ermittelten Spezifikationen. Die implementierten Interaktionen stellen gut gewählte Verbesserungen bzw. Erweiterungen dar, welche nach der Behebung der identifizierten Schwachstellen einer breiten Öffentlichkeit zugänglich gemacht werden können.

Durch die implementierte Feedbackfunktion besteht auch im aktiven Betrieb der Anwendung die Möglichkeit, potentiell wichtige Verbesserungsvorschläge der Nutzer einzusammeln, was eine ständige Verbesserung der Anwendung und Identifizierung von weiteren Schwachstellen möglich macht.

Die tatsächlich erzielten Fortschritte lassen sich jedoch erst mit einer großen Anzahl an Nutzern verifizieren. In diesem Kapitel fand eine Evaluierung von ausgewählten analytischen Funktionalitäten statt, welche nur einen Teil der reimplementierten Funktionen betreffen. Hier müssen noch weitere Überprüfungen erfolgen, um ein finales Produkt mit einer breiten Nutzerakzeptanz zu ermöglichen.

#### **8.4. Fazit**

In diesem Kapitel wurde die Realisierung einer neuen Anwendung für den IÖR-Monitor auf vorher festgelegte Kritikpunkte untersucht und mit der alten Anwendung durch eine Gegenüberstellung verglichen. Um dieses Ziel zu erreichen, fanden zwei Evaluierungen statt, welche überprüften, ob die gestellten Anforderungen (siehe Kapitel 5.5) erfüllt wurden.

Als Ergebnis konnte eine Verbesserung der Nutzererfahrung und Performance mit dem zugrundeliegenden Konzept bestätigt werden. Aber auch Schwachstellen sind deutlich geworden. Diese müssen in die Weiterentwicklung der Anwendung einfließen und ein erneuter Test muss zeigen, ob diese tatsächlich zur Verbesserung beitragen.

## 9. Zusammenfassung und Ausblick

Zu Beginn dieses Kapitels wird eine Zusammenfassung der gesamten Arbeit gegeben und die gewonnenen Erkenntnisse bewertet. Zum Abschluss werden mögliche Weiterentwicklungen der entstandenen Anwendung beleuchtet und diskutiert.

### 9.1. Zusammenfassung

Das Ziel dieser Arbeit war die Konzeption und Implementierung einer neuen Visualisierung der Ergebnisse des IÖR-Monitors. Mit dem Einsatz modernster Webtechnologien sollte eine Anwendung entstehen, welche die Interaktions- und Visualisierungsmöglichkeiten der alten Anwendung erweitert bzw. verbessert. Eine wichtige Grundlage bildete dabei die Kombination aus einem Redesign mit den neuesten Kenntnissen des Webdesigns und dem Einsatz modernster Web-Technologien. Dieses Ziel konnte mit der vorliegenden Arbeit erfolgreich implementiert und evaluiert werden.

Im Kapitel zwei wurde der Leser mit den Grundlagen des Flächenmonitorings vertraut gemacht. Mit der Erörterung des Begriffs und dem Hervorheben der Bedeutung für die Bundesrepublik Deutschland, findet eine Einleitung in die Thematik statt. Als eine der bedeutendsten Institutionen, welche sich mit dem Flächenmonitoring befassen wird die Forschungsarbeit des Leibniz-Institut für ökologische Raumentwicklung und dessen Wichtigkeit vorgestellt.

Um die gewonnenen Forschungsergebnisse einer breiten Öffentlichkeit zugänglich zu machen, stellt das WebMapping ein geeignetes Instrument dar. Diese Form der Internet-Kartographie und deren Grundlagen werden im dritten Kapitel resümiert. Hierfür findet eine Begriffsklärung und eine Einführung in die zweite Generation des WebMapping, mit ihrer Bedeutung und Ansprüchen statt. Aufbauend auf diesem Hintergrund wird das Client-\Server-Modell vorgestellt, welches den architektonischen Eckpfeiler von Anwendungen dieser Natur bildet. Mit diesen Grundlagen erfolgt eine Vorstellung des IÖR-Monitors, dessen zwei Viewer und technische Basis dokumentiert werden.

Mit den veränderten Nutzeransprüchen und der rasanten Entwicklung von Webtechnologien entstanden für die Implementierung von WebMapping-Anwendungen neue Möglichkeiten aber auch Bedingungen. Das vierte Kapitel stellt die wichtigsten Technologien vor, anhand derer Anwendungen entstehen können, die in ihrer Reaktionsgeschwindigkeit Desktop-Anwendungen sehr nah kommen und eine Geräteunabhängigkeit realisieren.

Kapitel fünf versucht durch multiple Analysen mögliche Entwicklungspotentiale der zu konzipierenden Anwendung abzuleiten. Zur Erreichung dieses Ziels wird zu Beginn eine begonnene Benutzeranalyse des Leibniz-Institutes aufgegriffen und ausgewertet. Da das Institut bereits über eine intakte Serverinfrastruktur verfügt, kann diese eingesetzt werden. Jedoch entstehen diesbezüglich und durch Vorgaben des Institutes Rahmenbedingungen, welche in diesem Kapitel dokumentiert sind. Mit der durchgeföhrten Bestandsanalyse konnte eine weitere Generierung von möglichen Entwicklungspotentialen des IÖR-Monitors erfolgen. In diesem Abschnitt

findet eine Untersuchung ausgewählter WebMapping-Anwendungen, nach vorher festgelegten Kriterien statt. Die ermittelten Ergebnisse wurden mit der aktuellen Implementierung der Anwendung des Leibniz-Institut für ökologische Raumentwicklung gegenübergestellt. Als Resultat dieses Kapitels entstanden Anforderungskriterien, welche in den folgenden Kapiteln konzipiert und implementiert worden sind. Mit der Wahl von Leaflet konnte ein geeignetes Kartenframework ermittelt werden, auf dessen Basis die kartographischen Funktionalitäten entstehen.

Das sechste Kapitel beschreibt die konzeptionelle Umsetzung der gebildeten Anforderungen. Dazu wurde das Layout der neuen Oberfläche designt und die wichtigsten Funktionen theoretisch skizziert. Das Konzept basiert dabei auf aktuellen Kenntnissen und Möglichkeiten der Webentwicklung, welche als Grundlage für Entscheidungen einbezogen und erklärt werden. Als Resultat entstand ein Konzept einer Anwendung, welches eine Geräteunabhängigkeit realisierte und anhand eines *Clean-Webdesign* die Übersichtlichkeit erhöhte. Es wurde eine Vereinigung der Tabellen- und Kartenansicht mit Interaktionen konzipiert und eine Option geschaffen, zwei Rasterkarten zu vergleichen. Neben einer interaktiven Oberfläche entstanden auch grundlegend neue Interaktionsmöglichkeiten und Funktionen. In alle Überlegungen floss die Absicht ein, indirekte und direkte Zeigegeräte zu unterstützen.

In Kapitel sieben wurde die Implementierung der neuen Anwendung besprochen, hierbei wird der Entwurf vollständig umgesetzt. Die für die Implementierung eingesetzte Hard- und Software und genutzte externe Komponenten wurden aufgezeigt und die wesentlichen Funktionen und Abläufe erläutert. Mit der Implementierung fand das vorangegangene Konzept seine vollständige Umsetzung. Es entstand eine Anwendung, welche mit Hilfe von GeoJSON einen interaktiven Gebietsviewer umsetzt, welcher die thematische Indikatorkarte mit der Tabellenansicht fusionierte. Durch ausgewählte Open Source Software konnten Funktionen implementiert werden, die dem Nutzer ein breiteres funktionales Spektrum bieten, um mit der Anwendung zu interagieren.

Mit einem Smart-Client konnte ein ausgewogenes Verhältnis zwischen client- und serverseitiger Belastung erreicht werden. Rechenintensive Funktionen fanden mit PHP eine Realisierung auf dem Server und Visualisierungsaufgaben eine clientenseitige.

Die implementierte Anwendung entstand unter Berücksichtigung der Konstruktionsregeln des Software-Engineering. Die entworfene Software ist modular aufgebaut und der Funktionsumfang kann problemlos erweitert werden.

Mit der durchgeführten Evaluierung im letzten Kapitel wurde die implementierte Anwendung validiert und mit dem aktuellen IÖR-Monitor verglichen. Durch einen Performance Test konnte ermittelt werden, dass die neue Anwendung eine signifikant kürzere Visualisierungszeit der administrativen Gebietseinheiten realisiert und damit eines der gestellte Kriterien erfolgreich umsetzt.

Um die entwickelte WebMapping-Anwendung und deren Funktionen auf die Nutzerakzeptanz zu untersuchen, fand ein Nutzertest statt. In dieser Studie wurde unter der Verwendung eines Fragebogens und vorher gebildeter Aufgaben, ein Vergleich der beiden Anwendungen durch-

geführt. Als Resultat konnten die anvisierten Verbesserungen, durch eine intuitive und interaktive Oberfläche, bestätigt werden. Bei allen Probanden hinterließ die Reimplementierung dieser Arbeit den positiveren Eindruck im Vergleich zur alten Anwendung des IÖR-Monitors. Neben der Evaluierung traten auch Schwachstellen ans Licht, welche in aufbauende Entwicklungsprozessen einbezogen werden müssen.

Die wesentlichen Themen dieser Arbeit sind nun vorgestellt worden. Im nächsten und letzten Abschnitt werden als Ausblick kurz weitere Möglichkeiten aufgezeigt, die Potential zur Qualitätssteigerung der Anwendung haben.

## 9.2. Ausblick

In Folgenden werden einige technische und organisatorische Weiterentwicklungsmöglichkeiten der neuen Anwendung des IÖR-Monitors vorgestellt.

### 9.2.1. Optimierung der Performance

Die Gesamtdauer der Visualisierung auf Gemeinde-Ebene stellt noch kein zufriedenstellendes Ergebnis dar und zeigt in diesem Bereich noch Entwicklungspotential. Vor allem innerhalb des Rendervorgangs der Tabellenansicht, können Optimierungen erfolgen. Durch den Lösungsansatz, die vom Server empfangene Tabelle direkt darzustellen, muss der Client entsprechend viel Arbeit leisten.

Ein empfohlener und mit sehr großen Tabellen getesteter Ansatz stellt folgende Überlegung dar. Auf dem Server wird keine HTML-Tabelle generiert sondern eine JSON-Datei. Hat der Client die Datei empfangen, erstellt JQuery automatisch die Zeilen im Sichtbereich des Nutzers aus der JSON-Datei, wenn dieser durch die Tabelle scrollt. Hiermit muss der Browser nicht mehr alle Zeilen der Tabelle auf einmal rendern, sondern nur die benötigten [Jeney, 2012]. Ein Plugin welches diesen Ansatz bereits integriert anbietet, ist die Open Source Software *DataTables*<sup>43</sup>.

Ein weiteres auftretendes Problem der neuen Anwendung stellt die Visualisierung aller Gemeinden auf einem mobilen Endgerät dar. Auf diesem hat der Nutzer meist keinen gleichwertigen Arbeitsspeicher, wie bei einem Desktop-PC zur Verfügung. Eine Lösungsmöglichkeit dieser Problematik wäre eine Umstellung der Visualisierungsart bei leistungsschwachen Clienten. Hier könnte das GeoJSON auf dem MapServer generiert und als WMS an den Clienten gesendet werden. Großer Nachteil dieses Vorgehens ist, dass kein ansteuerbares Format mehr vorliegt, welches für Interaktionen zur Verfügung steht.

### 9.2.2. Funktionale Erweiterungen

Eine gute Möglichkeit um die Indikatorwerte neben einer thematischen Karte zu visualisieren, stellt die Generierung von Diagrammen dar. Für diese Visualisierungsaufgaben bietet die JavaScript-Bibliothek *D3*<sup>44</sup> einen großen Funktionsumfang an. Mit dem Einsatz dieser

<sup>43</sup><https://datatables.net>

<sup>44</sup><https://d3js.org>

Bibliothek können zukünftig interaktive Diagramme entstehen, welche die analytischen Möglichkeiten der Nutzer weiter ausbauen. Denkbar wäre die Erweiterung des MultiView um ein drittes Fenster, für die Aufnahme generierter Diagramme aus den Indikatorwerten. Mit der Einführung dieser Funktionalität müssen auch geeignete Interaktionen implementiert werden. Ein erster Ansatzpunkt bildet die bereits integrierte Synchronisierung der Ansichten, nach räumlichen und zeitlichen Aspekten.

Mit der Existenz von Indikatoren mit einem Bezug zu Gebäuden, wäre die Implementierung einer dreidimensionalen Gebäude-Darstellung eine sinnvolle Erweiterung. Für diese Aufgabe zeigt sich das von Marsch [2006] entwickelte Open Source Plugin als geeignet. Dieses generiert auf der Basis von *OSM-Buildings* dreidimensionale Gebäudeprimitive. In dieser Arbeit fanden bereits erste Validierungen statt, welche aber für eine komplette Realisierung unzureichend waren. In der folgenden Abbildung sind erste Ergebnisse zusehen. Hier wurde der Indikator *Siedlungs- und Verkehrsfläche* mit dreidimensionalen Gebäudemodellen visualisiert.



Abbildung 40: 3D-Visualisierung mit Rasterkarte

Im Moment ist es nur möglich Rasterkarten im Vergleichsmodus gegenüberzustellen. Hier wäre die Schaffung einer Möglichkeit interessant, administrative Gebietskarten mit Rasterkarten vergleichen zu können. Um dies zu realisieren, müsste eine Teilumstrukturierung der Anwendungs-Architektur erfolgen, da beide Karten unterschiedlich generiert werden. Mit der Programmierung eines PHP-Skript, welches die zwei benötigten Karten zusammenführend berechnet, könnte die Umsetzung erfolgen.

Alle Anwendungen der Bestandsanalyse stellten dem Nutzer die Möglichkeit zur Verfügung, auf die Kartengrundlage interaktive Zeichnungen anzufertigen, welche zum Erhalt als vektorielles Dateiformat exportiert werden konnten. Dieses Werkzeug wäre für einige Nutzergruppen interessant, um beispielsweise Indikatorkarten durch eigene Notizen oder erstellte Grafiken zu ergänzen und diese anschließend zu teilen.

## Literaturverzeichnis

- [Alby 2008] ALBY, Tom: *Web 2.0*. 3., überarbeitete Auflage. Carl Hanser Verlag, 2008
- [Babich 2017] BABICH, Nick: *UX Design: Drop-Downs in Forms*. <https://uxplanet.org/ux-design-drop-downs-in-forms-c6943ec30037>. Version: April 2017
- [Bach 2014] BACH, Christian: *tablesorter*. <https://github.com/christianbach/tablesorter>. Version: 2014
- [Bai 2009] BAI, Giulio: *10 Ways to Instantly Increase Your jQuery Performance*. <https://code.tutsplus.com/tutorials/10-ways-to-instantly-increase-your-jquery-performance--net-5551>. Version: 2009
- [Baldonado u. a. 2000] BALDONADO, M. Q. ; WOODRUFF, A. ; KUCHINSKI, A.: Guidelines for Using Multiple Views in Information Visualization, Proceedings of the working conference on Advanced Visual Interfaces. In: *Proc. of Inf. Processing & Management*. New York : ACM Press, 2000, S. 110–119
- [Bats 2015] BATS, Simon: *Leaflet.FileLayer*. <https://github.com/makinacorpus/Leaflet.FileLayer>. Version: 2015
- [Bats 2016] BATS, Simon: *Leaflet.Spin*. <https://github.com/makinacorpus/Leaflet.Spin>. Version: 2016
- [Bayerisches Staatsministerium der Finanzen, für Landesentwicklung und Heimat 2017] BAYERISCHES STAATSMINISTERIUM DER FINANZEN, FÜR LANDESENTWICKLUNG UND HEIMAT: *BayernAtlas*. <https://geoportal.bayern.de/bayernatlas>. Version: 2017
- [Becquet 2016] BECQUET, Benjamin: *Leaflet Magnifying Glass*. <https://github.com/bbecquet/Leaflet.MagnifyingGlass>. Version: 2016
- [BfN 2010] BfN: *Monitoring und Indikatoren für den Naturschutz*. [https://www.bfn.de/0315\\_monitoring.html](https://www.bfn.de/0315_monitoring.html). Version: 2010
- [Birkmann 2005] BIRKMANN, J.: Monitoring. In: *Handwörterbuch der Raumplanung*. Hannover : AKADEMIE FÜR RAUMFORSCHUNG UND LANDESPLANUNG, 2005, S. 668–674
- [BKG 2017] BKG: *TopPlus*. <https://www.bkg.bund.de/DE/Ueber-das-BKG/Geoinformation/Sonderleistungen/TopPlus/topplus.html>. Version: 2017
- [Black Duck Software 2017] BLACK DUCK SOFTWARE, Inc.: *Compare Projects*. [https://web.archive.org/web/20140808124119/https://www.openhub.net/p/compare?project\\_0=OpenLayers&project\\_1=Leaflet](https://web.archive.org/web/20140808124119/https://www.openhub.net/p/compare?project_0=OpenLayers&project_1=Leaflet). Version: 2017
- [Blankerts 2006] BLANKERTS, Arne: *Web-2.0-Anwendungen mit JSON und JSON-RPC*. <https://entwickler.de/online/web/web-2-0-anwendungen-mit-json-und-json-rpc-129462.html>. Version: 2006
- [Blumauer u. Pellegrini 2009] BLUMAUER, Andreas ; PELLEGRINI, Tassilo: *Social Semantic Web Web 2.0 – Was nun?* Berlin Heidelberg : X.media.press, 2009
- [BMUB 2017] BMUB: *Flächenverbrauch – Worum geht es?* [www.bmub.bund.de/P2220/](http://www.bmub.bund.de/P2220/). Version: 2017
- [Bundesministerium des Innern 2012] BUNDESMINISTERIUM DES INNERN: 3. Geo-Fortschrittsbericht. 2012. – Forschungsbericht

- [Butler 2008] BUTLER, H.: *The GeoJSON Format*. <http://geojson.org/geojson-spec.html>. Version: 2008
- [Caldwell 2013] CALDWELL, A.: *Responsive Web Design Examples with CSS Tips and Tricks*. <http://brolik.com/blog/responsive-web-design-examples-with-css-tips-and-tricks/>. Version: 2013
- [Chantelau u. Brothuhn 2010] CHANTELAU, Klaus ; BROTHUHN, René: *Multimediale Client-Server-Systeme*. Berlin Heidelberg : Springer, 2010
- [Copeland 2016] COPELAND, Brandon: *leaflet-measure*. <https://github.com/ljagis/leaflet-measure>. Version: 2016
- [Derrough 2013] DERROUGH, Jonathan: *Instant Interactive Map Designs with Leaflet JavaScript Library How-to*. Birmingham, UK : Packt Pub, 2013
- [Dickmann 2004] DICKMANN, Frank: *Einsatzmöglichkeiten neuer Informationstechnologien für die Aufbereitung und Vermittlung geographischer Informationen / das Beispiel karten-gestützte Online-Systeme*. Göttingen : Goltze, 2004
- [Ecma International 2013] ECMA INTERNATIONAL: *The JSON Data Interchange Format*. <https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>. Version: 2013
- [Eszter u. a. 2010] ESZTER, Simonné-Dombóvári ; SCHMIDT, Manuela ; GARTNER, Georg: Kartenanwendungen im Web. Version: 2010. <https://doi.org/10.1007/BF03340532>. Wiesbaden : Springer Fachmedien, 2010 (HMD Praxis der Wirtschaftsinformatik 47), 59–67
- [Fähnrich u. Franczyk 2010] FÄHNRICH, Klaus-Peter ; FRANCZYK, Bogdan: *Service Science – Neue Perspektiven für die Informatik*. Bd. 2. Leipzig : Köllen Druck+Verlag GmbH, 2010 <http://cs.emis.de/LNI/Proceedings/Proceedings176/>
- [Fürster 2011] FÖRSTER, Jörg: Neue Analyse- und Visualisierungsmöglichkeiten im Monitor der Siedlungs- und Freiraumentwicklung (IÖR-Monitor). In: *Flächennutzungsmonitoring III. Erhebungen - Analyse - Bewertung* Bd. 58. Berlin : Rhombos-Verlag, 2011, S. 115–121
- [Förtsch 2017] FÖRTSCH, Daniela: *Nutzergruppenanalyse des IÖR-Monitor und Hinweise zu dessen Weiterentwicklung*. Mai 2017
- [Garrett 2005] GARRETT, Jesse J.: *Ajax: A New Approach to Web Applications*. <http://adaptivepath.org/ideas/ajax-new-approach-web-applications/>. Version: 2005
- [Gartner 2009] GARTNER, Georg: Web Mapping 2.0. In: *Kartographie und GIS* 151.Jg. (2009), S. 14
- [Gibson u. Earle 2006] GIBSON, R. ; EARLE, S.: *Google Map Hacks*. Sebastopol : O'Reilly Media, 2006
- [Goellner 2016] GOELLNER, Simon: *jQuery UI Slider Pips*. <https://github.com/simeydotme/jquery-ui-Slider-Pips>. Version: 2016
- [Goodchild 2007] GOODCHILD, M.F.: Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0. Version: 2007. <http://ijsdir.jrc.it/index.php/ijsdir/article/viewFile/28/22>. In: *Intern. Journal of Spatial Data Infrastructures Research*. 2007
- [Google 2017] GOOGLE: *Google Maps*. <https://www.google.de/maps>. Version: 2017

- [Google 2017] GOOGLE, Inc.: *Introduction*. <https://material.io/guidelines/material-design/introduction.html#introduction-principles>. Version: 2017
- [Gundelsweiler u. Reiterer 2008] GUNDELSWEILER, Fredrik ; REITERER, Harald: Zoombasiertes Interaktionskonzept für die Suche in großen, heterogenen Bilddatenbanken. In: *Mensch & Computer*. Oldenburg Verlag, 2008, S. 390–400
- [Haar u. Keck 2003] HAAR, Tobias ; KECK, Gabriele: Offene Konkurrenz. Version: 2003. Masterarbeit. In: *iX*. Hannover : Heise Medien gruppe, 2003 (9 03)
- [Hahn 2017] HAHN, Martin: *Webdesign : das Handbuch zur Webgestaltung*. 2., aktualisierte Auflage. Bonn : Rheinwerk Design, 2017
- [Haklay 2010] HAKLAY, Mordechai: *Interacting with Geospatial Technologies*. Oxford : John Wiley & Sons Ltd., 2010
- [Harrower 2001] HARROWER, Mark: *Animated Maps – Overview of Animated Maps*. 2001
- [Harrower u. Sheesley 2005] HARROWER, Mark ; SHEESLEY, Benjamin: Designing Better Map Interfaces: A Framework for Panning and Zooming. In: *Transactions in GIS*. Oxford : Blackwell Publishing Ltd, 2005, S. 77– 89
- [Hegner 2003] HEGNER, Marcus: Methoden zur Evaluation von Software. Version: 2003. [https://www.gesis.org/fileadmin/upload/forschung/publikationen/gesis\\_reihen/iz\\_arbeitsberichte/ab\\_29.pdf](https://www.gesis.org/fileadmin/upload/forschung/publikationen/gesis_reihen/iz_arbeitsberichte/ab_29.pdf). In: *IZ-Arbeitsbericht Nr. 29*. Köln : GE-SIS - InformationsZentrum Sozialwissenschaften, 2003
- [Henning u. Vogler 2011] HENNING, S. ; VOGLER, R.: WebMapping: Der Einsatz von digitalen, interaktiven Karten in Schule und Bildung. In: *GW-UNTERRICHT*. Salzburg : Institut für GIScience, Österreichische Akademie der Wissenschaften, 2011 (Nr. 123), S. 86–99
- [Henzen u. Bernhard 2013] HENZEN, Christin ; BERNHARD, Lars: Usability für Geoportal am Beispiel der Konzeption des Geoportal Sachsen. In: *Kartographische Nachrichten* (2013), Mai, 262–269. [http://www.gdi.sachsen.de/inhalt/info/echo/Artikel\\_KN\\_Geoportal.pdf](http://www.gdi.sachsen.de/inhalt/info/echo/Artikel_KN_Geoportal.pdf)
- [Herrmann 2013] HERRMANN, Eric: *Semantic UI: Wie Bootstrap, aber mit anderem Vokabular*. <https://entwickler.de/online/semantic-ui-wie-bootstrap-aber-mit-anderem-vokabular-137079.html>. Version: 2013
- [Hoffmann 2013] HOFFMANN, Manuela: *Modernes Webdesign*. 1. Bonn : Galileo Press, 2013
- [INSPIRE 2017] INSPIRE: *INSPIRE KNOWLEDGE BASE*. <https://inspire.ec.europa.eu/id/document/tg/1c>. Version: 2017
- [IÖR 2008] IÖR: *Methodik*. <https://www.ioer.de/langzeitmonitoring/homepage.html>. Version: 2008
- [IÖR-Monitor 2014] IÖR-MONITOR: *Flächennutzung*. <http://www.ioer-monitor.de/glossar/f/flaechennutzung/>. Version: 2014
- [Jakub 2016] JAKUB, Jankiewicz: *jQuery Splitter*. <https://github.com/jcubic/jquery-splitter>. Version: 2016
- [Jeney 2012] JENEY, E.: *Optimizing JQuery DataTables Performance*. Version: 2012. <http://madebyknight.com/optimizing-datatables-performance/>

- [Jotz 2015] JOTZ, Melanie: *Qualitativ & quantitativ: Fragebögen können den Usability-Test bereichern!* Version: 2015. <http://www.usabilityblog.de/2015/12/qualitativ-quantitativ-frageboegen-koennen-den-usability-test-bereichern/>
- [Kallas 2001] KALLAS, Michael: *Kategorien freier und unfreier Software.* <https://www.gnu.org/philosophy/categories.de.html>. Version: 2001
- [Karla 2007] KARLA, Jürgen: Implementierung von Regelkreisen in Geschäftsmodellen für Web 2.0-Publikumsdienste. In: *HMD Praxis der Wirtschaftsinformatik*. Wiesbaden : Springer Fachmedien, 2007 ( 44), S. 17–26
- [Kestler 2017] KESTLER, Petra: *Flächenmonitoring in der EU.* [http://www.umweltbundesamt.at/aktuell/presse/lastnews/newsarchiv\\_2011/news110321/](http://www.umweltbundesamt.at/aktuell/presse/lastnews/newsarchiv_2011/news110321/). Version: 2017
- [Klauer 2002] KLAUER, R. H.: Potenziale und Techniken eines kommunalen Interneteinsatzes. (2002), Nr. Kommunale Geo-Informationssysteme. Basiswissen, Praxisberichte und Trends, S. 181 – 189
- [Kleijn 2016] KLEIJN, Alexandra: Open-Source-Lizenzen. Version: 2016. <https://www.heise.de/ct/artikel/Open-Source-Lizenzen-221957.html?seite=2>. In: c't. Hannover : Heise Medien Gruppe, 2016
- [Klöckner ] KLÖCKNER, Jens: *OpenStreetMap.* <http://www.wegeundpunkte.de/gps.php?content=osm>
- [Knoll u. Niederl 2015] KNOLL, Mathias ; NIEDERL, Franz: Modernes Web Design. Version: 2015. <http://bizepaper.fh-joanneum.at/KMU6M-Sammelband.pdf>. In: *KMU goes mobile Sammelband*. Graz : Verlag der FH JOANNEUM Gesellschaft mbH, 2015, 104 –109
- [Koch 2009] KOCH, Stefan: *Einführung, Programmierung und Referenz – inklusive Ajax.* 5. Heidelberg : dpunkt.verlag, 2009
- [Koch u. Bollmann 2002] KOCH, W. G. ; BOLLMANN, J.: *Lexikon der Kartographie und Geomatik. Band 1 und Band 2.* Heidelberg, Berlin : Spektrum Akademischer Verlag GmbH, 2002
- [Kollmann u. Michaelis 2015] KOLLMANN, Tobias ; MICHAELIS, Alexander: Responsive Webdesign. Version: 2015. <https://elibrary.vahlen.de/10.15358/0340-1650-2015-7-50.pdf>. In: *WiSt - Wirtschaftswissenschaftliches Studium*. München Frankfurt a.M. : C.H.BECK und Vahlen, 2015, 406–409
- [Korduan u. Zehner 2008] KORDUAN, P. ; ZEHNER, M. L.: *Geoinformation im Internet - Technologien zur Nutzung raumbezogener Informationen im WWW.* Rostock : Wichmann Herbert, 2008
- [Koussoulakou u. Kraak 1992] KOUSSOULAKOU, A. ; KRAAK, M. J.: Spatio-temporal maps and cartographic communication. In: *The Cartographic Journal*. 1992, S. 101–108
- [Kraak u. MacEacren 1994] KRAAK, M. J. ; MACEACREN, A.M: Visualization of the 'Temporal Component of spatial Data. In: *T.C. Waugh. R.G. Healey (Eds.): Proc. of 6. Intl. Symp. on Spatial Data Handling.* Taylor & Francis, 1994, S. 391–409
- [Kriese u. Bohnsack 2006] KRIESE, U. ; BOHNSACK, K.: *Aktiv für Landschaft und Gemeinde!- Leitfaden für eine nachhaltige Siedlungsentwicklung.* Rheinbach : Stiftung Europ. Naturerbe, 2006

- [Krüger 2011] KRÜGER, J. D.: *Conversion Boosting mit Website-Testing*. 1. Aufl. Heidelberg ; München ; Landsberg ; Frechen ; Hamburg : Mitp, 2011
- [Köhler 2012] KÖHLER, Christiane: *WebGIS-basierte Visualisierung der Ergebnisse des Monitors der Siedlungs- und Freiraumentwicklung (IÖR-Monitor) in einem Detailviewer auf Grundlage des MapServers und OpenLayers*. Hochschule für Technik und Wirtschaft Dresden, 2012
- [Landesbetrieb Geoinformation und Vermessung. Kartographie und Gestaltung; LGV Hamburg 2017] LANDESBETRIEB GEOINFORMATION UND VERMESSUNG. KARTOGRAPHIE UND GESTALTUNG: LGV HAMBURG: *Geo - Online*. <https://geoportal-hamburg.de/geoportal/geo-online>. Version: 2017
- [Lawrence u. Tavakol 2007] LAWRENCE, Dave ; TAVAKOL, Soheyla: *Balanced Website Design*. London : Springer, 2007
- [Leibniz-Institut für ökologische Raumentwicklung 2017a] LEIBNIZ-INSTITUT FÜR ÖKOLOGISCHE RAUMENTWICKLUNG: *Detailviewer*. <http://maps.ioer.de/detailviewer/raster/>. Version: 2017
- [Leibniz-Institut für ökologische Raumentwicklung 2017b] LEIBNIZ-INSTITUT FÜR ÖKOLOGISCHE RAUMENTWICKLUNG: *Übersichtsviewer*. <http://www.ioer-monitor.de/karten/karten/>. Version: 2017
- [Lukic 2015] LUKIC, Jack: *Semantic-UI*. <https://github.com/Semantic-Org/Semantic-UI>. Version: 2015
- [Macotte 2010] MACOTTE, E.: *Responsive Web Design*. <https://alistapart.com/article/responsive-web-design>. Version: 2010
- [Maguire 2007] MAGUIRE, David J.: *GeoWeb 2.0 and Volunteered GI*. Bd. Workshop on Volunteered Geographic Information. Santa Barbara, 2007 [http://www.ncgia.ucsb.edu/projects/vgi/docs/position/Maguire\\_paper.pdf](http://www.ncgia.ucsb.edu/projects/vgi/docs/position/Maguire_paper.pdf)
- [Maier 2010] MAIER, Sebastian: *Mobil und übersichtlich: So designst du Tabellen für liquid-responsive*. <https://www.smooster.com/de/blog/mobil-und-uebersichtlich-so-designst-du-tabellen-fuer-liquid-responsive.html>. Version: 2010
- [Maike 2015] MAIKE, Luhmann: *R für Einsteiger : Einführung in die Statistiksoftware für die Sozialwissenschaften*. 1. Aufl. Weinheim ; Basel : Beltz, 2015
- [Maness 2006] MANESS, Jack: *Library 2.0 Theory: Web 2.0 and Its Implications for Libraries*. <http://www.webology.org/2006/v3n2/a25.html>. Version: 2006
- [Manlai u. a. 2007] MANLAI, You ; CHUN-WEN, Chen ; HANTSAI, Liu ; HSUAN, Lin: A Usability Evaluation of Web Map Zoom and Pan Functions. Version: 2007. <http://www.ijdesign.org/ojs/index.php/IJDesign/article/viewFile/31/4>. In: *International Journal of Design* Vol.1. Yunlin : National Yunlin University of Science and Technology, 2007
- [Marchionini 1997] MARCHIONINI, Gary: *Information Seeking in Electronic Environments*. Cambridge Press, 1997
- [Marsch 2006] MARSCH, Jon: *OSMBuildings*. <https://github.com/OSMBuildings/OSMBuildings/tree/master/>. Version: 2006

- [Martin u. Richter 2008] MARTIN, Tobias ; RICHTER, André: Was gute Webseiten ausmacht. Version: 2008. <http://static.onleihe.de/content/businessvillage/20081001/978-3-938358-68-9/v978-3-938358-68-9.pdf>. In: *Edition Praxis. Wissen*. Göttingen : Business Village, 2008
- [McConnachie 2007] McCONNACHIE, D.: *GeoNames founder opens up the GIS world.* [https://www.goodgearguide.com.au/article/203442/geonames\\_founder.opens\\_up\\_gis\\_world](https://www.goodgearguide.com.au/article/203442/geonames_founder.opens_up_gis_world). Version: 2007
- [Meidl 2014] MEIDL, Oliver: *Globales Webdesign Anforderungen und Herausforderungen an Globale Webseiten*. Wiesbaden : Springer Gabler, 2014
- [Meinel u. Siedentop 2007] MEINEL, G. ; SIEDENTOP, S.: *Erhebung und indikatorgestützte Bewertung der Siedlungsstruktur und ihrer Entwicklung – Konzept „Deutschlandmonitor Siedlungs- und Freiraumentwicklung“*. Salzburg, 2007
- [Meinel 2009] MEINEL, Gotthard: *Konzept eines deutschlandweiten Monitors der Siedlungs- und Freiraum- entwicklung auf der Grundlage von ATKIS. Vortrag: 22.1.2009*. Leibniz-Institut für ökologische Raumentwicklung, 2009
- [Meinel 2013] MEINEL, Gotthard: Anforderungen an ein Flächennutzungsmonitoring. In: *Flächennutzungsmonitoring V, Methodik – Analyseergebnisse – Flächenmanagement*. Berlin : Rhombos, 2013 (Band 61)
- [Meinel u. Schumacher 2009] MEINEL, Gotthard ; SCHUMACHER, Ullrich: *Flächennutzungsmonitoring Konzepte – Indikatoren – Statistik*. Aachen : Shaker Verlag, 2009
- [Meinel u. a. 2014] MEINEL, Gotthard ; SCHUMACHER, Ullrich ; BEHNISCH, Martin: *Flächennutzungsmonitoring VI Inneneentwicklung – Prognose – Datenschutz*. Berlin : Rhombos, 2014 (Band 65)
- [Mintert 2009] MINTERT, Stefan: Hilf, wenn Du kannst. In: *iX* (2009), Nr. 07/09. <https://www.heise.de/ix/artikel/Hilf-wenn-Du-kannst-794658.html>
- [Mitchell 2005] MITCHELL, Tyler: *Web Mapping Illustrated*. 1st ed. Sebastopol, CA : O'Reilly Media, 2005 [http://proquest.tech.safaribooksonline.de/book/web-design-and-development/0596008651/web-mapping-illustrated/id289684?query=\(Web+Mapping+Illustrated\)\)#snippet](http://proquest.tech.safaribooksonline.de/book/web-design-and-development/0596008651/web-mapping-illustrated/id289684?query=(Web+Mapping+Illustrated))#snippet)
- [Mittlböck u. a. 2012] MITTLBÖCK, Manfred ; MORPER-BUSCH, Lucia ; ATZL, Caroline ; KLUG, Hermann: *Task-orientierte Web-Maps zur kompakten Visualisierung kartographischer Inhalte*. Berlin/Offenbach : VDE Verlag GmbH, 2012
- [Mohorovičić 2013] MOHOROVIČIĆ, S.: Implementing responsive web design for enhanced web presence. In: *Information & Communication Technology Electronics & Microelectronics (MIPRO), 36th International Convention*. Optaija : IEEE Computer Society Press, 2013, S. 1206–1210
- [Moore 2014] MOORE, Jordan: Responsive web design, Volume 2. Version: 2014. <http://proquest.tech.safaribooksonline.de/9783944540627>. In: *Smashing Magazine*. Freiburg, 2014
- [Moore u. Walz 2016] MOORE, Michael ; WALZ, Andy: *Choosing a Mapping Framework*. <https://luminfire.com/2016/05/10/choosing-mapping-framework/>. Version: 2016
- [Muehlenhaus 2014] MUEHLENHAUS, Ian: *Web cartography : map design for interactive and mobile devices*. Boca Roca : CRC Press, 2014

- [Mun 2014] MUN, Terry: *Sticky Table Headers & Columns*. <https://tympanus.net>. Version: 2014
- [Nielsen 1998] NIELSEN, Jakob: *Usability Engineering*. Nachdr. Boston : AP Professional, 1998
- [Nielsen 1999] NIELSEN, Jakob: *The Top 10 Web Design Mistakes of 1999*. Version: Mai 1999. <https://www.nngroup.com/articles/the-top-ten-web-design-mistakes-of-1999/>
- [Nielsen 2008] NIELSEN, Jakob: *Right-Justified Navigation Menus Impede Scannability*. <https://www.nngroup.com/articles/right-justified-navigation-menus/>. Version: 2008
- [Nordan 2017] NORDAN, Robert: *Leaflet.MiniMap*. <https://github.com/Norkart/Leaflet-MiniMap>. Version: 2017
- [Norman 1988] NORMAN, D: *The psychology of everyday things*. New York : Basic Books, 1988
- [Olsen 2007] OLSEN, Steven D.: *Ajax und Java*. 1. Aufl., Dt. Ausg. Köln : O'Reilly, 2007
- [Open Source Initiative 2007] OPEN SOURCE INITIATIVE: *The Open Source Definition*. <https://opensource.org/docs/osd>. Version: 2007
- [Ormerod u. Lansdale 1994] ORMEROD, T. ; LANSDALE, M.: *Understanding Interfaces. A Handbook of Human-Computer Dialogue*. San Diego : Academic Press, 1994
- [O'Reilly 2017] O'REILLY, Tim: *What is Web 2.0? Design patterns and business models for the next generation of software*. O'Reilly Media, Inc, 2017
- [Pangakulyanont 2015] PANGSAKULYANONT, T.: *Monkey-Patching, Overriding, and Decorating Methods in JavaScript*. <http://me.dt.in.th/page/JavaScript-override/>. Version: 2015
- [Perez 2012] PEREZ, Antonio S.: *OpenLayers Cookbook*. Birmingham : Packt Pub, 2012
- [Plewe 1997] PLEWE, B.: *GIS Online - Information Retrieval, Mapping and the Internet*. Santa Fe : OnWorld Press, 1997
- [Plewe 2007] PLEWE, B: Web cartography in the United States. In: *Cartography and Geographic Information Science*. 2007, S. 133–136
- [Post 2017] POST, Julian: *Clean Webdesign*. <https://contunda.de/clean-webdesign>. Version: 2017
- [Preim u. Dachselt 2010] PREIM, Bernhard ; DACHSELT, Raimund: *Band 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung*. Heidelberg Dordrecht London New York : Springer-Verlag, 2010
- [Preim u. Dachselt 2015] PREIM, Bernhard ; DACHSELT, Raimund: *Band 2: User Interface Engineering, 3D-Interaktion, Natural User Interfaces*. Berlin, Heidelberg : Springer, 2015
- [Sanchez 2015] SANCHEZ, Ivan: *Slides for FOSS4G 2015 talk "Leaflet vs OpenLayers"*. <https://github.com/IvanSanchez/leaflet-vs-openlayers-slides>. Version: 2015
- [van de Sand 2017] SAND, Kerstin van d.: *Ein klarer Blick auf die Welt*. <https://arcgis.esri.de/ein-klarer-blick-auf-die-welt/>. Version: 2017

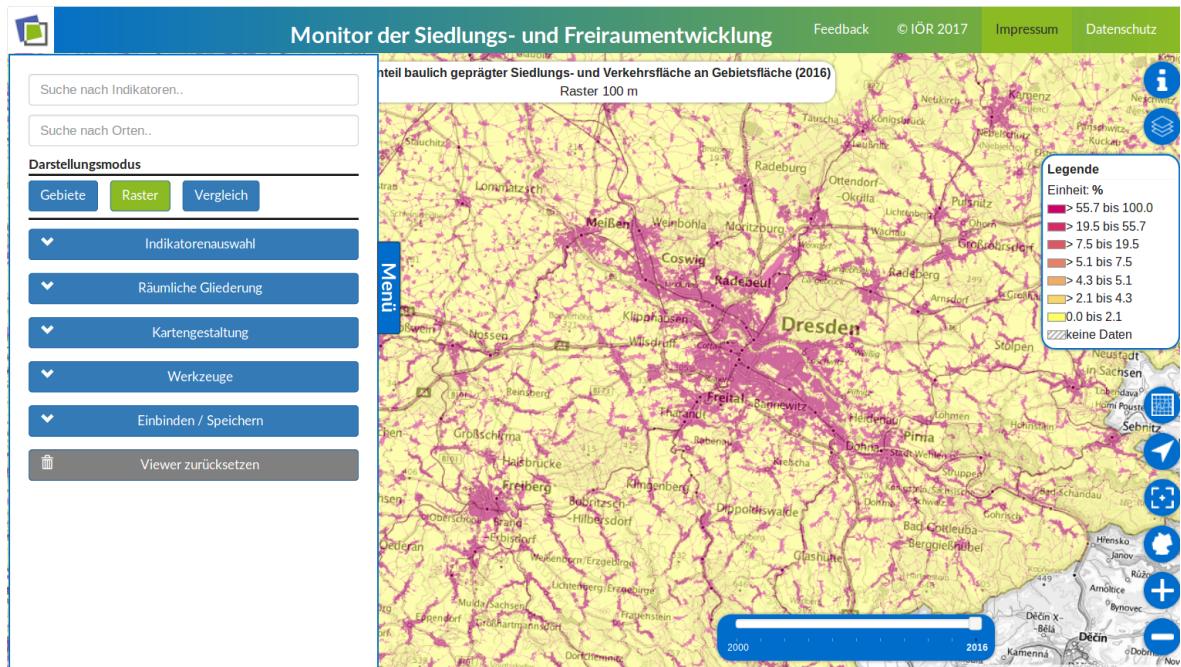
- [Schlegel 2013] SCHLEGEL, Thomas: *Multi-Touch Interaktion durch Berührung.* Berlin Heidelberg : Springer Vieweg, 2013 <https://link.springer.com/content/pdf/10.1007%2F978-3-642-36113-5.pdf>
- [Schneider 2010] SCHNEIDER, Peter: *Einbindung von Geobasisdaten als WMS - Dienste.* Version: 2010. [https://www.hft-stuttgart.de/Studienbereiche/Vermessung/Bachelor-Vermessung- Geoinformatik/Aktuell/Veranstaltungen/edvstadtplanung2010/edvstadtplanung2010\\_ws6\\_schneider.pdf](https://www.hft-stuttgart.de/Studienbereiche/Vermessung/Bachelor-Vermessung- Geoinformatik/Aktuell/Veranstaltungen/edvstadtplanung2010/edvstadtplanung2010_ws6_schneider.pdf)
- [Schnürer u. a. 2015] SCHNÜRER, Raimund ; SIEBER, René ; ÇÖLTEKIN, Arzu: The Next Generation of Atlas User Interfaces: A User Study with “Digital Natives”. In: *Modern Trends in Cartography*. Heidelberg New York Dordrecht London : Springer, 2015
- [Schumann u. Kreuseler 2003] SCHUMANN, Heidrun ; KREUSELER, Matthias: Fokus und Kontext-Darstellung im geographischen Kontext. In: *Proceedings GeoVis*. Hanover, 2003
- [Schweizerische Eidgenossenschaft 2017] SCHWEIZERISCHE EIDGENOSSENSCHAFT: *Geo Admin CH*. <https://map.geo.admin.ch>. Version: 2017
- [Shneiderman 1996] SHNEIDERMAN, B: The eyes have it: a task by data type taxonomy for information visualizations. In: *Spencer Sipple R (ed) Proceedings of the IEEE symposium on visual languages, Boulder*. Los Alamitos : IEEE Computer Society Press, 1996, S. 336–343
- [Shneiderman u. Plaisant 2010] SHNEIDERMAN, Ben ; PLAISANT, Catherine: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 5. ed. München : Pearson Addison-Wesley, 2010
- [Skupin u. Hagelman 2005] SKUPIN, André ; HAGELMAN, Ron: Visualizing Demographic Trajectories with Self-Organizing Maps. In: *GeoInformatica*. New Orleans : Springer Science + Business Media, 2005 (Volume 9), S. 159–179
- [Sossou 2017] SOSSOU, Ulrich: *Bootstrap Tour*. <https://github.com/sorich87/bootstrap-tour>. Version: 2017
- [Spurlock 2013] SPURLOCK, Jake: *Bootstrap*. Sebastopol, CA : O'Reilly Media, 2013 <http://proquest.tech.safaribooksonline.de/9781449344573>
- [Staatsbetrieb Geobasisinformation und Vermessung Sachsen 2017] STAATSBETRIEB GEOBASISINFORMATION UND VERMESSUNG SACHSEN: *Geo Viewer Sachsen*. <https://geoviewer.sachsen.de>. Version: 2017
- [statista 2017a] STATISTA: *Absatz von Smartphones in Deutschland in den Jahren 2009 bis 2017 (in Millionen Stück)*. <https://de.statista.com/statistik/daten/studie/77637/umfrage/absatzmenge-fuer-smartphones-in-deutschland-seit-2008/>. Version: 2017
- [statista 2017b] STATISTA: *Marktanteile führender Suchmaschinen in Deutschland in den Jahren 2014 bis 2016*. <https://de.statista.com/statistik/daten/studie/167841/umfrage/marktanteile-ausgewahlter-suchmaschinen-in-deutschland/>. Version: 2017
- [Stephan 2010] STEPHAN, Thesemann: *Einführung in das Design multimedialer Webanwendungen*. 1. Aufl. Wiesbaden : Vieweg+Teubner Verlag, 2010
- [Storch 1999] STORCH, H.: "WebGIS" oder "WWW-Mapping"? Die Grenzen von WWW-Strategien für den öffentlichen Zugang zu raumbezogenen Umweltinformationen. In: *Angewandte geographische Informationsverarbeitung XI : Beiträge zum AGIT-Symposium Salzburg 1999*. Heidelberg : Wichmann, 1999, S. 510–521

- [Street 2017] STREET, Arthur: *leaflet-side-by-side*. <https://github.com/digidem/leaflet-side-by-side>. Version: 2017
- [Stuhec-Meglic 2016] STUHEC-MEGLIC, K.: *Was ist Web 2.0?* <https://webconsulting-stuhec.com/blog/was-ist-web-2.0>. Version: 2016
- [ThinkGeo 2015] THINKGEO: *Leaflet vs. OpenLayers 3: Which is the better client-side JavaScript mapping library?* <http://blog.thinkgeo.com/2015/09/08/leaflet-vs-openlayers-3-which-is-the-better-client-side-javascript-mapping-library/>. Version: 2015
- [Thissen 2001] THISSEN, F.: *Screen-Design-Handbuch: Effektiv informieren und kommunizieren mit Multimedia*. 2. überarbeitete und erweiterte Aufl. Berlin : Springer Verlag, 2001
- [UBA 2014] UBA: *Flächensparen – Böden und Landschaften erhalten*. <http://www.umweltbundesamt.de/themen/boden-landwirtschaft/flaechenparen-boeden-landschaften-erhalten#textpart-3>. Version: 2014
- [Vdovkin 2012] VDOVKIN, Andreas: *JQuery kurz & gut*. 2. Köln : O'Reilly Verlag, 2012 <http://proquest.tech.safaribooksonline.de/book/programming/javascript/9783868992816>
- [Verclas u. Linnhoff-Popien 2012] VERCLAS, Stephan V. ; LINNHOFF-POPIEN, Claudia: *Smart Mobile Apps*. Heidelberg Dordrecht London New York : Springer, 2012
- [Volker u. a. 2007] VOLKER, H. ; SCHROTH, C. ; STANOEVSKA-SLAVEVA, K.: *Web 2.0-Entwicklung — ewige Beta-Version*. Version: 2007. [http://www.ics.uci.edu/%7Efielding/pubs/dissertation/fielding\\_dissertation.pdf](http://www.ics.uci.edu/%7Efielding/pubs/dissertation/fielding_dissertation.pdf). In: *HMD Praxis der Wirtschaftsinformatik*. Wiesbaden : Springer Fachmedien, 2007 (Volume 44), 78–87
- [Ward u. a. 2015] WARD, Brendan ; STEVENSON-MOLNAR, Nik ; KARIMI-ASLI, Kaveh: *Leaflet.ZoomBox*. <https://github.com/consbio/Leaflet.ZoomBox>. Version: 2015
- [Warnerdam 2013] WARNERDAM, Frank: *Raster Data*. <http://mapserver.org/input/raster.html>. Version: 2013
- [Wilke 2017] WILKE, Henry: *News 2017 Juni 30-Hektar-Tag: Kein Grund zum Feiern*. <https://www.nabu.de/news/2017/06/22630.html>. Version: 2017
- [Zverev 2017] ZVEREV, Ilya: *Leaflet.TileLayer.Grayscale*. <https://github.com/Zverik/leaflet-grayscale>. Version: 2017
- [Śmiałek 2017] ŚMIAŁEK, Krzysztof: *SweetAlert2*. <https://github.com/limonte/sweetalert2>. Version: 2017

## A. Anhang

### A.1. Screenshots der Anwendung

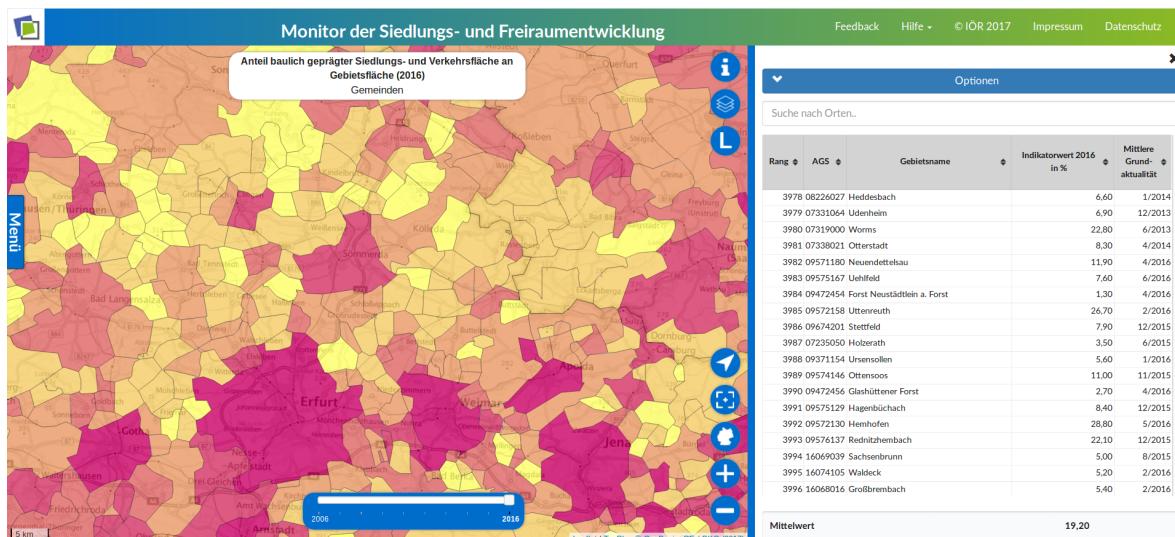
Die vollständige Anwendung ist unter der folgenden URL verfügbar: [http://maps.ioer.de/detailviewer/raster\\_test/New\\_Surface](http://maps.ioer.de/detailviewer/raster_test/New_Surface)



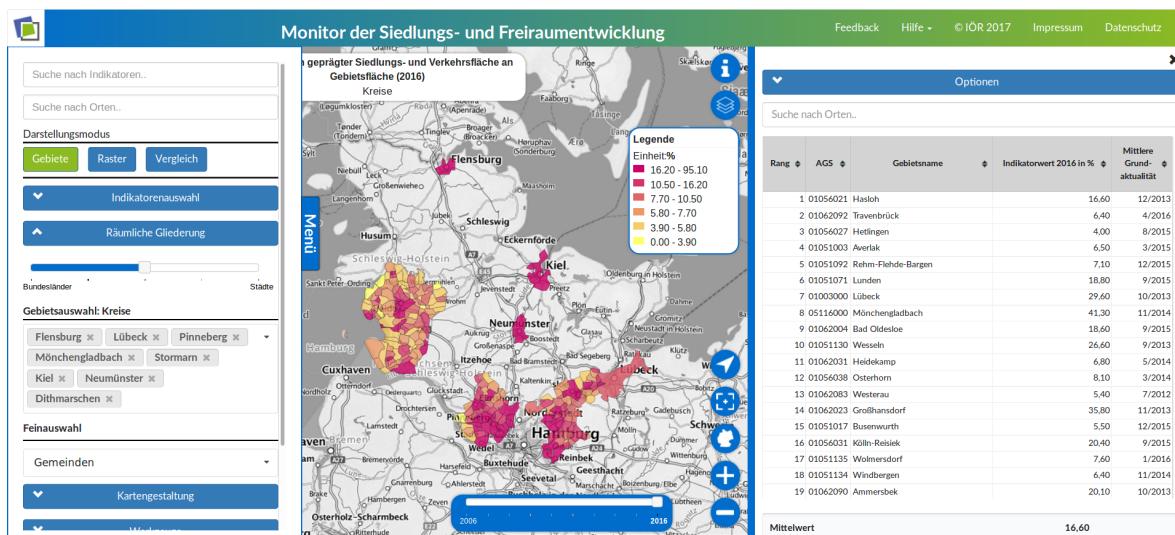
Raster-Modus mit Navigationsmenü



Vergleichs-Modus mit einer Gegenüberstellung von zwei Indikatoren. links: Anteil des amtlich festgesetzten Überschwemmungsgebiets dar rechts: Bodenversiegelung



Gebiete-Modus mit der Visualisierung aller Gemeinden für die terrestrische Ausdehnung Deutschlands



Gebiete-Modus mit der Funktionalität sich eigene administrative Gebietseinheiten zusammenzustellen

## A.2. Fragebogen des Nutzertests

Probanden-Nr.: \_\_\_\_\_

### ALLGEMEINES

1. Ich bin  männlich /  weiblich und \_\_\_\_\_ Jahre alt.
2. Seit \_\_\_\_\_ Jahren benutze ich Web-GIS Anwendungen.
3. Wie häufig verwendest du Web-GIS Anwendungen privat oder im Studium/Beruf?

|                          |                          |                          |                          |                          |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| täglich                  | wöchentlich              | monatlich                | seltener                 | nie → Weiter mit Frage 5 |
| <input type="checkbox"/> |

4. Wie gut kommst du im Allgemeinen mit den bisher von dir genutzten Web-GIS Anwendungen zurecht?

|                          |                          |                          |                          |                          |                          |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| problemlos               |                          |                          |                          |                          | schlecht                 |
| <input type="checkbox"/> |

5. Wie eingehend kennst du den IÖR-Monitor?

|                          |                          |                          |                          |                          |                          |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| ausführlich              |                          |                          |                          |                          | gar nicht                |
| <input type="checkbox"/> |

### TEILAUFGABEN

6. Bewerte die folgenden Aussagen anhand einer Skala zwischen 1 (volle Zustimmung) bis 5 (starke Ablehnung).

  - (i) Vergleich von zwei Zeitschnitten in einem bestimmten Gebiet
  - (ii) Ermittlung von Maximalwerten für administrative Gebietseinheiten
  - (iii) Zusammenstellen und analysieren von administrative Gebietseinheiten
  - (iv) Farbliche Anpassung

Die Aufgabenstellung war einfach und verständlich. (i)  (ii)  (iii)  (iv)

In der Neuen Oberfläche  
... konnte ich die Aufgabe besonders effizient lösen.

In der Alten Oberfläche  
... konnte ich die Aufgabe besonders effizient lösen.

7. Hättest du mit Hilfe der Tour die Aufgaben besser lösen können ?

|                          |                          |
|--------------------------|--------------------------|
| ja                       | nein                     |
| <input type="checkbox"/> | <input type="checkbox"/> |

**GESAMTEINDRUCK**

8. Wie gut bist du insgesamt mit den Anwendungen zurechtgekommen ?

|         | sehr gut                 | befriedigend             | mangelhaft               |
|---------|--------------------------|--------------------------|--------------------------|
| Neue OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Alte OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

9. Bitte bewerte die Übersichtlichkeit der Oberflächen.

|         | sehr gut                 | befriedigend             | mangelhaft               |
|---------|--------------------------|--------------------------|--------------------------|
| Neue OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Alte OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

10. Wie gefiel dir die Reaktionsgeschwindigkeit der Anwendungen?

|         | sehr gut                 | befriedigend             | mangelhaft               |
|---------|--------------------------|--------------------------|--------------------------|
| Neue OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Alte OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

11. Wie verständlich waren Beschriftungen und Symbole in den Oberflächen ?

|         | sehr gut                 | befriedigend             | mangelhaft               |
|---------|--------------------------|--------------------------|--------------------------|
| Neue OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Alte OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

12. Was fiel dir in den Anwendungen als gut gelungen oder positiv auf?

Neue OF:

---



---

Alte OF:

---



---

13. Was war in den Anwendungen störend oder unverständlich?

Neue OF:

---



---

Alte OF:

---



---

14. Kannst du die Anwendungen anderen Nutzern empfehlen?

|         | völlig                   |                          |                          | keinesfalls              |
|---------|--------------------------|--------------------------|--------------------------|--------------------------|
| Neue OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Alte OF | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

## **Danksagung**

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mir während der Erstellung meiner Masterarbeit geholfen haben und mich moralisch unterstützt haben.

Dabei möchte ich meinen Betreuern danken, welche mir bei dieser wissenschaftlichen Arbeit zur Seite standen. Auf Seiten der TU Dresden möchte ich meinem Betreuer Prof. Dirk Burghardt dafür danken, dass er sich stets Zeit genommen hat und hilfreiche Ratschläge geben konnte.

Mein besonderer Dank gilt meinem weiteren Betreuern am IÖR. Dr. Gotthard Meinel konnte mich durch sein Engagement bei der Bearbeitung des Themas motivieren hat und mir sehr nützliche Hilfestellungen geben können.

Weiter möchte ich Jörg Gössel dafür danken, dass er bei technischen Problemen zügig eine Lösung finden konnte und Updates zeitnah durchgeführt hat.

Christin Michel und Martin Schorcht danke ich für das Korrekturlesen meiner Arbeit und den konstruktiven Hinweisen.

Mein besonderer Dank gilt Julia Krause, die mir mit ausführlichen Korrekturarbeiten zu dieser Arbeit geholfen und mich auch immer wieder aufgebaut hat, wann immer es notwendig war.

Zu guter Letzt möchte ich ganz besonders meinen Eltern danken, die mich während des gesamten Zeitraums meines Studiums unterstützte.