



**Hochschule Neubrandenburg**  
**Studiengang Master Geodäsie und Geoinformatik**

**Entwicklung eines webbasierten GIS zur  
Überwachung und Analyse von Extensionsklüften  
an der marokkanischen Atlantikküste**

**Masterarbeit**

vorgelegt von: *Tino Schuldt*

Zum Erlangen des akademischen Grades  
**„Master of Engineering“ (M.Eng.)**

Erstprüfer: Prof. Dr.-Ing. Andreas Wehrenpfennig

Zweitprüfer: Prof. Dr. Ralf Löwner

Eingereicht am: 04.12.2017

URN: urn:nbn:de:gbv:519-thesis2017-0581-0

## **Danksagung**

Für die Unterstützung dieser Masterarbeit möchte ich mich zunächst bei einigen Personen bedanken.

Ich bedanke mich zuerst einmal bei Prof. Dr.-Ing. Andreas Wehrenfennig und Prof. Dr. Ralf Löwner für die Betreuung meiner Masterarbeit.

Weiterhin bedanke ich mich bei allen Personen, die mich mit Korrekturlesen unterstützt haben.

Ein besonderer Dank gilt meiner Familie, die mich während der Bearbeitungszeit unterstützt haben.

## Kurzfassung

Daten mit einem räumlichen Bezug (Geodaten) werden für die Überwachung und Analyse von geologischen Problemstellungen immer relevanter. Eine Problemstellung betrifft die Ausbreitung von Extensionsklüften an der marokkanischen Atlantikküste durch die Platten tektonik. Mithilfe eines webbasierten Geoinformationssystem (Web-GIS) können die erhobenen Geodaten zentral gespeichert und durch einheitliche Schnittstellen mit dem Web browser zur Überwachung und Analyse genutzt werden, ohne das vorher eine Softwareinstallation notwendig ist. Weiterhin können diese bereitgestellten Daten mit einer beliebigen GIS Software zur weiteren Nutzung der Daten verwendet werden. Ziel der Arbeit ist es, ein Web-GIS für die Überwachung und Analyse von Extensionsklüften an der marokkanischen Atlantikküste zu entwickeln.

## Abstract

Data with a spatial reference (geodata) are becoming increasingly relevant for the monitoring and analysis of geological problems. One problem concerns the expansion of extension clefts on the Moroccan Atlantic coast by plate tectonics. With the help of a web-based geoinformation system (Web-GIS) the collected geo-data can be centrally stored and used for monitoring and analysis without the need to install any software beforehand through uniform interfaces with the web browser. Furthermore, these data can be used with any GIS software for further use of the data. The aim of the work is to develop a Web-GIS for the monitoring and analysis of extension clefts on the Moroccan Atlantic coast.

## Inhaltsverzeichnis

<b>Danksagung .....</b>	II
<b>Kurzfassung .....</b>	III
<b>Abstract .....</b>	III
<b>Inhaltsverzeichnis.....</b>	<b>IV</b>
<b>1 Einführung.....</b>	<b>1</b>
1.1 Problemstellung.....	1
1.2 Ziel der Masterarbeit.....	2
1.3 Anforderungen.....	2
1.4 Erwartung an das System.....	3
1.5 Analyse der Bewertungskriterien für die Softwarekomponenten .....	3
1.6 Analyse der Web-Mapping-Anwendung.....	4
<b>2 Untersuchungsgebiet .....</b>	<b>5</b>
2.1 Geografischer Überblick .....	5
2.2 Extensionsklüfte .....	6
2.3 Geologische Entstehung .....	7
<b>3 Grundlagen eines Web-GIS .....</b>	<b>8</b>
3.1 Geodaten .....	8
3.1.1 Vektordaten .....	8
3.1.2 Rasterdaten.....	9
3.2 Metadaten .....	9
3.3 4D-Koordinaten .....	9
3.4 Geo-Informationssysteme (GIS) .....	9
3.5 Web-Kartografie .....	10
3.5.1 Statische Anwendungen.....	10
3.5.2 Interaktive Anwendungen.....	10
3.5.3 Kartenserver .....	11
3.5.4 Geodatenserver.....	12
3.5.5 Web-Mapping-Frameworks.....	12
3.6 Normen und Standards des Geoinformationswesens .....	12
3.6.1 Bedeutung .....	12
3.6.2 Organisationen .....	13
3.6.3 Web Map Service (WMS).....	14
3.6.4 Web Feature Service (WFS) .....	18
3.7 Inspire-Richtlinie .....	22
<b>4 Analyse der Softwarekomponenten.....</b>	<b>23</b>
4.1 Bewertungskriterien für Softwarekomponenten .....	23

4.2	Geodatenbanken .....	23
4.2.1	PostgreSQL mit der PostGIS-Erweiterung.....	23
4.2.2	MySQL mit der Spatial-Erweiterung .....	25
4.2.3	SQLite mit der SpatialLite-Erweiterung.....	26
4.2.4	Auswertung der Geodatenbanken .....	27
4.3	Kartenserver.....	28
4.3.1	GeoServer .....	28
4.3.2	MapServer / TinyOWS .....	30
4.3.3	deegree .....	33
4.3.4	Auswertung der Kartenserver .....	35
4.4	Web-Mapping-Anwendungen und Frameworks .....	37
4.4.1	OpenLayers .....	37
4.4.2	Leaflet.....	38
4.4.3	GeoExt.....	38
4.4.4	MapFish.....	39
4.4.5	Mapbender .....	40
4.4.6	Auswertung der Web-Mapping-Anwendungen und Frameworks .....	42
4.5	Betriebssystem.....	43
<b>5</b>	<b>Konzept und Entwurf des Systems.....</b>	<b>44</b>
5.1	Serviceorientierte Architektur (SOA).....	44
5.2	Ausgewählte Softwarekomponenten .....	44
5.3	Systemarchitektur und Schnittstellen.....	44
5.4	Kommunikation.....	46
5.5	Datenbankmodell .....	47
5.6	Import-Schnittstelle.....	48
5.7	Räumliche Analyse.....	49
5.8	Aufbereiten der Daten .....	51
5.9	Darstellung der zeitabhängigen Geodaten.....	52
5.10	Analyse der Extensionsklüfte.....	52
<b>6</b>	<b>Einrichtung des Systems .....</b>	<b>53</b>
6.1	Erste Überlegungen.....	53
6.2	Grundlage .....	53
6.3	Installation des Betriebssystems.....	53
6.4	Installationsskripte .....	54
6.5	Konfigurationsskripte .....	55
6.5.1	Konfigurationskript des Systems.....	55
6.5.2	Konfiguration von PostgreSQL / PostGIS .....	55
6.5.3	Konfiguration von TinyOWS.....	55
6.5.4	Konfiguration von MapServer.....	57

6.5.5	Konfiguration von Mapbender3 .....	58
6.5.6	Erstellen einer Anwendungen .....	59
6.6	Import der Geodaten .....	61
6.7	Weitere Analysefunktionen .....	62
<b>7</b>	<b>Nutzung des Systems .....</b>	<b>63</b>
7.1	Aufrufen des Web-GIS .....	63
7.2	Importieren von Daten .....	63
7.3	Darstellung der Geodaten .....	64
7.4	Analyse der Geodaten .....	67
7.5	Nutzung mit lokaler GIS-Software .....	70
7.6	Bearbeiten der importierten Daten .....	70
<b>8</b>	<b>Ergebnisse .....</b>	<b>71</b>
<b>9</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>72</b>
9.1	Zusammenfassung .....	72
9.2	Fazit .....	74
9.3	Ausblick .....	74
<b>Glossar .....</b>	<b>VII</b>	
<b>Quellenverzeichnis .....</b>	<b>IX</b>	
<b>Abbildungsverzeichnis .....</b>	<b>XII</b>	
<b>Tabellenverzeichnis .....</b>	<b>XIII</b>	
<b>Anhang .....</b>	<b>XIV</b>	
Anhang A – Ordnerstruktur .....	XIV	
Anhang B – Inhalt der CD .....	XVI	

# 1 Einführung

Durch eine Kooperation der marokkanischen Universität „Cadi Ayyad de Marrakech“ mit der Hochschule Neubrandenburg entstanden einige Projekte. Ein Projekt beschäftigt sich mit der Überwachung und Analyse von Extensionsklüften an der marokkanischen Atlantikküste. Im Rahmen des Projektes soll ein Geoinformationssystem mit aktuellen Webtechnologien und mit freier Software bzw. Open-Source-Software (FOSS) konzipiert und umgesetzt werden, mit dem die Ausbreitung der Extensionsklüfte überwacht und analysiert werden kann.

Geoinformationssysteme (GIS) existieren seit dem Karten von Menschen gezeichnet werden. Die Anforderungen an Geodaten und Karten haben sich in den letzten Jahren deutlich gewandelt. Durch die Digitalisierung werden vielseitige Informationen erfasst und bereitgestellt. Immer mehr dieser Daten werden zeitnah verändert und ergänzt. Dadurch nimmt auch die Informationsdichte stetig zu. Diese Daten können im Internet für jeden bereitgestellt werden, wodurch einheitliche Schnittstellen notwendig sind. Durch diese Entwicklung entstand die Geodateninfrastruktur (GDI), die ein Netzwerk zum Austausch von Geodaten mit verteilten Geodatenbeständen ist. Gewährleistet wird der Austausch der Geodaten durch aufgestellte Normen und Standards. Dadurch wird zunehmend auf webbasierte Komponenten gesetzt. Vorteil eines solchen webbasierten Geoinformationssystems (Web-GIS) ist die ständige Erreichbarkeit durch das Internet mit einem beliebigen Webbrowser und eine flexible Austauschbarkeit einzelner Softwarekomponenten. Das macht die Nutzung lokaler GIS-Software, die zwingend auf dem Rechner installiert werden muss, zur Erstellung und Darstellung der Geodaten überflüssig. Lokale GIS-Software kann darüber hinaus verwendet werden, um diese Daten zu analysieren. Durch diesen Trend wurden einige FOSSGIS-Produkte hervorgebracht, die bei der Wahl der geeigneten Softwarekomponenten untersucht werden sollen.

## 1.1 Problemstellung

Die Implementierung eines webbasierten Geoinformationssystems ist oft sehr aufwendig und für Ungeübte in diesem Themenbereich schwierig. Hingegen ist die Installation der benötigten Systemkomponenten (Datenbank, Kartenserver, Web-Mapping-Anwendung) einfach zu realisieren. Lediglich das Einrichten des Systems an individuelle Anforderungen stellt einen hohen Aufwand bei der Umsetzung dar.

Die erhobenen Geodaten besitzen zusätzlich eine zeitliche Komponente, die bei der Realisierung des Projektes berücksichtigt werden muss. Der Import der Geodaten in das System soll durch ein Konzept realisiert werden.

## **1.2 Ziel der Masterarbeit**

Ziel dieser Arbeit ist die Konzeption und Umsetzung eines webbasierten Geoinformationssystems (Web-GIS) zur Überwachung und Analyse von Extensionsklüften an der marokkanischen Atlantikküste. Die Daten sollen einfach in das System integriert und bereitgestellt werden können.

Folgende Ziele sind in dieser Arbeit zu erreichen:

- Auswahl und Analyse von geeigneter Softwarekomponenten
- Umsetzung eines webbasierten Geoinformationssystems mit den ausgewählten Softwarekomponenten
- Entwicklung von Methoden für die Integration, Darstellung und Erfassung der zeitabhängigen Geodaten

## **1.3 Anforderungen**

Folgende Anforderungen sind für die Umsetzung des Projektes einzuhalten:

- Die Wahl der geeigneten Softwarekomponenten liegt ausschließlich bei FOSS-Produkten ohne zusätzliche Kosten
- Die Softwarekomponenten sollen unabhängig im System interagieren und damit sicherstellen, dass die einzelnen Komponenten auch durch andere ersetzt werden können
- Standardisierte Schnittstellen sollen den Datenaustausch gewährleisten, damit das System mit anderen Geoinformationssystemen kommunizieren kann
- Die Geodaten sollen durch eine selbst definierte Schnittstelle in das System importiert werden können
- Die Nutzung des Systems erfolgt durch die Web-Mapping-Anwendung und QGIS als lokale GIS-Software
- Die grafische Benutzeroberfläche der Web-Mapping-Anwendung und deren Funktionen sollen durch den Benutzer anpassbar sein

## **1.4 Erwartung an das System**

Das System soll für die Überwachung und weitere Analyse der Extensionsklüfte eingesetzt werden. Außerdem dient es der Visualisierung der erhobenen Geodaten. Die zeitliche Komponente der Geodaten ist hierbei zu berücksichtigen. Heutige GIS-Systeme unterstützen problemlos 3D-Geodaten. Hingegen erfordern 4D-Geodaten mit einer zeitlichen Komponente weitere individuelle Konzepte bei der Umsetzung. Das System sollte einfach zu bedienen sein und im Web für jeden zur Verfügung stehen. Hierfür kann sich an die Inspire-Richtlinie, die bereits in Europa die Nutzung der Geodaten beschreibt, orientiert werden. Die benötigten Geodaten werden in Marokko erhoben und benötigen eine vorherige Verarbeitung für das System. Hierfür eignet sich ein einfaches Format, um die Daten in das System zu importieren. Dabei ist die Einhaltung eines einheitlichen Koordinatenreferenzsystems für die Geodaten notwendig, der bei diesem Format berücksichtigt werden muss und ggf. durch eine Koordinatentransformation erfolgt.

## **1.5 Analyse der Bewertungskriterien für die Softwarekomponenten**

Notwendig für die Wahl der geeigneten Softwarekomponenten sind Kriterien zur Bewertung. Damit die Softwarekomponenten auf einem freien Betriebssystem laufen können, ist das erste Kriterium die unterstützten Betriebssysteme der Softwarekomponenten. Das zweite Kriterium betrifft den Aufwand bei der Installation und Konfiguration mit den Softwarekomponenten, welcher möglichst gering sein sollte. Auch zusätzlicher Programmieraufwand sollte so gering wie möglich sein. Alle Softwarekomponenten unterliegen einer Lizenz, hierbei wäre es sinnvoll die Lizenzen zu vergleichen. Die Daten müssen im System zur Auswertung abgelegt werden, daher ist ein Kriterium bei der Wahl von unterstützten Geometrien wichtig. Weiterhin relevant sind der Funktionsumfang und die Features der Softwarekomponenten, um Extensionsklüfte darstellen zu können. Für diese Darstellung der Geodaten ist das Kriterium der Visualisierung erforderlich. Anhand der Anforderungen ist die Konformität ein wichtiger Aspekt für die standardisierten Schnittstellen. Das System sollte schnelle Interaktionen mit der Karte bieten, dafür ist die Performanz des Kartenservers ein weiteres Kriterium. Auch wichtig sind Dokumentationen und unterstützte Datenformate bei der Wahl des Kartenservers.

## 1.6 Analyse der Web-Mapping-Anwendung

Die Web-Mapping-Anwendung stellt die nutzbare Anwendung im Webbrowser dar. Für die Funktionalitäten der Anwendung wird in der Abb. 1.1 das Anwendungsfalldiagramm (Use-Case-Diagramm) mit den beiden Nutzern „User“ und „Administrator“ dargestellt. Der User verfügt über alle Möglichkeiten der reinen Betrachtung der dargestellten Geodaten, ohne diese explizit verändern zu können. Dabei werden dem User verschiedene Elemente zur Navigation der Karte und der Layer, zur Auslesung der Geodaten, zum Drucken von Bereichen, zur Messung, zum Hinzufügen von weiteren WMS-Diensten und zum Ändern von Maßstab und Referenzsystem geboten. Der Administrator kann alle Bedienelemente und Layer anpassen, sowie neue Geodaten importieren.

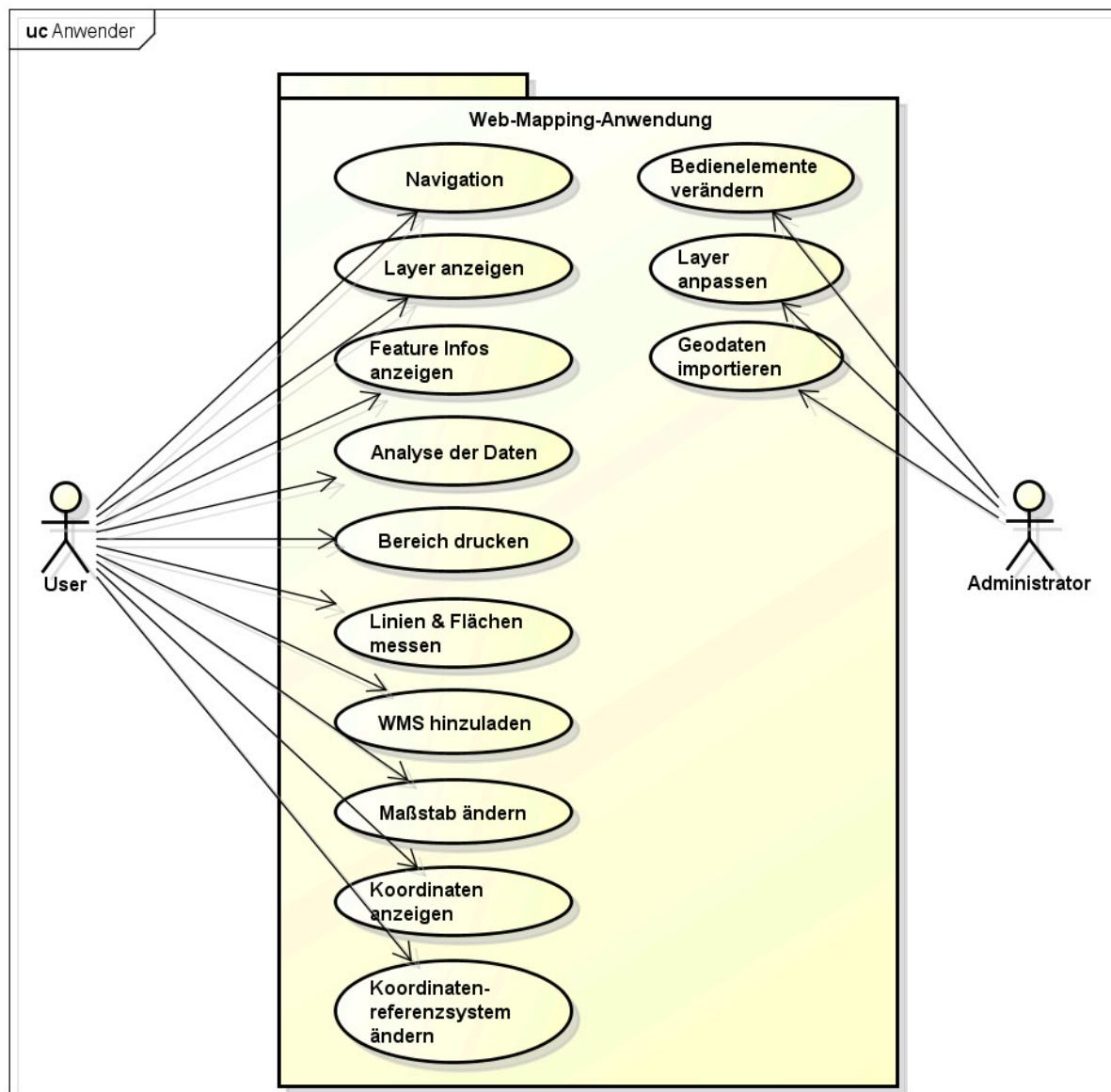


Abb. 1.1: Anwendung der Web-Mapping-Anwendung mit den Nutzern

powered by Astah

## 2 Untersuchungsgebiet

### 2.1 Geografischer Überblick

Das Königreich Marokko ist ein Staat im Nordwesten Afrikas, welches im Westen an den Atlantischen Ozean, im Norden an das Mittelmeer und im Osten an Algerien grenzt (siehe Abb. 2.1). Die Landeshauptstadt Rabat ist seit 1912 die Residenz der marokkanischen Könige. Das Land hat eine Fläche von 446.550 km<sup>2</sup> und eine Einwohnerzahl von ca. 33 Millionen. (vgl. [Aus17])

Geprägt wird die Natur durch eine weite flachwellige Steppenlandschaft (Marokkanische Mëseta) im Norden und Osten. Während im Norden des Landes Mittelmeerklima herrscht, nehmen im Süden zur Sahara hin die Niederschläge rapide ab. Marokko hat eine 400 km lange und steile Küste am Mittelmeer und eine 1400 km lange und deutlich flacheren Küstenraum am Atlantischen Ozean. Die Haupteinnahmequellen des Landes bestehen aus Landwirtschaft, dem Bergbau und der Tourismusbranche. (vgl. [Ler10])



Abb. 2.1: Karte von Marokko (übernommen von [Col15])

## 2.2 Extensionsklüfte

Der Begriff Extensionsklüfte wird beschrieben aus den beiden Begriffen Extension und Klüfte.

**Extension** beschreibt generell eine Längenänderung in Form von Verlängerung, Ausdehnung oder Deformierung. Dieser Deformationswert (Extensionswert) einer Linie ergibt sich aus dem Verhältnis zwischen Längenänderung  $\Delta l$  und Ursprungslänge  $l_u$ . (vgl. [Spe00a])

**Klüfte** sind feine Trennflächen im Gestein, die durch winzige Bewegungen der Tektonik entstehen. Die Arten der Ausbreitungen sind in der Abb. 2.2 dargestellt und entstehen durch drei verschiedene Wege. (vgl. [Spe00b])

- Die Kluftflächen weichen bei einem Bruch senkrecht zur Bruchebene auseinander
- Parallele Bewegung, die senkrecht zur Bruchfront verläuft, führt zum Scherbruch Typ A.
- Parallele Bewegung, die parallel zur Bruchfront verläuft, führt zum Scherbruch Typ B.

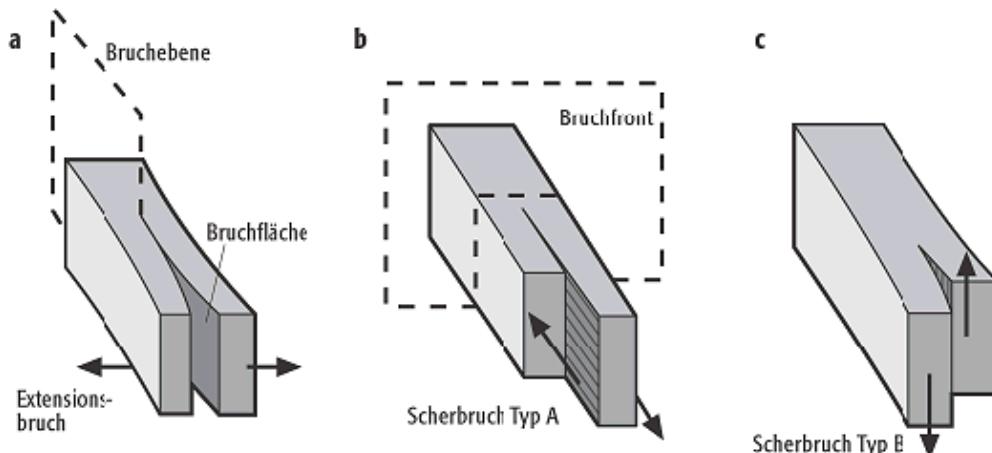


Abb. 2.2: Drei verschiedene Arten der Bruchausbreitung (übernommen von [Spe00b])

Charakterisiert werden Extensionsklüfte durch zwei ebene Bruchflächen, bei denen keine merklichen Verschiebungen stattfinden. Ein entsprechender Extensionsbruch breite sich wie auf der Abb. 2.3 dargestellt senkrecht zur Richtung der kleinsten Hauptspannung  $\sigma_3$  aus. Dadurch können Rückschlüsse auf die Richtung der minimalen Hauptspannung  $\sigma_3$  zur Zeit der Kluftentstehung gezogen werden. Hingegen bleibt die räumliche Orientierung der Hauptspannungen  $\sigma_1$  und  $\sigma_2$  meist unklar, da diese in der Kluftebene liegen. (vgl. [Spe00b])

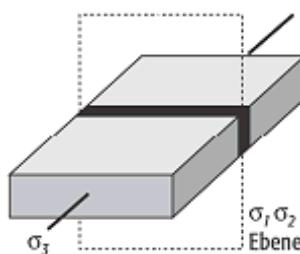


Abb. 2.3: Extensionskluft in Bezug seiner drei Hauptspannungen (übernommen von [Spe00b])

### ***2.3 Geologische Entstehung***

Extensionsklüfte sind für gewöhnlich geöffnet und entstehen hauptsächlich durch die Platten-tektonik, Faltung (durch Schichtbiegung), Bruchbildung bei Entlastung oder bei Abkühlung. In diesen offenen Rissen können sich enge Gesteinsgänge mit Mineralien füllen (meistens Quarz oder Kalzit). Diese Klüfte existieren in unterschiedlichen Größenordnungen vom Mikrobereich bis hin zum Zentimeter- und Meterbereich. Klüfte verlaufen manchmal gekrümmmt in einheitlichen Gesteinen. Dabei gehen unterschiedliche Bereiche verschiedener Bruchklassen ineinander über. Unter Einfluss von hohen Porenflüssigkeitsdrucken breiten sich Klüfte in tieferen Bereichen der Erde (ab 5 km und mehr) aus und entstehen während der Überlagerung und Kompaktion (Überlagerungsdruck) in Sedimentgesteinen. Abgebaut werden die hohen Porenflüssigkeitsdrucke durch das Aufreißen der Klüfte. (vgl. [Spe00b])

### 3 Grundlagen eines Web-GIS

Bevor die Analyse der Softwarekomponenten erfolgt, werden in diesem Kapitel die grundlegenden Begriffe, die für das Verständnis eines webbasierten Geoinformationssystems (Web-GIS) und für die Umsetzung der Extensionsklüfte benötigt werden, erklärt.

#### 3.1 Geodaten

Als Geodaten werden Daten mit einem eindeutigen Raumbezug bezeichnet. Dieser Raumbezug kann direkt (z. B. durch Koordinaten) oder indirekt (z. B. durch Beziehungen) erfolgen. Diese Daten beschreiben einzelne Objekte der Landschaft wie z. B. Gegenstände, Geländeformen und Infrastrukturen an der Erdoberfläche. Einteilen lassen sich diese aus informationstechnischer Sicht in: (vgl. [Wik17d])

- Geometrie (Lage und Form der Objekte)
- Topologie (explizit gespeicherte räumliche Beziehungen)
- Semantik (Sachdaten zur Beschreibung der Daten)
- Dynamik (Zeitliche Veränderungen der Objekte)

Mit der Nutzung von GIS-Funktionalitäten lassen sich Geodaten miteinander verknüpfen, um neue Informationen ableiten zu können. Diese sind notwendig, um Extensionsklüfte beschreiben zu können. Aufgeteilt werden Geodaten in die zwei großen Teilkomplexe Geobasisdaten und Geofachdaten. (vgl. [Pro01], Stichwort: Geodaten)

1. **Geobasisdaten** beschreiben interessensneutral die Landschaft (Topografie) und Liegenschaften der Erdoberfläche und dienen als Basis für Geofachdaten. Dazu gehören Daten der Vermessungsverwaltung als Grundlage für viele Anwendungen. (vgl. [Pro01], Stichwort: Geobasisdaten)
2. **Geofachdaten** sind Daten aus einem spezifischen Fachgebiet (z. B. Statistik, Boden, Naturschutz etc.). Aber auch anwendungsspezifische Daten eines Fachanwenders z. B. Leitungsdaten oder Kundendaten eines Versorgungsunternehmens. (vgl. [Pro01], Stichwort: Geofachdaten)

#### 3.1.1 Vektordaten

Nach Objekten strukturierte Geodaten werden als Vektordaten bezeichnet. Dabei handelt es sich durch Koordinaten definierte Punkt-, Linien- und Flächenobjekte. Repräsentiert werden die Objekte durch eine Liste von geordneter X- und Y-Koordinaten. Bei 3D-Koordinaten wird die Liste um eine weitere Z-Koordinate erweitert. (vgl. [Pro01], Stichwort: Vektordaten)

Die gemessenen Extensionsklüfte sind dementsprechend Vektordaten, die durch Punkt- und Flächenobjekte mit den passenden Koordinaten gespeichert werden.

### **3.1.2 Rasterdaten**

Rasterdaten sind nicht nach Objekten strukturiert, sondern entstehen vorwiegend durch das Scannen und Georeferenzieren von analogen Karten, Bildern und bei Luft- oder Satellitenaufnahmen. Diese bestehen aus einer Anzahl von Bildpunkten (Pixel), wobei jedem Bildpunkt einem Wert zugeordnet wird. Die Summe aller Bildpunkte ergibt das kartografische Gesamtbild und dient dadurch vorwiegend als räumliche Bezugsgrundlage. In vielen GIS-Anwendungen eignen sich Rasterdaten für die Hintergrundinformationen. (vgl. [Pro01], Stichwort: Rasterdaten)

### **3.2 Metadaten**

Metadaten sind Daten über Daten und sind in der ISO-Norm 19115 definiert. Diese sind notwendig zur Charakterisierung der Geodaten und beschreiben deren Eigenschaften. Zu den Eigenschaften gehören Koordinatensystem, Raster- oder Vektorformat, Datenqualität und viele weitere Angaben. (vgl. [Pro01], Stichwort: Metadaten)

### **3.3 4D-Koordinaten**

Die 3D-Raumkoordinaten (X,Y,Z) der Extensionsklüfte werden mit einer zusätzlichen zeitlichen Komponente  $t$  gespeichert. Diese Komponente wird den Zeitpunkt für die Erhebung der Geodaten darstellen und ist für die Ausbreitung der Extensionsklüfte zwingend benötigt. Mit dem Datenbestand können dann zeitlich verändernde geowissenschaftliche Fragestellungen über mehrere Zeiträume analysiert und weitere Objekte für die Darstellung geschaffen werden.

### **3.4 Geo-Informationssysteme (GIS)**

Ein rechnergestütztes System zur Verarbeitung mit Geodaten, bestehend aus Hardware, Software, Daten und deren Anwendung, wird als Geoinformationssystem bezeichnet. Mit diesem System können raumbezogene Daten digital erfasst, verändert, gespeichert, organisiert, modelliert, analysiert und auch grafisch repräsentiert werden. Die genutzten Daten liegen hierfür in Form von Raster- oder Vektordaten vor. (vgl. [Pro01], Stichwort: Geoinformationssystem)

Ein Geoinformationssystem (GIS) unterscheidet sich im Wesentlichen an der Art der Datenhaltung und der verwendeten Datenstrukturen von anderen Informationssystemen (IS). Getrennt werden die Daten in Sach- und Geometriedaten. Strukturiert wird nach dem Ebenenkonzept, wobei die Geometriedaten in die Layer gespeichert und durch Überlagerung zusammengeführt werden können. Demgegenüber werden Sachdaten häufig in Form von Tabellen in einer relationalen Datenbank gespeichert. (vgl. [Spe01])

### **3.5 Web-Kartografie**

Die Methode der Web-Kartografie stellt kartografische Informationen über eine Webseite zur Verfügung. Bei der Art der Bereitstellung wird zwischen einer statischen und interaktiven Anwendung unterschieden. (vgl. [Mit08], S.8-9)

Der Begriff „Web-Mapping“ fällt häufig bei dieser Anwendung und beinhaltet verschiedene Leistungsstufen von einer statischen Visualisierung bis zum GIS mit Onlinefunktionalitäten. (vgl. [Pro01], Stichwort: Webmapping)

#### **3.5.1 Statische Anwendungen**

Bei statischen Anwendungen werden vorgefertigte Karten auf einer Webseite eingebunden, deren Darstellung mit jedem Browser möglich ist. Die verwendeten Karten werden beispielsweise von einer Papierkarte gescannt oder aus Vektordaten generiert und als Rasterdatei in einem gängigen Bildformat (z. B. GIF, PNG oder JPEG) ausgegeben. (vgl. [Pro01], Stichwort: Statische Karte)

Für diese Art der Anwendung sind nur wenig Fertigkeiten mit dem Web notwendig, da nur Rasterdaten für den Nutzer zur Verfügung gestellt werden. In den Anfängen der Webkartografie bildeten statische Anwendungen den Grundstein für spätere interaktive Anwendungen. Mit zunehmender Entwicklung der Webtechnologien wurden weitere Werkzeuge für die Erstellung einer Web-Mapping-Anwendung bereitgestellt. Interaktive Anwendungen wurden praktikabler und ersetzten die statischen Anwendungen weitgehend. (vgl. [Mit08], S.8-9)

#### **3.5.2 Interaktive Anwendungen**

Im Gegensatz zu statischen Anwendungen handelt es sich bei interaktiven Anwendungen um eine webbasierte und komplexe Anwendung. Interaktionen mit der Karte sind dadurch möglich. Beispielsweise durch Auswahl von bestimmten Kartenebenen oder durch Zoomen in einem bestimmten Bereich der Karte. Die Karte wird bei jeder Interaktion aktualisiert. Für das Betreiben und Einrichten solcher Seiten mit Interaktion sind im Gegensatz zu statischen Anwendungen besondere Fähigkeiten mit dem Web notwendig. (vgl. [Mit08], S.9)

Dieses System lässt sich als Drei-Schichten-Modell darstellen (siehe Abb. 3.1), wobei eine Schicht dem Client und die beiden anderen Schichten den Servern zuzuordnen sind.

- **Präsentationsschicht** stellt die Bedienelemente für den Nutzer dar
- **Fachlogikschicht** stellt Module für die Datenverarbeitung zur Verfügung
- **Datenhaltungsschicht** speichert die Geodaten auf einem Speicher, um diese wieder abrufen zu können

Im Bereich Web-GIS hat sich diese Architektur aus Gründen der Kapselung von Internet-technologien und der besseren Skalierbarkeit von Servern bewährt. (vgl. [Uiz15])

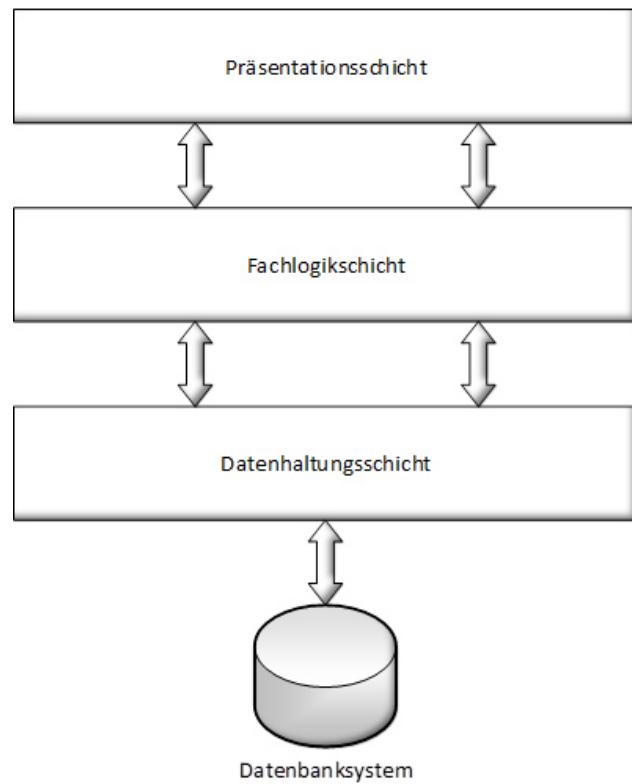


Abb. 3.1: Drei-Schichten-Architektur eines GIS (übernommen von [Uiz15])

### 3.5.3 Kartenserver

Ein Kartenserver ist eine zusätzliche Komponente der Fachlogikschicht, die den Umgang mit Geodaten erleichtert und Karten dynamisch erzeugt.

In der Abb. 3.2 wird der Vorgang zwischen dem Client und dem Server über das Internet dargestellt. Der Client fordert eine Karte vom Webserver an und der Webserver überreicht die Anfrage an den Kartenserver. Der Kartenserver nimmt die Anfrage entgegen und trägt alle nötigen Daten vom Datenbanksystem zusammen. Danach überträgt dieser eine fertige Karte zurück an den Webserver, welche wiederum an den Client überreicht wird. Als Resultat wird die fertige Karte im Webbrower am Client angezeigt. (vgl. [Uiz15])

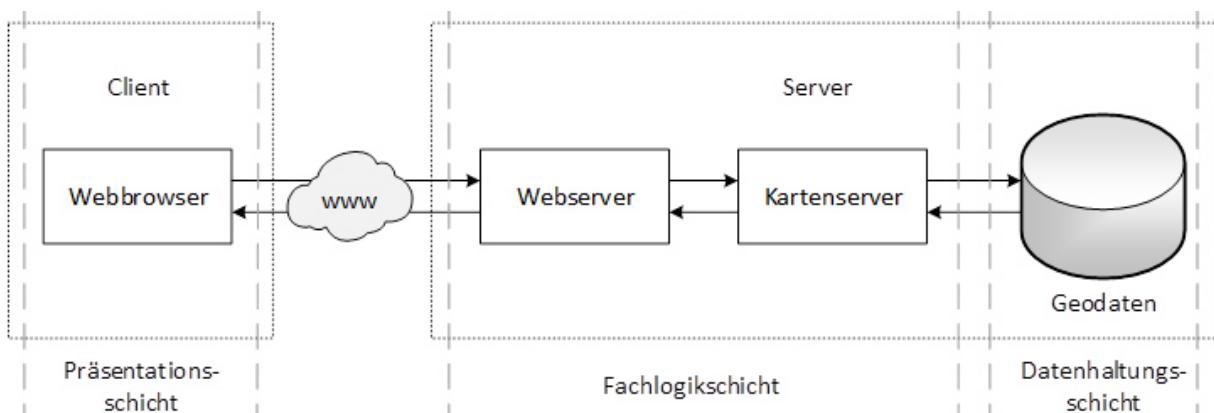


Abb. 3.2: Kommunikation zwischen Client und Server (übernommen von [Uiz15])

### **3.5.4 Geodatenserver**

Der Geodatenserver liegt in der Datenhaltungsschicht und beinhaltet die Geodaten, die für die Kommunikation mit dem Kartenserver notwendig sind, in einem Datenbankmanagementsystem. Vektordaten werden darin gespeichert, um räumliche Analysen zu ermöglichen. Diese sind unter anderem notwendig für die Berechnung der Ausbreitung von Extensionsklüften.

### **3.5.5 Web-Mapping-Frameworks**

Um die Kommunikation mit dem Web-GIS zwischen der Präsentationsschicht und der Fachlogikschicht realisieren zu können, werden Webanwendungen benötigt. Web-Mapping-Frameworks bieten weitreichende Möglichkeiten, um Webanwendungen zu programmieren. Dazu zählen vorgefertigte Komponenten mit Funktionen, wie z. B. Navigations- und Analysewerkzeuge und Schnittstellen zum Import von Geodaten aus unterschiedlichen Datenquellen.

## **3.6 Normen und Standards des Geoinformationswesens**

Die Entwicklung von Normen und Standards im Geoinformationswesen führte zu der Kompatibilität der Geoinformationssysteme und ist relevant für die Infrastruktur eines heutigen Web-GIS. Zunächst wird der Unterschied zwischen den beiden Begriffen Normen und Standards erklärt. Danach wird auf die bestehenden Organisationen und Institute für die Normung und Standardisierung eingegangen. Anschließend werden die bekanntesten internationalen Spezifikationen dargestellt.

### **3.6.1 Bedeutung**

Es wurde gezeigt, dass Normen und Standards entscheidend für heutige Web-GIS sind. Bevor auf wichtige Organisationen und Institute eingegangen wird, ist zunächst der Unterschied zwischen den Begriffen Normen und Standards zu klären.

Eine Norm ist eine technische Spezifikation, die von einer anerkannten Normenorganisation wiederholt zur Anwendung geführt wird. Die Nutzung steht jedem frei zur Verfügung, wird aber grundsätzlich nicht zwingend vorgeschrieben. Unterschieden werden Normen in nationale, europäische und internationale Norm. (vgl. [Pro01], Stichwort: Norm)

Als Standard wird eine breit akzeptierte und angewandte Regel bezeichnet. Diese kann aus einer Normungsarbeit hervorgehen oder wird durch eine weite Verbreitung als De-facto-Standard gesetzt. Ein solcher Standard erhöht die Flexibilität, Funktionalität und Produktivität eines Informationssystems und ermöglicht die Kommunikation zwischen verschiedenen Informationssystemen. (vgl. [Pro01], Stichwort: Standard)

### **3.6.2 Organisationen**

Auf internationaler Ebene sind die beiden Organisationen OGC und ISO für die Definition von Geoinformationen von großer Bedeutung.

Das **Open Geospatial Consortium** (OGC) ist eine internationale Non-Profit-Organisation, mit dem Ziel, offene Standards für den Austausch räumlicher Daten zu schaffen und für jedermann frei zur Verfügung zu stellen. Bei der Entwicklung werden lediglich nur Spezifikationen geschaffen und keine eigene Anwendungssoftware. Diese OGC-Standards werden bereits in vielen Bereichen wie Umwelt, Verteidigung, Gesundheit, Landwirtschaft, Meteorologie und vieles mehr verwendet. 1994 wurde die Organisation gegründet (ehemals unter den Namen Open GIS Consortium bis zum Jahre 2004) und vereint viele Mitglieder aus unterschiedlichen Bereichen (z. B. GIS-Anbieter, Dienstleister, IT-Firmen, Universitäten). Durch die zunehmende Bedeutung von Geodaten im Bereich der IT, finden sich in diesem Konsortium auch andere Firmen wieder, wie z. B. Microsoft, Google und der Datenbankhersteller Oracle. (vgl. [Pro01], Stichwort: Open Geospatial Consortium; [Mit08], S.234)

Die unabhängige und nicht staatliche **internationale Organisation für Normung** (ISO) wurde 1947 gegründet und koordiniert weltweit die Normung. Die Mitglieder bringen Experten zusammen, um das Wissen zu teilen und Lösungen für globale Herausforderungen zu entwickeln. Es beinhaltet 163 nationale Normungsgremien und ist die Mutterorganisation der meisten nationalen Normungsinstitute. (vgl. [Pro01], Stichwort: ISO)

Das ISO/TC 211 arbeitet seit 1994 an der Normung auf dem Gebiet der Geoinformation. Seit 1997 beginnt die enge Kopplung mit dem OGC. Die ISO-Norm 191xx wurde von ihnen mit 20 verschiedenen Standards erarbeitet. Sie arbeiten in 5 Arbeitsausschüssen (Working Group), die sich wie folgt zusammensetzen. (vgl. [Pro01], Stichwort: ISO/TC 211)

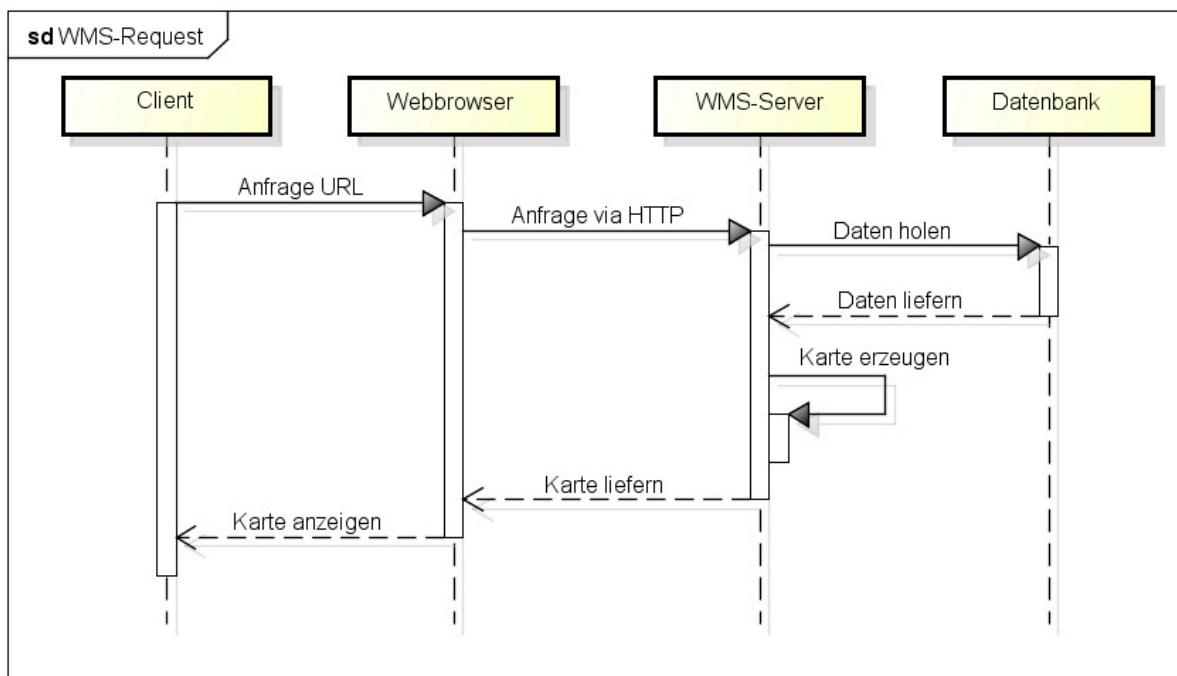
- WG1.** Referenzmodell für Standards auf dem Gebiet Geoinformation
- WG2.** Raumbezogene Datenmodelle und Operationen
- WG3.** Verwaltung raumbezogener Daten
- WG4.** Raumbezogene Dienste
- WG5.** Profile und Funktionale Standards

Im nachfolgenden sind einige relevante ISO-Normen für das Geoinformationswesen dargestellt.

- **ISO 19107 - Spatial Schema** - Beschreibung von Geometrie und Topologie
- **ISO 19115 - Metadata** - Elemente zur Beschreibung von Geodaten
- **ISO 19128 - Web Map Service Interface** - Webservice für Rasterdaten
- **ISO 19142 - Web Feature Service** - Webservice für Vektordaten
- **ISO 19125 - Geographic Information** - SQL-Schema für Ablage, Abholung, Abfrage und Aktualisierung geo-referenzierter Informationen über SQL

### 3.6.3 Web Map Service (WMS)

Der Web Map Service (WMS) ist eine von OGC verabschiedete Spezifikation, um die raster- und vektorbasierten Geodaten als Rasterkarten im Web zu publizieren. Das Sequenzdiagramm in der Abb. 3.3 soll diesen Vorgang verdeutlichen. Der Client wählt zunächst einen beliebigen rechteckigen Ausschnitt aus dem Geodatenbestand und gibt seine Anfrage an den Webbrowser weiter. Als Technologie für die Kommunikation zwischen Webbrowser und WMS-Server wird das Hypertext Transfer Protocol (HTTP) genutzt. Es wird eine Anfrage als HTTP-Request vom Webbrowser an den WMS-Server übermittelt. Dies geschieht im Browser durch Hinzufügen von weiteren Parametern an die URL oder wird von GIS-Programmen automatisch ergänzt. In der Abb. 3.4 wird ein solcher allgemeiner Aufbau am Beispiel Niedersachsens Geobasisdaten verdeutlicht. Die Parameter werden durch das &-Zeichen getrennt. Alle Parameter werden nach dem Key-Value Prinzip übergeben, bei dem ein Schlüsselwort ein oder mehrere Werte beinhaltet kann. Als Nächstes folgt die Funktionsweise nach Kapitel 3.5.3, bei dem der WMS-Server die Daten aus der Datenbank holt und die Karte erzeugt. Anschließend wird der WMS-Server die Karte an den Webbrowsers leiten, wodurch diese beim Client angezeigt wird. (vgl. [Pro01], Stichwort: Web Map Service; [Lgb17a])



powered by Astah

Abb. 3.3: Sequenzdiagramm eines WMS-Service mit dem Request GetMap (vgl. [Kad09], S.16)

URL der WMS-Schnittstelle <http://www.geobasisdaten.niedersachsen.de/bestand?SERVICE=WMS&VERSION=1.1.1&REQUEST=GetCapabilities> Parameter für den Request

Abb. 3.4: Allgemeiner Aufbau einer Anfrage (übernommen von [Gdi15], S.2)

Über drei Methoden wird auf diese standardisierte Schnittstelle mit einem HTTP-Request zugegriffen:

- **GetCapabilities** (Leistungsverzeichnis)

Die Tabelle 3.1 zeigt die Parameter zur Abfrage der Fähigkeit eines WMS. Die Informationen der Antwort (XML-Dokument) enthalten neben allgemeinen Angaben zum Anbieter auch die abfragbaren Layer und unterstützten Koordinatenreferenzsysteme der Karte. Eine Abfrage kann wie folgt aussehen:

```
http://www.geobasisdaten.niedersachsen.de/bestand?SERVICE=WMS&REQUEST=GetCapabilities&VERSION=1.1.1&FORMAT=application/vnd.ogc.xml
```

Parameter	Beschreibung	Pflicht
<b>SERVICE =</b>	Der Dienst, der Kartenbilder liefert. In diesem Fall immer <b>WMS</b> .	Ja
<b>REQUEST =</b>	Zeige alles, was als Antwort geliefert wird. In diesem Fall immer <b>GetCapabilities</b> .	Ja
<b>VERSION =</b>	Antwort in einer bestimmten Version der Schnittstelle. (z. B. 1.1.0, <b>1.1.1</b> , 1.3.0)	Nein
<b>FORMAT =</b>	Öffnen der Antwort direkt in einem passenden Programm als XML-Format. Zum Beispiel: <b>application/vnd.ogc.xml</b>	Nein

Tabelle 3.1: WMS-Parameter für den Request GetCapabilities (vgl. [Gdi15]; [Geo17a])

Ein Layer erlaubt nur dann zusätzliche Sachinformationsabfragen mit dem Parameter GetFeatureInfo, wenn im XML-Dokument beim Element <Layer> hinter dem Parameter „queryable“ die Zahl „1“ steht (siehe Abb. 3.5).

```
<Layer queryable="1" opaque="0" cascaded="0">
  <Name>dtk100_v_bkg_b</Name>
  <Title>DTK100-V BKG Bestand</Title>
  <Abstract>Bestand - Topographische Karte 1:100.000 vom BKG</Abstract>
  <SRS>EPSG:31467</SRS>
  <LatLonBoundingBox minx="6.39421" miny="51.0403" maxx="12.0626" maxy="54.2203"/>
  <BoundingBox SRS="EPSG:31467" minx="3.33e+06" miny="5.66e+06" maxx="3.7e+06" maxy="6.01e+06"/>
  <Style>
    <Name>default</Name>
    <Title>default</Title>
    <LegendURL width="356" height="2016">
      <Format>image/png</Format>
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
        xlink:href="https://www.geobasisdaten.niedersachsen.de/mapgate/bestand?
          version=1.1.1&service=WMS&request=GetLegendGraphic&layer=dtk100_v_bkg_b&format=image/png"/>
    </LegendURL>
  </Style>
</Layer>
```

Abb. 3.5: Layer erlaubt die Abfrage von zusätzlichen Sachinformationen

- **GetMap** (Karte)

Die Tabelle 3.2 stellt die Parameter zur Abfrage eines georeferenzierten Rasterbildes dar. Optionen dieser Anfrage sind u. a. der gewünschte Kartenlayer, die Darstellung der Layer, das Koordinatensystem, der Kartenausschnitt, die Größe der Kartenausgabe und das Ausgabeformat. Eine Abfrage kann wie folgt aussehen:

```
http://www.geobasisdaten.niedersachsen.de/bestand?SERVICE=WMS&REQUEST=GetMap&VERSION=1.1.1&SRS=EPSG:31467&LAYERS=rasterdaten&STYLES=&BBOX=3545127.777778,5803954.166667,3552127.77778,5808426.388889&WIDTH=720&HEIGHT=460&FORMAT=image/png
```

Parameter	Beschreibung	Pflicht
<b>SERVICE =</b>	Der Dienst, der Kartenbilder liefert. In diesem Fall immer <b>WMS</b> .	Ja
<b>REQUEST =</b>	Zeige alles, was als Antwort geliefert wird. In diesem Fall immer <b>GetMap</b> .	Ja
<b>VERSION =</b>	Antwort in einer bestimmten Version der Schnittstelle. (z. B. 1.1.0, <b>1.1.1</b> , 1.3.0)	Ja
<b>SRS =</b> <b>CRS =</b>	Das zurückgelieferte Projektionssystem. (z. B. <b>EPSG:31467</b> ) Info: <b>SRS</b> bei Version 1.1.1 und <b>CRS</b> bei Version 1.3.0	Ja
<b>LAYERS =</b>	Eine bestimmte Kartenebene zurückliefern. (z. B. <b>dtk100_v_nf</b> )	Ja
<b>STYLES =</b>	Ein vordefinierten Stil wählen. Bleibt meist leer, wenn der Standardstil gewählt werden soll.	Ja
<b>BBOX =</b>	Gibt das angegebene Rechteck (Bounding-Box), basierend auf dem gewählten Projektionssystem aus. Die Angabe erfolgt durch Koordinatenpaare, die einzelnen Koordinatenpaare werden dann durch Komma getrennt. Nachkommastellen erfolgen durch anfügen eines Punktes.	Ja
<b>WIDTH =</b>	Die Karte wird als angegebene Pixelbreite auf dem Bildschirm angezeigt. (z. B. <b>720</b> )	Ja
<b>HEIGHT =</b>	Die Karte wird als angegebene Pixelhöhe auf dem Bildschirm angezeigt. (z. B. <b>460</b> )	Ja
<b>FORMAT =</b>	Gibt das Rasterbild in einem bestimmten Format zurück. (z. B. <b>image/png</b> )	Ja
<b>EXCEPTIONS =</b>	Ausgabeformat bei Fehlern. Standardmäßig: <b>application/vnd.ogc.se_xml</b>	Nein
<b>BGCOLOR =</b>	Hintergrundfarbe für die Karte. Standardmäßig: FFFFFF (weiß)	Nein
<b>TRANSPARENTE =</b>	Hintergrund transparent setzen. Standardmäßig: false	Nein
<b>TIME =</b>	Zeitwert des angeforderten Layers.	Nein
<b>SLD =</b>	URL zur Styled-Layer-Deskriptor (SLD) XML-Datei, welche die Kartenebenen gestalten	Nein
<b>SLD_BODY =</b>	Enthält den tatsächlichen URL-kodierten XML-Text, welche die Kartenebenen gestalten	Nein

Tabelle 3.2: WMS-Parameter für den Request GetMap (vgl. [Gdi15]; [Geo17a])

- **GetFeatureInfo** (Sachinformationen)

Der einzige optionale Service eines WMS zur Abfrage nach thematischen Sachinformationen der zugrunde liegenden Daten wird mit GetFeatureInfo realisiert. Die Parameter dafür sind in der Tabelle 3.3 dargestellt. Eine Abfrage kann wie folgt aussehen:

```
http://www.geobasisdaten.niedersachsen.de/bestand?SERVICE=WMS&REQUEST=GetFeatureInfo&VERSION=1.1.1&SRS=EPSG:31467&LAYERS=dtk100_v_b&BBOX=3226086.957000,5650000.000000,3773913.043000,6000000.000000&WIDTH=720&HEIGHT=460&QUERY_LAYERS=dtk100_v_b&X=360&Y=241&FEATURE_COUNT=1&INFO_FORMAT=text/html
```

Parameter	Beschreibung	Pflicht
<b>SERVICE =</b>	Der Dienst, der Kartenbilder liefert. In diesem Fall immer <b>WMS</b> .	Ja
<b>REQUEST =</b>	Zeige alles, was als Antwort geliefert wird. In diesem Fall immer <b>GetFeatureInfo</b> .	Ja
<b>VERSION =</b>	Antwort in einer bestimmten Version der Schnittstelle. (z. B. 1.1.0, <b>1.1.1</b> , 1.3.0)	Ja
<b>SRS =</b> <b>CRS =</b>	Das zurückgelieferte Projektionssystem. (z. B. <b>EPSG:31467</b> ) Info: <b>SRS</b> bei Version 1.1.1 und <b>CRS</b> bei Version 1.3.0	Ja
<b>LAYERS =</b>	Die abfragbare Kartenebene. (z. B. <b>dtk100_v_b</b> )	Ja
<b>BBOX =</b>	Gibt das angegebene Rechteck (Bounding-Box), basierend auf dem gewählten Projektionssystem aus. Die Angabe erfolgt durch Koordinatenpaare, die einzelnen Koordinatenpaare werden dann durch Komma getrennt. Nachkommastellen erfolgen durch anfügen eines Punktes.	Ja
<b>WIDTH =</b>	Die Karte wird als angegebene Pixelbreite auf dem Bildschirm angezeigt. (z. B. <b>720</b> )	Ja
<b>HEIGHT =</b>	Die Karte wird als angegebene Pixelhöhe auf dem Bildschirm angezeigt. (z. B. <b>460</b> )	Ja
<b>QUERY_LAYERS =</b>	Sachinformationsabfrage auf eine Kartenebene. (z. B. <b>dtk100_v_b</b> )	Ja
<b>X =</b> <b>I =</b>	Pixelpunkt für Sachinformationsanfragen (Ursprung: Oben-Links). Info: <b>X</b> bei Version 1.1.1 und <b>I</b> bei Version 1.3.0	Ja
<b>Y =</b> <b>J =</b>	Pixelpunkt für Sachinformationsanfragen (Ursprung: Oben-Links). Info: <b>Y</b> bei Version 1.1.1 und <b>J</b> bei Version 1.3.0	Ja
<b>FEATURE_COUNT =</b>	Maximale Anzahl der Sachinformationen wird für den ausgewählten XY-Wert dargestellt. (Ohne Angabe des Parameters beträgt dieser Wert <b>1</b> )	Nein
<b>INFO_FORMAT =</b>	Sachinformationen in diesem Format am Bildschirm darstellen. (z. B. <b>text/html</b> )	Nein
<b>EXCEPTIONS =</b>	Ausgabeformat bei Fehlern. Standardmäßig: <b>application/vnd.ogc.se_xml</b>	Nein

Tabelle 3.3: WMS-Parameter für den Request GetFeatureInfo (vgl. [Gdi15]; [Geo17a])

Durch die Abfrage GetCapabilities werden zunächst Informationen über den WMS-Server gesammelt und mit GetMap die benötigte Karte in Rasterform übermittelt. Die Funktionsweise von GetMap ist in der Abb. 3.3 als Sequenzdiagramm dargestellt.

Unterstützung findet die WMS-Schnittstelle inzwischen bei allen großen GIS-Herstellern und auch bei vielen FOSS-Projekten für Geoinformationssysteme.

### **3.6.4 Web Feature Service (WFS)**

Für viele Web-GIS reicht der WMS bei der Web-Mapping-Anwendung bereits aus. Dennoch sind WMS für Analysen ungeeignet. Für Analysen mit vektoriellen Daten eignet sich daher der WFS. Speziell bei der Analyse mit lokaler Software ist der Einsatz von WFS unumgänglich. Auch das verändern der Geodaten durch Transaktionen ist damit möglich. Der WFS-T („Transactional“) bietet diese Funktionalitäten zum verändern, löschen und hinzufügen von Geodaten.

Der Web Feature Service (WFS) ist eine von OGC verabschiedete Spezifikation zum Zugriff auf vektorielle Geodaten über HTTP und ermöglicht das Suchen nach raumbezogenen Informationen aus beliebigen objektorientierten Vektordatenbeständen. Unterstützt werden beliebige Kodierungen von Attributen. Die Geometrie der Objekte werden im Format GML (Geography Markup Language) zur Verfügung gestellt. (vgl. [Pro01], Stichwort: Web Feature Service; [Lgb17b])

Alle nach OGC definierten WFS-Operationen für die Versionen 1.0.0, 1.1.0 und 2.0.0 werden in der Tabelle 3.4 dargestellt. Die ersten sechs Operationen sind mit jeder Version möglich, alle anderen funktionieren nur mit einer bestimmten Version.

Operation	WFS 1.0.0	WFS 1.1.0	WFS 2.0.0
<b>GetCapabilities</b>	X	X	X
<b>DescribeFeatureType</b>	X	X	X
<b>GetFeature</b>	X	X	X
<b>Transaction</b>	X	X	X
<b>LockFeature</b>	X	X	X
<b>GetFeatureWithLock</b>	X	X	X
<b>GetGMLObject</b>		X	
<b>GetPropertyValues</b>			X
<b>ListStoredQueries</b>			X
<b>CreateStoredQuery</b>			X
<b>DropStoredQuery</b>			X
<b>DescribeStoredQueries</b>			X

Tabelle 3.4: Alle nach OGC definierten WFS-Operationen (vgl. [Gdi13], S.6-7)

Auch der WFS bietet vordefinierte Schnittstellen für einen HTTP-Request, die den Zugriff auf Geodaten vereinfachen sollen:

- **GetCapabilities** (Leistungsverzeichnis)

Die Tabelle 3.5 zeigt die Parameter zur Abfrage der Fähigkeit eines WFS. Die Informationen der Antwort (als XML-Dokument) enthalten neben allgemeinen Angaben zum Anbieter auch die abfragbaren Feature Types und die möglichen Operationen. Beispiel einer Abfrage:

```
http://geoserv.weichand.de:8080/geoserver/wfs?SERVICE=WFS&REQUEST=GetCapabilities&VERSION=1.1.0
```

Parameter	Beschreibung	Pflicht
<b>SERVICE =</b>	Der Dienst, der Informationen liefert. In diesem Fall immer <b>WFS</b> .	Ja
<b>REQUEST =</b>	Zeige alles, was als Antwort geliefert wird. In diesem Fall immer <b>GetCapabilities</b> .	Ja
<b>VERSION =</b>	Antwort in einer bestimmten Version der Schnittstelle. (z. B. 1.0.0, <b>1.1.0</b> , 2.0.0)	Nein

Tabelle 3.5: WFS-Parameter für den Request GetCapabilities (vgl. [Gdi13], S.8; [Geo17b])

- **DescribeFeatureType** (Struktur)

Die Tabelle 3.6 stellt die Parameter zur Abfrage der Struktur einzelner Feature Types dar. Beispiel einer Abfrage:

```
http://geoserv.weichand.de:8080/geoserver/wfs?SERVICE=WFS&REQUEST=DescribeFeatureType&VERSION=1.1.0&TYPENAME=bvv:gmd_ex
```

Parameter	Beschreibung	Pflicht
<b>SERVICE =</b>	Der Dienst, der Informationen liefert. In diesem Fall immer <b>WFS</b> .	Ja
<b>REQUEST =</b>	Zeige alles, was als Antwort geliefert wird. In diesem Fall immer <b>DescribeFeatureType</b> .	Ja
<b>VERSION =</b>	Antwort in einer bestimmten Version der Schnittstelle. (z. B. 1.0.0, <b>1.1.0</b> , 2.0.0)	Ja
<b>TYPENAMES =</b>	Abfrage einer Liste (durch Komma getrennt) mit Namen der Feature Types für die Beschreibung.	Ja
<b>EXCEPTIONS =</b>	Ausgabeformat bei Fehlern. Standardmäßig: <b>application/vnd.ogc.se_xml</b>	Nein
<b>OUTPUT-FORMAT =</b>	Sachinformationen in diesem Format am Bildschirm darstellen. (z. B. <b>text/html</b> )	Nein

Tabelle 3.6: WFS-Parameter für den Request DescribeFeatureType (vgl. [Gdi13], S.8; [Geo17b])

- **GetFeature** (Sachinformationen)

In der Tabelle 3.7 werden die Parameter zur Abfrage der Daten (Feature Instanzen) mit der Möglichkeit einer näheren Spezifikation für die Eigenschaften des Features und der räumlichen Informationen dargestellt. Die Informationen des zurückgegebene-

nen XML-Dokuments kann dann durch die Anwendung weiterverarbeitet und die Ergebnisse dem Nutzer anschaulich präsentiert werden. Beispiel einer Abfrage:

```
http://geoserv.weichand.de:8080/geoserver/wfs?SERVICE=WFS&REQUEST=GetFeature&VERSION=1.1.0&TYPENAME=bvv:gmd_ex&SRSGNAME=EPSG:31468&BBOX=4450407.80325,5324432.56981,4481624.47895,5346224.56688,EPGS:31468
```

Parameter	Beschreibung	Pflicht
<b>SERVICE =</b>	Der Dienst, der Informationen liefert. In diesem Fall immer <b>WFS</b> .	Ja
<b>REQUEST =</b>	Zeige alles, was als Antwort geliefert wird. In diesem Fall immer <b>GetFeature</b> .	Ja
<b>VERSION =</b>	Antwort in einer bestimmten Version der Schnittstelle. (z. B. 1.0.0, <b>1.1.0</b> , 2.0.0)	Ja
<b>TYPENAMES =</b>	Abfrage einer Liste (durch Komma getrennt) mit Namen der Feature Types für die Beschreibung.	Ja
<b>STARTINDEX =</b>	Index, ab dem die Treffermenge übermittelt werden soll. Standardmäßig der Wert <b>1</b> .	Nein
<b>MAXFEATURES =</b>	Maximale Anzahl der zurückgegebenen Features.	Nein
<b>OUTPUT-FORMAT =</b>	Sachinformationen in diesem Format zurückliefern. Standardwert: <b>application/gml+xml; version=3.2</b>	Nein
<b>SRSGNAME =</b>	Wahl des Koordinatensystems bei Features mit Geometrie. (z. B. <b>urn:ogc:def:crs:EPSG:31468</b> )	Nein
<b>COUNT =</b>	Maximale Trefferanzahl, die zurückgeliefert wird.	Nein
<b>SORTBY =</b>	Angabe der Sortierung nach einen Attributwert („ASC“ für aufsteigend und „DESC“ für absteigend).	Nein
<b>PROPERTY-NAME =</b>	Attribut eines Feature Types, auf das ein Filter angewendet wird.	Nein
<sup>1</sup> <b>FILTER =</b>	Angabe eines Filterausdrucks nach dem Filter Encoding Standard.	Nein
<sup>1</sup> <b>FEATUREID =</b>	Indikator eines bestimmten Features, welcher abgefragt werden soll.	Nein
<sup>1</sup> <b>BBOX =</b>	Nur Features, die innerhalb des angegebenen Rechteck (BoundingBox) liegen, werden abgefragt.	Nein

Tabelle 3.7: WFS-Parameter für den Request GetFeature (vgl. [Gdi13], S.8-9; [Geo17b])

- **Transaction** (Transaktionen)

Transaktionsoperation zum Erstellen, Verändern und Löschen von Features. Jede Transaktion besteht aus mehreren Operationen, die in einer vorbestimmten Reihenfolge ausgeführt werden. Transaktionen sind atomar, wenn ein Element ausfällt, dann wird die Transaktion abgebrochen und die Daten werden nicht verändert. Server die Transaktionen unterstützen, werden als WFS-T Server bezeichnet. Alle Parameter für die Nutzung sind in der Tabelle 3.8 dargestellt. Beispiel einer Abfrage:

---

<sup>1</sup> Nur einer der Parameter „FILTER“, „FEATUREID“ oder „BBOX“ kann gleichzeitig angegeben werden.

### 3. Grundlagen eines Web-GIS

```
http://geoserv.weichand.de:8080/geoserver/wfs?SERVICE=WFS&REQUEST=Transaction&VERSION=1.1.0&OPERATOR=DELETE&TYPENAME=bvv:gmd_ex
```

Parameter	Beschreibung	Pflicht
<b>SERVICE =</b>	Der Dienst, der Informationen liefert. In diesem Fall immer <b>WFS</b> .	Ja
<b>REQUEST =</b>	Zeige alles, was als Antwort geliefert wird. In diesem Fall immer <b>Transaction</b> .	Ja
<b>VERSION =</b>	Antwort in einer bestimmten Version der Schnittstelle. (z. B. 1.0.0, <b>1.1.0</b> , 2.0.0)	Ja
<b>OPERATION =</b>	Transaktionsoperation zum ausführen. Zum Beispiel <b>delete</b> , <b>update</b> , <b>insert</b> .	Ja
<b>TYPENAME =</b>	Liste (durch Komma getrennt) mit Namen der Feature Types für die Operation. (Optional, wenn FEATUREID genutzt wird.)	Ja
<b>RELEASEACTION =</b>	Rücksetzen von Sperren bei Features. Werte können hierbei <b>ALL</b> oder <b>SOME</b> sein.	Nein
<b>FILTER =</b>	Angabe eines Filterausdrucks nach dem Filter Encoding Standard.	Nein
<b>FEATUREID =</b>	Indikator eines bestimmten Features für die Operation.	Nein
<b>BBOX =</b>	Nur Features, die innerhalb des angegebenen Rechteck (BoundingBox) liegen, werden für die Operation genutzt.	Nein

Tabelle 3.8: WFS-Parameter für den Request Transaction (vgl. [Vre02], S.71)

- **LockFeature** (Sperrfunktion)

Ein LockFeature bietet einen Sperrmechanismus, um die Konsistenz bei Bearbeitungsvorgängen zu gewährleisten. Wenn ein WFS diesen Vorgang unterstützt, wird dieser in den GetCapabilities angegeben. In der Tabelle 3.9 werden alle Parameter dargestellt. Beispiel einer Abfrage:

```
http://geoserv.weichand.de:8080/geoserver/wfs?SERVICE=WFS&REQUEST=LockFeature&VERSION=1.1.0&TYPENAME=bvv:gmd_ex
```

Parameter	Beschreibung	Pflicht
<b>SERVICE =</b>	Der Dienst, der Informationen liefert. In diesem Fall immer <b>WFS</b> .	Ja
<b>REQUEST =</b>	Der Name des Request <b>LockFeature</b> .	Ja
<b>VERSION =</b>	Antwort in einer bestimmten Version der Schnittstelle. (z. B. 1.0.0, <b>1.1.0</b> , 2.0.0)	Ja
<b>TYPENAME =</b>	Liste (durch Komma getrennt) mit Namen der Feature Types die gesperrt werden sollen. (Optional, wenn FEATUREID genutzt wird.)	Ja
<b>EXPIRY =</b>	Maximaldauer der Sperrung in Minuten.	Nein
<b>LOCKACTION =</b>	Sperren für alle oder ausgewählte Features. Werte können hierbei <b>ALL</b> oder <b>SOME</b> sein.	Nein
<b>FILTER =</b>	Angabe eines Filterausdrucks nach dem Filter Encoding Standard.	Nein
<b>FEATUREID =</b>	Indikator eines bestimmten Features für die Sperrung.	Nein
<b>BBOX =</b>	Nur Features, die innerhalb des angegebenen Rechteck (BoundingBox) liegen, werden gesperrt.	Nein

Tabelle 3.9: WFS-Parameter für den Request LockFeature (vgl. [Vre02], S.69-70)

### ***3.7 Inspire-Richtlinie***

Die **Infrastructure for Spatial Information in Europe (INSPIRE)**, zu deutsch Geodateninfrastruktur in Europa, soll die Nutzung von Geodaten in Europa erleichtern. Diese Inspire-Richtlinie, die ausschließlich für Europa dient, kann als Orientierung für Projekte in Marokko genutzt werden.

Verabschiedet wurde die Richtlinie durch das Europäische Parlament und trat am 15. Mai 2007 in Kraft. Diese definiert den rechtlichen Rahmen zum Aufbau von Geodateninfrastrukturen und ist für alle Mitgliedstaaten verbindlich. Jeder Mitgliedstaat hat dafür auf nationaler Ebene die dafür notwendigen Gesetze geschaffen. Praktisch beschreibt diese einheitlich die Geodaten und die Bereitstellung im Internet, mit den dazugehörigen Diensten wie Suche, Visualisierung und Download. Das Ziel ist es, relevante und hochwertige Geoinformationen auf europäischer Ebene für jeden verfügbar zu machen und damit die Barrieren für die Nutzung zu beseitigen. Dies nützt der Wissenschaft, der Verwaltung, der Wirtschaft und auch den Bürgerinnen und Bürgern. Besonders fordert dies der Wirtschaft für private Unternehmen, da die öffentlichen Geodaten einfacher nutzbar sind. Auch bei Investitionsentscheidungen bieten die aktuell Gehaltenden Geodaten Planungssicherheit. (vgl. [Pro01], Stichwort: INSPIRE; [Gdi17])

## 4 Analyse der Softwarekomponenten

Dieses Kapitel beschäftigt sich mit der Fragestellung, welche Softwarekomponenten für den Aufbau des Web-GIS geeignet sind. Dazu werden mehrere Softwarekomponenten anhand der Anforderungen analysiert, die für diesen Einsatz am besten geeignet sind.

### 4.1 Bewertungskriterien für Softwarekomponenten

Wie bereits im Kapitel 3.5.2 behandelt, besteht ein webbasiertes Geoinformationssystem aus den drei Hauptkomponenten. Für den Aufbau eines webbasierten Geodatenportals sollen nur FOSS-Produkte eingesetzt werden. Die Bewertungskriterien wurden bereits in Kapitel 1.5 festgelegt und sind in der Tabelle 4.1 unterteilt in die drei Hauptkomponenten.

Geodatenbanken	Kartenserver	Web-Mapping-Frameworks
Lizenzmodell		
Aufwand der Installation		
Unterstützte Betriebssysteme	Features	
Aufwand in der Konfiguration	Programmieraufwand	
Unterstützte Geometrien	Visualisierung	
Funktionsumfang	Konformität	
	Performanz	
	Dokumentation	
	Datenformate und Datenbanken	

Tabelle 4.1: Bewertungskriterien der Softwarekomponenten

### 4.2 Geodatenbanken

Die Grundlage, und somit die erste Komponente, für die Datenspeicherung von Geodaten ist die Datenbank. Einfache Datenbanken reichen für diesen Zweck nicht aus. Oft werden daher bereits bestehende Datenbanksysteme mit einer zusätzlichen Komponente für räumliche Daten erweitert. Die Daten werden in tabellarischer Form als Spalten und Zeilen oder Attribute und Werte abgelegt. Dabei können die Felder unterschiedliche Datentypen enthalten. Mehrere Tabellen können untereinander in Bezug gesetzt werden, dadurch sind die Daten flexibel modellierbar und es besteht zudem die Möglichkeit komplexe Suchanfragen zu stellen. Die Verknüpfungen erfolgen durch die in den Tabellen auftauchenden Informationen. Vorteile bietet die Datenbank in Hinblick auf Redundanzfreiheit (verhindern doppelter Informationen), Konsistenzprüfung (Korrektheit der gespeicherten Informationen) und Mehrbenutzerfähigkeit. (vgl. [Mit08], S.312)

#### 4.2.1 PostgreSQL mit der PostGIS-Erweiterung

PostgreSQL ist ein objektrelationales Datenbankmanagementsystem, bei dem die Daten über einen Datenbankserver bereitgestellt werden. Dieser Datenbankserver verarbeitet die Datenbankabfragen des Clients, stellt die Daten zusammen und übergibt diese an den Client zurück. Die PostGIS-Erweiterung stellt zahlreiche GIS-Funktionen direkt auf dem Server bereit und nutzt damit einer der Vorteile einer serverbasierten Datenbank. Neben dem Spei-

#### 4. Analyse der Softwarekomponenten

---

chern von räumlichen Daten besitzt PostGIS die Möglichkeiten der Datenbearbeitung direkt auf dem Server. (vgl. [Mit08], S.313-314)

**Lizenzmodell:** Lizenziert unter einer eigenen Lizenz (PostgreSQL License), welches der BSD- oder MIT-Lizenz ähnelt.

**Unterstützte Betriebssysteme:**

- Linux
- Mac OS
- Windows
- FreeBSD

**Aufwand der Installation:** Unter Linux wird die Paketverwaltung genutzt, um eine kompilierte Version zu installieren. Dadurch ist der Aufwand bei der Installation sehr gering.

**Aufwand der Konfiguration:** Die Konfiguration von PostgreSQL erfolgt über die Kommandozeilenebene oder durch grafische Werkzeuge wie „PGAdmin3“. Nach der Installation kann die Datenbank direkt genutzt werden, jedoch wäre es sinnvoll ein eigenes Passwort für den Standardbenutzer „postgres“ zu setzen, um den Zugriff einzuschränken. Durch simple Befehle oder durch die grafischen Werkzeuge können sehr einfach neue Benutzer und Datenbanken angelegt und „Extensions“ für räumliche Analysen installiert werden.

**Unterstützte Geometrien:** In der Tabelle 4.2 stehen die Geometrien der PostGIS-Extension zur Verfügung. Alle Geometrien basieren auf dem OGC-Standard „OpenGIS Simple Feature Specification For SQL“ von 1997. Die neuste Norm SQL/MM Spacial (ISO 13249-3) ergänzt den Standard durch weitere Geometrien, ist aber im Gegensatz zum Simple-Feature auf objektorientierte oder objektrelationale Datenbanken beschränkt. (vgl. [Koc09], S.2-3)

Geometrie	SQL Repräsentation
Punkt	`POINT(10 10)`
Linien	`LINESTRING(10 10, 20 20, 30 40)`
Polygone	`POLYGON((10 10, 10 20, 20 20, 20 15, 10 10))`
Mehrere Punkte	`MULTIPOINT(10 10, 20 20)`
Mehrere Linien	`MULTILINESTRING((10 10, 20 20), (15 15, 30 15))`
Mehrere Polygone	`MULTIPOLYGON(((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60)))`
Sammlung aus mehreren Geometriertypen	`GEOMETRYCOLLECTION(POINT(10 10),POINT(30 30), LINESTRING(15 15, 20 20))`

Tabelle 4.2: PostGIS Geometrien (vgl. [Dal05], S.10)

**Funktionsumfang:** Die PostGIS-Erweiterung enthält einen umfangreichen Satz an Funktionen für die Analyse und Bearbeitung der räumlichen Daten. Im Nachfolgenden werden nur die grundlegenden Operationen aufgelistet, die für das System relevant sind. Orientiert wird sich an die Norm „SQL/MM Spatial“. Um die Identifikation der räumlichen Funktionen und Operatoren zu vereinfachen, werden alle Funktionsbezeichnungen mit dem Präfix „ST\_“ (Spatial Type) erweitert. (vgl. [Mit08], S.330)

- Neue Geometrien erzeugen
- Räumliche Funktionen (Distanzberechnung, Flächenberechnung)
- Koordinaten Transformationen
- Analyse und Operationen für topologische Beziehungen zwischen Geometrien

### **4.2.2 MySQL mit der Spatial-Erweiterung**

Seit 1994 wurde MySQL von dem schwedischen Unternehmen MySQL AB entwickelt und ist eine sehr verbreitete Open-Source-Datenbank, die häufig bei Webanwendungen zusammen mit dem Webserver Apache und PHP eingesetzt wird. Im Jahr 2008 wurde MySQL von Sun Microsystems übernommen. Sun wurde dann im Jahre 2010 durch Oracle aufgekauft und gehört nun zu Oracle. (vgl. [Wik17a])

**Lizenzmodell:** Lizenziert wird unter die GPL (GNU General Public License). Damit erlaubt es vielfältige Nutzung für privat sowie kommerziell. Lizenzgebühren für die Nutzung der Software sind nicht gestattet. Jede Änderung an der Software führt dazu, dass die Software wieder unter die GPL vertrieben und auf Anfrage der Quellcode zur Verfügung gestellt werden muss.

#### **Unterstützte Betriebssysteme:**

- Linux
- Mac OS
- Windows
- FreeBSD

**Aufwand der Installation:** Die Installation erfolgt ähnlich wie bei PostgreSQL durch die Paketverwaltung. Die Spatial-Erweiterung ist nach der Installation direkt nutzbar. Der Aufwand bei der Installation ist dadurch sehr gering.

**Aufwand der Konfiguration:** Die Konfiguration erfolgt auch hier auf Kommandozeilenebene oder durch grafische Werkzeuge wie „phpMyAdmin“. Bearbeitet wird die Nutzer- und Rechteverwaltung unter Linux mit SQL-Befehlen.

**Unterstützte Geometrien:** Es werden alle Geometrietypen wie bei PostGIS unterstützt. Auch hier basieren alle Geometrietypen nach dem OGC-Standard „OpenGIS Simple Feature Specification For SQL“.

**Funktionsumfang:** Auch hier sind viele Funktionen, wie bei PostGIS, zu finden. Einige Funktionen, die in PostGIS existieren, sind in MySQL Spatial erst ab der Version 5.7.6 implementiert worden. Dazu gehört die sphärische Distanzberechnung und Flächenberechnung. Auch hier wird sich an die Norm „SQL/MM Spacial“ orientiert. (vgl. [Mys17], Stichwort: ST\_Distance\_Sphere, ST\_Area)

### **4.2.3 SQLite mit der SpatiaLite-Erweiterung**

Im Gegensatz zu den beiden bereits vorgestellten Datenbanken ist SQLite eine Programm-bibliothek, die ein relationales Datenbanksystem enthält. Dadurch, dass mit der Erweiterung auch Geodaten gespeichert und verarbeitet werden können, wird diese zum Zwecke der Analyse hier aufgelistet. Das Besondere an SQLite ist, dass die gespeicherte Datei, die die Datenbank enthält, nur wenige Kilobyte groß ist und auch sehr einfach im Dateisystem gespeichert werden kann. SQLite bietet keine vollständige Implementierung des SQL-Standards, für viele Anwendungsfälle sind die vorhandenen Funktionen aber ausreichend. (vgl. [Wik17b])

**Lizenzmodell:** Das Lizenzmodell ist PD (Public Domain) und kann damit uneingeschränkt genutzt werden.

#### **Unterstützte Betriebssysteme:**

- Linux
- Mac OS X
- Windows
- FreeBSD

**Aufwand der Installation:** Die Installation erfolgt ähnlich wie bei PostgreSQL und MySQL durch die Paketverwaltung. Dadurch ist der Aufwand bei der Installation sehr gering.

**Aufwand der Konfiguration:** Die Konfiguration erfolgt ebenfalls hier auf Kommandozeilenebene oder durch grafische Werkzeuge wie SQLite Manager oder SQLite Database Browser. Neue Datenbanken anlegen und verwalten erfolgt mit SQL-Befehlen.

**Unterstützte Geometrien:** Es werden alle Geometrietypen, wie bei PostGIS, unterstützt. Auch hier basieren alle Geometrietypen nach dem OGC-Standard „OpenGIS Simple Feature Specification For SQL“.

**Funktionsumfang:** Hier sind dieselben Funktionen wie bei PostGIS zu finden.

#### **4.2.4 Auswertung der Geodatenbanken**

Grundsätzlich sind alle drei Geodatenbanken für den Einsatz geeignet. Die Funktionen und Geometrien sind durch Normen und Standards geregelt, an dem sich die Datenbanken halten. Die einzigen Unterschiede liegen bei der Art der zu konfigurierenden Datenbank und dem Umfang. PostgreSQL und MySQL haben zudem ein Managementsystem zum Anlegen von Benutzern und Rollen, welches bei SQLite nicht vorhanden ist. SQLite hingegen ist eine Programmbibliothek und damit auch ohne zusätzlichen Server nutzbar.

**Lizenzmodell:** Hier unterscheiden sich die Lizenzmodelle. Während SQLite völlig uneingeschränkt nutzbar ist, steht MySQL unter der GPL und PostgreSQL unter einer eigenen Lizenz. Bei Betrachtung der Lizenzen eignet sich SQLite am meisten.

**Unterstützte Betriebssysteme:** Hinsichtlich der unterstützten Betriebssysteme ist kein Unterschied festzustellen. Alle genannten Geodatenbanken unterstützen eine Vielzahl von Betriebssystemen. Dadurch kann für die spätere Entscheidung dieses Kriterium vernachlässigt werden.

**Aufwand der Installation:** Alle erwähnten Geodatenbanken sind durch die interne Paketverwaltung von Linux sehr schnell installierbar. Dadurch zeigen sich auch hier keine wesentlichen Unterschiede.

**Aufwand der Konfiguration:** Alle beschriebenen Geodatenbanken unterstützen SQL und sind durch die Kommandozeile konfigurierbar. Auch viele grafische Programme zur Konfiguration der Datenbanken sind verfügbar. Unterschiede sind nur beim Thema Rechte- und Sicherheitseinstellungen erkennbar. Während SQLite kein Rechtesystem anbietet und nur lokal arbeitet, muss bei PostgreSQL und MySQL das Standardpasswort geändert werden, um den Zugriff durch Dritte verhindern zu können. Alle Befehle für die Konfiguration der Datenbank sind durch die Entwickler dokumentiert und können bei Unklarheiten genutzt werden.

**Unterstützte Geometrien:** Alle Geometrietypen der drei Geodatenbanken basieren nach dem OGC-Standard „OpenGIS Simple Feature Specification For SQL“. Dadurch sind keine Unterschiede bei den Geometrien festzustellen.

**Funktionsumfang:** Hierbei existieren nur geringe Unterschiede. Während die PostGIS-Erweiterung den größten Funktionsumfang an räumlichen Daten bietet, stellen die anderen Geodatenbanken nur geringe Abweichungen in der Komplexität der Funktionen.

Im Prinzip ist der Einsatz mit jeder der drei Geodatenbanken möglich. SQLite bietet kein Rollen- und Rechtekonzept und sollte daher in diesem Projekt nicht genutzt werden. MySQL verfügt im Umgang mit Geodaten seit kurzem viele neue Funktionalitäten, welche die PostGIS-Erweiterung bereits länger beinhaltet. Zudem ist PostGIS führend im Bereich GIS. Daher wäre die PostgreSQL mit der PostGIS-Erweiterung zu empfehlen. (vgl. [Wri16])

### 4.3 Kartenserver

Das Nächste ist die Entscheidung des Kartenservers. Der Kartenserver stellt die Daten über das Web bereit. Drei Kartenserver mit unterschiedlichen Merkmalen wurden hierfür ausgewählt. Alle Kartenserver sind OGC-Konform und werden anhand der Bewertungskriterien analysiert.

#### 4.3.1 GeoServer

Der GeoServer stellt Karten und Daten verschiedener Formate als Webservice zur Verfügung. Entwickelt wird das Projekt durch eine Vielzahl von Einzelpersonen und Organisationen als Open-Source-Software auf der Plattform GitHub mit OGC-Konformen Standards. Mittlerweile wird GeoServer unter der OSGeo weiterentwickelt. (vgl. [Osg17a]; [Git17a])

**Lizenzmodell:** Lizenziert wird unter GPL (GNU General Public License).

**Betriebssysteme:** Das Projekt basiert auf Java und ist damit unabhängig von den Plattformen. Der GeoServer unterstützt alle möglichen Plattformen (Windows, Linux, Mac), auf denen auch Java installiert werden kann.

**Aufwand der Installation:** Der GeoServer benötigt mind. Java 6, um auf einer Plattform laufen zu können. Dabei wird lediglich das Java Runtime Environment (JRE) benötigt. Java lässt sich auf vielen Betriebssystemen einfach installieren. Es gibt zwei Arten, wie der GeoServer in Betrieb genommen werden kann. Die erste Variante erfolgt durch das Starten eines Skriptes. Dadurch wird ein Jetty-Container aufgebaut, in denen der GeoServer so lange läuft, bis das Skript beendet wird. Die zweite Variante ist die Installation von einem Java-Server (Tomcat), in dem das WAR-Archiv abgelegt und automatisch entpackt wird. Das WAR steht für „Web Application Archive“ und ist ein Dateiformat, welches die Webanwendung nach der Java-Servlet-Spezifikation beschreibt. Unter Linux wird die interne Paketverwaltung für Tom-

## 4. Analyse der Softwarekomponenten

cat genutzt. Der GeoServer wird als WAR-Archiv heruntergeladen und in das entsprechende Verzeichnis von Tomcat verschoben.

**Visualisierung:** Die Visualisierung der Geodaten erfolgt in der grafischen Benutzeroberfläche durch Styles. Hierfür können bereits vorhandene Styles genutzt oder eigene durch den Style-Editor erstellt werden. Der im Editor genutzte Styled Layer Descriptor (SLD) beschreibt dabei die Symbolik nach Definition der OGC.

**Aufwand der Konfiguration:** Der GeoServer besitzt eine browserbasierte grafische Benutzeroberfläche (siehe Abb. 4.1). Darüber wird dieser Kartenserver vollständig konfiguriert und verwaltet. Der Vorteil besteht darin, dass alle Konfigurationen über vordefinierte Eingabemasken vorgenommen werden können. Dadurch ist die Konfiguration sehr einfach. Jeder Layer muss durch die Oberfläche publiziert werden. Viele Seiten bieten ebenso eine Hilfestellung, welche die Funktionalitäten erklären.

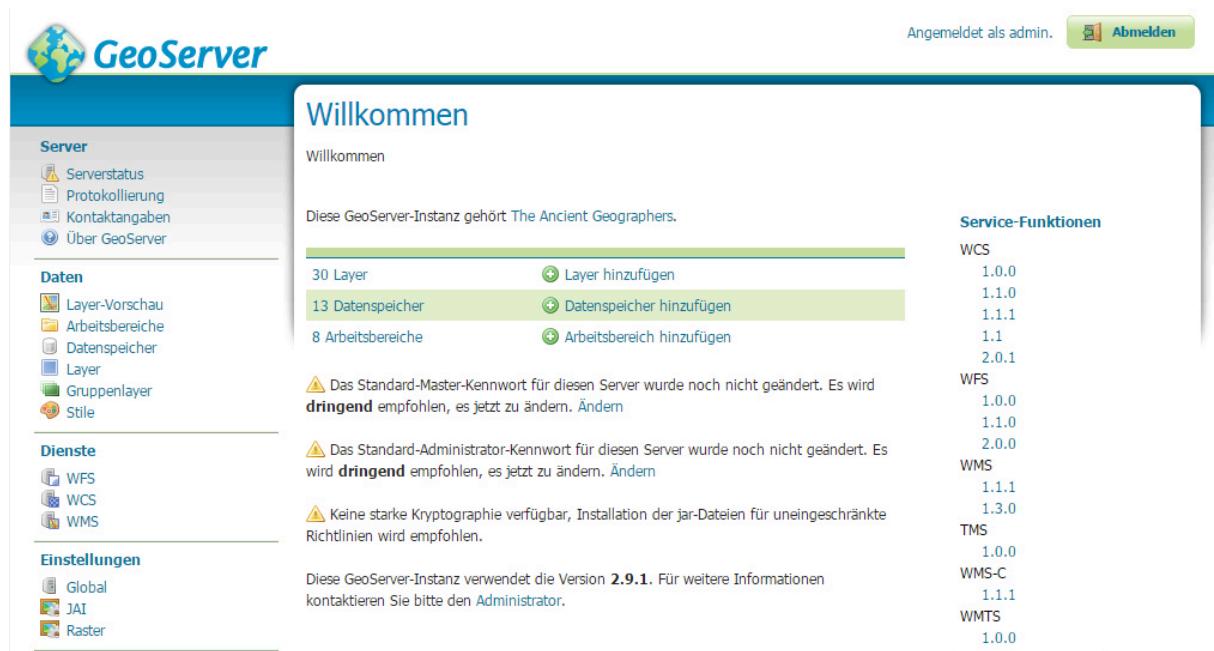


Abb. 4.1: Grafische Benutzeroberfläche von GeoServer

**Konformität (Unterstützte Normen und Standards):** GeoServer ist OGC-Konform und gewährleistet eine umfassende Unterstützung von Normen und Standards im Geoinformati onsraum. Folgende OGC-Standards bzw. ISO-Normen werden mit GeoServer unterstützt:

- WMS (Web Map Service)
- WFS / WFS-T (Web Feature Service / Web Feature Service - Transaction)
- WCS (Web Coverage Service)
- WPS (Web Processing Service)
- GML (Geography Markup Language)

## 4. Analyse der Softwarekomponenten

---

- KML (Keyhole Markup Language)
- SLD (Style Layer Descriptor)
- FE (Filter Encoding)

**Performanz:** GeoServer ist ein dauerhaft laufender Prozess. Der Vorteil besteht darin, dass die Verbindung zur Datenbank permanent steht. (vgl. [Sch12], S.45)

**Dokumentation:** Alle Dokumentationen sind online verfügbar und sehr umfangreich. Es ist nur die englische Sprache vorhanden.

**Datenformate und Datenbanken:** Viele Raster- und Vektorformate werden mit GeoServer unterstützt. Auch können Datenbanken als Bezugsquelle für die Daten angegeben werden. Zu den unterstützten Datenbanken und Formaten zählen:

- Datenbankanbindungen: PostgreSQL/PostGIS, MySQL, Oracle, Microsoft SQL Server, DB2
- Vektorformate: Shapefiles, GML
- Rasterformate: ArcGrid, GeoTIFF, Gtopo30, ImageMosaic, WorldImage

### 4.3.2 MapServer / TinyOWS

Der zweite Kartenserver, der analysiert wird, ist MapServer. In der Mitte der 1990er wurde MapServer an der University of Minnesota unter den Namen UMN-MapServer entwickelt. Wie GeoServer wird auch MapServer von Entwicklern auf der ganzen Welt gepflegt und gehört ebenfalls zur OSGeo. (vgl. [Wik17c])

**Lizenzmodell:** Die Software steht unter der freien MIT-Lizenz (Massachusetts Institute of Technology). Lediglich der Urheberrechtsvermerk muss enthalten bleiben und für Fehler der Software wird nicht gehaftet.

**Betriebssysteme:** Plattformübergreifend und kann auf einer Vielzahl von Betriebssystemen installiert werden:

- Linux
- Mac OS X
- Windows
- FreeBSD

**Aufwand der Installation:** Die Installation erfolgt als CGI-Anwendung im Webserver. Unter Linux kann die interne Paketverwaltung für die Einrichtung genutzt werden. Der Nachteil be-

#### 4. Analyse der Softwarekomponenten

steht darin, dass nicht die neuste Version dadurch installiert wird. Um immer die aktuellste Version zu nutzen, muss MapServer direkt auf dem Betriebssystem kompiliert werden.

**Visualisierung:** Alle Einstellungen zur Visualisierung der Geodaten erfolgen durch die Konfigurationsdatei. Es gibt keine vorgegebene Symbolik für Punkte, hingegen bieten Linien und Polygone einfache schwarze Umrisse zur Darstellung. Jedes Symbol muss in der Konfigurationsdatei definiert und im Layer darauf verwiesen werden. Auch die Angabe einer SLD-Datei zur Beschreibung der Darstellung ist möglich.

**Aufwand der Konfiguration:** Im Gegensatz zu GeoServer verfügt MapServer über keine grafische Benutzeroberfläche. Die komplette Konfiguration erfolgt mithilfe von Mapfiles. Da für die komplette Konfiguration lediglich Mapfiles verwendet werden, kann der Kartenserver zudem einfach mit Automatismen konfiguriert werden. Der Aufbau eines Mapfiles orientiert sich an XML. In der nachfolgenden Abb. 4.2 ist ein kleines Mapfile dargestellt, welches den Aufbau verdeutlichen soll:

```
MAP
  SIZE 800 600
  EXTENT -180 -90 180 90
  WEB
    METADATA
      OWS_TITLE "My Layers"
      OWS_SRS "EPSG:4326 EPSG:31467"
      OWS_ENABLE_REQUEST "*"
    END
  END
  SYMBOL
    NAME "CIRCLE"
    TYPE ellipse
    POINTS
      1 1
    END
  END
  LAYER
    NAME "mylayer"
    TYPE POINT
    STATUS ON
    DATA "geom from layer using unique gid"
    CLASS
      NAME "locations"
      STYLE
        SYMBOL "CIRCLE"
        COLOR 255 0 0
      END
    END
  END
END
```

Das Diagramm zeigt den Strukturbaum eines Mapfiles. Der gesamte Block ist als 'MAP' markiert. Unter 'WEB' befindet sich ein blauer Block mit 'METADATA'-Einstellungen. Der 'SYMBOL'-Block ist orange und enthält eine 'NAME'-Einstellung 'CIRCLE' und einen 'POINTS'-Abschnitt mit der Werte '1 1'. Der 'LAYER'-Block ist gelb und enthält eine 'NAME'-Einstellung 'mylayer', eine 'TYPE'-Einstellung 'POINT', eine 'STATUS'-Einstellung 'ON' und eine 'DATA'-Einstellung 'geom from layer using unique gid'. Der 'CLASS'-Block ist pink und enthält eine 'NAME'-Einstellung 'locations'. Unter 'CLASS' befindet sich ein grüner Block mit 'STYLE'-Einstellungen, die auf 'SYMBOL "CIRCLE"' und 'COLOR 255 0 0' zeigen. Am Ende des Mapfiles steht ein weiterer 'END'.

Abb. 4.2: Aufbau eines Mapfiles

#### 4. Analyse der Softwarekomponenten

---

Der Code für die Konfiguration ist in Blöcken eingeteilt. Der erste Block wird immer mit „MAP“ eingeleitet und am Ende mit „END“ beendet. Hier erfolgen dann Angaben zum Mapfile wie Größe und Ausdehnung der Karte. Die nächsten Blöcke sind im MAP-Block verschachtelt. Der WEB-Block enthält Informationen zum Speicherort der generierten Karten sowie einige Metadaten über das Mapfile. Die Metadaten können durch Angabe eines Präfixes „WMS\_“ speziell für WMS-Dienste oder „WFS\_“ für WFS-Dienste angepasst werden. Ein Präfix „OWS\_“ fasst WMS und WFS zu einem zusammen. Im SYMBOL-Block können eigene Symbole für die Karte definiert werden. Die LAYER-Blöcke beinhalten alle Informationen zum Layer. Hierbei folgen Angaben wie Name, Geometriertyp, der Status und die SQL-Anweisung für die Geometriespalte. In einem Mapfile können beliebig viele LAYER-Blöcke enthalten sein. Jeder LAYER-Block kann dabei wiederum mehrere CLASS-Blöcke umfassen. Die CLASS-Blöcke enthalten den STYLE-Block, der in den Features das Design verändert. Durch Bedingungen in CLASS-Blöcken sind zudem auch verschiedene Designs für Features möglich. Für die einzelnen Blöcke existieren noch jede Menge weitere Parameter, die je nach Einsatzzweck genutzt werden können.

**Konformität (Unterstützte Normen und Standards):** MapServer unterstützt die wichtigsten Normen und Standards des Geoinformationswesens. Zu den verfügbaren OGC-Standards und ISO-Normen zählen:

- WMS (Web Map Service)
- WFS (Web Feature Service)
- WCS (Web Coverage Service)
- GML (Geography Markup Language)
- KML (Keyhole Markup Language)
- SLD (Style Layer Descriptor)
- FE (Filter Encoding)

Im Gegensatz zu GeoServer unterstützt der MapServer kein WFS-T. Mithilfe von TinyOWS wird diese Funktion für den MapServer ergänzt. TinyOWS wird dabei ähnlich, wie MapServer als CGI-Anwendung installiert.

**Performanz:** Im Kontrast zum GeoServer ist der MapServer kein dauerhaft laufender Prozess. Das bedeutet, dass der MapServer bei jedem Kartenaufruf erneut gestartet wird. Dieser Vorgang wirkt sich negativ auf die Performanz aus. Das verdeutlicht sich besonders bei dem Zugriff auf die Datenbank, da immer eine neue Verbindung mit der Datenbank hergestellt werden muss. Abhilfe schafft die Optimierung der CGI-Schnittstelle mit FastCGI. Mithilfe von FastCGI kann die Antwortzeit halbiert und dadurch die Performanz verdoppelt werden. (vgl. [Sch12], S.45-46)

**Dokumentation:** Alle Dokumentationen sind online in englischer Sprache verfügbar und sehr umfangreich.

**Datenformate und Datenbanken:** MapServer unterstützt viele Raster- und Vektorformate. Außerdem wird eine Verbindung zu diversen Datenbanken geboten. Zu den genannten Datenbankverbindungen und Formaten zählen:

- Datenbankanbindungen: PostgreSQL/PostGIS, MySQL, SpatiaLite, Oracle Spatial, ESRI
- Vektorformate: Shapefiles, GML, KML, GPX
- Rasterformate: GeoTIFF, BMP, GIF, JPEG

### 4.3.3 deegree

Das Java-Framework deegree ist eine freie Software Initiative für räumliche Dateninfrastrukturen und bietet Werkzeuge für das Geodatenmanagement, den Datenzugriff, die Visualisierung und Sicherheitsmechanismen. Das Projekt implementiert die Standards des OGC und hält sich an die Normen der ISO. Entwickelt wurde es von der Firma lat/lon und das Geografische Institut der Universität Bonn. Seit 2010 ist es ein OSGeo-Projekt. (vgl. [Osg17b]; [Wup17])

**Lizenzmodell:** Lizenziert wird es unter LGPL (GNU Lesser General Public License). Das ist eine entschärzte Version der GPL und verhindert damit, dass die ganze Software bei Veränderungen unter dieser Lizenz herausgegeben werden muss.

**Betriebssysteme:** Deegree basiert auf Java und ist damit unabhängig von den Plattformen. Dadurch werden alle möglichen Plattformen (Windows, Linux, Mac) unterstützt, auf denen auch Java installiert werden kann.

**Aufwand der Installation:** Deegree benötigt mind. Java 6, um auf einer Plattform laufen zu können. Dabei wird das Java Runtime Environment (JRE) und das Java Development Kit (JDK) benötigt. Java lässt sich auf vielen Betriebssystemen einfach installieren. Es gibt zwei Arten, wie deegree in Betrieb genommen werden kann. Die erste Variante erfolgt durch das Starten eines Skriptes. Dadurch wird ein Jetty-Container aufgebaut, in den deegree so lange läuft, bis das Skript beendet wird. Die zweite Variante ist die Installation von einem Java-Server (Tomcat), in dem das WAR-Archiv abgelegt und automatisch entpackt wird. Unter Linux wird die interne Paketverwaltung für Tomcat genutzt. Deegree wird als WAR-Archiv heruntergeladen und in das entsprechende Verzeichnis von Tomcat verschoben.

**Visualisierung:** Deegree nutzt SLD zur Einfärbung von geografischen Objekten.

**Aufwand der Konfiguration:** Deegree besitzt eine browserbasierte, grafische Benutzeroberfläche (siehe Abb. 4.3). Darüber wird dieser Kartenserver vollständig konfiguriert und verwaltet. Einige Seiten verfügen über vordefinierte Eingabemasken, bei denen nur die Parameter gesetzt werden müssen. Andere Seiten bieten lediglich einen XML-Editor an, bei dem der XML-Code manuell geändert werden soll. Dieser XML-Code ist zwar lesbar, dennoch sind für die Bearbeitung weitreichende Kenntnisse erforderlich. Alle diese Konfigurationsdateien sind im Ordner „webapps/ROOT/WEB-INF/workspace“ als XML-Format abgelegt.

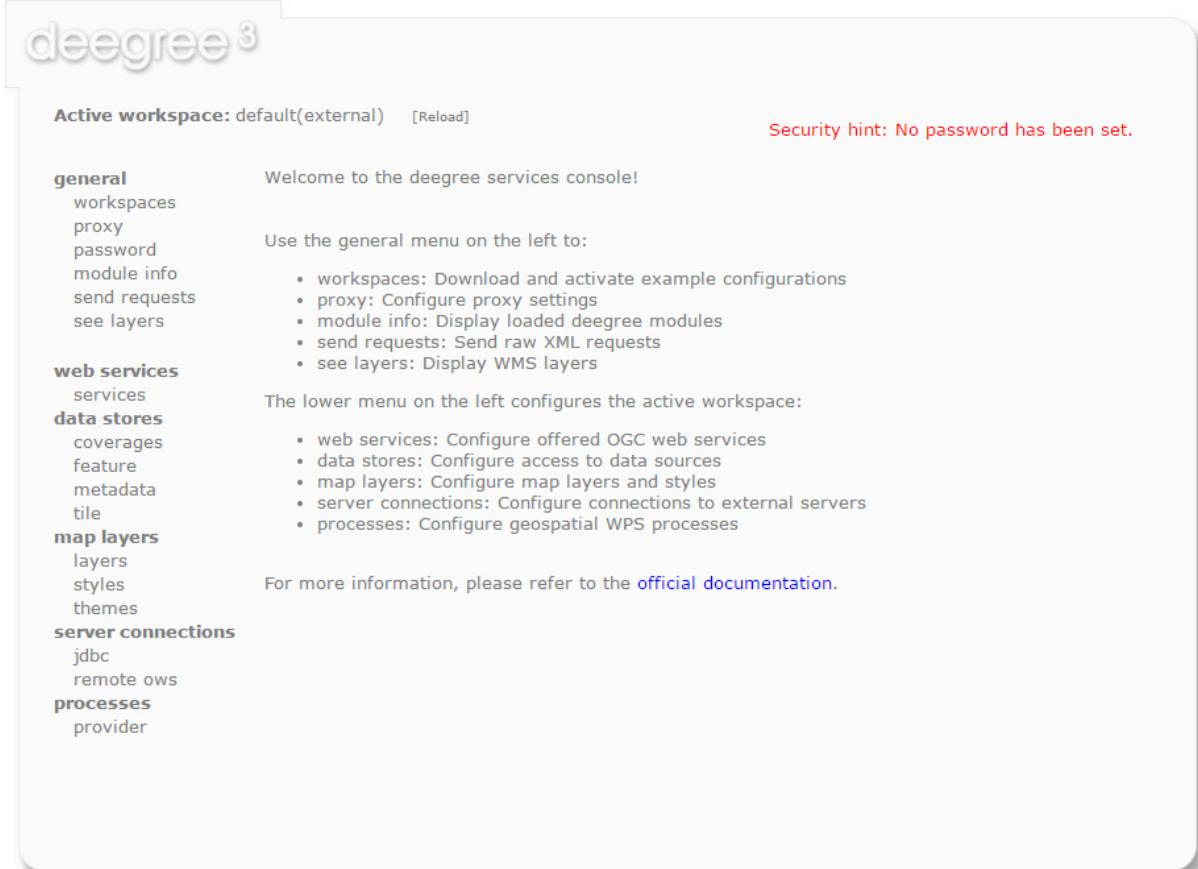


Abb. 4.3: Grafische Benutzeroberfläche von deegree

**Konformität (Unterstützte Normen und Standards):** Deegree ist OGC-Konform und gewährleistet eine umfassende Unterstützung von Normen und Standards. Folgende OGC-Standards bzw. ISO-Normen zählen dazu:

- WMS (Web Map Service)
- WMTS (Web Map Tile Service)
- WFS / WFS-T (Web Feature Service / Web Feature Service - Transaction)
- CSW (Catalogue Service-Web)
- WPS (Web Processing Service)
- GML (Geography Markup Language)
- SLD (Style Layer Descriptor)

#### 4. Analyse der Softwarekomponenten

---

- FE (Filter Encoding)

**Performanz:** Deegree ist ein dauerhaft laufender Prozess, ähnlich wie der GeoServer. Das bringt den Vorteil, dass die Verbindung zur Datenbank permanent steht.

**Dokumentation:** Alle Dokumentationen sind online verfügbar und können heruntergeladen werden. Zudem sind diese sehr umfangreich. Es ist nur die englische Sprache vorhanden.

**Datenformate und Datenbanken:** Viele Raster- und Vektorformate werden mit deegree unterstützt. Auch können Datenbanken als Bezugsquelle für die Daten angegeben werden. Zu den kompatiblen Datenbanken und Formaten zählen:

- Datenbankanbindungen: PostgreSQL/PostGIS, Oracle Spatial, Microsoft SQL Server
- Vektorformate: Shapefiles, GML
- Rasterformate: GeoTIFF, BMP, GIF, JPEG, PNG

##### **4.3.4 Auswertung der Kartenserver**

Im Prinzip kann mit allen drei Kartenservern ein Web-GIS aufgebaut werden. Die gravierendsten Unterschiede zwischen den Kartenservern zeigen sich beim Aufwand der Konfiguration.

**Lizenzmodell:** Hier unterscheiden sich die Lizenzmodelle. GeoServer und deegree stehen unter der GPL bzw. der LGPL und MapServer unter der MIT-Lizenz. MapServer ist damit die beste Wahl, um es uneingeschränkt nutzen zu können.

**Betriebssysteme:** Hinsichtlich der unterstützten Betriebssysteme ist kein Unterschied festzustellen. Speziell GeoServer und deegree laufen auf jedem Betriebssystem, auf denen Java läuft. Dadurch kann für die spätere Entscheidung dieses Kriterium vernachlässigt werden.

**Aufwand der Installation:** Für MapServer wird ein Webserver wie Apache2 benötigt, um mit der CGI-Schnittstelle zu interagieren. Durch die Paketverwaltung gestaltet sich die Installation von MapServer sehr viel einfacher als bei GeoServer und deegree. Bei GeoServer und deegree muss zuerst Java auf dem System installiert und die Software in einem Servlet-Container gestartet werden. Um den zusätzlichen Ballast durch die Installation von Java zu sparen, sollte daher MapServer gewählt werden.

**Visualisierung:** Alle Kartenserver unterstützen den nach OGC definierten Styled Layer Descriptor (SLD). Der GeoServer bietet zudem noch vordefinierte Symbole an, die direkt genutzt werden können. Der MapServer bietet auch den SLD und darüberhinaus noch eine

#### 4. Analyse der Softwarekomponenten

---

eigene Lösung zur Darstellung der Geodaten mit den Mapfiles an. Für die Visualisierung der Geodaten eignen sich damit alle Kartenserver. Dieses Kriterium kann daher bei der Wahl der Kartenserver vernachlässigt werden.

**Aufwand der Konfiguration:** Beide Kartenserver GeoServer und deegree haben eine grafische Benutzeroberfläche, die es dem Nutzer ermöglicht, den Kartenserver zu konfigurieren. Bei der Wahl einer dieser Kartenserver muss sich auch in die grafische Benutzeroberfläche eingearbeitet werden. Das erhöht geringfügig den Aufwand bei der Einrichtung. Der MapServer hingegen wird komplett mit kleinen strukturierten Textdateien konfiguriert und ermöglicht dadurch eine automatisierte Konfiguration durch Skripte. Der Aufwand für die Erstellung der Skripte wäre zunächst höher, aber kann bei der späteren Einrichtung durch automatisierte Skripte deutlich niedriger sein als bei den anderen Kartenservern. Daher wäre hier der Einsatz von MapServer sinnvoller.

**Konformität (Unterstützte Normen und Standards):** Alle Kartenserver unterstützen die wichtigsten OGC-Standards WMS und WFS. Nur MapServer unterstützt kein WFS-T für transaktionale Operationen. WFS-T erlaubt das bearbeiten und verändern von Daten am Server. Mit der zusätzlichen Komponente TinyOWS kann MapServer um diese Funktion erweitert werden. Dadurch sind alle Kartenserver für die Auswahl geeignet.

**Performanz:** GeoServer und deegree sind ein dauerhaft laufender Prozess und sind damit permanent mit der Datenbank verbunden. Nur MapServer wird bei jedem Kartenauftruf neu gestartet, um eine Verbindung zur Datenbank herzustellen. Dieser Prozess verlangsamt MapServer zwar, durch FastCGI kann aber die Antwortzeit halbiert und dadurch die Performanz verdoppelt werden. Hinsichtlich der Performanz sind damit alle Kartenserver gleich gut.

**Dokumentation:** Alle drei Kartenserver bieten eine umfangreiche Dokumentation. Voraussetzung für alle Kartenserver sind Englischkenntnisse, um sich in der jeweiligen Dokumentation durchzuarbeiten.

**Datenformate und Datenbanken:** Hier zeigen sich nur geringe Unterschiede hinsichtlich der Datenformate zwischen den drei Kartenservern. Jedoch bieten alle Kartenserver eine Verbindung zur Datenbank an.

Am Besten würde sich der Kartenserver MapServer mit TinyOWS für die Unterstützung von WFS-T eignen. Der MapServer benötigt weniger Ressourcen und ist im Gegensatz zu Geo-

Server und deegree ohne Java-Installation lauffähig. Zudem ist MapServer mit kleinen Textdateien und Automatismen einfacher zu konfigurieren, wodurch der Aufwand deutlich geringer ist. Die Performanz von MapServer mit der CGI-Schnittstelle ist durch das ständige neu verbinden mit der Datenbank zu langsam, daher muss die FastCGI-Schnittstelle für eine permanente Verbindung eingesetzt werden. Auch berücksichtigt werden muss, dass TinyOWS nur die PostgreSQL-Datenbank unterstützt.

### **4.4 Web-Mapping-Anwendungen und Frameworks**

Die letzte Komponente eines Web-GIS ist die Web-Mapping-Anwendung. Diese Komponente ist die direkte Verbindung zwischen Interaktion im Web und das Benutzen der Geodaten mit dem Nutzer. Es existieren jede Menge Frameworks und Anwendungen, um eine Webanwendung aufzubauen. Dabei wird zwischen Lösungen mit hohem und geringen Programmieraufwand unterschieden. Die JavaScript-Bibliotheken bilden die Grundlage der Web-Mapping-Frameworks. Die nachfolgende Analyse bezieht sich auf freie Frameworks und Anwendungen zum Erstellen von Web-Mapping-Anwendungen.

#### **4.4.1 OpenLayers**

OpenLayers ist eine JavaScript-Bibliothek, die es ermöglicht unkompliziert dynamische Karten auf der eigenen Webseite einzubinden. Dazu stellt es typische Elemente für Navigation, Maßstab, Koordinaten, Messinstrumente sowie Marker auf der Karte bereit. Die Software entstand als kostenlose Alternative zu Google Maps. (vgl. [Bau11])

**Lizenzmodell:** OpenLayers steht unter der BSD-Lizenz und kann damit frei verwendet werden, solange der Copyrightvermerk enthalten ist.

**Aufwand der Installation:** Der Aufwand für die Installation ist sehr gering. Es existieren zwei Möglichkeiten und Versionen für die Nutzung. Die erste Möglichkeit ist die Einbindung der Bibliothek über das Internet. Die andere Möglichkeit besteht durch das Herunterladen und Ablegen auf den eigenen Webserver. Die zwei Versionen bestehen aus komprimierter und nicht-komprimierter JavaScript-Bibliothek. Bevorzugt wird das Ablegen auf den eigenen Webserver, um Abhängigkeiten von anderen Webseiten zu unterbinden. Außerdem sollte die komprimierte JavaScript-Bibliothek bevorzugt werden, da diese schneller vom Client geladen werden kann.

**Features:** Die wichtigsten Features von OpenLayers werden im Folgenden aufgelistet.

- Elemente für Navigation (Auswahl der Layer, Zoomen, Verschieben)
- Elemente für Analyse (Zeichen-, Messwerkzeuge, Objektabfragen)
- Einbindung von Geodaten GeoJSON, GML

- Zugriff auf Geodaten mit WMS, WFS und WTS-T
- Auswahl von Koordinatenreferenzsystem mit On-the-fly-Koordinatentransformation

**Programmieraufwand:** Mit dem Framework kann jede Art von Problemstellung umgesetzt werden, sofern JavaScript-Kenntnisse für die Programmierung vorhanden sind. Die dafür bereitgestellte API ist sehr umfangreich und auf der Webpräsenz von OpenLayers existieren viele Anwendungsbeispiele. Für einfache Anwendungen hält sich die Einarbeitungszeit in Grenzen. Wird aber eine größere und komplexe Anwendung benötigt, so steigt die Einarbeitungszeit und Komplexität der Programmierung. Hierfür sollte im Vorhinein eine Struktur des Programmcodes vorgenommen werden.

### 4.4.2 Leaflet

Leaflet ist eine JavaScript-Bibliothek, die es ermöglicht browserbasierte, mobil-freundliche und interaktive Karten zu erstellen. Der Vorteil von Leaflet ist der geringe Speicherverbrauch der Bibliothek mit umfangreichen Features. Weitergehende Funktionen werden durch Plugins hinzugefügt. (vgl. [Osg17c])

**Lizenzmodell:** Leaflet steht unter der BSD-Lizenz.

**Aufwand der Installation:** Die Installation ist ähnlich wie bei OpenLayers. Die JavaScript-Bibliothek ist im Gegensatz zu Openlayers nur um die 145 KB groß (Version: 1.0.3).

**Features:** Leaflet hat ähnliche Features wie OpenLayers. Die einzigen Unterschiede sind, dass Leaflet noch kein WFS unterstützt und nur eingeschränkt Koordinatensystemumrechnungen durchgeführt werden können. Abhilfe schaffen die vielen Plug-ins von Drittanbietern. Mithilfe der Plug-ins können die Basisfunktionen um viele weitere Funktionen wie WFS-T erweitert werden.

**Programmieraufwand:** Der Programmieraufwand ist ähnlich hoch wie bei OpenLayers.

### 4.4.3 GeoExt

GeoExt setzt auf die JavaScript-Bibliotheken OpenLayers (für die Karten im Web) und ExtJS (für interaktive Webanwendungen). Der Vorteil der Anwendung ist, dass das Kartenmaterial sehr einfach in ansprechenden und komplexen Oberflächen erstellt werden kann. (vgl. [Geo17c])

**Lizenzmodell:** GeoExt steht unter einer Doppellizenz zwischen GPL und BSD. Beim privaten Gebrauch steht es unter GPL und bei Besitz einer kommerziellen Lizenz von Sencha ExtJS kann es unter der BSD-Lizenz verwendet werden. (vgl. [Git17b])

**Aufwand der Installation:** Die Installation ist ähnlich wie bei OpenLayers und Leaflet. Es müssen lediglich alle drei Bibliotheken (OpenLayers, ExtJS, GeoExt) auf der Webseite eingebunden werden, entweder durch einen Verweis auf der Webseite oder durch das Ablegen der Projekte auf dem Webserver. Zwingend ist die Einhaltung der richtigen Version zwischen den Bibliotheken. Bei der Nutzung von inkompatiblen Versionen zwischen den Bibliotheken können ansonsten Fehler auftreten.

**Features:** GeoExt besitzt die gleichen Features wie OpenLayers, da GeoExt auf diese JavaScript-Bibliothek aufbaut. Zusätzlich bietet GeoExt folgende Features:

- Gestaltung der Oberfläche und Einteilung von Elementen
- Ebenenbaum (Layertree)
- Drucken von Karten
- Formulare für die Bearbeitung von Geodaten

**Programmieraufwand:** Der Programmieraufwand ist ähnlich hoch bzw. noch höher als bei OpenLayers. Durch die Integration von zwei weiteren Bibliotheken, auf der diese aufbaut, steigt der Aufwand für die Programmierung. Die Webpräsenz enthält einige Beispiele, an die sich orientiert werden kann.

### **4.4.4 MapFish**

MapFish basiert auf dem Pylons Python Web-Framework und erweitert diese um räumliche Funktionalitäten zum Abfragen und Bearbeiten von räumlichen Objekten. Außerdem nutzt MapFish das GeoExt mit den OpenLayers und ExtJS JavaScript-Bibliotheken. Zum Umsetzen von Web-Mapping-Anwendungen bietet es eine flexible und komplett Umgebung. (vgl. [Map17]; [Bau11])

**Lizenzmodell:** MapFish unterliegt der BSD-Lizenz.

**Aufwand der Installation:** Für die Installation wird Python mit der richtigen Version benötigt. Außerdem müssen die Umgebungsvariablen angepasst werden. Hierfür bietet MapFish notwendige Skripte im Internet an.

**Features:** MapFish nutzt verschiedene OGC-Normen wie WMS, WFS, WMC, KML und GML. Da auch OpenLayers und GeoExt zum Einsatz kommen, unterstützt MapFish alle Features von den anderen Web-Mapping-Anwendungen.

**Programmieraufwand:** Der Programmieraufwand ist bei MapFish sehr gering, da mit wenigen Befehlen neue Anwendungen erstellt und diese durch das Ändern von Konfigurationsdateien konfiguriert werden können. Die Dokumentation ist nicht sehr umfangreich, wodurch eine erhöhte Einarbeitungszeit entsteht.

### **4.4.5 Mapbender**

Mapbender ist ein Content Management System für Kartenanwendungen und Geodatendienste. Es basiert auf PHP und JavaScript mit aktuellen Technologien, wie das Web-Application-Framework Symfony2, JQuery und OpenLayers. Diese Software bietet vorkonfigurierte Anwendungen zur Anzeige, Navigation und Abfragen von Karten. Über die Weboberfläche können alle Anwendungen sehr einfach und individuell angepasst werden. Zusätzlich können Rollenkonzepte dafür definiert werden. (vgl. [Bau11])

**Lizenzmodell:** Mapbender steht ab der Version 3.0.4.0 unter der MIT-Lizenz. Vorher wurde die Software unter GPL lizenziert.

**Aufwand der Installation:** Für die Installation wird ein Webserver und PHP benötigt. Anschließend muss der Webserver so konfiguriert werden, dass Mapbender als Webverzeichnis eingebunden wird. Zum Schluss sind die Rechte zu setzen und die Konfigurationsdatei anzupassen.

**Features:** In der Abb. 4.4 ist die grafische Benutzeroberfläche von Mapbender3 dargestellt. In dieser Oberfläche können verschiedene Anwendungen erstellt und für jede Anwendung Nutzer- und Gruppenrechte gesetzt werden. Zusätzlich können Anwendungen im YAML-Format<sup>2</sup> eingelesen werden, sodass diese nicht in der grafischen Oberfläche mühsam zusammen geklickt werden müssen. In der „Datenquelle“ werden die URLs zu WMS und WFS eingetragen. Als Datenquelle wird derzeit lediglich nur der WMS unterstützt. Im Menüpunkt „Benutzer“ können zusätzlich neue Nutzer angelegt und im Menüpunkt „Gruppen“ können die Nutzer in Gruppen eingeteilt werden. In den Anwendungen können diese Nutzer und Gruppen unterschiedliche Funktionen enthalten. Jede Funktion wird als Element dargestellt. 27 Elemente stehen derzeit in Mapbender3 zur Verfügung. Die wichtigsten Elemente sind nachfolgend dargestellt. (vgl. [Map17b])

---

<sup>2</sup> Das YAML-Format („Yet Another Markup Language“) dient der Datenserialisierung von Textdateien und ist zudem noch für Menschen leicht lesbar. Angelehnt wurde dieses Format an XML.

#### 4. Analyse der Softwarekomponenten

- Koordinatenanzeige
- Feature Info
- GPS-Position
- Ebenenbaum (Layertree)
- Legende
- Mapkarte
- Navigationselemente
- Übersichtskarte
- Längen und Flächen berechnen
- Maßstabsauswahl, Maßstabsleiste und Maßstabsanzeige
- Suchfunktion

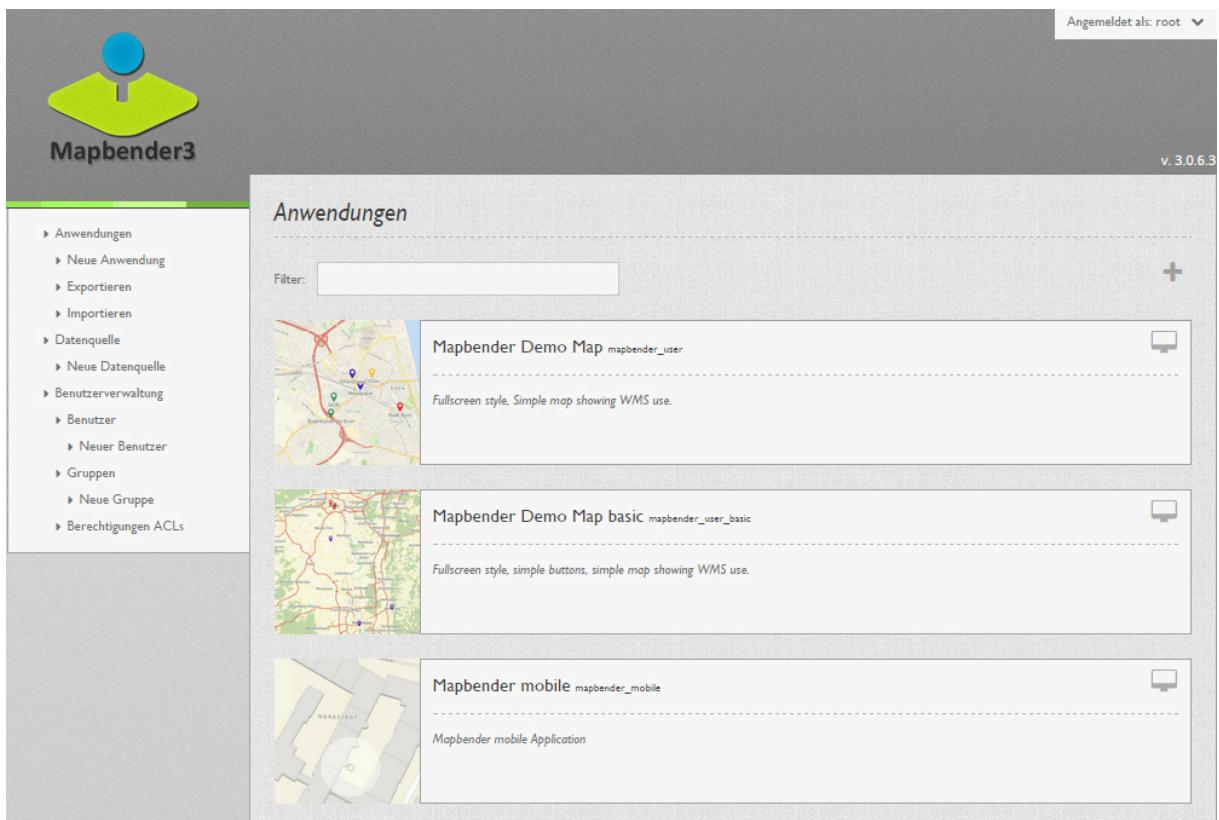


Abb. 4.4: Grafische Oberfläche von Mapbender3

**Programmieraufwand:** Die Konfiguration erfolgt komplett durch die grafische Oberfläche. Viele vorhandene Funktionalitäten können ganz einfach zu den Anwendungen hinzugefügt werden. Dadurch kann Mapbender3 auch ohne Programmierkenntnisse genutzt werden und es existiert somit kein Programmieraufwand.

#### **4.4.6 Auswertung der Web-Mapping-Anwendungen und Frameworks**

Mit allen Web-Mapping-Anwendungen und Frameworks können geeignete Oberflächen für das Web-GIS erstellt werden. Die größten Unterschiede zwischen Anwendungen und Frameworks liegen bei dem Aufwand der Programmierung und dem Einrichten der Anwendung.

**Lizenzmodell:** Die Lizenzmodelle unterscheiden sich nur geringfügig. OpenLayers, Leaflet und MapFish stehen unter der BSD-Lizenz und Mapbender steht seit der neusten Version 3.0.4.0 unter der MIT-Lizenz. Nur GeoExt steht unter einer Doppellizenz zwischen BSD und GPL. Die Lizenzen der Softwareanwendungen und Frameworks stellen damit kein Problem dar, sodass grundsätzlich jede Web-Mapping-Anwendung und jedes Framework genutzt werden kann.

**Aufwand der Installation:** Bei der Installation unterscheiden sich die Web-Mapping-Anwendungen und Frameworks. Während die Frameworks OpenLayers, Leaflet und GeoExt schnell installiert werden können, bedarf es bei den Anwendungen MapFish und Mapbender deutlich mehr Aufwand.

**Features:** Alle Web-Mapping-Anwendungen und Frameworks verfügen über eine Vielzahl von Features, die für das eigene Web-GIS vollkommen ausreichend sind. Fast alle verfügen über eine WMS- und WFS-Schnittstelle. Nur bei Mapbender fehlt derzeit die WFS-Schnittstelle.

**Programmieraufwand:** Hier existieren die größten Unterschiede zwischen Web-Mapping-Anwendungen und Frameworks. Die Frameworks OpenLayers, Leaflet und GeoExt sind JavaScript-Bibliotheken, bei denen die Anwendungen selbst zusammengestellt und programmiert werden müssen. Hingegen bieten MapFish und Mapbender fertige Lösungen an, damit auch mit weniger Programmieraufwand eine Anwendung für das Web-GIS erstellt werden kann.

Die Entscheidung bei der Wahl der Web-Mapping-Anwendung bzw. dem Framework ist deutlich schwieriger. Alle haben Vor- und Nachteile. Der Vorteil der Frameworks liegt eindeutig bei der Flexibilität, der Möglichkeit vieles umzusetzen und auch einfach zu erweitern. Der größte Nachteil ist der enorme Aufwand der Programmierung und die Einarbeitung in das Framework. Im Gegensatz liegt der Vorteil der Anwendungen bei dem schnellen Einsatz von Funktionen ohne zusätzlichen Programmieraufwand. Eine Anwendung ist viel schneller erstellt und eingerichtet als mit den Frameworks. Der einzige Nachteil ist, dass die Anwendungen schlecht durch neue Funktionen erweiterbar ist. Mapbender3 bietet weiterhin ein Rollen-

konzept und die Unterstützung von mehreren Benutzern. Zudem ist das Design im Nachhinein durch den Benutzer in der grafischen Oberfläche anpassbar.

Die Entscheidung fällt schließlich auf Mapbender3, da diese Web-Mapping-Anwendung alle Funktionalitäten bietet, die für ein Web-GIS notwendig sind. Mit einem Web-Mapping-Framework müssten alle Funktionen erst zu einer Anwendung programmiert werden. Der Programmieraufwand ist bei Mapbender3 nicht vorhanden und bietet dadurch im Gegensatz zu anderen Frameworks eine schnelle Einrichtung. Auch die Features sind völlig ausreichend für dieses System.

### **4.5 Betriebssystem**

Um die ausgewählten Softwarekomponenten auf einem System zu installieren, wird ein Betriebssystem benötigt. Alle Softwarekomponenten unterstützen eine Vielzahl von Betriebssystemen, dies schränkt die Wahl des Betriebssystems nicht ein. Die Anforderungen legen fest, dass sich die Auswahl auf ein freies Betriebssystems beschränkt, damit keine Lizenzkosten anfallen. Außerdem muss das Betriebssystem für Serversysteme ausgelegt sein. Zur Auswahl stehen daher viele Linux-Distributionen. Speziell für Serversysteme sind die beliebtesten Linux-Distribution Debian, Ubuntu, CentOS und OpenSuSE. Grundsätzlich kann jede dieser Linux-Distribution als Grundlage dienen.

Die Wahl fällt auf die aktuelle Linux-Distribution von Debian. Diese Linux-Distribution ist für die meisten Systemarchitekturen weit verbreitet und überzeugt durch eine hohe Stabilität sowie Sicherheit. (vgl. [Bel17])

## 5 Konzept und Entwurf des Systems

Nachdem die Softwarekomponenten analysiert wurden, wird in diesem Kapitel die Modellierung und Konzeption des Web-GIS mit den ausgewählten Softwarekomponenten vorgenommen. Zuerst wird auf die zugrunde liegende Architektur eingegangen. Danach werden die Komponenten des Systems anhand eines Komponentendiagramms und die Funktionsweise von WMS/WFS-Anfragen mithilfe von Sequenzdiagrammen dargestellt. Anschließend wird das zugrunde liegende Datenbankmodell entworfen und der Import der Daten mit der eigenen Schnittstelle beschrieben. Weiterhin wird auf die räumliche Analyse der Geodaten eingegangen, und wie diese zeitlich dargestellt werden können.

### 5.1 Serviceorientierte Architektur (SOA)

Für das System ist eine geeignete Architektur notwendig. Anhand der Anforderungen ist eine Architektur mit einer losen Kopplung der Komponenten auszuwählen.

Ein webbasiertes Geoinformationssystem folgt grundsätzlich dem Gedanken der losen Kopplung zwischen den Komponenten und den standardisierten Austausch der Geodaten. Durch eine Serviceorientierte Architektur (SOA) kann dieser Aufbau realisiert werden. Anstelle das Komponenten direkt aufeinander zugreifen, werden einheitliche Schnittstellen genutzt. Dies fördert die Flexibilität des gesamten Systems. Außerdem wird dadurch die Abhängigkeit der Komponenten minimiert und der reibungslose Austausch von Komponenten ist möglich. Der Grund für den Austausch von Softwarekomponenten ist oftmals wegen neuer Softwarekomponenten, die auf dem Markt kommen bzw. Komponenten, die für das System besser geeigneter wären oder neuen Versionen bereits genutzter Softwarekomponenten. (vgl. [Mel10], S.9 ff.)

### 5.2 Ausgewählte Softwarekomponenten

In Kapitel 4 wurden die Softwarekomponenten analysiert und ausgewählt, welche für den Aufbau des Web-GIS eingesetzt werden. Die Softwarekomponenten kommen mit wenigen Systemressourcen aus, sodass lediglich nur ein Webserver mit PHP benötigt wird. Die Auswahl der Softwarekomponenten ist wie folgt:

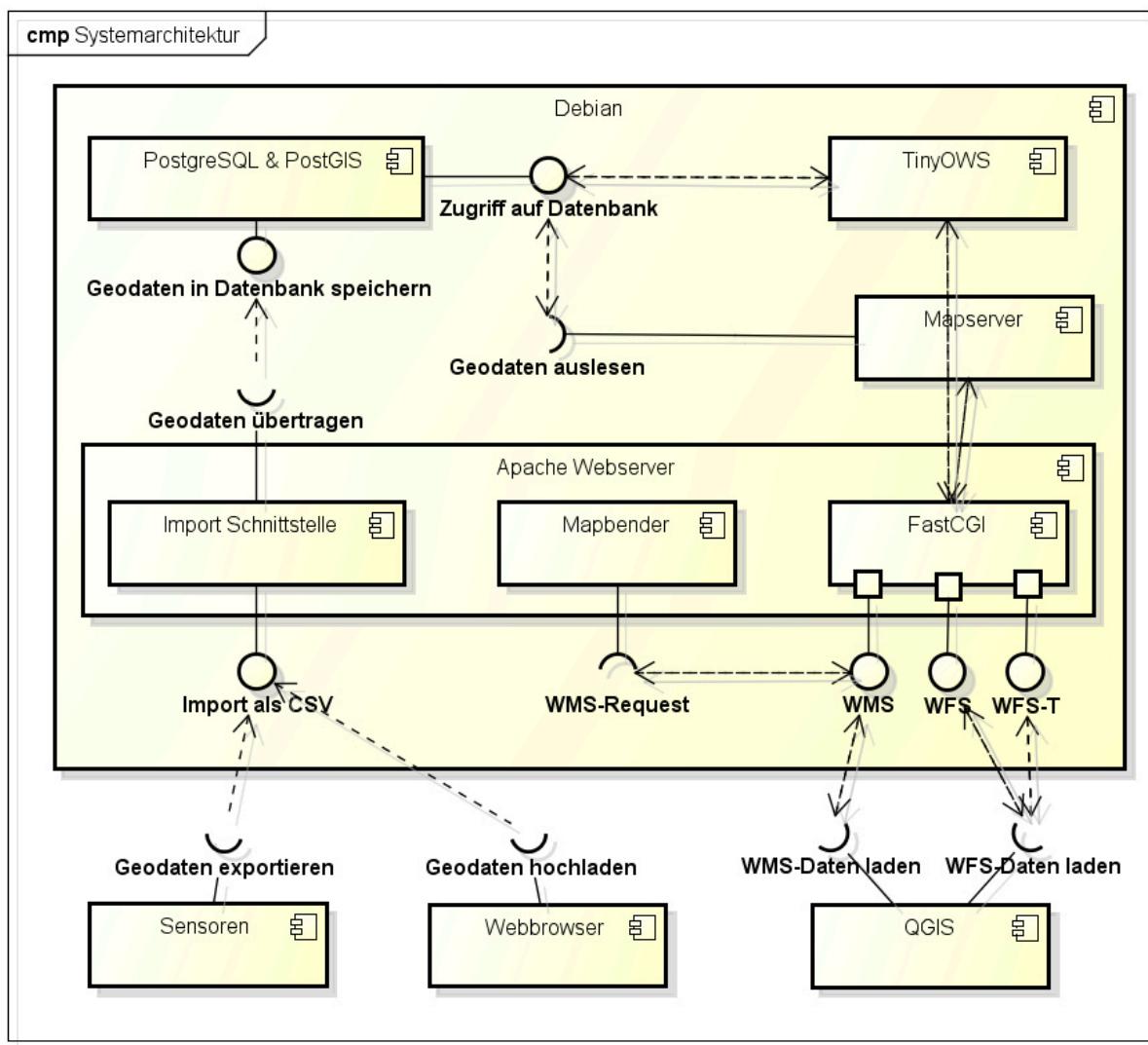
1. **Betriebssystem:** Debian 9.2.0
2. **Datenbank:** PostgreSQL 9.6.4 mit der PostGIS-Erweiterung 2.3.1
3. **Kartenserver:** Mapserver 7.0.4 (für WMS / WFS) und TinyOWS 1.1.1 (für WFS-T)
4. **Web-Mapping-Anwendung:** Mapbender 3.0.6.3

### 5.3 Systemarchitektur und Schnittstellen

Nachdem die Softwarekomponenten ausgewählt wurden, ist die Anordnung der Komponenten untereinander für das Gesamtsystem notwendig. Das Gesamtsystem wird in der Abb. 5.1 als Komponentendiagramm dargestellt. In dem Betriebssystem Debian werden alle Soft-

## 5. Konzept und Entwurf des Systems

warekomponenten sowie der Webserver mit der FastCGI-Schnittstelle installiert und eingerichtet. Die Geodaten können in das System durch eine Import-Schnittstelle von unterschiedlichen Quellen wie z. B. von Sensoren oder über den Webbrower importiert und in der Datenbank gespeichert werden. Der Import der Daten mit der Import-Schnittstelle erfolgt dabei in einem vordefinierten Format. Die FastCGI-Schnittstelle bietet die standardisierten Schnittstellen WMS, WFS, WFS-T zum Austausch der Geodaten. Der Kartenserver (Mapserver mit TinyOWS) verfügt über eine permanente Verbindung zur Datenbank und kommuniziert bei einer Anfrage mit der FastCGI-Schnittstelle im Webserver. Die Web-Mapping-Anwendung Mapbender stellt durch die WMS-Schnittstelle alle Geodaten von der Datenbank dar. Mithilfe von lokaler Software, wie z. B. QGIS, können die Geodaten durch die standardisierten Schnittstellen geladen und weiterführende Analysen durchgeführt werden.



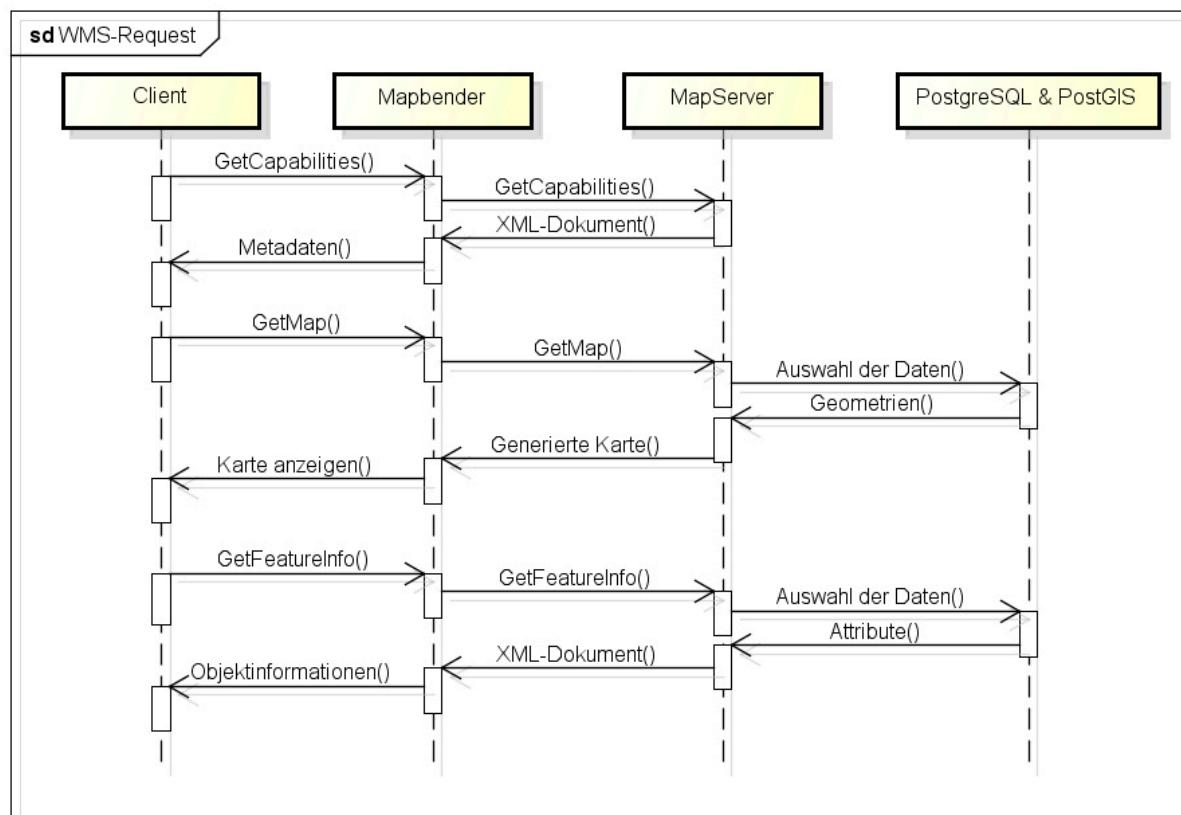
powered by Astah

Abb. 5.1: Systemarchitektur als Komponentendiagramm

## 5.4 Kommunikation

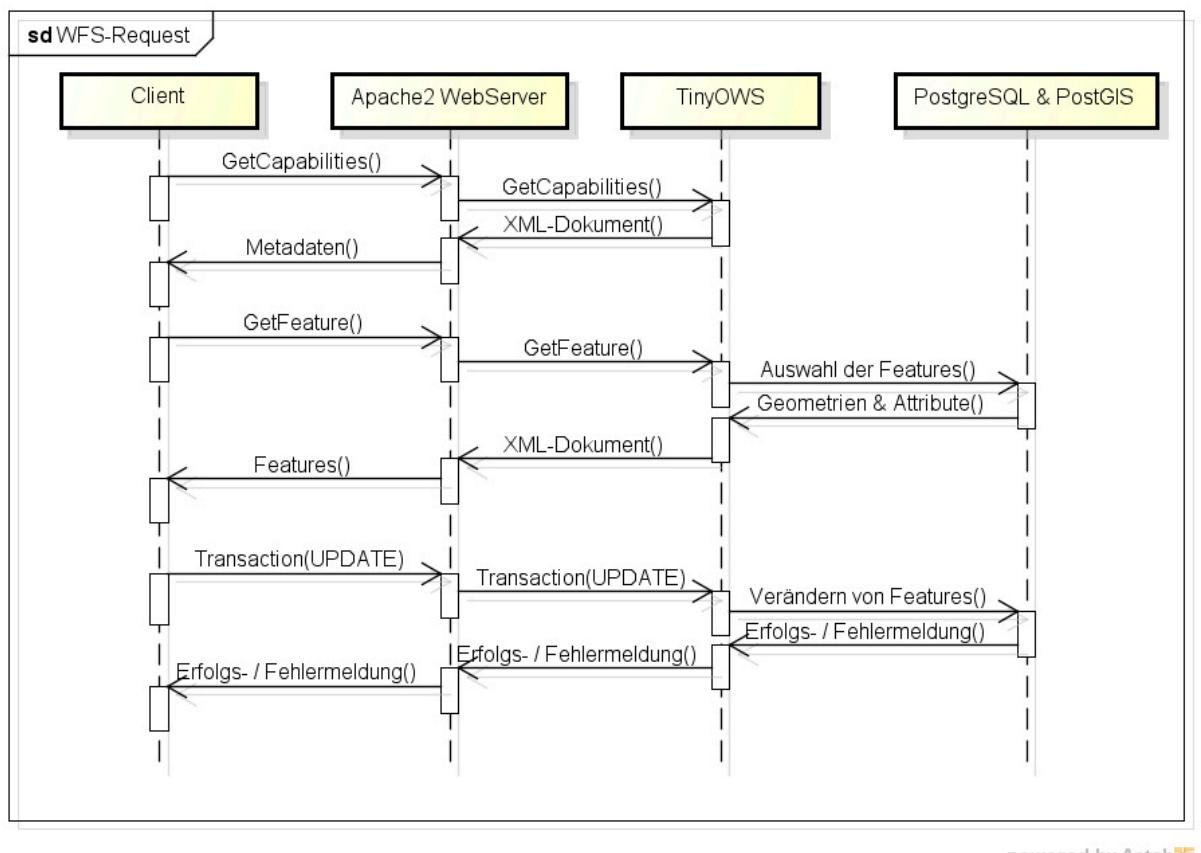
Für das System sind die standardisierten Schnittstellen für die Kommunikation mit anderen Komponenten zwingend erforderlich. Von zentraler Bedeutung sind hierbei die beiden standardisierten Schnittstellen WMS und WFS.

Die WMS-Schnittstelle wird zur Kommunikation sehr häufig im Bereich Web-GIS genutzt und bietet damit alle Möglichkeiten zur reinen Betrachtung und Auslesung der Geodaten. Der Vorgang der verschiedenen WMS-Requests ist mit den ausgewählten Softwarekomponenten in der Abb. 5.2 dargestellt. Hingegen bietet eine WFS-Schnittstelle die Möglichkeit alle Features als Vektordaten in QGIS zu importieren. Die Abb. 5.3 stellt den Vorgang mithilfe der TinyOWS-Komponente dar. Ein WFS-Request ist auch mit MapServer möglich, jedoch unterstützt nur TinyOWS zusätzlich die Transaktionen zum Verändern von Geodaten (WFS-T). Sinnvoll wäre daher TinyOWS nur durch autorisierte Nutzer verfügbar zu machen und MapServer als WMS- & WFS-Server für die Betrachtung und Analyse zu nutzen.



powered by Astah

Abb. 5.2: Funktionsweise der WMS-Requests im Sequenzdiagramm



powered by Astah

Abb. 5.3: Funktionsweise der WFS-Requests im Sequenzdiagramm

## 5.5 Datenbankmodell

Das Datenbankmodell ist für das Web-GIS von zentraler Bedeutung und muss den Anforderungen entsprechend entworfen werden, damit die Daten der Extensionsklüfte mit zeitlicher Angabe darauf abgelegt und zugeordnet werden können.

In Abb. 5.4 wird dieses Datenbankmodell dargestellt. Zwei Tabellen mit dem Primärschlüssel „gid“ dienen zur Speicherung der Geodaten. Die Geometriertypen „PointZ“ und „PolygonZ“ werden speziell für 3D-Punkte und 3D-Polygone genutzt. Für die Speicherung der Geometrien ist die Wahl eines Koordinatenreferenzsystems notwendig, dass auch bei 3D-Koordinaten genutzt werden kann. Das WGS84-Referenzsystem mit der SRID<sup>3</sup> 4326 ist ein solches Koordinatenreferenzsystem mit 3D-Koordinaten und wird im Datenbankmodell als Referenzsystem genutzt. Beide Tabellen mit der Geometrie PointZ und PolygonZ besitzen eine „id“ und „collected“ Spalte. In „collected“ wird das Datum der Aufnahme gespeichert. Die Spalte „id“ dient zur Identifikation von gleichen Punkten oder Polygonen mit verschiedenen Zeitpunkten, die in „collected“ gespeichert wurden. Um Angaben zur Person für beide Tabellen zu speichern, wird eine zusätzliche Relationstabelle „person“ aufgebaut. Eine weitere Relationstabelle „type“ beschreibt die Art des aufgenommenen Punktes. Zusätzliche Informa-

<sup>3</sup> Spatial Reference Identifier (zu Deutsch: „Referenz-Identifikationsnummer für räumliche Koordinatensysteme“) ist eine Identifikation für räumliche Koordinatensysteme und sind damit vergleichbar mit den EPSG-Codes der European Petroleum Survey Group Geodesy.

tionen für Polygone sind die Ausrichtung (zwischen 0°-180°), die maximale sowie minimale Kluftöffnung und die Tiefe.

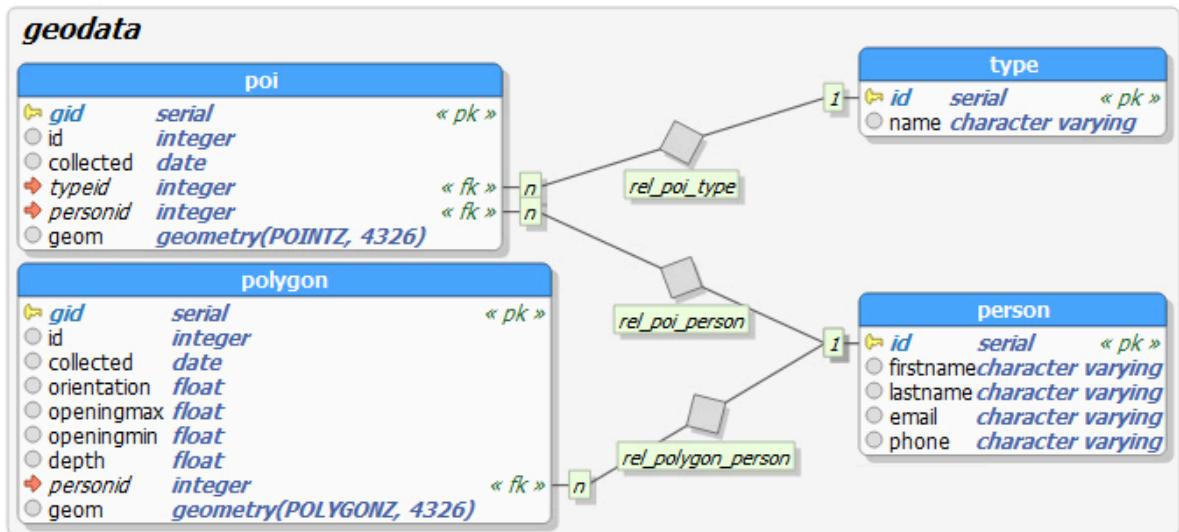


Abb. 5.4: PostgreSQL-Datenbankmodell

## 5.6 Import-Schnittstelle

Nachdem das Datenbankmodell entworfen wurde, müssen die Daten durch eine geeignete Schnittstelle in das Web-GIS importiert werden. Wie die Import-Schnittstelle mit dem Gesamtsystem interagiert ist bereits im Kapitel 5.3 beschrieben worden. Die Daten werden in einem einfachen Format geliefert und in das System importiert. Hierfür bietet sich ein einfaches CSV-Format an. Nach dem Datenbankmodell müssen alle Koordinaten in der CSV-Datei der SRID 4326 unterliegen, andernfalls müssen die Koordinaten vorher in dieses System durch eine Koordinatentransformation transformiert werden. In der Kopfzeile einer CSV-Datei werden die einzelnen Felder angegeben. Das Trennzeichen für die einzelnen Felder ist hierbei das Semikolon. Zeilenweise darunter erfolgen dann die eigentlichen Daten.

Das Importieren von Punkten ist relativ einfach, da lediglich die Felder `id`, `collected`, `typeid`, `personid`, `x`, `y` und `z` benötigt werden. Hingegen beim Importieren von Polygonen müssen die 3D-Koordinaten als eine Liste von Koordinatenpaaren angegeben werden. Dabei können beliebig viele Koordinatenpaare angegeben werden. Diese Koordinatenpaare werden in der Reihenfolge X, Y, Z angegeben und sind durch das Komma voneinander getrennt. Am Anfang und am Ende der Liste müssen die gleichen Koordinatenpaare stehen, damit ein geschlossenes Polygon entstehen kann. Der Import der Daten für die Relationstabelle „type“ ist sehr simpel, da lediglich nur ein Feld angegeben wird. Die Relationstabelle „person“ enthält neben den Vor- und Nachnamen noch weitere Angaben zur Person.

Im Folgenden werden vier Beispieldateien für den Import als CSV-Datei dargestellt, an denen sich orientiert werden kann:

## 5. Konzept und Entwurf des Systems

---

- **Importieren von Punkten**

```
id;collected;typeid;personid;x;y;z
1;2016-07-22;2;1;-9.262581;32.322909;10.00000
1;2017-07-22;2;1;-9.262580;32.322908;10.00000
2;2014-03-22;1;2;-9.263060;32.322646;11.00004
2;2015-04-22;1;2;-9.263061;32.322645;11.00003
2;2016-05-22;1;2;-9.263062;32.322644;11.00002
2;2017-06-22;1;2;-9.263063;32.322643;11.00001
```

- **Importieren von Polygonen**

```
id;collected;orientation;openingmax;openingmin;depth;personid;edges
1;2015-07-22;90;15;13;3;1;-9.26201 32.322800 10.00000, -9.26199
32.322802 10.00000, -9.26196 32.322802 10.00000, -9.26195 32.322801
10.00000, -9.26196 32.322800 10.00000, -9.26199 32.322800 10.00000,
-9.26201 32.322800 10.00000
1;2016-07-22;90;15;13;3;1;-9.26203 32.322801 10.00000, -9.26200
32.322803 10.00000, -9.26195 32.322803 10.00000, -9.26193 32.322802
10.00000, -9.26195 32.322799 10.00000, -9.26200 32.322799 10.00000,
-9.26203 32.322801 10.00000
1;2017-07-22;90;15;13;3;1;-9.26205 32.322802 10.00000, -9.26201
32.322804 10.00000, -9.26194 32.322804 10.00000, -9.26190 32.322803
10.00000, -9.26194 32.322798 10.00000, -9.26201 32.322798 10.00000,
-9.26205 32.322802 10.00000
```

- **Importieren von Personennamen**

```
firstname;lastname;email;phone
Tino;Schuldt;lg11127@hs-nb.de;0152111111
Andreas;Wehrenpfennig;wehrenpfennig@hs-nb.de;0152111112
Ralf;Loewner;loewner@hs-nb.de;0152111113
```

- **Importieren von Punktarten**

<b>name</b>
Festpunkt
Messpunkt

## 5.7 Räumliche Analyse

Nachdem die Daten in die Datenbank importiert worden sind, muss für das Web-GIS noch die Ausbreitung der Extensionsklüfte berechnet werden. Hierfür bieten sich serverseitige Funktionen zur Berechnung an. PostGIS stellt viele Funktionen, die im Kapitel 4.2.1 bereits erläutert wurden, zur Verfügung. Diese werden unter anderem genutzt, um die Ausbreitung zu bestimmen und darzustellen.

Speziell für Punkte eignen sich die folgenden Funktionen:

- **ST\_Force2D** zum Umwandeln in 2D-Koordinaten
- **ST\_X**, **ST\_Y** und **ST\_Z** um den Punkt in X-, Y- und Z-Koordinaten zurückzugeben
- **ST\_Distance** zum Berechnen der Abstände zweier Punkte
- **ST\_MakeLine**, um ein Linien-Geometriertyp zwischen beiden Punkten zu erstellen

## 5. Konzept und Entwurf des Systems

---

Mit diesen Funktionen wird die Ausbreitung zwischen einem gemessenen Punkt aus zwei verschiedenen Zeitpunkten berechnet und mit einem Pfeil als Linien-Geometriertyp visualisiert. Das Resultat kann dann in einer Web-Mapping-Anwendung wie in der Abb. 5.5 dargestellt werden.

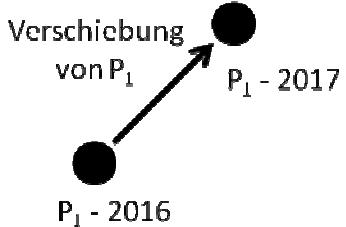


Abb. 5.5: Verschiebung eines Punktes

Für die Berechnung der Ausbreitungsfläche aus dem Polygon-Geometriertyp, der aus mehreren Knotenpunkten bestehen, werden außerdem noch weitere PostGIS-Funktionen benötigt. Folgende Funktionen eignen sich hierfür:

- **ST\_Force2D** zum Umwandeln in 2D-Koordinaten
- **ST\_Area** zum Berechnen der Fläche des Polygons
- **ST\_Perimeter** zum Berechnen des Umfangs des Polygons
- **ST\_Centroid** um den Schwerpunkt eines Polygons zu berechnen
- **ST\_Distance** zum Berechnen der Abstände zweier Schwerpunkte
- **ST\_MakeLine** um eine Linie zwischen zwei Schwerpunkten zu zeichnen
- **ST\_Difference** um ein neues Polygon zu erzeugen, das die Überlappung zweier Polygone darstellt

Mit diesen Funktionen wird die Distanz zwischen zwei Schwerpunkten berechnet und mit einem Pfeil als Linien-Geometriertyp dargestellt. Weiterhin wird durch Differenz zweier Polygone die Fläche berechnet, die einer Ausbreitung gleicht. Das Resultat kann dann in einer Web-Mapping-Anwendung wie in der Abb. 5.6 dargestellt werden.

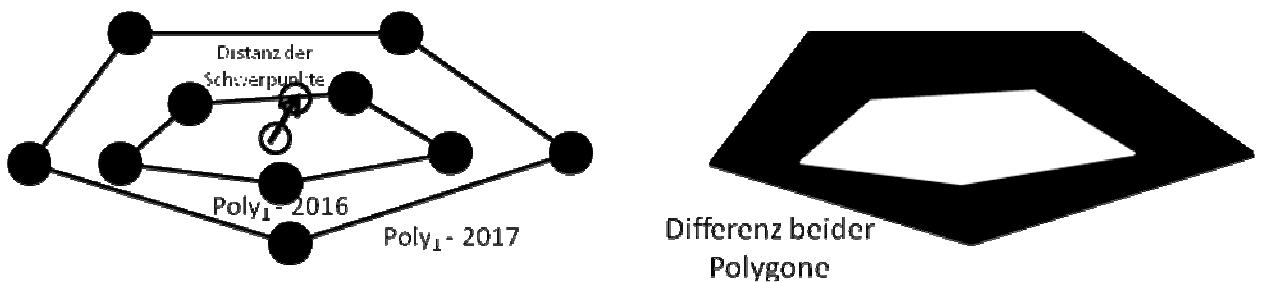


Abb. 5.6: Ausbreitung eines Polygons

## 5.8 Aufbereiten der Daten

Nachdem die notwendigen PostGIS-Funktionen erläutert wurden, müssen diese auf die Daten angewendet und in einer separaten Tabelle gespeichert werden. Diese separate Tabelle sollte nur temporär gesichert werden, um Redundanzen in der Datenbank zu vermeiden. PostgreSQL bietet dafür sogenannte Sichten an.

Sichten (engl. Views) sind SQL-Anweisungen auf die Datenbank, die eine Ergebnistabelle mit einem Namen erstellen, ohne dass dabei eine neue reale Tabelle erstellt wird. Dadurch können Berechnungen bzw. Verknüpfungen unter mehreren Tabellen durch eine einfache SELECT-Abfrage durchgeführt und als vordefinierte Ergebnistabelle ausgegeben werden. Mehrere Sichten können aufeinander aufbauen und sind notwendig für die Anzeige der Geodaten mit der GetFeatureInfo-Abfrage, sowie für die räumliche Berechnung der Geodaten mit den PostGIS-Funktionen.

In den Sichten werden alle Felder benutzt, die auch später beim Auslesen der Daten erscheinen sollen. Die Tabelle 5.1 stellt diese Felder anhand der beiden Geometrietypen zusammen.

Punkt	Polygon
Aufnahmezeitpunkt	Aufnahmezeitpunkt
Jahreszahl	Jahreszahl
Personenname	Personenname
Art des Punktes	Orientierung
Koordinaten X,Y,Z	Maximale Kluftöffnung
	Minimale Kluftöffnung
	Klufttiefe
	Fläche
	Umfang
	Schwerpunkt
Distanz zweier Punkte	Distanz zweier Schwerpunkte
Linie zwischen den Punkten als Geometrieobjekt	Linie zwischen den Schwerpunkten als Geometrieobjekt
	Differenz beider Polygone als Geometrieobjekt
	Fläche der Differenz
Maximale Distanz aller Punkte	Maximale Fläche
Anzahl der berechneten Distanzen	Anzahl der berechneten Flächen
Mittelwert der Distanzen pro Jahr	Mittelwert der Flächengröße pro Jahr

Tabelle 5.1: Felder der Sichten

## **5.9 Darstellung der zeitabhängigen Geodaten**

Die Funktionen und Darstellungen bei der räumlichen Analyse wären damit geklärt. Weiterhin ungeklärt ist die Auswahl der Daten für das Aufbereiten. Für das Aufbereiten und damit auch für die Darstellung der Daten wären vorgegebene Zeitintervalle sinnvoll wie z. B. eine jährliche Ausbreitung. Für die Darstellung zur Ausbreitung der Extensionsklüfte in der Web-Mapping-Anwendung werden zwei Möglichkeiten in Betracht gezogen.

Bei der **maximalen Ausbreitung** wird dem frühesten und spätesten Aufnahmezeitpunkt eines Punktes bzw. Polygons durch die räumlichen Funktionen verglichen. Dadurch kann die maximale Ausbreitung bestimmt und durch zwei Punkte oder Polygone dargestellt werden. Unterschiedliche Farben bei der Darstellung der Geometrien ermöglichen zudem das direkte Erkennen vom frühesten und spätesten Aufnahmezeitpunkt.

Bei der **jährlichen Ausbreitung** wird der späteste Aufnahmezeitpunkt jeden Jahres gewählt und mit dem spätesten Aufnahmezeitpunkt des nächsten Jahres verglichen. Dadurch kann die Ausbreitung für jedes Jahr bzw. auch die durchschnittliche Ausbreitung pro Jahr oder mögliche Trends für die nächsten Jahre berechnet werden.

## **5.10 Analyse der Extensionsklüfte**

In der Web-Mapping-Anwendung wäre es sehr aufwändig alle Extensionsklüfte einzeln durchzugehen, um die größte Ausbreitung zu ermitteln. Daher wäre eine einfache Analyse der Extensionsklüfte sinnvoll. Dadurch, dass die Geodaten verschiedene Darstellungsmöglichkeiten (jährliche und maximale Ausbreitung) besitzen, muss dies bei der Analyse der Extensionsklüfte berücksichtigt werden. Das könnte durch eine Auswahlbox realisiert werden, welche die Wahl zwischen maximaler, jährlicher und durchschnittlich jährlicher Ausbreitung gewährleistet. Weiterhin muss bei der Analyse nach den Geometriertypen (Punkt, Polygon) unterschieden werden. Bei dieser Analyse werden dann alle Extensionsklüfte dargestellt, bei denen die Ausbreitung am höchsten ist. Somit ist eine schnelle Analyse zu allen Extensionsklüften möglich.

## 6 Einrichtung des Systems

Nachdem das Konzept und die Entwürfe des Systems beschrieben wurden, wird sich in diesem Kapitel mit der Einrichtung des Systems beschäftigt. Die ausgewählten Softwarekomponenten sollen hierbei installiert und konfiguriert werden, sodass diese zusammen mit den Webseiten für den Import und die Analyse ein nutzbares Gesamtsystem ergeben.

### 6.1 Erste Überlegungen

Die Implementierung der Softwarekomponenten soll möglichst einfach und effizient erfolgen, sodass mit wenig Aufwand das System schnell einsetzbar ist. Um dies zu realisieren, sollte die Installation und Konfiguration automatisch, wie bei einem Setup-Programm, erfolgen. Dafür wird zusätzliche Software in Form von Skripten benötigt. Da eine grafische Oberfläche (GUI) bei dem Betriebssystem optional und damit nicht unbedingt notwendig ist, sollten die Programme auch ohne GUI nutzbar sein. Die Grundlage der Installation und Konfiguration bilden Shell-Skripte, die häufig bei Linux-Systemen eingesetzt werden, um Befehle zu automatisieren. Die Skripte sollten je nach Aufgabe als eigenständige Datei fungieren, damit jedes Skript eine klar definierte Aufgabe übernimmt. Zudem ermöglicht es die Flexibilität für den Austausch von einzelnen Softwarekomponenten und die Anpassung der Skripte an richtiger Stelle. Eine Konfigurationsdatei an zentraler Stelle enthält alle Informationen, die durch den Anwender eingetragen werden müssen.

### 6.2 Grundlage

Als Grundlage der Skripte dient das Betriebssystem Linux. Durch die Analyse im Kapitel 4.5 wurde sich für die Linux-Distribution Debian entschieden. Die Installation und Konfiguration aller Softwarekomponenten erfolgt durch Shell-Skripte mit dem Kommandozeileninterpreter „Bash“, da dieser bei vielen Linux-Systemen standardmäßig eingesetzt wird. Wie übliche Setup-Programme benötigt die Installationsroutine selbstverständlich Administratorrechte (Root-Rechte) auf dem System. Hilfreich sind hierbei Befehle wie „sudo“ zum Starten der Skripte als Nutzer mit entsprechenden Rechten oder das Wechseln zum „root“ mit dem Befehl „su“.

### 6.3 Installation des Betriebssystems

Debian steht als kleines (~300 MB) oder vollständiges Installationsimage auf der Webpräsenz <https://www.debian.org> für 32- und 64 Bit-Systeme sowie verschiedene Hardwarearchitekturen zur Verfügung. Für die Installation stehen zwei Möglichkeiten zur Verfügung.

**Datenträger:** Das Installationsimage wird in einem CD-/DVD-Brenner mit entsprechender Software auf einem bootfähigen Datenträger gebrannt. Als Alternative kann mit entspre-

chender Software auch ein bootfähiger USB-Stick erstellt werden. Beim Starten des Rechners ist dann der bootfähige Datenträger einzulegen und von dort zu starten.

**Virtuelle Maschine:** Hierbei wird ein zweites Betriebssystem in einem Programm visualisiert und kann wie ein richtiges Betriebssystem genutzt werden. Weitverbreitete Software dafür sind VMware, VirtuelBox und Windows Virtual PC. Nach der Installation der Software wird das Installationsimage ausgewählt und die Installation des Betriebssystems erfolgt wie auf einem echten Rechner.

### **6.4 Installationsskripte**

Nachdem die Installation des Betriebssystems durchgeführt wurde, können die Skripte für die Installation des Systems ausgeführt werden. Das Installationsskript mit dem Namen „install.sh“ führt die komplette Installation und Konfiguration der Softwarekomponenten aus. Dabei werden weitere Skripts im Ordner „script“ aufgerufen. Die Skripte sind gekennzeichnet durch die Präfixe „install\_“, „config\_“ und „create\_“. Die Softwarekomponenten werden zuerst installiert und direkt danach konfiguriert, bevor eine neue Softwarekomponente installiert wird. Die Installation von vielen Softwarekomponenten erfolgt durch die interne Paketverwaltung (APT - Advanced Packaging Tool). Softwarekomponenten, die nicht in der Paketverwaltung vorhanden sind, müssen für das System zuerst kompiliert werden. Dies betrifft die Komponente TinyOWS. Die Reihenfolge für die Abarbeitung der Softwarekomponenten ist wie folgt:

1. Installation von Apache2, PHP und Fcgid
2. Konfigurieren von Apache2 und Fcgid
3. Installation von PostgreSQL und der PostGIS-Erweiterung
4. Konfigurieren von PostgreSQL
5. Konfigurieren von PostGIS
6. Tabellen erstellen in PostgreSQL
7. Sichten erstellen in PostgreSQL
8. Installation von Mapserver
9. Installation durch das Kompilieren von TinyOWS
10. Konfigurieren von TinyOWS für den autorisierten Zugriff auf die Tabellen
11. Mapfiles erstellen
12. HTML-Dateien für die GetFeatureInfo-Abfrage eines WMS erstellen
13. Installation von Mapbender3
14. Konfiguration von Mapbender3
15. Eine vordefinierte Mapbender3 Applikation erstellen
16. Webseite für die Analyse der Geodaten erstellen
17. Webseite für den Import der Geodaten erstellen

## 6.5 Konfigurationsskripte

### 6.5.1 Konfigurationskript des Systems

Bevor die Installation gestartet wird, muss die Konfigurationsdatei im Ordner „config“ angepasst werden. Hierbei müssen folgende Angaben verändert werden:

- IP-Adresse des Servers, um von außerhalb erreichbar sein zu können
- Passwort für den Standard Postgres-Nutzer
- Nutzernname, Passwort und Name der Datenbank
- Nutzername und Passwort für die Authentifizierung mit WFS-T
- Nutzername und Passwort für die grafische Oberfläche der Web-Mapping-Anwendung (Mapbender3)
- Passwort für den Import der Geodaten mit der Webseite
- Metadaten für den WMS/WFS

### 6.5.2 Konfiguration von PostgreSQL / PostGIS

Als Erstes muss die Konfiguration der Datenbank erfolgen. Mit dem Skript „config\_postgres.sh“ wird die Datenbank bei der Installation eingerichtet und mit dem Skript „config\_postgis.sh“ wird die PostGIS-Erweiterung der Datenbank hinzugefügt. Danach werden Tabellen und Sichten durch die Skripte „create\_tables.sh“ und „create\_views.sh“ erstellt. Folgende Aufgaben werden durch diese Skripte bei der Konfiguration bewältigt:

1. Das Standardpasswort des Datenbankbenutzers „postgres“ wird geändert, um unerlaubten Zugriff Dritter zu verhindern.
2. Ein neuer Datenbanknutzer mit den Rechten „Superuser“ wird angelegt.
3. Eine neue Datenbank „efdb“ mit dem Schema „geodata“ für die Geodaten und Sichten wird erstellt.
4. In die Datenbank werden zwei Relationstabellen (type, person) und zwei Tabellen mit Geometriespalte (poi, polygon) erstellt.
5. Zum Schluss werden mehrere Sichten für die maximale und jährliche Ausbreitung auf die Tabellen erstellt.

### 6.5.3 Konfiguration von TinyOWS

Nachdem die Datenbank steht, wird die WFS-T Schnittstelle mit der Softwarekomponente TinyOWS eingerichtet. Mit dem Skript „config\_tinyows.sh“ wird die Konfigurationsdatei von TinyOWS („tinyows.xml“) erstellt und in das Verzeichnis „etc“ verschoben. Die Konfigurationsdatei enthält Angaben zur Verbindung mit der Datenbank (Host, Port, Datenbankname, Username und Passwort) sowie Einstellungen zu allen Layern. Alternativ kann die Konfiguration von TinyOWS auch durch Mapfiles vorgenommen werden, um einen WFS-T aufzubau-

## 6. Einrichtung des Systems

---

en. Die Konfigurationsdatei ist im Gegensatz zu Mapfiles viel übersichtlicher und leichter anpassbar. Die Konfigurationsdatei ist wie folgt aufgebaut:

```
<tinyows online_resource="http://192.168.174.149/cgi-bin/wfs-t/tinyows" schema_dir="/usr/local/share/tinyows/schema/">
  <pg host="localhost" user="efuser" password="efpassword" dbname="efdb" port="5432"/>
    <metadata name="TinyOWS Server" title="TinyOWS Server" />
    <layer retrievable="1"
      writable="1"
      ns_prefix="layer"
      ns_uri="http://192.168.174.149"
      schema="geodata"
      name="poi"
      title="poi"
    />
    <layer retrievable="1"
      writable="1"
      ns_prefix="layer"
      ns_uri="http://192.168.174.149"
      schema="geodata"
      name="polygon"
      title="polygon"
    />
</tinyows>
```

Wurde die Konfiguration erfolgreich ausgeführt, dann kann die Funktionsfähigkeit mit dem Befehl `./tinyows --check` im „cgi-bin“-Verzeichnis geprüft werden. Als Ausgabe erscheint dann die Versionsnummer, die PostGIS-Versionsnummer, die Verbindungsdaten und alle verfügbare Layer.

```
TinyOWS version: 1.1.0
FCGI support: Yes
Config File Path: /etc/tinyows.xml (TinyOWS XML)
PostGIS Version: 2.3.1
PostGIS dsn: host=localhost user=efuser password=efpassword
dbname=efdb port=5432
Output Encoding: UTF-8
Database Encoding: UTF8
Schema dir: /usr/local/share/tinyows/schema/
Display bbox: Yes
Estimated extent: No
Check schema: Yes
Check valid geoms: Yes
Available layers:
- geodata.poi (4326) -> layer:poi [RW]
- geodata.polygon (4326) -> layer:polygon [RW]
```

TinyOWS ist nun einsatzbereit und kann von jedem genutzt werden, der die URL zu TinyOWS kennt. Um den unerlaubten Zugriff vor Dritten zu verhindern, wird TinyOWS in ein neues Verzeichnis `wfs-t` abgelegt und dieses Verzeichnis mit einer Basisauthentifizierung

abgesichert. In Apache2 wird dafür eine neue Konfigurationsdatei `auth-tinyows.conf` im Verzeichnis `/etc/apache2/sites-available/` mit dem folgenden Inhalt abgelegt:

```
Directory /usr/lib/cgi-bin/wfs-t/>
AuthType Basic
AuthName "Authentication for WFS-T"
AuthUserFile /etc/apache2/.htpasswd
require valid-user
</Directory>
```

Anschließend wird ein neuer User für den Zugriff erstellt, die Konfigurationsdatei aktiviert und der Apache2 neu gestartet. Für die Nutzung von TinyOWS ist nun eine Authentifizierung notwendig.

```
> htpasswd -c -b /etc/apache2/.htpasswd tinyuser tinypassword
> a2ensite auth-tinyows.conf
> service apache2 reload
```

### **6.5.4 Konfiguration von MapServer**

Im nächsten Schritt der Installationsroutine werden die Konfigurationsdateien für den MapServer erstellt. Das Skript „create\_mapfiles.sh“ erzeugt dafür drei verschiedene Mapfiles. Diese Mapfiles werden als Konfiguration für den WMS/WFS genutzt und beschreiben die jährliche und maximale Ausbreitung der Extensionsklüfte. Das dritte Mapfile stellt lediglich alle aufgenommenen Datensätze ohne eine genutzte räumliche Funktion dar und kann beispielsweise mit lokaler Software für weitere Analysen verwendet werden.

Beschrieben wurde der generelle Aufbau eines Mapfiles bereits im Kapitel 4.3.2. Die folgenden Parameter sind notwendig für die Konfiguration der Mapfiles und werden bei der Konfiguration aus der Konfigurationsdatei des Projektes entnommen.

- Name des Mapfiles
- Koordinatenreferenzsystem der vorhandenen Geodaten
- Speicherort zum Ablegen der erzeugten Grafiken
- Metadaten zum Layer (Titel, Abstract, unterstützte Koordinatenreferenzsysteme SRS usw.)
- URL zum Erreichen des WMS/WFS
- Namen der Layer
- Geometriertyp der Layer
- Verbindung zur Datenbank (Host, Datenbank, Username, Passwort)
- Name der Geometriespalte
- Pfad zur HTML-Datei zum Abfragen von Feature-Informationen

Diese Mapfiles sollten nun im „cgi-bin“-Verzeichnis abgelegt werden, da in einem Mapfile Informationen zur Verbindung mit der Datenbank enthalten sind und in diesem Verzeichnis der Zugang für das Lesen der Mapfiles gesperrt ist.

Anschließend werden die HTML-Dateien für die GetFeatureInfo-Anfrage eines WMS erstellt. Dafür ist das Skript „create\_wms\_getfeatureinfo.sh“ zuständig. Die HTML-Dateien müssen in einem Verzeichnis sein, von dem die Web-Mapping-Anwendung diese erreichen kann. Idealerweise bietet sich dafür das Verzeichnis „/var/www/html/“ an. Der Aufbau einer HTML-Datei kann beliebig gestaltet sein, lediglich der Header der HTML-Datei muss mit „<!-- MapServer Template -->“ gekennzeichnet sein, damit die Web-Mapping-Anwendung diese erkennt. Die Attribute der abfragbaren Layer sind durch Platzhalter mit eckigen Klammern gekennzeichnet.

### 6.5.5 Konfiguration von Mapbender3

Nachdem der WMS/WFS einsatzbereit ist, kann die Konfiguration der Web-Mapping-Anwendung vorgenommen werden. Der Mapbender3 wird durch das Skript „config\_mapbender3.sh“ auf dem Betriebssystem eingerichtet. Die dafür notwendigen Schritte sind auf der Webpräsenz von Mapbender<sup>4</sup> zu entnehmen und sind wie folgt:

- Aktivieren des Apache Moduls rewrite mit `a2enmod rewrite`
- Entpacken des Mapbender3 Archiv ins Verzeichnis `/var/www/mapbender3`
- Anlegen der Konfigurationsdatei in Apache2 `/etc/apache2/sites-available/mapbender3.conf` mit folgendem Inhalt:

```
Alias /mapbender3 /var/www/mapbender3/web/
<Directory /var/www/mapbender3/web/>
    Options MultiViews FollowSymLinks
    DirectoryIndex app.php
    AllowOverride none
    Order allow,deny
    Allow from all

    RewriteEngine On
    RewriteBase /mapbender3/
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^(.*)$ app.php [QSA,L]
</Directory>
```

- Aktivieren der Konfigurationsdatei mit `a2ensite mapbender3.conf`
- Neustarten von Apache2 mit `service apache2 reload`
- Ändern der Zugriffsrechte mit:  
`> sudo chmod -R ugo+r /var/www/mapbender3`  
`> sudo chown -R www-data:www-data /var/www/mapbender3`

---

<sup>4</sup> Installationsanleitung zum Konfigurieren von Mapbender3 auf der Webseite:  
[http://doc.mapbender3.org/de/book/installation/installation\\_ubuntu.html](http://doc.mapbender3.org/de/book/installation/installation_ubuntu.html)

```
> sudo chmod -R ug+w /var/www/mapbender3/web/uploads
```

- Kopieren der Konfigurationsdatei „parameters.yml.dist“ nach „parameters.yml“ im Verzeichnis /var/www/mapbender3/app/config/
- Einfügen von Parameters zur Verbindung der Mapbender3-Datenbank

```
parameters:  
    database_driver:    pdo_pgsql  
    database_host:      localhost  
    database_port:     5432  
    database_name:     mapbender3  
    database_path:     ~  
    database_user:     mb3user  
    database_password: mb3password
```

- Wechseln ins Verzeichnis /var/www/mapbender3 zum Ausführen von folgenden Doctrine-Befehlen, um die Mapbender3-Datenbank zu erstellen:

```
> app/console doctrine:database:create  
> app/console doctrine:schema:create  
> app/console assets:install web  
> app/console fom:user:resetroot --username=root --  
  password=root --email=root@root.de ---silent  
> app/console doctrine:fixtures:load --  
  fix-  
  tures=./mapbender/src/Mapbender/CoreBundle/DataFixtures/ORM/  
  Epsg/ --append  
> app/console doctrine:fixtures:load --  
  fix-  
  tures=./mapbender/src/Mapbender/CoreBundle/DataFixtures/ORM/  
  Application/ --append<
```

### 6.5.6 Erstellen einer Anwendungen

Wie bereits im Kapitel 4.4.5 beschrieben, können vorkonfigurierte Anwendungen mit Mapbender3 im YAML-Format eingelesen werden, die sich als Datei im Verzeichnis /var/www/mapbender3/app/config/applications befinden. Das Erstellen von einer vorkonfigurierten Mapbender Anwendung geschieht über das Skript „create\_mapbender\_application.sh“. Dieses Skript erstellt eine YAML-Datei, die dann über einen Konsolenbefehl eingelesen wird. In dieser Datei werden die Layer angepasst, dass diese mit dem eigenen WMS genutzt werden können. Dafür muss nur die URL zum jeweiligen Mapfile angegeben und die verfügbaren Layer ausgewählt werden. Außerdem wird ein weiterer WMS von terrestris<sup>5</sup>, für die Anzeige der OpenStreetMap Daten, hinzugefügt und dient damit als Hintergrundkarte für das webbasierte Geoinformationssystem.

---

<sup>5</sup> Die Firma terrestris GmbH & Co. KG stellt frei zugängliche Web-Services zur Verfügung, die ohne Kosten genutzt werden können. Kostenloser Dienst unter: <http://ows.terrestris.de/osm/service>

## 6. Einrichtung des Systems

---

Alle notwendigen Elemente der Web-Mapping-Anwendung werden anhand der Analyse im Kapitel 1.6 ausgewählt und sind in der YAML-Datei beschrieben. Diese Elemente sind in der Tabelle 6.1 dargestellt und werden dann, wie auf der Abb. 6.1, bei der Web-Mapping-Anwendung angeordnet. In der grafischen Benutzeroberfläche der Web-Mapping-Anwendung kann diese Anordnung später beliebig verändert werden.

Elemente	Anordnung	Beschreibung
<b>Button für FeatureInfo</b>	Toolbar	Dialog für die
<b>Button für Print client</b>	Toolbar	Dialog für Bereichseinstellungen öffnen
<b>Button für Image Export</b>	Toolbar	Dialog für Kartenansicht speichern öffnen
<b>Button für WMS Loader</b>	Toolbar	Dialog für das hinzufügen weiterer WMS öffnen
<b>Button für GPS Position</b>	Toolbar	Auf die eigene Position navigieren.
<b>Button für Line Ruler</b>	Toolbar	Dialog für Linienmessung öffnen
<b>Button für Area Ruler</b>	Toolbar	Dialog für Flächenmessung öffnen
<b>Button für Redlining</b>	Toolbar	Dialog für Redlining öffnen
<b>Layertree</b>	Sidepane	Zeigt alle verfügbaren Layer an
<b>Legend</b>	Sidepane	Zeigt eine Legende der Symbole an
<b>HTML</b>	Sidepane	Einbinden eines Inlineframe für die Analyse
<b>HTML</b>	Sidepane	Einbinden eines Inlineframe für den Import der Geodaten
<b>Main Map</b>	Content	Die Karte
<b>Navigation toolbar</b>	Content	Steuerungselemente für die Navigation
<b>FeatureInfo</b>	Content	Auslesen von Attributen
<b>WMS Loader</b>	Content	Hinzufügen weiterer WMS-Quellen
<b>Print client</b>	Content	PDF-Druck eines Bereiches
<b>Image export</b>	Content	Speichern der aktuellen Kartenansicht
<b>Line Ruler</b>	Content	Messen einer Linie
<b>Area Ruler</b>	Content	Messen einer Fläche
<b>Redlining</b>	Content	Geometrien temporär auf die Karte zeichnen
<b>Overview</b>	Content	Kleine Übersichtskarte
<b>Scale bar</b>	Content	Maßstabsleiste
<b>Activity Indicator</b>	Footer	Stellt ein Ladebild dar, bis alle Aktivitäten fertig geladen wurden
<b>Coordinates Display</b>	Footer	Koordinaten anzeigen
<b>SRS Selector</b>	Footer	Koordinatenreferenzsystem ändern
<b>Scale Display</b>	Footer	Maßstabsanzeige
<b>Scale Selector</b>	Footer	Maßstab ändern
<b>HTML</b>	Footer	Einbinden von HTML-Code

Tabelle 6.1: Elemente der Web-Mapping-Anwendung

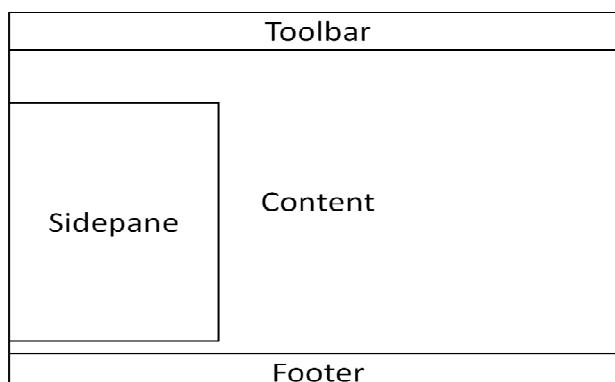


Abb. 6.1: Anordnung der Elemente auf dem Bildschirm

## 6.6 Import der Geodaten

Nachdem das System eingerichtet ist, wird die eigene Webseite für den Import der Daten auf dem Webserver abgelegt. Beim Import der Geodaten wird eine neue temporäre Tabelle in der Datenbank angelegt. In diese Tabelle werden alle Daten der CSV-Datei eingelesen und gespeichert. Anschließend werden aus dieser Tabelle alle Daten kopiert und in die ausgewählte Tabelle der Datenbank hinzugefügt.

Wie in der Abb. 6.2 dargestellt, wird diese Webseite als Inlineframe im Sidepane der Web-Mapping-Anwendung eingebunden. Um den Prozess des Datenimportes vor Unbefugten zu schützen, ist hierbei ein Passwort notwendig. Dieses Passwort wird beim Import in einer Textbox angegeben oder kann direkt in der URL mit dem Parameter „pw“ erfolgen. Mit einer Auswahlbox wird die Art des Imports ausgewählt. Anschließend wird die CSV-Datei auf dem eigenen PC gewählt und kann hochgeladen werden. Die Webseite speichert dann die CSV-Datei temporär auf dem Server, wo sie dann in ein vorgegebenes Verzeichnis kopiert und für das Auslesen zur Verfügung steht. Das Auslesen der Datei erfolgt dann mit PHP-Befehlen und das Speichern in die PostgreSQL-Datenbank mit SQL-Anweisungen.

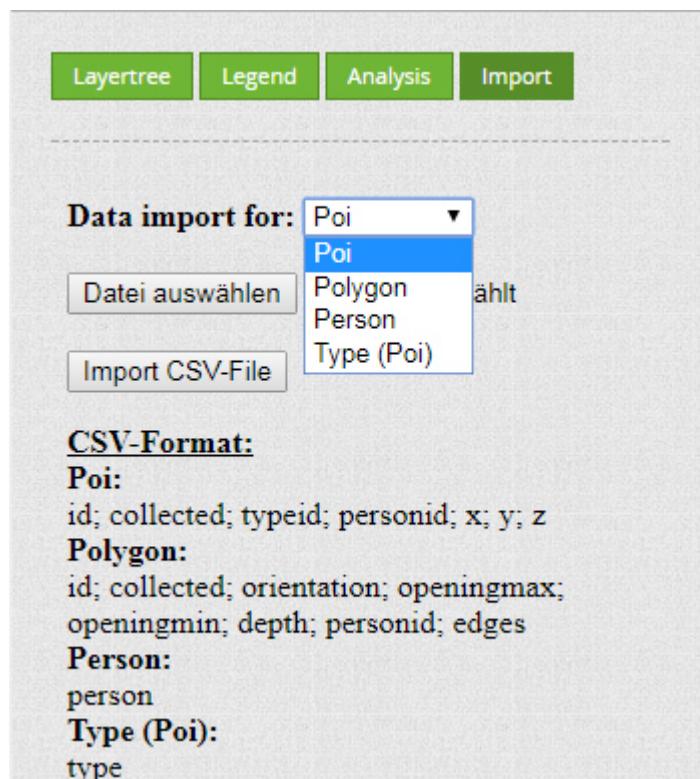


Abb. 6.2: Datenimport durch eine Webseite

## 6.7 Weitere Analysefunktionen

Zum Schluss erfolgt die Einrichtung der eigenen Webseite für die Analyse der Extensionsklüfte. Dafür wird die eigene Webseite auf dem Webserver abgelegt. Diese Webseite wird dann im Sidepane der Web-Mapping-Anwendung, wie in der Abb. 6.3 dargestellt, als Inline-frame eingebunden. Für die Analyse bietet eine Auswahlbox folgende drei Möglichkeiten:

- **Maximale** Distanz oder Flächengröße zwischen dem frühesten und spätesten Aufnahmezeitpunkt eines Punktes oder Polygons
- **Jährliche** Distanz oder Flächengröße zwischen jeweils dem spätesten Aufnahmezeitpunkt jeden Jahres
- **Jährliche durchschnittliche** Distanz oder Flächengröße

Anschließend kann in der ersten Textbox nach Distanzgrößen zwischen Punkten gesucht werden, die größer als die eingegebene Zahl in Metern sind. Die zweite Textbox ermöglicht das Suchen nach Flächengrößen zwischen Polygonen, die größer als die eingegebene Zahl in m<sup>2</sup> sind. Die Ausgabe beinhaltet dann die Distanzen bzw. Flächengrößen und den dazugehörigen Zeitraum. Durch einen Mausklick auf die ID wird das entsprechende Objekt in der Web-Mapping-Anwendung markiert. Durch diese Analyse kann herausgefunden werden, welche Extensionsklüfte sich am weitesten ausbreiten und in welchem Zeitraum.

The screenshot shows a user interface for geodata analysis. At the top, there are four green buttons labeled 'Layertree', 'Legend', 'Analysis', and 'Import'. Below them, a horizontal dashed line separates the navigation from the main content area. The 'Analysis' tab is currently selected. In the main area, there are two sets of input fields. The first set is for 'Poi distance [m] greater than:' and the second for 'Polygon area [m<sup>2</sup>] greater than:'. Each set includes an input field and a 'search' button. The background of the interface is light gray.

Abb. 6.3: Analyse der Geodaten durch die eigene Webseite

## 7 Nutzung des Systems

Die genauen Schritte bis hin zur fertigen Einrichtung des Systems auf einem Linux-Betriebssystem wird im Anhang „Anwenderdokumentation“ beschrieben. Nachdem die Einrichtung des Systems erfolgte, ist das Web-GIS einsatzbereit für die Nutzung.

### 7.1 Aufrufen des Web-GIS

Der Webbrower des Host-Betriebssystems wird geöffnet und die Webseite mit der IP-Adresse des Systems (beispielsweise: <http://192.168.174.149/mapbender3>) wird aufgerufen. Danach wird sich auf der grafischen Oberfläche als Administrator eingeloggt und die Anwendung gestartet. In der Abb. 7.1 wird dann die Web-Mapping-Anwendung mit den OpenStreetMap Daten dargestellt.

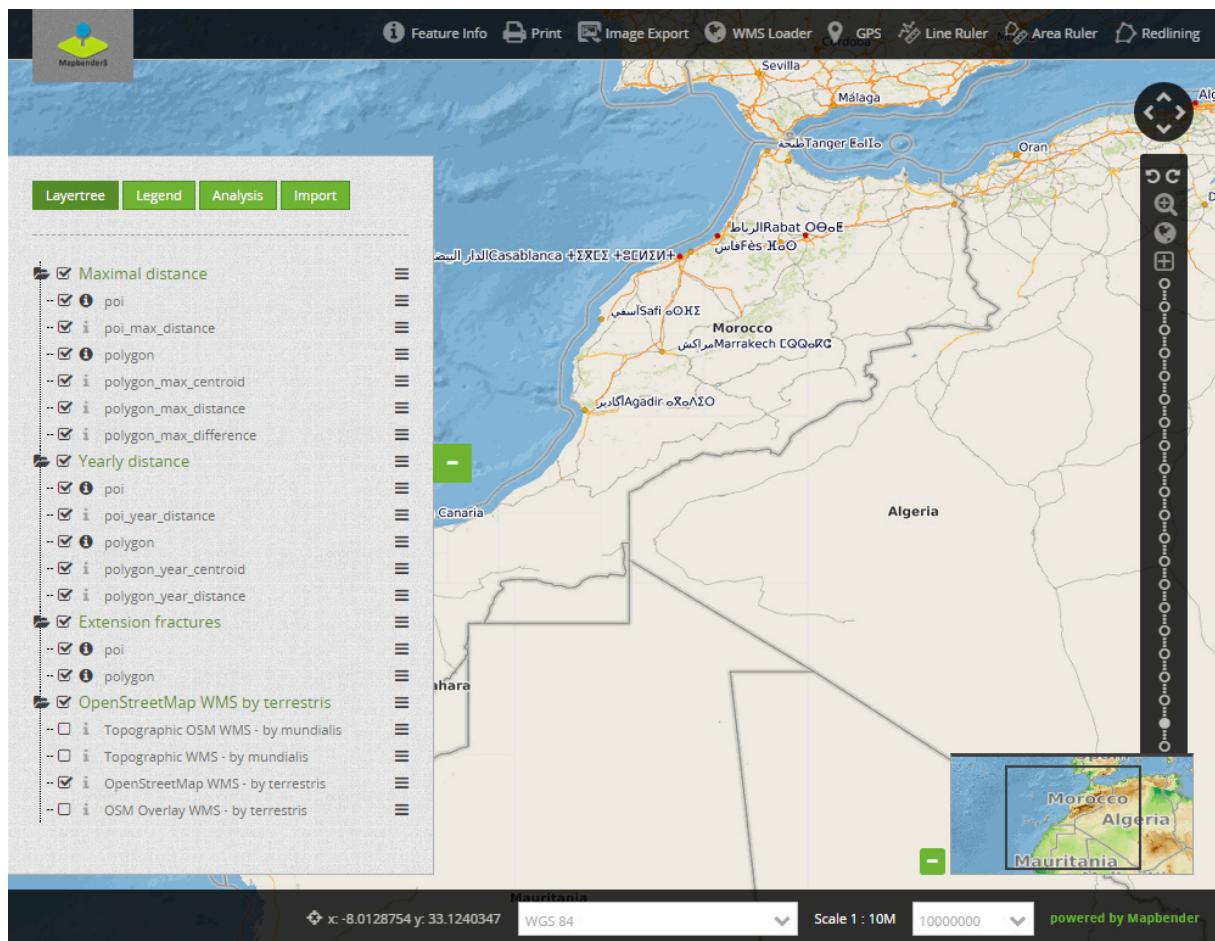


Abb. 7.1: Darstellung der Anwendung

### 7.2 Importieren von Daten

Im nächsten Schritt müssen nun die eigenen Daten in das System importiert werden. Mit der Schaltfläche „Import“ im Sidepane werden die Daten auf dem Server übertragen und in die Datenbank hinzugefügt. Wie im Kapitel 6.6 bereits beschrieben, werden die Daten als CSV-Dateien im vordefinierten Format mit dem entsprechenden Koordinatenreferenzsystem

(WGS84 mit der SRID 4326) eingelesen. Im Folgenden werden dafür die Beispieldaten im Ordner „import\_data“ des Projektes genutzt.

Zuerst müssen die Daten über die Personennamen und Punktarten hinzugefügt werden. In der Auswahlbox wird „Person“ gewählt und die entsprechende CSV-Datei mit den Personennamen über die Schaltfläche „Datei auswählen“ ausgewählt. Zum Schluss wird mit „Import CSV-File“ der Upload abgeschlossen. Nach dem erfolgreichen Importieren werden alle Personennamen mit der dazugehörigen ID aufgelistet. Diese Identifikationsnummer muss in den CSV-Dateien der Geodaten anstelle des Namens angegeben werden. Danach werden die Daten für die Punktarten durch Wahl von „Type (Poi)“ in der Auswahlbox importiert. Auch hier entspricht die ID dem Namen der Punktarten und muss in der CSV-Datei für Punkte anstelle des Namens angegeben werden.

Nachdem alle Personennamen und Punktarten in die Datenbank importiert wurden, werden die CSV-Dateien mit den Geodaten ausgewählt. In der Auswahlbox können Punkt und Polygone importiert werden. Zuerst werden die Punkte importiert und anschließend die Polygone. Die Daten befinden sich nun in der Datenbank und werden nach der nächsten Interaktion mit der Karte oder nach einem aktualisieren der Webseite angezeigt.

### **7.3 Darstellung der Geodaten**

Nachdem die Daten im System importiert wurden, können diese in der Web-Mapping-Anwendung genutzt werden. Auf der linken Seite können die Geodaten mit einer maximalen und jährlichen Ausbreitung bzw. jeden gemessenen Datensatz angezeigt werden.

Zuerst wird die maximale Ausbreitung gewählt und alle anderen Kategorien ausgeblendet. Diese Kategorie stellt jeweils den ältesten und frühesten Datensatz zusammen und zeigt diesen auf der Karte an. Nach der visuellen Definition in den Mapfiles wird hierbei der älteste gemessene Datensatz blau und der neuste Datensatz rot dargestellt. Ein grüner Pfeil zeigt bei den Punkten die Richtung und Distanz der Ausbreitung an (siehe Abb. 7.2). Bei Polygone stellen ein grüner Punkt den Schwerpunkt der Polygone dar, wobei auch hier ein Pfeil die Richtung und Distanz der Schwerpunkte anzeigt (siehe Abb. 7.3). Ein weiterer Layer stellt die Differenzfläche der beiden Polygone dar.

Mit der FeatureInfo-Abfrage (das i-Symbol in der Toolbar) werden Sachdaten zum Objekt abgefragt und als vordefinierte HTML-Seite dargestellt. Bei Punkten werden damit die vordefinierten Felder Layer, Aufnahmzeitpunkt, Punktart, Personenname, XYZ-Koordinaten, Referenz-Aufnahmzeitpunkt und die Ausbreitung in Metern ausgegeben. Hingegen enthalten Polygone noch zusätzliche Informationen über die Ausrichtung in Grad, maximale und minimale Kluftöffnung, Tiefe, Fläche in Metern, Umfang in Metern, Distanz der Schwerpunkte und die Ausbreitungsfläche in Metern.

Als Nächstes wird die jährliche Ausbreitung gewählt und die maximale Ausbreitung wieder ausgeblendet. Diese Kategorie zeigt jeweils den letzten Datensatz jeden Jahres an und ver-

## 7. Nutzung des Systems

gleicht diese miteinander. Zusätzlich wird die durchschnittliche Distanz bei Punkten und die durchschnittliche Ausbreitungsfläche bei Polygonen berechnet.

In der Abb. 7.4 wird ein Festpunkt in einem Aufnahmezeitraum von vier Jahren dargestellt. Mit der FeatureInfo-Abfrage auf dem letzten Punkt wird die Ausbreitung zwischen diesen Punkt und den gleichen Punkt im Jahr davor berechnet. Es wird die maximale und die durchschnittliche Distanz angezeigt. Für die Berechnung werden die einzelnen Distanzen summiert und durch deren Anzahl dividiert.

Die Abb. 7.5 stellt Polygone in einem Aufnahmezeitraum von drei Jahren dar. Die FeatureInfo-Abfrage auf das äußerste Polygon zeigt die Fläche zwischen diesem Polygon und mit dem Jahr davor an. Weiterhin wird auch die maximale und die durchschnittliche Ausbreitungsfläche angezeigt. Die durchschnittliche Ausbreitungsfläche berechnet sich aus den summierten Ausbreitungsflächen dividiert durch deren Anzahl.

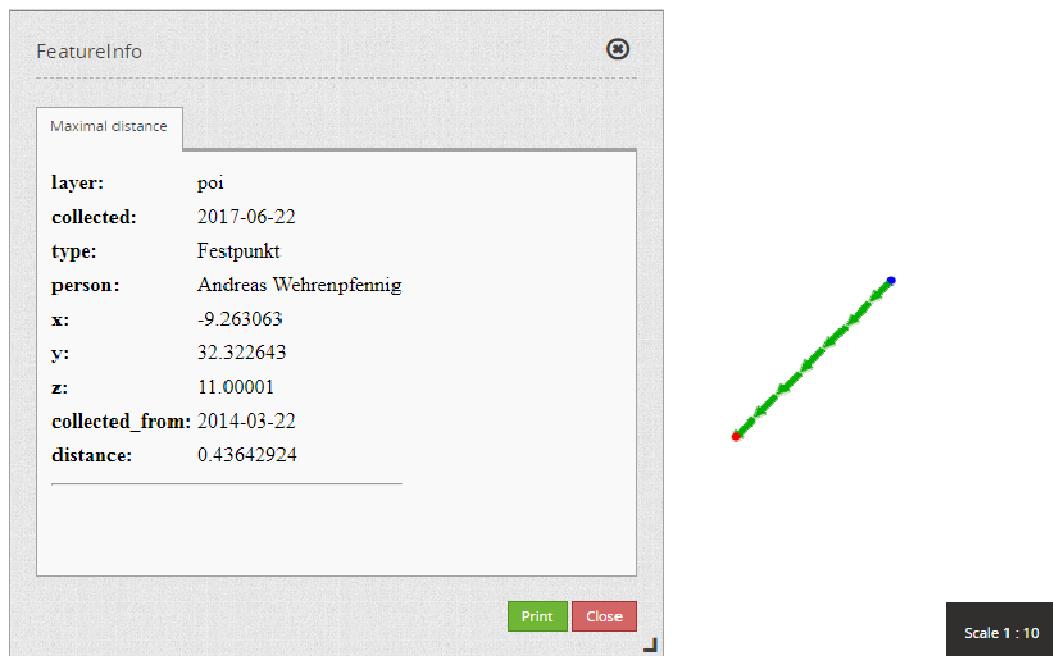


Abb. 7.2: Maximale Ausbreitung eines gemessenen Punktes

## 7. Nutzung des Systems

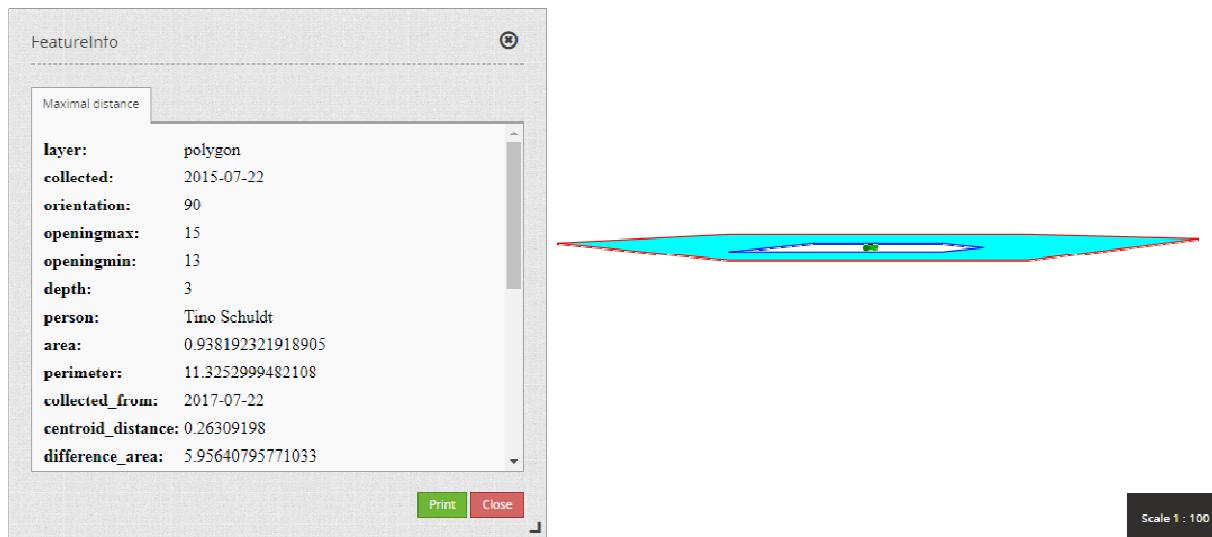


Abb. 7.3: Maximale Ausbreitung der Extensionsklüfte dargestellt als Polygon

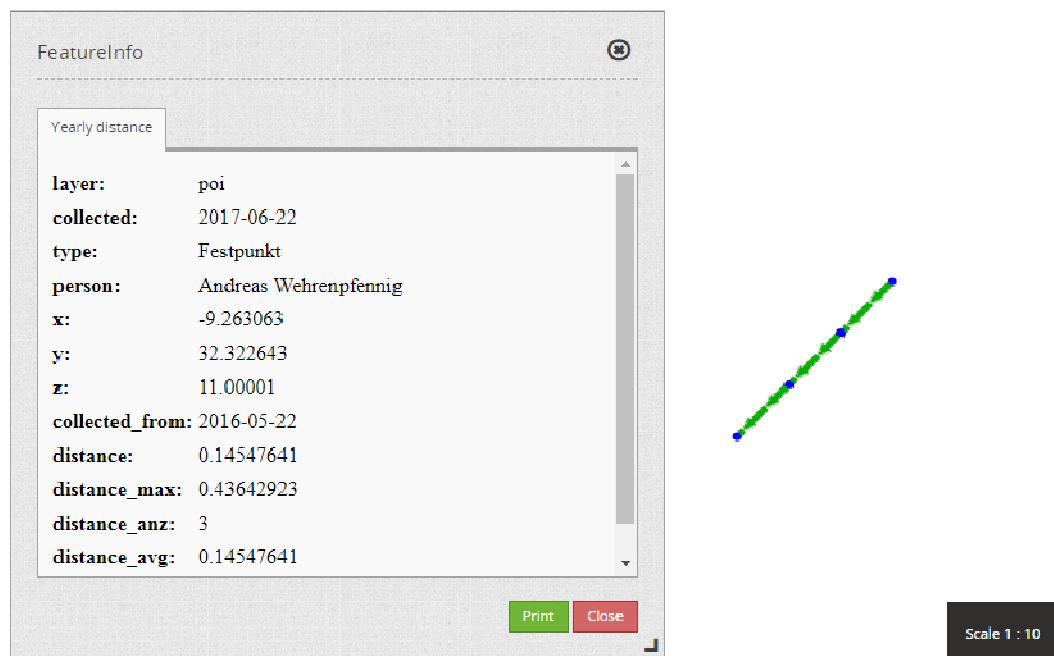


Abb. 7.4: Jährliche Ausbreitung eines gemessenen Punktes

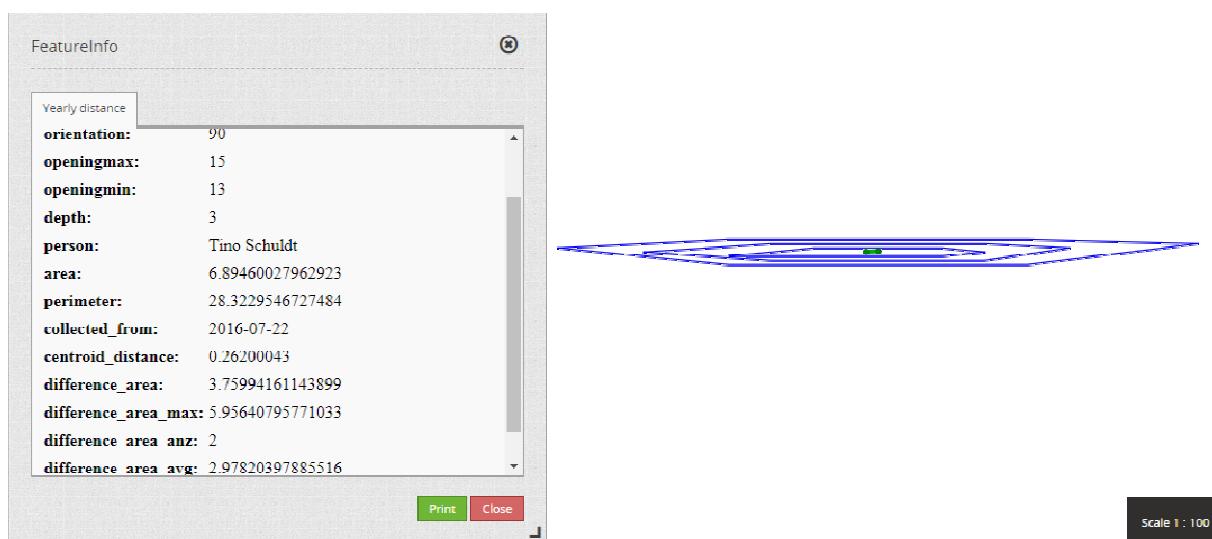


Abb. 7.5: Jährliche Ausbreitung der Extensionsklüfte dargestellt als Polygon

## 7.4 Analyse der Geodaten

Nachdem die Geodaten einzeln ausgelesen werden können, wäre es bei einer Vielzahl von Geodaten schwierig diese auszulesen und zu vergleichen. Hierfür wurde eine Webseite gebaut, die eine Analyse über alle Datensätze ermöglicht. Diese Analyse erfolgt in der Web-Mapping-Anwendung als eingebaute Webseite im Sidepane „Analysis“. Im Kapitel 6.7 wurden bereits die grundlegenden Funktionen dieser Webseite erläutert.

Zuerst werden alle Punkte gesucht, welche einen bestimmten maximalen Distanzwert überschreiten. Dafür wird die Analyse Methode „Maximal“ gewählt und im ersten Textfeld der Abb. 7.6 beispielsweise 0.01 m angegeben. Anschließend wird auf „search“ geklickt. Nun werden alle Punkte in einer Liste dargestellt, die eine größere Ausbreitungsdistanz haben als der angegebene Wert. Mit einem Mausklick auf die ID wird der Punkt mit einer roten Pinna del in der Web-Mapping-Anwendung dargestellt.

Als Nächstes wird nach Polygonen gesucht. Hierfür wird im zweiten Textfeld der Abb. 7.7 der Wert 0.01 m<sup>2</sup> eingegeben und auf „search“ geklickt. Als Resultat werden dann alle Polygone angezeigt, die eine größere Ausbreitungsfläche haben, als der eingegebene Wert in m<sup>2</sup>. Durch einen Mausklick auf die ID wird das Polygon vergrößert dargestellt.

Die nächste Analysemethode ist die jährliche Ausbreitung, womit im ungefähren Zeitraum von einem Jahr alle Punkte und Polygone ausgewertet werden. Hierbei wird die Methode „Yearly“ ausgewählt. Auch hier wird zunächst im ersten Textfeld der Abb. 7.8 ein Wert von 0.01 m eingetragen und mit „search“ bestätigt. In der Ausgabetafel werden nun alle jährlichen Distanzen der Punkte mit dem dazugehörigen Zeitraum angezeigt, die größer als der eingegebene Wert sind. Ein Klick auf die ID stellt den letzten Punkt des Zeitbereichs dar.

Die Analyse der Polygone erfolgt auch hier im zweiten Textfeld der Abb. 7.9. Es wird der Wert 0.01 m<sup>2</sup> eingegeben und der danebenstehende Button geklickt. Die Liste enthält nun alle Polygone, deren Ausbreitungsfläche größer als der eingetragene Wert ist.

Die letzte Analysemethode wird mit „Average yearly“ gewählt und beschreibt die durchschnittliche jährliche Ausbreitung. Diese Methode kann genutzt werden, um herauszufinden, welche Punkte und Polygone sich im Durchschnitt am weitesten bewegen. Im ersten Textfeld der Abb. 7.10 wird nach einem durchschnittlichen Wert von 0.01 m gesucht. Die Ergebnisliste enthält alle durchschnittlichen Distanzen der Punkte, die größer als dieser Wert sind. In das zweite Textfeld der Abb. 7.11 wird ebenfalls der Wert 0.01 m<sup>2</sup> eingetragen und danach gesucht. Angezeigt werden nun alle Polygone, die eine größere durchschnittliche Ausbreitungsfläche haben als der eingetragene Wert.

## 7. Nutzung des Systems

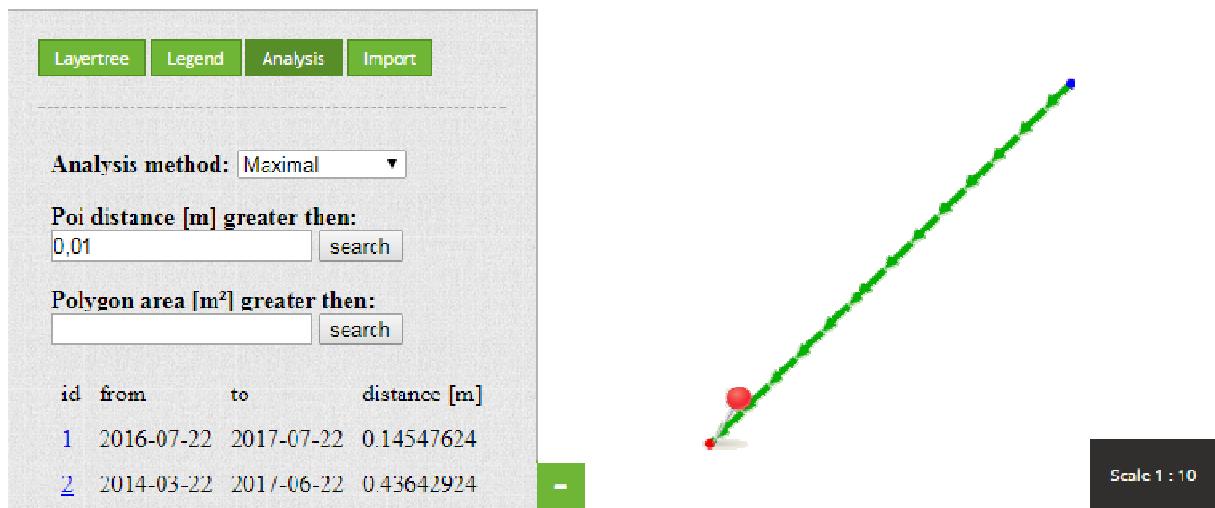


Abb. 7.6: Analyse über die maximale Ausbreitung von Punkten



Abb. 7.7: Analyse über die maximale Ausbreitung von Polygonen

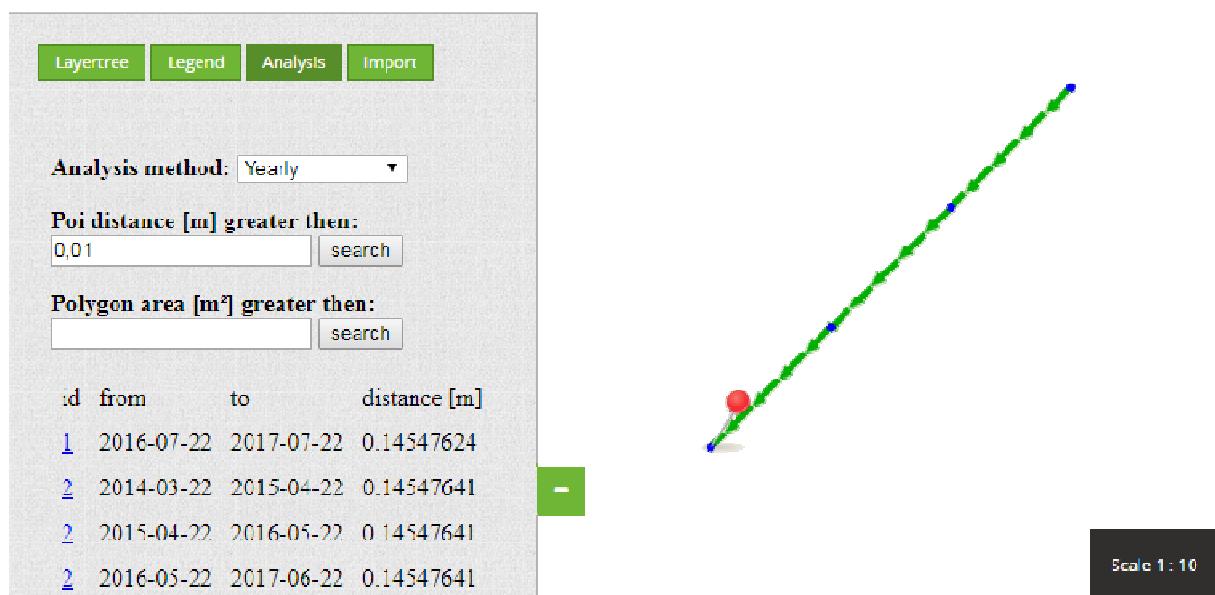


Abb. 7.8: Analyse über die jährliche Ausbreitung von Punkten

## 7. Nutzung des Systems



Abb. 7.9: Analyse über die jährliche Ausbreitung von Polygonen



Abb. 7.10: Analyse über die durchschnittliche Ausbreitung von Punkten



Abb. 7.11: Analyse über die durchschnittliche Ausbreitung von Polygonen

## **7.5 Nutzung mit lokaler GIS-Software**

Nachdem die Analyse in der Web-Mapping-Anwendung erfolgte, kann auch lokale GIS-Software über die entsprechenden WFS-Schnittstellen des Systems zur Analyse genutzt werden. Nach den Anforderungen wird hierbei die freie GIS-Software QGIS eingesetzt.

Zuerst wird ein neuer WFS-Layer über die Menüleiste in QGIS hinzugefügt. Anschließend wird eine neue Verbindung gewählt und im nächsten Fenster eine der folgenden URLs zum WFS-Mapfile eingegeben:

- [http://192.168.174.149/cgi-bin/mapserv?map=mapserver/ef\\_all.map](http://192.168.174.149/cgi-bin/mapserv?map=mapserver/ef_all.map)&
- [http://192.168.174.149/cgi-bin/mapserv?map=mapserver/ef\\_max.map](http://192.168.174.149/cgi-bin/mapserv?map=mapserver/ef_max.map)&
- [http://192.168.174.149/cgi-bin/mapserv?map=mapserver/ef\\_year.map](http://192.168.174.149/cgi-bin/mapserv?map=mapserver/ef_year.map)&

Danach wird sich mit dem WFS verbunden und alle dargestellten Layer ausgewählt. Zum Schluss werden alle Layer durch „Hinzufügen“ hinzugefügt. Diese Geodaten sind nun in QGIS importiert und können für weitere lokale Analysen genutzt werden. Durch die Attributabelle sind alle Informationen als Tabellenform ersichtlich. Wie bei der Web-Mapping-Anwendung sind auch hier die Attribute der einzelnen Features abfragbar.

## **7.6 Bearbeiten der importieren Daten**

Für das nachträgliche Bearbeiten der importierten Daten gibt es verschiedene Möglichkeiten. Die erste Möglichkeit beschreibt den Einsatz eines WFS-T mit der lokalen GIS-Software QGIS. Dadurch können die Geodaten für die Punkte und Polygone in der Attributabelle verändert und gespeichert werden. Eine andere Möglichkeit ist der Einsatz von Administrations-tools, die direkt auf die Datenbank zugreifen. Hierbei wird die Open-Source-Software pgAdmin3 eingesetzt.

Um den WFS-T Dienst von TinyOWS in Anspruch zu nehmen, muss die URL zu TinyOWS gewählt und für die Authentifizierung Benutzernamen und Passwort angegeben werden.

- URL: <http://192.168.174.149/cgi-bin/wfs-t/tinyows>
- Benutzername: tinyuser
- Passwort: tinypassword

Anschließend werden die beiden Layer ausgewählt und hinzugefügt. Alle Punkte und Polygone werden nun dargestellt und können mit der Attributabelle ausgelesen werden. Bei einer Veränderung der Geometrie bzw. der Sachdaten in der Attributabelle werden diese auch auf dem Server verändert. Hierbei können Änderungen durch das Bearbeiten einzelner Felder in der Attributabelle oder durch das Verändern der Geometrie, zum Beispiel durch Verschiebung der Geodaten, vorgenommen werden.

## 8 Ergebnisse

Mit den erstellten Shell-Skripten war die Einrichtung eines webbasierten Geoinformationssystems zur Überwachung und Analyse der Extensionsklüfte einfach und schnell umsetzbar. Die zentrale Konfigurationsdatei ist schnell an die jeweilige Projektumgebung unter Debian angepasst und befindet sich im Ordner „config“ des Projektes. Die Ordnerstrukturen des Projektes befinden sich im Anhang A – Ordnerstruktur.

Eine vordefinierte Anwendung wird in YAML-Format erstellt und danach durch Konsolenbefehle in die Mapbender3 Datenbank eingelesen. Dies erleichtert das Einrichten eigener Anwendungen, ohne, dass Anwendungen manuell in der grafischen Oberfläche zusammengeklickt werden müssen. Der Import von zwei HTML-Elementen klappte nicht problemlos, so dass ein zweites HTML-Element für den Import noch zusätzlich in der grafischen Oberfläche erstellt werden musste. Dies ist in wenigen Schritten erreicht und durch eine zusätzliche Berechtigung ist das Element nur für ausgewählte angemeldete Nutzer verfügbar. Hierbei können auch verschiedene Benutzer mit dieser Berechtigung erstellt werden. Das Importieren der Beispieldaten durch die Webseite ist sehr schnell erledigt. Nur eine Aufbereitung der entsprechenden Daten vorher als CSV-Format ist notwendig. Nachträgliches Verändern der Daten ist entweder durch Zugriff auf die Datenbank oder durch Nutzung des WFS-T möglich. Die Darstellung der importierten Geodaten erfolgt durch eine jährliche oder maximale Ausbreitungsansicht. Je nach Art sind auch die ausgelesenen Informationen unterschiedlich aufbereitet. So wird bei der jährlichen Ausbreitung zusätzlich eine durchschnittliche Ausbreitung berechnet, während bei einer maximalen Ausbreitung nur zwei Datensätze einer Geometrie verglichen werden. Die Darstellung der Punkte durch kleine Kreise ist bei einem kleinen Maßstab gut sichtbar. Hingegen werden Polygone bei kleinem Maßstab nicht angezeigt und erscheinen unsichtbar. Erst beim Vergrößern des Kartenausschnittes wurden diese angezeigt. Um dieses Problem zu umgehen, wurde ein weiterer Layer, der den Schwerpunkt der Polygone als grünen Punkt darstellt, hinzugefügt.

Die webbasierte Analysefunktion bietet eine Suche nach der größten Ausbreitung von Punkten und Polygonen. Durch Angabe eines Schwellwertes werden alle Ausbreitungsdistanzen bzw. Ausbreitungsflächen gesucht, die größer als dieser Schwellwert sind und als Liste dargestellt. Dadurch können sehr einfach alle Punkte und Polygone gefunden werden, bei dem die Ausbreitung einen bestimmten Wert überschreiten.

Ein Vorteil der webbasierten Geoinformationssysteme sind die Standards WMS und WFS. Durch den öffentlichen WFS können die Daten direkt in einer lokalen GIS-Software für weitere Analysen genutzt werden. Das Importieren der WFS-Layer mit der lokalen GIS-Software QGIS ist sehr einfach und verlief ohne Probleme. Es können dieselben Mapfiles als Quelle genutzt werden, die auch als WMS-Quelle dienten.

## 9 Zusammenfassung und Ausblick

Im abschließenden Kapitel werden noch einmal die Ergebnisse der Arbeit zusammengefasst und ein Ausblick für weitere Verbesserungen oder Erweiterungen auf das Projekt erstellt.

### 9.1 Zusammenfassung

Ein Web-GIS besteht aus einer Drei-Schichten-Architektur, bestehend aus Datenhaltungsschicht, Fachlogikschicht und Präsentationsschicht. Daraus ergeben sich die lose gekoppelten Softwarekomponenten für die Datenbank, Kartenserver und Web-Mapping-Anwendung. Im ersten Teil dieser Arbeit wurde nach freien Softwareprodukten für den Einsatz eines Web-GIS nach dieser Drei-Schichten-Architektur gesucht. Geprüft wurden die Softwarekomponenten hinsichtlich der Funktionalitäten für die Speicherung und weiteren Analysefunktionen für 3D-Geodaten. Hierbei stellt sich heraus, dass viele Softwareprodukte ähnlich aufgebaut sind und damit auch im Nachhinein problemlos ausgetauscht werden können. Die Untersuchungen von PostgreSQL/PostGIS, MySQL Spatial und SQLite/SpatialLite zeigten nur wenige Unterschiede hinsichtlich dieser Funktionalitäten. Ähnlich verhielt es sich bei der Analyse der Kartenserver. Technisch sind alle Kartenserver geeignet und unterscheiden sich im Wesentlichen nur durch die eingesetzten Technologien auf dem Betriebssystem. Bei der Analyse von Web-Mapping-Anwendung und Frameworks stellte sich der enorme Aufwand für den Einsatz von Frameworks heraus. Eigene Anwendungen müssen selbst nach eigenen Wünschen programmiert werden. Hingegen bieten Web-Mapping-Anwendungen bereits alles für einen schnellen Aufbau. Die Wahl fiel schließlich auf die freien Softwarekomponenten PostgreSQL als Datenbanksystem mit der Erweiterung PostGIS für viele weitere wichtige Funktionen im Umgang mit Geodaten, Mapserver und TinyOWS für die leicht konfigurierbaren Kartenserver und WMS-/WFS-Dienste, Mapbender als bereits fertige Web-Mapping-Anwendung und Debian als Server-Betriebssystem. Damit sind alle Softwarekomponenten für die Einrichtung eines Web-GIS unter Berücksichtigung der FOSS-Produkte und Unabhängigkeit einzelner Komponenten ausgewählt.

Der anschließende Schritt befasste sich mit dem Konzept des Systems. Hierfür wurde sich an die serviceorientierte Architektur, für die lose Koppelung zwischen Komponenten und standardisierten Austausch der Geodaten, orientiert. Mit den ausgewählten Softwarekomponenten wurde ein Gesamtsystem konzipiert, das sowohl alle standardisierten Schnittstellen (WMS, WFS, WFS-T) sowie eine eigene Schnittstelle für den Datenimport bereitstellt und mit anderen Softwarekomponenten untereinander kommuniziert. Das dafür geplante Datenbankmodell musste die Speicherung von 4D-Koordinaten unterstützen. Die Datenbank von PostgreSQL mit der PostGIS-Erweiterung unterstützt dafür weitere Datentypen „POINTZ“ und „POLYGONZ“ für die Speicherung von 3D-Koordinaten. Die vierte zusätzliche zeitliche Komponente wird mithilfe eines weiteren Feldes für den Zeitstempel (Aufnahmezeitpunkt)

und einer eindeutigen ID für den Datensatz umgesetzt. Dieser Ansatz realisiert mehrere Datensätze mit gleicher ID und unterschiedlich vielen Zeitstempeln. Im weiteren Konzept des Systems beinhaltet die Web-Mapping-Anwendung zwei Nutzergruppen mit unterschiedlichen Funktionalitäten. Die erste Nutzergruppe „User“ stellt umfangreiche Funktionen zur allgemeinen Nutzung der Daten dar. Darüber hinaus stellt die zweite Nutzergruppe „Administrator“ weitere interne Funktionen zur Anpassung oder Import von Daten zur Verfügung. Dieser Import der Daten erfolgt mit einer eigenen Importschnittstelle in das System. Realisiert wurde dies mit Dateien im CSV-Format, wobei diese Daten vorher aufbereitet werden müssen. Außerdem müssen alle Geodaten in das Koordinatenreferenzsystem WGS84 transformiert werden. Die CSV-Daten werden dann durch die eigene Webseite importiert.

Ein weiteres Gebiet der Arbeit befasste sich mit den Analysen der Geodaten. Die erforderliche räumliche Analyse zur Darstellung sowie zur Berechnung von Distanzen und Flächen erfolgten durch die PostGIS-Funktionen der Datenbank. Diese Funktionen können diverse räumliche Berechnungen durchführen und neue Geometrieobjekte erzeugen. Dadurch konnten alle Informationen berechnet und in der Web-Mapping-Anwendung dargestellt werden. Diese Darstellung erfolgte auf zwei verschiedenen Wegen. Ein Weg beschreibt die maximale Ausbreitung. Dort wird von einem Punkt oder Polygon nur der früheste und späteste Datensatz dargestellt und alle Informationen diesbezüglich berechnet. Der andere Weg beschreibt die jährliche Ausbreitung von einem Punkt oder Polygon, bei dem jedes Jahr der späteste Datensatz genommen und den anderen Jahren gegenübergestellt wird. Somit sind Berechnungen wie die durchschnittliche Ausbreitung möglich und kann für weitere Trendanalysen genutzt werden.

Der letzte Punkt befasst sich mit der Einrichtung aller Softwarekomponenten. Um diesen Schritt so einfach wie möglich zu halten, wurden diverse Shell-Skripte erstellt, die das System unter der neusten Debian 9.2 Version installieren und einrichten. Mit einem Startskript wird die komplette Installation und Einrichtung des Systems vorgenommen. Dafür muss nur die zentrale Konfigurationsdatei vor der Installation an die Projektumgebung angepasst werden. Beispielsweise durch Änderung der IP-Adresse, den Pfaden und Paketnamen. Durch diese Konfigurationsdatei sind auch zukünftige Updates und Veränderungen des Betriebssystems kein Problem für eine schnelle Einrichtung des Systems. Nach der Installation und Einrichtung mussten noch zwei Kleinigkeiten in der Web-Mapping-Anwendung erledigt werden. Zum Einen musste ein HTML-Element für den Import hinzugefügt werden. Zum Anderen mussten alle WMS-Quellen aktualisiert werden, damit die Legende und alle Metadaten der Layer angezeigt werden können. Danach standen alle Funktionen der Web-Mapping-Anwendung zur Verfügung. Die Nutzung der Daten erfolgt in der Web-Mapping-Anwendung oder in QGIS mit der WFS-Schnittstelle. Eine weitere Webseite ermöglicht die Suche nach Punkten und Polygonen, welche einen bestimmten Ausbreitungswert überschritten. Dadurch

ist die Überwachung und Analyse von allen Extensionsklüften in der Web-Mapping-Anwendung möglich.

Zusammenfassend ist zu sagen, dass alle Ziele unter Berücksichtigung der Anforderungen erreicht worden sind. Es wurde ein Web-GIS für die Überwachung und Analyse von Extensionsklüften geschaffen, die auf Basis von FOSS-Produkten entwickelt wurde. Das System ist durch diverse Skripte und der Importschnittstelle schnell einsatzbereit.

### **9.2 Fazit**

Gezeigt wurde die schnelle und einfache Einrichtung eines Web-GIS für die Überwachung und Analyse von Extensionsklüften. Begonnen bei der Auswahl geeigneter Softwarekomponenten bis hin zur Entwicklung eines Konzeptes zur Einrichtung eines Web-GIS für Extensionsklüfte. Das System wird mit wenigen Handgriffen ausgerollt und ist dadurch auch für Nutzer ohne tiefgreifende Kenntnisse nutzbar. Lediglich die Konfigurationsdatei muss vor Beginn der Einrichtung angepasst werden.

Das Ziel ein Web-GIS für die Überwachung und Analyse von Extensionsklüften zu entwickeln ist damit erreicht worden. Entstanden ist ein System, das sehr einfach einzurichten sowie zu nutzen ist.

### **9.3 Ausblick**

Das System kann beliebig erweitert und optimiert werden. Zwar ist das System bereits lauffähig, dennoch können weitere Ideen und Ansätze das System erweitern und optimieren. Die folgenden Ansätze und Ideen können in Zukunft am System sinnvoll weiterentwickelt werden.

#### **Web-Mapping-Anwendung für mobile Geräte**

Heutzutage sind Nutzer mit Geräten wie Smartphones oder Tablets im Internet unterwegs. Optimal wäre eine speziell an mobile Geräte angepasste Anwendung. Der Mapbender bietet dafür ein spezielles Mobile Template an, das dafür genutzt werden kann. Dieses Template sollte dann als zweite Anwendung eingerichtet werden. Damit diese Anwendung während der Installation eingerichtet wird, müsste ein neues Skript für die Erstellung der Anwendung geschrieben und in das bestehende Installationsskript integriert werden. Hierbei kann sich an das Skript „create\_mapbender\_application.sh“ orientiert werden.

#### **Automatisierte Importquellen für die Geodaten**

Ein weiterer Punkt betrifft das automatisierte Importieren der Geodatenaus Sensoren. Hierfür könnte eine weitere Schnittstelle implementiert werden, welche die Geodaten automatisiert einliest.

### **Verbesserungen an der Importschnittstelle mit Personennamen**

Ein zusätzlicher Punkt betrifft die CSV-Dateien für die Importschnittstelle. Darin werden die Personennamen durch eine ID eingetragen, nachdem die Personen ins System importiert wurden. Dies ist in der Praxis nicht sehr praktikabel, daher sollten die Personennamen beim importieren stattdessen auf der Webseite, beispielsweise durch eine Auswahlbox, auswählbar sein. Dadurch muss die ID der Person nicht umständlich in den CSV-Dateien eingetragen werden und es wird stattdessen auf der Webseite der Personename gewählt.

### **Analyse der Tendentwicklung**

Der letzte Punkt ist die Analyse von Tendentwicklungen der Extensionsklüfte. Beispielsweise könnte noch überprüft werden, wie weit sich in der Zukunft die Extensionsklüfte ausbreiten könnten anhand der durchschnittlichen jährlichen Ausbreitung. Hierbei wären weiterführende Überlegungen für die Berechnung und Darstellung im Web-GIS notwendig. Andernfalls könnte lokale GIS-Software für diesen Einsatz genutzt werden.

## Glossar

<b>API</b>	Das „Application Programming Interface“ ist eine Schnittstelle für den Zugriff auf die eigenen Funktionen.
<b>CGI</b>	Das „Common Gateway Interface“ ist eine Schnittstelle für den Datenaustausch zwischen Webserver und dritter Software.
<b>CSW</b>	Catalogue Service for the Web führt die Suche nach räumlichen Datensätzen und Diensten aus.
<b>Domain</b>	Ist ein weltweit eindeutiger Name einer Webpräsenz.
<b>Dokument</b>	Ein Text, Bild o. ä. das sich im Binärkode auf einem Dateisystem abgelegt befindet.
<b>FOSS</b>	Hybrider Begriff für freie Software und Open-Source-Software.
<b>Georeferenzieren</b>	Zuweisung von raumbezogenen Informationen auf einer Erdab-bildung.
<b>HTML</b>	Textbasierte Auszeichnungssprache zur Strukturierung digitaler Dokumente.
<b>Hyperlinks</b>	Querverweis zu einem anderen elektronischen Dokument oder Webseite.
<b>Inlineframe</b>	HTML-Element zur Strukturierung von Webseiten, um andere Inhalte auf die Webseite einzubetten.
<b>Java</b>	Objektorientierte und plattformunabhängige Programmierspra-che.
<b>Layer</b>	Der Layer (die Ebene) bezeichnet eine zusammengehörige Zu-sammenfassung von gleichen raumbezogenen Objekten.
<b>Referenzsystem</b>	Koordinatensysteme die sich mit einem Datum auf die reale Welt beziehen.
<b>SDL</b>	Styled Layer Descriptor beschreibt ein XML-Schema zur Darstel-lung von Kartenebenen, der nach OGC definiert wurde.
<b>Servlet</b>	Bezeichnet das Nutzen von Java-Klassen innerhalb des Webser-vers.
<b>Shell</b>	Text- bzw. kommandobasierte Benutzung eines Computersys-tems.
<b>Skript</b>	Zusammenführung von einzelnen Befehlen in eine ausführbare Datei.
<b>SQL</b>	Bezeichnet eine Datenbanksprache für relationale Datenbanken.

<b>SRID</b>	Der „Spatial Reference Identifier“ identifiziert räumliche Referenzsysteme.
<b>URL</b>	Bezeichnung für Netzwerkressourcen.
<b>Webpräsenz</b>	Bezeichnung für einen Internetauftritt, einen virtuellen Platz im Internet, an dem sich mehrere Webseiten und Dokumente befinden.
<b>WCS</b>	Web Coverage Service definiert eine standardisierte Schnittstelle um interoperablen Zugriff auf räumliche Daten zu ermöglichen.
<b>Webseite</b>	Eine Seite die mit Angabe einer URL aufgerufen werden kann.
<b>WFS</b>	Web Feature Service, der den direkten Zugriff auf räumliche Datenobjekte ermöglicht.
<b>WMS</b>	Web Map Service, der gerenderte Karten aus räumlichen Daten liefert.
<b>WMTS</b>	Web Map Tile Service, welcher vorberechnete Kartenkacheln (Tiles) liefert.
<b>WPS</b>	Web Processing Service führt Berechnungen auf räumlichen Daten aus.
<b>XML</b>	Die „Extensible Markup Language“ ist eine Auszeichnungssprache zur Darstellung von hierarchisch strukturierten Daten.
<b>YAML</b>	Vereinfachte Auszeichnungssprache zur Datenserialisierung mit Anlehnung an XML.

## Quellenverzeichnis

- [Aus17] Auswärtiges Amt (02.2017): "Marokko". URL: <https://www.auswaertiges-amt.de/DE/Aussenpolitik/Laender/Laenderinfos/01-Laender/Marokko.html> (Abgerufen am 08.11.2017).
- [Bau11] Baudson, Christoph; Stamm, Peter; Feilner, Markus (02.2011): "Die Web-GIS-Frameworks OpenLayers, Mapfish und Mapbender". URL: <http://www.linux-magazin.de/Ausgaben/2011/02/Webmapping> (Abgerufen am 08.11.2017).
- [Bel17] Beltov, Martin (07.2017): "Die besten Linux-Server Distributionen in 2017". URL: <https://sensorstechforum.com/de/best-linux-server-distributions-2017/> (Abgerufen am 08.11.2017).
- [Col15] Boualla, Othmane; Mehdi, Khalid; Zourarah, Bendahou (12.2015): "Collapse dolines susceptibility mapping in Doukkala Abda (Western Morocco) by using GIS matrix method (GMM)", Springer International Publishing Switzerland 2015
- [Dal17] Dallüge, Uwe (06.2005): "PostGIS 2.0 Tutorial (Grundlagen)". URL: [http://static.openjump.de/media/documentation/extern/PostGIS\\_Tutorial\\_20.pdf](http://static.openjump.de/media/documentation/extern/PostGIS_Tutorial_20.pdf) (Abgerufen am 08.11.2017).
- [Gdi13] Geodateninfrastruktur Bayern am Landesamt für Digitalisierung, Breitband und Vermessung (11.2013): "Einrichten von Web Feature Services - Leitfaden". URL: [http://www.gdi.bayern.de/file/pdf/922/Leitfaden\\_WebFeatureServices\\_1\\_0.pdf](http://www.gdi.bayern.de/file/pdf/922/Leitfaden_WebFeatureServices_1_0.pdf) (Abgerufen am 08.11.2017).
- [Gdi15] Koordinierungsstelle GDI-NI beim Landesamt für Geoinformation und Landesvermessung (04.2015): "Formulieren von Anfragen an einen WebMapService (WMS)". URL: [http://www.geodaten.niedersachsen.de/download/26325/Formulieren\\_von\\_Anfragen\\_an\\_einen\\_WMS.pdf](http://www.geodaten.niedersachsen.de/download/26325/Formulieren_von_Anfragen_an_einen_WMS.pdf) (Abgerufen am 08.11.2017).
- [Gdi17] Geodateninfrastruktur Deutschland: "Geodaten aus ganz Europa nutzen". URL: <http://www.geoportal.de/DE/GDI-DE/INSPIRE/inspire.html?lang=de> (Abgerufen am 08.11.2017).
- [Geo17a] GeoServer: "WMS reference". URL: <http://docs.geoserver.org/stable/en/user/services/wms/reference.html> (Abgerufen am 08.11.2017).
- [Geo17b] GeoServer: "WFS reference". URL: <http://docs.geoserver.org/stable/en/user/services/wfs/reference.html> (Abgerufen am 08.11.2017).
- [Geo17c] GeoExt 3: "JavaScript Toolkit for Rich Web Mapping Applications". URL: <https://geoext.github.io/geoext3/> (Abgerufen am 08.11.2017).
- [Git17a] GitHub: "GeoServer". URL: <https://github.com/geoserver/geoserver> (Abgerufen am 08.11.2017).
- [Git17b] GitHub: "GeoExt". URL: <https://github.com/geoext/geoext3/blob/master/LICENSE-FAQ.md> (Abgerufen am 08.11.2017).

- [Kad09] Kadner, Daniel (12.2009): "Konzeption und Integration der Open Geospatial Consortium Web Services in die Open Service Process Platform". URL: [https://tu-dresden.de/bu/umwelt/geo/geoinformatik/ressourcen/dateien/dateien/kadner\\_download/GB.pdf?lang=de](https://tu-dresden.de/bu/umwelt/geo/geoinformatik/ressourcen/dateien/dateien/kadner_download/GB.pdf?lang=de) (Abgerufen am 08.11.2017).
- [Koc09] Koch, Thomas (2009): "Arbeiten mit Geodaten in Oracle und MySQL - Gemeinsamkeiten und Unterschiede". URL: [https://www.doag.org/formes/pubfiles/1387569/docs/Konferenz/2009/vortraege/Oracle%20&%20SUN/149-2009-K-SUN-Koch-Arbeiten\\_mit\\_Geodaten\\_in\\_Oracle\\_und\\_MySQL\\_M.pdf](https://www.doag.org/formes/pubfiles/1387569/docs/Konferenz/2009/vortraege/Oracle%20&%20SUN/149-2009-K-SUN-Koch-Arbeiten_mit_Geodaten_in_Oracle_und_MySQL_M.pdf) (Abgerufen am 08.11.2017).
- [Ler10] Lernhelper (2010): "Königreich Marokko". URL: <https://www.lernhelper.de/schuelerlexikon/geografie/artikel/koenigreich-marokko> (Abgerufen am 28.08.2017).
- [Lgb17a] Landesvermessung und Geobasisinformation Brandenburg (LGB): "Web Map Service (WMS)". URL: [https://www.geobasis-bb.de/geodaten/geo\\_wms\\_uebersicht.htm](https://www.geobasis-bb.de/geodaten/geo_wms_uebersicht.htm) (Abgerufen am 08.11.2017).
- [Lgb17b] Landesvermessung und Geobasisinformation Brandenburg (LGB): "Web Feature Service (WFS)". URL: [https://www.geobasis-bb.de/geodaten/geo\\_wfs\\_uebersicht.htm](https://www.geobasis-bb.de/geodaten/geo_wfs_uebersicht.htm) (Abgerufen am 08.11.2017).
- [Map17a] MapFish: "MapFish". URL: <http://www.mapfish.org> (Abgerufen am 08.11.2017).
- [Map17b] Mapbender3: "MapbenderCoreBundle". URL: <http://doc.mapbender3.org/de/bundles/Mapbender/CoreBundle/index.html> (Abgerufen am 08.11.2017).
- [Mel10] Melzer, Ingo (03.2010): "Service-orientierte Architekturen mit Web Services: Konzepte - Standards - Praxis", Springer-Verlag, 4. Auflage
- [Mit08] Mitchell, Tyler; Emde, Astrid (01.2008): "Web-Mapping mit Open-Source-GIS-Tools", O'Reilly Verlag GmbH & Co. KG, 1. Auflage, ISBN: 978-3-897-21723-2
- [Mys17] MySQL: "Spatial Function Reference". URL: <https://dev.mysql.com/doc/refman/5.7/en/spatial-function-reference.html> (Abgerufen am 10.11.2017).
- [Osg17a] OSGeo-Live: "GeoServer". URL: [https://live.osgeo.org/de/overview/geoserver\\_overview.html](https://live.osgeo.org/de/overview/geoserver_overview.html) (Abgerufen am 08.11.2017).
- [Osg17b] OSGeo-Live: "deegree". URL: [https://live.osgeo.org/de/overview/deegree\\_overview.html](https://live.osgeo.org/de/overview/deegree_overview.html) (Abgerufen am 08.11.2017).
- [Osg17c] OSGeo-Live: "Leaflet". URL: [https://live.osgeo.org/de/overview/leaflet\\_overview.html](https://live.osgeo.org/de/overview/leaflet_overview.html) (Abgerufen am 08.11.2017).
- [Pro01] Professur für Geodäsie und Geoinformatik (GG) AUF Universität Rostock: "Geoinformatik-Service". URL: <http://www.geoinformatik.uni-rostock.de/lexikon.asp> (Abgerufen am 08.11.2017).

- [Sch12] Schröder, Simon (05.2012): "GeoServer oder MapServer? Ein Vergleich von Leistungsfähigkeit und -umfang beider Kartenserver", Masterarbeit an der Universität Osnabrück
- [Spe00a] Spektrum Akademischer Verlag, Heidelberg (2000): "Extension". URL: <http://www.spektrum.de/lexikon/geowissenschaften/extension/4478> (Abgerufen am 08.11.2017).
- [Spe00b] Spektrum Akademischer Verlag, Heidelberg (2000): "Klüfte". URL: <http://www.spektrum.de/lexikon/geowissenschaften/kluefte/8471> (Abgerufen am 08.11.2017).
- [Spe01] Spektrum Akademischer Verlag, Heidelberg (2001): "Geoinformationssystem". URL: <http://www.spektrum.de/lexikon/kartographie-geomatik/geoinformationssystem/1771> (Abgerufen am 08.11.2017).
- [Uiz15] Umwelt und Informationstechnologie Zentrum GmbH (11.2015): "WebGIS - Allgemeine Systemarchitektur". URL: <http://uizentrum.de/webgis-allgemeine-systemarchitektur/> (Abgerufen am 08.11.2017).
- [Vre02] Vretanos, Panagiotis (05.2002): "Web Feature Service Implementation Specification". URL: [http://portal.opengeospatial.org/files/?artifact\\_id=7176](http://portal.opengeospatial.org/files/?artifact_id=7176) (Abgerufen am 08.11.2017).
- [Wik17a] Seite „MySQL“. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 1. September 2017. URL: <https://de.wikipedia.org/w/index.php?title=MySQL&oldid=168673158> (Abgerufen am 08.11.2017).
- [Wik17b] Seite „SQLite“. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 25. Oktober 2017. URL: <https://de.wikipedia.org/w/index.php?title=SQLite&oldid=170317729> (Abgerufen am 08.11.2017).
- [Wik17c] Seite „MapServer“. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 27. August 2017. URL: <https://de.wikipedia.org/w/index.php?title=MapServer&oldid=168523982> (Abgerufen am 08.11.2017).
- [Wik17d] Seite „Geoobjekt“. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 26. Januar 2017. URL: <https://de.wikipedia.org/w/index.php?title=Geoobjekt&oldid=162021200> (Abgerufen am 19.11.2017).
- [Wri16] Wright, Adam (03.2016): "Mit PostGIS Geoinformationen in PostgreSQL integrieren". URL: <http://www.it-administrator.de/themen/storage/fachartikel/204043.html> (Abgerufen am 08.11.2017).
- [Wup17] Wuppertaler Umwelt- und Geodatenportal: "Über deegree". URL: <http://geoportal.wuppertal.de/legal.html> (Abgerufen am 08.11.2017).

## Abbildungsverzeichnis

Abb. 1.1: Anwendung der Web-Mapping-Anwendung mit den Nutzern .....	4
Abb. 2.1: Karte von Marokko (übernommen von [Col15]).....	5
Abb. 2.2: Drei verschiedene Arten der Bruchausbreitung (übernommen von [Spe00b]).....	6
Abb. 2.3: Extensionskluft in Bezug seiner drei Hauptspannungen (übernommen von [Spe00b]) .....	6
Abb. 3.1: Drei-Schichten-Architektur eines GIS (übernommen von [Uiz15]) .....	11
Abb. 3.2: Kommunikation zwischen Client und Server (übernommen von [Uiz15]) .....	11
Abb. 3.3: Sequenzdiagramm eines WMS-Service mit dem Request GetMap (vgl. [Kad09], S.16) .....	14
Abb. 3.4: Allgemeiner Aufbau einer Anfrage (übernommen von [Gdi15], S.2).....	14
Abb. 3.5: Layer erlaubt die Abfrage von zusätzlichen Sachinformationen .....	15
Abb. 4.1: Grafische Benutzeroberfläche von GeoServer .....	29
Abb. 4.2: Aufbau eines Mapfiles .....	31
Abb. 4.3: Grafische Benutzeroberfläche von deegree .....	34
Abb. 4.4: Grafische Oberfläche von Mapbender3 .....	41
Abb. 5.1: Systemarchitektur als Komponentendiagramm .....	45
Abb. 5.2: Funktionsweise der WMS-Requests im Sequenzdiagramm .....	46
Abb. 5.3: Funktionsweise der WFS-Requests im Sequenzdiagramm.....	47
Abb. 5.4: PostgreSQL-Datenbankmodell .....	48
Abb. 5.5: Verschiebung eines Punktes .....	50
Abb. 5.6: Ausbreitung eines Polygons .....	50
Abb. 6.1: Anordnung der Elemente auf dem Bildschirm .....	60
Abb. 6.2: Datenimport durch eine Webseite .....	61
Abb. 6.3: Analyse der Geodaten durch die eigene Webseite .....	62
Abb. 7.1: Darstellung der Anwendung .....	63
Abb. 7.2: Maximale Ausbreitung eines gemessenen Punktes .....	65
Abb. 7.3: Maximale Ausbreitung der Extensionsklüfte dargestellt als Polygon .....	66
Abb. 7.4: Jährliche Ausbreitung eines gemessenen Punktes .....	66
Abb. 7.5: Jährliche Ausbreitung der Extensionsklüfte dargestellt als Polygon .....	66
Abb. 7.6: Analyse über die maximale Ausbreitung von Punkten .....	68
Abb. 7.7: Analyse über die maximale Ausbreitung von Polygonen.....	68
Abb. 7.8: Analyse über die jährliche Ausbreitung von Punkten .....	68
Abb. 7.9: Analyse über die jährliche Ausbreitung von Polygonen.....	69
Abb. 7.10: Analyse über die durchschnittliche Ausbreitung von Punkten .....	69
Abb. 7.11: Analyse über die durchschnittliche Ausbreitung von Polygonen.....	69

## Tabellenverzeichnis

Tabelle 3.1: WMS-Parameter für den Request GetCapabilities (vgl. [Gdi15]; [Geo17a]) .....	15
Tabelle 3.2: WMS-Parameter für den Request GetMap (vgl. [Gdi15]; [Geo17a]) .....	16
Tabelle 3.3: WMS-Parameter für den Request GetFeatureInfo (vgl. [Gdi15]; [Geo17a]) .....	17
Tabelle 3.4: Alle nach OGC definierten WFS-Operationen (vgl. [Gdi13], S.6-7) .....	18
Tabelle 3.5: WFS-Parameter für den Request GetCapabilities (vgl. [Gdi13], S.8; [Geo17b])	19
Tabelle 3.6: WFS-Parameter für den Request DescribeFeatureType (vgl. [Gdi13], S.8; [Geo17b]) .....	19
Tabelle 3.7: WFS-Parameter für den Request GetFeature (vgl. [Gdi13], S.8-9; [Geo17b]) ..	20
Tabelle 3.8: WFS-Parameter für den Request Transaction (vgl. [Vre02], S.71).....	21
Tabelle 3.9: WFS-Parameter für den Request LockFeature (vgl. [Vre02], S.69-70) .....	21
Tabelle 4.1: Bewertungskriterien der Softwarekomponenten .....	23
Tabelle 4.2: PostGIS Geometrien (vgl. [Dal05], S.10) .....	24
Tabelle 5.1: Felder der Sichten .....	51
Tabelle 6.1: Elemente der Web-Mapping-Anwendung .....	60

## Anhang

### **Anhang A – Ordnerstruktur**

Die Ordnerstruktur des Projektes ist wie folgt:

```
> ~/work
  > config
    > install_config.sh
  > data
    > analysis.php
    > import.php
    > mapbender_screenshot.png
  > download
    > mapbender3-starter-3.0.6.3.tar.gz
    > tinyows-v.1.1.1.tar.gz
  > import_data
    > import_person.sh
    > import_poi.sh
    > import_polygon.sh
    > import_type.sh
    > testdata_person.txt
    > testdata_poi.txt
    > testdata_polygon.txt
    > testdata_type.txt
  > log
  > script
    > config_basic.sh
    > config_mapbender3.sh
    > config_postgis.sh
    > config_postgres.sh
    > config_tinyows.sh
    > create_analysis.sh
    > create_import.sh
    > create_mapbender_application.sh
    > create_mapfiles.sh
    > create_tables.sh
    > create_views.sh
    > create_wms_getfeatureinfo.sh
    > head_admin.sh
    > install_basic.sh
    > install_mapbender3.sh
    > install_mapserver.sh
    > install_postgres.sh
    > install_tinyows.sh
    > load_function.sh
  > tmp
  > install.sh
```

Das Webverzeichnis, in dem sich Mapbender3, die HTML-Dateien für die GetFeatureInfo-Abfrage und die Webseiten für den Import und die Analyse befinden, ist wie folgt strukturiert:

```
> /var/www/
  > html
    > analysis
      > config.php
      > index.php
    > import
      > config.php
      > index.php
      > tmp
    > mapserver
      > getfeatureinfo
        > geodata
          > poi.html
          > poi_max.html
          > poi_year.html
          > polygon.html
          > polygon_max.html
          > polygon_year.html
          > blank.html
        > tmp
      > index.html
  > mapbender3
    > app
      > config
        > applications
        > parameters.yml
```

Das CGI-Verzeichnis mit allen Dateien für den WMS / WFS ist wie folgt aufgebaut:

```
> /usr/lib/cgi-bin/
  > mapserver
    > ef_all.map
    > ef_max.map
    > ef_year.map
  > wfs-t
    > tinyows
  > mapserv
```

### ***Anhang B – Inhalt der CD***

- Pflichtenheft im PDF/A-Format
- Masterarbeit im PDF/A-Format
- Anwenderdokumentation im PDF/A-Format
- Projektdateien im ZIP-Format
- Beispieldaten im ZIP-Format

## **Eidesstattliche Erklärung**

Hiermit versichere ich, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Neubrandenburg, den

*Unterschrift*