

Hochschule Neubrandenburg
Studiengang Master Geoinformatik und Geodäsie

Entwicklung von automatisierten Mechanismen zur nutzerbezogenen Einrichtung eines web-basierten Geoinformationssystems

Master-Thesis

20. August 2013

vorgelegt von

Name: Milbratz Vorname: Markus

Akademischer Grad: Bachelor of Engineering (B. Eng.)

Geb. am: 08.06.1984 in: Neubrandenburg

Zum Erlangen des akademischen Grades
"Master of Engineering (M. Eng.)"

Erstprüfer: Prof. Dr.-Ing. Andreas Wehrenpfennig

Zweitprüfer: Prof. Dr.-Ing. Ernst Heil

Bearbeitungszeitraum: 15. März 2013 - 20. August 2013

urn:nbn:de:gbv:519-thesis2013-0111-7

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Neubrandenburg, den 20. August 2013

Danksagung

Der Weg bis zu dem Verfassen dieser Master-Thesis war sehr lang und gekennzeichnet durch viel Arbeit. In dieser Zeit, die auch häufig mit der Vernachlässigung des privaten Lebens verbunden war, wurde ich durch meinen engsten Angehörigen stets motiviert. Insbesondere bedanke ich mich an dieser Stellen bei meiner Familie und meiner Lebensgefährtin, die mich durch das fünfjährige Studium begleitet haben und mir stets deren volle Unterstützung gegeben haben. Ebenfalls bedanke ich mich bei Prof. Dr.-Ing. Andreas Wehrenpfennig für die zuverlässige und sehr persönliche Betreuung seitens der Hochschule.

Kurzfassung

Geodaten sind ein fester Bestandteil der gegenwärtigen digitalen Welt. Die Analyse räumlicher Daten ermöglicht es, aussagekräftige Schlussfolgerungen aus komplexen ökonomischen und ökologischen Prozessen zu ziehen. Bevor aber solche Analysen durchgeführt werden können, sind zunächst Daten zu erheben. Ein Weg dies zu realisieren, liegt in der Verwendung web-basierter Technologien und Softwareprodukten, mit denen Geodaten verwaltet und verteilt werden können. Solche Umgebungen werden auch als web-basierte Geoinformationssysteme (Web-GIS) bezeichnet. Der Aufbau eines Web-GIS unter Berücksichtigung nutzerspezifischer Vorgaben ist häufig mit der Entwicklung individueller Konzepte und einem erhöhten Implementierungsaufwand verbunden. Im Folgenden wird ein Konzept gezeigt, um ein web-basiertes GIS in wenigen Schritten einzurichten. Die Einrichtung setzt den Fokus auf zwei Aspekte. Zum einen sind die bereits erwähnten nutzerbezogenen Anforderungen zu berücksichtigen, um das System funktional und inhaltlich anzupassen. Zum anderen sind die Prozesse, die für die Integration und Konfiguration der beteiligten Systemkomponenten notwendig sind, weitgehend zu automatisieren. Als Ergebnis steht eine Softwarelösung zur Verfügung, mit der mit geringem Aufwand eine maßgeschneiderte web-basierte Umgebung für die Verarbeitung von Geodaten geschaffen werden kann.

Abstract

Geodata are an integral part of the current digital world. The analysis of spatial data allows to draw meaningful conclusions out of complex economic and environmental processes. Before the analysis can be done, data needs to be collected. One way to realize this, lies in the usage of web-based technology und softwareproducts for managing and deploying geodata. Such environments are also known as web-based geographic information systems, short Web-GIS. Building up a Web-GIS by regarding to user-specific requirements is frequently connected to the development of individual concepts. In the following there is shown a concept to set up a Web-GIS in a few steps. The setup focusses on two aspects. On the one hand the user-specific requisitions are taken into consideration to modify the functionalities and content of the system. On the other hand the processes for the integration and configuration of the involved system components will be largely automated. The desired result is to deploy a software solution that allows to set up a customized web-based environment for processing geodata.

Inhaltsverzeichnis

1 Einführung	1
1.1 Motivation	1
1.2 Ziel	2
1.3 Anforderungen	2
2 Grundlagen und Begriffsklärung	3
2.1 Geodaten	3
2.1.1 Vektordaten	3
2.1.2 Rasterdaten	3
2.2 Geoinformationssysteme	4
2.3 Web-Kartografie	4
2.3.1 Statische Anwendungen	4
2.3.2 Dynamische Anwendungen	4
2.3.3 Kartenserver	5
2.3.4 Geodatenserver	5
2.3.5 Web-Mapping-Frameworks	5
2.4 Normen und Standards des Geoinformationswesens	6
2.4.1 Bedeutung von Normen und Standards	6
2.4.2 Wichtige Organisationen und Institute	6
2.4.3 Web Map Service - WMS	8
2.4.4 Web Feature Service - WFS	10
3 Analyse web-basierter Kartenportale	14
3.1 Karten-Viewer	14
3.1.1 Google Maps	14
3.1.2 GAIA-MV	15
3.2 Kartenportale mit Editierfunktionen	16
3.2.1 Open Street Map	16
3.2.2 Klarschiff-HRO	18
3.3 Schlussfolgerungen für das Design des Portals	19
4 Analyse und Bewertung ausgewählter Softwarekomponenten	20
4.1 Bewertungskriterien	20
4.2 Geodatenbanken	21
4.2.1 PostgreSQL/PostGIS	21
4.2.2 SQLite / SpatialLite	22
4.2.3 MySQL Spatial	23
4.2.4 Auswertung der Geodatenbanken	24
4.3 Kartenserver	25
4.3.1 MapServer/TinyOWS	25
4.3.2 degree	28
4.3.3 Geoserver	30
4.3.4 Auswertung der Analyse der Kartenserver:	32
4.4 Web-Mapping-Werkzeuge	33
4.4.1 OpenLayers	33
4.4.2 GeoExt	34
4.4.3 MapFish	35
4.4.4 Mapbender	37
4.4.5 Auswertung der Analyse der Web-Mapping-Frameworks:	38
4.4.6 Betriebssystem des Servers	38

5	Konzeptioneller Entwurf des Systems	40
5.1	Grundgedanke der Architektur - Service-orientierte Architektur (SOA)	40
5.2	Komponenten	40
5.3	Kommunikation und Schnittstellen	41
5.4	Konfigurationswerkzeuge	43
5.4.1	Datenbankkonfiguration	43
5.4.2	Konfiguration von Mapserver und TinyOWS	44
5.4.3	Konfiguration der Web-Mapping-Anwendung	44
5.5	Abläufe und Struktur der Web-Mapping-Anwendung	45
5.6	Einrichtung des Web-GIS	48
6	Implementierung des Systems	49
6.1	Grundlegende Herangehensweise	49
6.2	Vorbemerkung zu der Implementierung	49
6.3	Installation des Betriebssystems	49
6.4	Installationsskript	50
6.5	Konfigurationsskripte	51
6.5.1	Datenbankkonfiguration	51
6.5.2	Konfiguration von MapServer/TinyOWS	52
6.5.3	Konfiguration der Web-Mapping-Anwendung	53
6.6	Skript für die Ersteinrichtung des Web-GIS	54
6.7	Zusätzliche Skripte für Anpassungen des Web-GIS	55
6.8	Web-Mapping-Anwendung	55
6.9	Konfiguration des Webservers	57
6.10	Aufbau und Inhalte des Installationspakets	57
7	Exemplarische Umsetzungen eines Web-GIS	58
7.1	Problemstellungen für die Umsetzungen des Web-GIS	58
7.2	Installation des Betriebssystems	59
7.3	Übertragen des Installationspakets	62
7.4	Konfiguration und Ausführung des Setup-Programms	62
7.5	Konfiguration des Webservers	65
7.6	Ergebnisse der Umsetzungen	65
7.7	Testen der Installationen	66
8	Ergebnisse	69
9	Fazit	71
10	Ausblick	72
	Abbildungsverzeichnis	74
	Tabellenverzeichnis	74
	Inhaltsverzeichnis der beigelegten CD	75
	Literaturverzeichnis	76

1 Einführung

1.1 Motivation

Die Bedeutung von Geodaten hat in der heutigen IT-Landschaft einen hohen Stellenwert eingenommen. Die geographische Referenzierung von Informationen findet in verschiedensten Gebieten Anwendung. Bereiche wie die Navigation, Routenplanungen und das Umweltmonitoring basieren auf einem räumlichen Bezug der Informationen. Die stetige Entwicklung räumlicher Informationen hat eine Vielzahl an Standards und Normen hervorgebracht und führt zu einer wachsenden Vernetzung und Interoperabilität der Geodaten. Ein zentraler Aspekt, der im Rahmen dieser Entwicklung unumgänglich ist, ist der Begriff der Geodateninfrastruktur (GDI). Dabei handelt es sich um ein interoperables Netzwerk von verteilten Geodatenbeständen. Normen und Standards gewährleisten, dass die unterschiedlichen Datenbestände untereinander genutzt werden können. Die GDI sorgt zudem dafür, dass verstärkt auf web-basierte Komponenten gesetzt wird. Auch wenn lokale Geoinformationssysteme (GIS) aufgrund ihrer Rechenleistung und Funktionsumfang weiterhin ein essentieller Bestandteil sind, um Geodaten zu verwalten und zu analysieren, stellen web-basierte GIS eine Alternative dar, um Geodaten zu verwerten. Der größte Vorteil solcher web-basierter Systeme liegt darin, dass diese über einen Webbrowser genutzt werden können und somit eine lokale Installation von GIS-Software überflüssig machen. Auch wenn sich dies auf die Leistung des Systems nachteilig auswirkt, da die Rechenleistung von Webbrowsern wesentlich geringer ist, können mittels Web-GIS Geodaten visualisiert und rudimentär analysiert werden. Dieser Trend hält weiter an und hat eine Reihe von freien GIS-Softwareprodukten hervorgebracht.

Eine Schwierigkeit web-basierter GIS ist deren Implementierung. Auch wenn die reine Integration der Systemkomponenten (z. Bsp. Datenbank, Kartenserver, Web-Mapping-Anwendung) eher einfach zu realisieren ist, liegt die meiste Arbeit in der Systemkonfiguration und der Entwicklung einer maßgeschneiderten Web-Mapping-Anwendung. Um dies auch für ungeübte Nutzer zu ermöglichen, ist ein Konzept zu entwickeln, welches diese Aspekte möglichst erleichtert und einen schnellen Einsatz des Web-GIS realisiert.

Ein anderer bedeutender Aspekt dieser Arbeit liegt in den Interaktionsmöglichkeiten des Web-GIS. Zusätzlich zu der Navigation durch die Geodatenbestände (Betrachtung der Karten und Extraktion von Objektinformationen) soll das Hinzufügen neuer Informationen ermöglicht werden. Crowdsourcing-Plattformen wie "Open Street Map" bauen auf die Aktivität der Nutzer auf. Jeder hat die Möglichkeit Daten zu dem Bestand hinzuzufügen und so die Bandbreite der verfügbaren Informationen zu erweitern. Dieses Prinzip wird beispielsweise in Kommunen eingesetzt. Derartige Portale dienen der Erfassung von Problematiken innerhalb der Gemeinde. Zu diesen zählen z. Bsp. illegale Mülldeponien, Schlaglöcher und Gefahrenzonen im Verkehrsnetz. Die Bürger haben die Möglichkeit diese Gefahren über ein interaktives Kartenportal einzutragen. Diese Informationen werden dann von der Verwaltung (Entsorgungsunternehmen, Straßenverkehrsamt) genutzt, um auf Basis der Verortung das Problem zu beseitigen und so zur Sicherheit in der Gemeinde beizutragen. Es ist aber zu beachten, dass Crowdsourcing auch gewisse Gefahren birgt. Ohne eine Überprüfung der eingetragenen Daten kann ein solches Portal missbraucht werden und es können wesentlich falsche Informationen eingetragen werden. Durch Sicherheitsmechanismen, wie einer Authentifizierung, kann ein neuer Eintrag zurückverfolgt werden und der Nutzer, der das Portal entgegen der Richtlinien nutzt, gesperrt werden. Dennoch kann selbst unter Berücksichtigung sämtlicher Sicherheits- und Überprüfungsmechanismen eine missbräuchliche Nutzung nicht ausgeschlossen werden. Portale wie "Open Street Map" setzen daher auf die Ehrlichkeit und Kompetenz der Nutzer.

Unter Berücksichtigung der zwei zentralen Punkte, einer Infrastruktur für die Geodaten und einer Web-Mapping-Anwendung für die Darstellung und Erfassung von georeferenzierten Informationen, wird ein System entworfen und implementiert, welches durch einen hohen Grad an Flexibilität und Komfort den Einsatz optimiert. Insbesondere die Einrichtung des Systems ist so einfach wie möglich zu gestalten, damit auch ungeübte Nutzer das System aufsetzen können.

1.2 Ziel

Das Ziel ist die Konzeption und Realisierung eines web-basierten Geoinformationssystems. Der Fokus liegt auf dem universellen Einsatz und der nutzerbezogenen automatisierten Einrichtung des Systems. Das Web-GIS soll es ermöglichen, unabhängig von den verwendeten Geodaten ein System zur Verfügung zu stellen, mit dem über vorgefertigten Komponenten Daten integriert werden können und die Gestaltung der Benutzeroberfläche der Web-Mapping-Anwendung durch wenig Aufwand den eigenen Bedürfnissen angepasst werden kann. Sämtliche Anpassungen sollen ohne eine zusätzliche Programmierung durch den Nutzer durchgeführt werden können. Um welche Themen es sich schließlich bei den Geodaten handelt, ist für den Entwurf des Systems weniger relevant, da das gesamte Konzept auf Anpassungsfähigkeit und Flexibilität beruht. Zusammenfassend sind die folgenden Themengebiete Gegenstand der Arbeit:

- Entwicklung eines Konzepts zum Aufbau eines web-basierten Geoinformationssystems (Datenhaltung, Datenbereitstellung, Benutzerschnittstelle in Form einer Web-Mapping-Anwendung)
- Analyse und Auswahl geeigneter Software-Komponenten für den Aufbau des Systems
- Entwicklung von Werkzeugen zur automatisierten Implementierung der Softwarekomponenten
- Entwicklung von Werkzeugen und Methoden für die Integration von Geodaten
- Entwicklung einer Web-Mapping-Anwendung für die Darstellung und Erfassung von Geodaten
- Erstellung von Methoden für die Modifizierung der Benutzeroberfläche der Web-Mapping-Anwendung
- Exemplarische Erstellung eines Web-GIS auf Basis der erarbeiteten Methoden und Werkzeuge

1.3 Anforderungen

Die Realisierung ist an diverse Bedingungen geknüpft, die bei der Umsetzung einzuhalten sind. Zu diesen Rahmenbedingungen zählen:

- Bei der Wahl der Software-Komponenten sind nur freie Produkte zugelassen. Ob deren Quellcode offen ist (Open Source), ist nicht relevant.
- Die Kosten und der Aufwand bei der Einrichtung des Systems ist gering zu halten.
- Die Architektur des gesamten Systems ist flexibel zu gestalten und die einzelnen Komponenten sind lose zu koppeln.
- Der Datenaustausch ist über standardisierte Schnittstellen zu realisieren, um einen hohen Grad an Interoperabilität zu externen Geodatenbeständen und anderen GIS zu gewährleisten.
- Bei der Auswahl der Systemkomponenten ist zu berücksichtigen, dass vorkonfigurierte Muster verwendet werden können, um den Einrichtungsaufwand zu optimieren, indem z. Bsp. Muster verwendet werden.
- Das gesamte System soll möglichst leicht-gewichtet sein, um die benötigten Ressourcen zu minimieren.
- Die Anpassung der Benutzeroberfläche der Web-Mapping-Anwendung ist mit so wenig Aufwand wie möglich zu realisieren.

2 Grundlagen und Begriffsklärung

In diesem Kapitel werden die grundlegenden Begriffe, die für das Konzept und dessen Umsetzung des Web-GIS von Bedeutung sind, näher erläutert. Ziel ist es ein Verständnis für die Thematik zu erschließen und Schwerpunkte hervorzuheben.

2.1 Geodaten

Der Grundpfeiler eines GIS stellen Geodaten dar. Geodaten sind definiert als Informationen mit einem räumlichen Bezug, auch bezeichnet als Georeferenzierung. Geodaten unterteilen sich in zwei Kategorien:

1. **Geobasisdaten:** Diese Teilmenge der Geodaten umfasst Interessen-neutrale Geoinformationen. Dazu zählt primär die Topographie (Landschaft) der Erdoberfläche. Diese Kategorie stellt die Datenbasis für eine Vielzahl von Geo-Anwendungen dar und bildet die Grundlage für die zweite Kategorie, den Geofachdaten. (Vgl. [Pro13], Stichwort: Geobasisdaten)
2. **Geofachdaten:** Diese Teilmenge besitzt stets einen Bezug zu einer fach-spezifischen Thematik. Die Daten werden stets mit Bezug zu einer bestimmten Fachdisziplin erhoben. (Vgl. [Pro13], Stichwort: Geofachdaten)

Im Rahmen des Web-GIS werden beide Typen Anwendung finden. Die Geobasisdaten dienen als räumliche Orientierung und werden in Form von Hintergrundkarten (z. Bsp. Satellitenbilder) genutzt. Die Fachdaten umfassen jene Aspekte, die für den Provider des Web-GIS relevant sind. Ein anderer Punkt, der zu untersuchen ist, sind die vorherrschenden Formate von Geodaten.

2.1.1 Vektordaten

Vektordaten basieren auf der Beschreibung von räumlichen Objekten hinsichtlich ihrer Geometrie (Koordinaten) und nicht-räumlichen Eigenschaften. Bei den Geometrietypen handelt es sich um Punkte, Linien und Flächen. Bei der Verbindung der Geometrien mit spezifischen Informationen spricht man von linearen Eigenschaften der Objekte. Diese Art der Modellierung findet vor allem in Datenbanken Anwendung. Der große Vorteil von Vektordaten liegt in der Flexibilität der Modellierung. Je nach Anforderung können sehr einfache bis hoch komplexe Sachverhalt abgebildet werden. Ein Nachteil liegt in der Datenerhebung. Abhängig von der Komplexität des Datenmodells kann das Füllen des Modells mit Daten sehr aufwendig sein. Schließlich sollte man einen Kompromiss eingehen. Das Modell soll zum einen einen genügend großen Informationsraum abbilden und zum anderen die Möglichkeit bieten, Daten in einem angemessenen zeitlichen Rahmen einzutragen. (Vgl. [Pro13], Stichwort: Vektordaten)

2.1.2 Rasterdaten

Die Modellierung von Rasterdaten ist im Gegensatz zu der von Vektordaten eingeschränkt. Sie erfolgt im ersten Schritt durch eine äquidistante Diskretisierung der Informationen und im zweiten Schritt dessen Quantisierung. Rasterdaten bauen sich in Form einer Matrix auf. Jedes Element dieser Matrix (z. Bsp. Pixel) ist durch dessen Zeile und Spalte in seiner Position spezifiziert. Der Vorgang der Quantisierung ordnet jedem Element einen numerischen Wert zu. Dieser Wert kann z. Bsp. ein Messwert oder ein Farbwert sein. In der Geoinformatik werden Rasterdaten vor allem in der Photogrammetrie und Fernerkundung genutzt, um die Erdoberfläche hinsichtlich einer bestimmten Eigenschaft zu beschreiben. Der Nachteil von Rasterdaten liegt in der eingeschränkten

Modellierung. Dafür können aber mittels dieses Datentyps sehr große Areale effizient abgebildet werden. (Vgl. [Pro13], Stichwort: Rasterdaten)

2.2 Geoinformationssysteme

Geo-Informationssysteme sind definiert als computergestützte Systeme (Hardware, Programme, Daten), mit denen räumliche Informationen in digitaler Form erfasst, gespeichert, verwaltet, modelliert und untersucht werden können. Die Verarbeitung der Daten erfolgt grafisch und alphanumerisch. GIS definieren somit diverse Technologien und Softwareprodukte für die Arbeit mit Geodaten. Die stetig wachsende Komplexität räumlicher Informationen führt zu der Spezialisierung von GIS. Es existieren diverse fachspezifische Systeme. Zu diesen Systemen zählen u. a. Umwelt-Informationssysteme (UIS), Raum-Informationssysteme (RIS) und weitere Fach-Informationssysteme (FIS). (Vgl. [Pro13], Stichwort: Geo-Informationssystem)

2.3 Web-Kartografie

Der Begriff der Web-Kartografie bezeichnet Methoden der Bereitstellung von geografischen Informationen über das Internet. Die Bereitstellung der Informationen erfolgt über sog. Kartenanwendungen, die auch als Web-Mapping-Anwendungen bezeichnet werden. Derartige Anwendungen sind die Schnittstelle zum Nutzer. Was eine Web-Mapping-Anwendung an Leistungsumfang zur Verfügung stellt, hängt in erster Linie von dem Typ ab. Generell unterscheidet man zwischen zwei Arten von Web-Mapping-Anwendungen. (Vgl. [Tyl08], S. 8)

2.3.1 Statische Anwendungen

Statische Anwendungen bilden die Anfänge der Web-Kartografie. In der Regel bestand eine solche Anwendung aus einer Webseite in der diverse statische Karten in Form von Bildern, die in einem Rasterformat auf der Server abgelegt sind, eingebunden wurden. Über diesen Weg war es zwar möglich auf räumliche Informationen zuzugreifen, die fehlende Interaktivität schränkte den Nutzer aber stark ein, insbesondere hinsichtlich der Selektion von spezifischen Informationen zu Objekten der Karten. Als die Entwicklung web-basierter Technologien voran schritt und immer mehr Werkzeuge für die Erstellung eigener Web-Mapping-Anwendungen verfügbar waren, stellten sich statische Kartenanwendungen als nicht praktikabel heraus. Die neuen dynamischen Anwendungen eröffneten völlig neue Wege, mit web-basierten Kartenanwendungen zu arbeiten. (Vgl. [Tyl08], S. 8)

2.3.2 Dynamische Anwendungen

Der Unterschied zu den statischen Anwendungen liegt in der Interaktivität. Während der Anwender bei statischen Anwendungen nur den Kartenausschnitt betrachten und ggf. herunterladen konnte, kann man bei dynamischen Kartenanwendungen wie in einem GIS arbeiten. Zu einigen der Interaktionsmöglichkeiten zählen u. a.:

- Verändern des Kartenausschnitts durch Verschieben und Zoomen
- Ein- und Ausblenden von thematischen Karten (Layer)
- Abfragen von Informationen zu Objekten innerhalb der Karte
- Flächen- und Streckenmessung

Um die neuen Interaktionen zu realisieren, genügte es nicht mehr eine simple Webseite anzulegen und die Karten auf dem Server zu hinterlegen. Die Tatsache, dass der Kartenausschnitt bei jeder

Veränderung durch den Nutzer neu generiert werden muss, benötigt eine neue Architektur (Abbildung 1) und zusätzliche Softwarekomponenten. Zu diesen Komponenten zählen ein Programm für die Generierung der Karten (Kartenserver), eine Quelle für die Geodaten (Geodatenserver) und eine dynamische Web-Mapping-Anwendung. (Vgl. [Tyl08], S. 9)

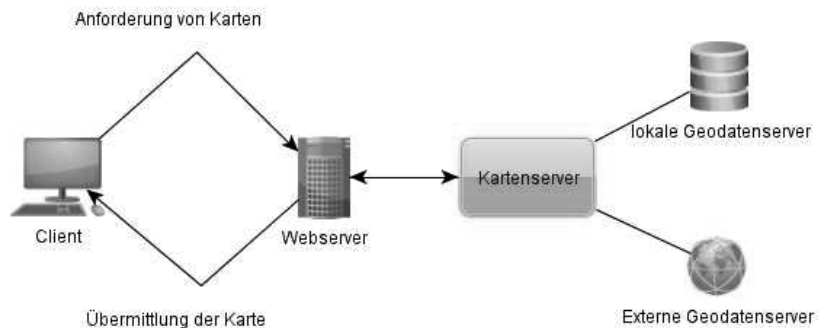


Abbildung 1: Architektur dynamischer Web-Kartografie

Im Folgenden werden die in der Abbildung 1 aufgezeigten Komponenten in ihrem Zweck und deren Arbeitsweise erläutert.

2.3.3 Kartenserver

Kartenserver entstanden aus der Notwendigkeit heraus, Karten dynamisch zu erzeugen. Ein Kartenserver ist ein Kartografieprogramm, das in der Regel im Kontext eines Webservers läuft. Der Webserver agiert als Vermittler. Er nimmt die Kartenanfrage entgegen und reicht diese an den Kartenserver weiter. Der Kartenserver erstellt auf Basis der Anforderung und dessen Parameter die Karte und gibt das Ergebnis über den Webserver an den Client zurück. Im Zuge der Standardisierungen von Geoinformationen (siehe Kapitel 2.4) sind diverse Festlegungen getroffen worden, was ein Kartenserver zu leisten hat und wie die Schnittstellen für die Kommunikation zu gestalten sind. Der "Web Map Service (WMS)" und "Web Feature Service (WFS)", die eine besondere Bedeutung im Rahmen des Web-GIS einnehmen, sind mittlerweile allgegenwärtig und stellen die Basis für den heutigen Austausch von Geodaten dar.

2.3.4 Geodatenserver

Um zu gewährleisten, dass der Kartenserver in der Lage ist, Karten zu erzeugen, werden Geodaten benötigt. Unabhängig von dem Format der Daten (Vektoren, Raster) werden die Daten mittels Geodatenserver verwaltet. Der Geodatenserver kann sowohl lokal implementiert sein, als auch über ein Netzwerk erreichbar sein. Der Trend, dem Geodaten unterliegen, geht in Richtung der Vektor-basierten Formate. Der Vorteil liegt darin, dass diese sehr effizient mittels räumlichen Datenbanksystemen verwaltet werden können. Zudem ist die Vielfalt der Analysefunktionen signifikant höher, im Gegensatz zu den Rasterformaten.

2.3.5 Web-Mapping-Frameworks

Die Benutzerschnittstelle eines Web-GIS wird über Web-Anwendungen realisiert. In Anbetracht der Tatsache, dass die Komplexität von GIS hinsichtlich des Funktionsumfang sehr groß ist, stellt die Erstellung einer Web-Mapping-Anwendung hohe Anforderungen an dessen Implementierung. Web-Mapping-Frameworks sind eine Möglichkeit, Web-Anwendungen mit dem Fokus auf Geodaten zu

programmieren. Die Frameworks bestehen aus einer Vielzahl an vorgefertigten Komponenten. Diese stellen beispielsweise Funktionen, wie Navigations- und Analysewerkzeuge, zur Verfügung und Routinen für den Import von Geodaten. Als Entwickler von Web-Mapping-Anwendungen kann somit der Fokus primär auf das Design und die Abstimmung der Komponenten untereinander gelegt werden. Des Weiteren können sehr komfortabel eine Fülle an unterschiedlichen Geodatenquellen genutzt werden.

Die weltweite Vernetzung der Geodatenbestände und deren Nutzung in verschiedensten Softwareprodukten führte zu der Entwicklung von Normen und Standards, um den Zugriff und die Bereitstellung von Geodaten zu vereinheitlichen. Welche Vorgaben für die Entwicklung relevant sind und welche Auswirkungen sich auf die Architektur ergeben, wird im Folgenden betrachtet. (Vgl. [Tyl08] S. 8 ff.)

2.4 Normen und Standards des Geoinformationswesens

Ein zentraler Aspekt heutiger web-basierter GIS und der zugehörigen Infrastruktur (Kartenserver, Geodatenserver) führte zu einer Entwicklung vieler Standards und Normen, mit dem Ziel die Geodaten-Landschaft interoperabel zu gestalten. Da diese für die eigene Entwicklung des Web-GIS relevant sind, wird ein Überblick über die zu verwendenden Standards und Normen gegeben.

2.4.1 Bedeutung von Normen und Standards

Die vorhergehenden Abschnitte haben bereits gezeigt, dass einheitliche Vorgaben in Form von Normen und Standards ein essentieller Bestandteil des gegenwärtigen Geoinformationswesens sind. Bevor auf die für Geoinformationen relevanten Vorgaben eingegangen wird, ist zuerst der Unterschied zwischen den Begriffen Normen und Standards zu klären.

Der Begriff der Norm ist für Deutschland durch das "Deutsche Institut für Normung (DIN)" definiert worden. Gemäß der "DIN 820-1, 1994" lautet die Definition wie folgt: "Normung ist die planmäßige, durch die interessierten Kreise gemeinschaftlich durchgeführte Vereinheitlichung von materiellen und immateriellen Gegenständen zum Nutzen der Allgemeinheit" [Pro]. Eine Grundvoraussetzung für eine Norm, ist die Ausrichtung nach den aktuellen Stand von Technik und Wissen, um die Vorteile bei der Anwendung der Norm zu stärken. Hinsichtlich der Rechtsverbindlichkeit von Normen ist folgendes zu beachten: Die Anwendung von Normen steht jedem frei, somit besteht prinzipiell kein Zwang sich nach diesen zu richten. Wenn aber in einem Vertrag auf Normen verwiesen wird, so sind diese rechtsverbindlich und einzuhalten.

Standards zeichnen sich dadurch aus, dass materielle und immaterielle Gegenständen durch Institutionen genormt werden. Dies kann im Gegensatz zu Normen unter Ausschluss der Öffentlichkeit erfolgen und dient in der Regel dazu, Entwicklungs- und Produktionsprozesse in Unternehmen zu optimieren.

Grundsätzlich ist die Unterscheidung zwischen diesen zwei Begriffen fließend und ggf. nicht zu treffen. Im angelsächsischen Raum sind die Begriffe Standard und Norm identisch, im Gegensatz zu den o. g. deutschen Definitionen.

Häufig tritt der Fall ein, dass Standards, die sich aufgrund ihrer starken Verbreitung etabliert haben, durch Normungsinstitute, wie der "Internationalen Organisation für Normung (ISO)", in Normen umgewandelt werden. Dieser Prozess trat bei den für die Geoinformatik relevanten Standards ein. Das "Open Geospatial Consortium (OGC)" ist die treibende Kraft und entwickelte eine Vielzahl an Standards, um Technologien und ein Netzwerk für geo-kodierte Informationen zu schaffen. (Vgl. [Pro])

2.4.2 Wichtige Organisationen und Institute

Die wichtigsten Organisationen für Regulierungen geoinformations-spezifischer Themen sind das "Open Geospatial Consortium (OGC)" und die "Internationalen Organisation für Normung (ISO)".

- **Open Geospatial Consortium (OGC)** Im Jahr 1994 gründete sich das OGC. Es ist eine gemeinnützige Organisation mit dem Ziel die weltweite Verbreitung, den Austausch und die Verwaltung von Geodaten für alle Interessengruppen (z. Bsp. Wirtschaft und Forschung) auf einen gemeinsamen Standard zu bringen. Ein Grund für die große Bedeutung der OGC liegt in den Mitgliedern. Mit derzeit 481 aktiven Mitgliedern aus Wirtschaft und Forschung zählen unter anderem: (Vgl. [Ope13a]):
 - ESRI
 - European Space Agency (ESA)
 - Deutsches Zentrum für Luft- und Raumfahrt - DLR
 - GFZ Potsdam
 - OpenGEO
 - Oracle USA
 - Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV)

Diese kleine Auswahl an aktiven Mitgliedern zeigt die Bedeutung der angewandten Standards. Die wichtigsten Veröffentlichungen bilden die "OpenGIS-Implementation Specifications". Dabei handelt es sich um frei verfügbare Implementierungsspezifikationen. Diese beinhalten an erster Stelle Schnittstellendefinitionen. Zudem werden Vorgaben getroffen, die die Darstellung von Informationen (Datenformate), Datenmodellierungen und den Bereich der Metadaten (Beschreibung von Daten) festlegen. Unter Verwendung dieser Vorgaben sind Anwendungen in der Lage mit anderen Applikationen, die ebenfalls diesen Spezifikationen unterliegen, zu kommunizieren und Daten auszutauschen. Um herauszufinden ob eine Anwendung tatsächlich den Spezifikationen entspricht, werden sog. "Konformitäts-Tests" durchgeführt. Ein Softwarehersteller reicht sein Produkt bei der OGC ein und diese testet im ersten Schritt, ob alle Vorgaben der Spezifikation eingehalten wurden. Der zweite Test umfasst die Interoperabilität zu anderen OGC-konformen Produkten. Sind beide Tests erfolgreich abgeschlossen, spricht man von OGC-Konformität (engl. OGC-compliant) und der Hersteller erhält für das Produkt ein Zertifikat, welches bescheinigt, dass das Produkt geltenden Standards entspricht.

Zur Umsetzung des Vorhabens, ein globales interoperables Netz für Geoinformationen zu schaffen, baut das OGC auf gewisse Arbeitsmethoden auf. Die enge Zusammenarbeit mit Entwicklern sorgt für realitätsnahe und umsetzbare Spezifikationen, eine permanente Analyse des aktuellen Standes der Technik und des Marktes ermöglicht schnelle Reaktionen auf gegenwärtige Trends und schließlich führt die Zusammenarbeit mit großen Unternehmen wie ESRI und Oracle zu einer wachsenden Akzeptanz und Bekanntheit der Standards. (Vgl. [Gü05])

- **Internationale Organisation für Normung (ISO)** Um die internationale Anerkennung und Verbreitung der Standards voranzutreiben, wurden eine Vielzahl der OGC-Standards durch die ISO genormt. Für diese Zwecke wurde das "Technische Komitee 211 (TC 211)" ins Leben gerufen. Das TC 211 ist für den Bereich der Geoinformationen zuständig und ist für die formale Umsetzung der Standards in Normen verantwortlich. Die Zusammenarbeit zwischen der OGC und dem TC 211 ist sehr eng und wird u. a. dadurch optimiert, dass diverse Personen in beiden Gremien tätig sind. Neben den technischen Aspekten sind auch viele Begriffe definiert worden, um die Kommunikation zu verbessern. So wurden z. Bsp. der Begriff der Features (Objekte) erarbeitet und Regelungen beim Umgang mit Koordinatenreferenzsystemen getroffen. Die für das Geoinformationswesen relevanten Normen gehören der 191xx-Familie an. Die folgende Liste zeigt einige der wichtigsten Normen:

- **ISO 19107 - Spatial Schema** Beschreibung von Geometrien 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 die Ablage, Abholung, Abfrage und Aktualisierung geo-referenzierter Informationen über SQL

(Vgl. [Int13]) Ein wichtiger Punkt ist die Versionierung der Normen. Da sich im Laufe der Zeit bestimmte Vorgaben als weniger praktikabel erwiesen haben oder neue Aspekte eingearbeitet werden müssen, liegen die Normen in verschiedenen Versionen vor. Die Abwärtskompatibilität ist stets gewährleistet. Im Rahmen der Entwicklung des Web-GIS kommt zwei Normen eine zentrale Bedeutung zu. Der "Web Map Service - WMS" und der "Web Feature Service - WFS" werden im Folgenden in ihren Grundzügen näher vorgestellt.

2.4.3 Web Map Service - WMS

Der Web Map Service - WMS (auch als Web Map Server bezeichnet) ist eine der wichtigsten und am weitesten verbreitetsten OGC-Spezifikationen. Der WMS dient der Visualisierung von Raster- und Vektor-basierten Geodaten in Form von Rasterkarten. Die Spezifikation gibt die Art und Weise vor, wie Geodaten bereitgestellt und abgerufen werden. Diese Schnittstellendefinition basiert auf drei parametrisierten Methoden und gibt an, wie eine Anforderung zu gestalten ist. Die Anforderungen, auch als Requests bezeichnet, erfolgen über HTTP und können direkt als URL in einen Webbrowser eingegeben werden. Die Parameter teilen sich in Pflichtangaben und optionalen Angaben auf. Im folgenden werden die drei Methoden (GetMap, GetCapabilities und GetFeatureInfo) vorgestellt und diverse Parameter in ihrem Zweck erläutert. Den Parametern ist die URL des WMS voran zustellen (z. Bsp. `www.MeinServer.de/MeinWebMapService?`). Des Weiteren sind alle Parameter mit einem "&" miteinander zu verbinden.

- **GetCapabilities** Der erste Request wird genutzt, um einen Überblick über den Leistungsumfang und den Inhalt des entsprechenden WMS zu erhalten. Unter Verwendung eines Internetbrowsers erhält man nach der Anfrage ein XML-Dokument, das u. a. Metadaten (Verantwortliche, Kontaktadressen, verfügbare Koordinatenreferenzsysteme) und spezifische Informationen zu den Karten (Layer) enthält. Der Request ist wie folgt gestaltet (Tabelle 1):

<u>Request Parameter</u>	<u>Erforderl.(E)/Opt.(O)</u>	<u>Beschreibung</u>
REQUEST= GetCapabilities	E	Gibt die Methode des Request an
VERSION= version	O	Angabe der WMS-Version
SERVICE= WMS	E	Angabe des Service-Typs

Tabelle 1: GetCapabilities-Request-Parameter ([Mar11] S. 30)

Unter Verwendung von Programmen für die Verarbeitung von XML-Dokumenten und diversen GIS kann das Capabilities-Dokument in eine besser lesbare Form konvertiert werden.

- **GetMap** Der GetMap-Request beinhaltet als Rückgabe eine Rasterkarte. Gemäß der Tabelle 2 sind die folgenden Parameter zugelassen.

Request Parameter	Erforderl.(E)/Opt.(O)	Beschreibung
REQUEST= GetMap	E	Gibt die Methode des Request an
VERSION= version	E	Angabe der WMS-Version
LAYERS= layer_list	E	Liste der Namen der Layer
SRS=EPSG	E	Angabe des Koordinatenreferenzsystems in Form eines EPSG-Codes ₁
BBOX= minx,miny,maxx,maxy	E	Kartenbox, Werte richten sich nach der Einheit des EPSG-Codes
WIDTH=output_width	E	Breite der Karte in Pixel
HEIGHT=output_height	E	Höhe der Karte in Pixel
STYLES=style_list	E	Stil (grafisch) der Karte
FORMAT=output_format	E	Ausgabeformat der Karte (jpeg, png...)
BGCOLOR=color_value	O	Hintergrundfarbe der Karte
TRANSPARENT=TRUE FALSE	O	Transparenz der Karte festlegen
SLD=sld_url	O	URL zum SLD-Dokument ₂
EXCEPTIONS=exception_format	O	Ausgabeformat der Fehlermeldung

Tabelle 2: GetMap-Request-Parameter ([Mar11] S. 30)

₁ Verzeichnis von Koordinatensystemen in Form numerischer Codes

₂ Styled Layer Descriptor - Standard für die Gestaltung von Kartenelementen auf Basis von XML

- **GetFeatureInfo** Der WMS verfügt über die Fähigkeit Informationen zu Koordinaten zurückzugeben. Dies kann im Kontext einer Web-Mapping-Anwendung genutzt werden, um Abfragen auszuführen. Es können folgende Parameter genutzt werden (Tabelle 3).

Request Parameter	Erforderl.(E)/Opt.(O)	Beschreibung
VERSION=version	E	Request-Version
REQUEST=GetFeatureInfo	E	Name des Request
GetMapRequest-Parameter		Hier sind die entsprechenden GetMap-Parameter zu nutzen.
QUERY_LAYERS=layer_list	E	Liste der abzufragenden Layer
INFO_FORMAT=output_format	E	Ausgabetyp (MIME-Typ) des Request
FEATURE_COUNT=number	O	Anzahl der abzufragenden Features
I=pixel_column	E	X-Koordinate (in Pixel) innerhalb der Karte
J=pixel_row	E	Y-Koordinate (in Pixel) innerhalb der Karte
EXCEPTIONS=exception_format	O	Ausgabeformat bei Fehler

Tabelle 3: GetFeatureInfo-Parameter ([Mar11], S. 31)

Im Rahmen der Web-Mapping-App werden diese Parameter durch unterschiedliche Objekte verwaltet, sodass eine explizite Angabe der Parameter obsolet ist. So werden bei einer Veränderung des Kartenausschnitts die Parameter automatisch aktualisiert und die Anforderung für die neue Karte dementsprechend ausgeführt. (Vgl. [Rei05])

2.4.4 Web Feature Service - WFS

In der Regel reicht für eine Vielzahl der Web-Mapping-Anwendungen der WMS vollkommen aus, wenn es nur darum geht, eine visuelle Verarbeitung der Geodaten vorzunehmen. Für den Fall, dass die Daten auch analytisch ausgewertet werden müssen, stößt man mit dem WMS an seine Grenzen. Insbesondere die Interpretation der Karte kann zum Teil nicht eindeutig erfolgen, da man nicht immer sagen kann, wofür welches Symbol steht und welche konkreten Eigenschaften einem Objekt zugeordnet sind. Um die Fähigkeit Analysen durchzuführen zu ermöglichen, ist der WFS entwickelt worden. Auf Basis vektorieller Daten können vielfältige Analysen, Verarbeitungen und auch Manipulationen vorgenommen werden. Das zentrale Element im Rahmen des WFS sind Features. Diese Geo-Objekte sind strukturierte vektorielle Geodaten. Wie bereits erwähnt, ist es auch möglich, Daten zu manipulieren (verändern, löschen, hinzufügen). Diese Funktionalitäten fallen unter den WFS-T (T steht für "Transactional"). Da das zu entwickelnde Web-GIS auch in der Lage sein soll, über die Web-Mapping-Anwendung Daten zu pflegen, ist der WFS-T unumgänglich. Dabei ist insbesondere bei der Wahl einer WFS-Implementierung zu beachten, dass Transaktionen auch möglich sind und die angeschlossenen Komponenten die Transaktionen zulassen. (Vgl. [And05])

Der WFS definiert, wie der WMS, die Schnittstelle. Es sind verschiedenen Request-Methoden implementiert. Zu diesen zählen die Anforderungen GetCapabilities, DescribeFeatureType, GetFeature, LockFeature, GetGMLObject und Transaction. Den Parametern ist ebenfalls die URL zu dem WFS voran zustellen.

- **GetCapabilitiesWFS** Das Ergebnis der Anforderung ist ebenfalls ein XML-Dokument (siehe WMS) mit dem Unterschied, dass detaillierte Informationen zu den Objekten mitgeliefert werden. Es wird u. a. angegeben, wie die Objekte (Features) aufgebaut sind, welche Attribute die Features besitzen und welche Operationen über den Features ausgeführt werden können. Der

Aufbau der Anforderung ist der Tabelle 4 zu entnehmen.

<u>Request Parameter</u>	<u>Erforderl.(E)/Opt.(O)</u>	<u>Beschreibung</u>
REQUEST= GetCapabilities	E	Gibt die Methode des Request an
VERSION= version	E	Angabe der WFS-Version
SERVICE= WFS	E	Angabe des Service-Typs

Tabelle 4: GetCapabilities-Request-Parameter ([Mar11] S. 30)

- **DescribeFeatureType** Da in der Regel verschiedene Objekttypen innerhalb eines WFS vorliegen, kann es von Interesse sein, eine detaillierte Beschreibung eines spezifischen Objekttyps auszuwerten. Für diesen Zweck kann diese Methode genutzt werden (Parameter aus der Tabelle 5).

<u>URL-Komponente</u>	<u>E/O</u>	<u>Default</u>	<u>Beschreibung</u>
REQUEST= DescribeFeatureType	E		Name des Request
TYPENAME= Name des Objekttyps	O		Namen der zu beschreibenden Feature Types (bei mehreren durch Komma getrennt)
OUTPUTFORMAT	O	text/xml; subtype= gml/3.1.1	Typ des Ausgabeformats

Tabelle 5: DescribeFeatureType-Parameter ([Mar11] S.43)

- **GetFeature** Eine weitere Methode ermöglicht die Anforderung bestimmter Features des WFS. Dieser Mechanismus wird genutzt, um in einem Web-GIS Objekte zu selektieren. Die Tabelle 6 enthält eine Übersicht der Parameter.

<u>URL-Komponente</u>	<u>E/O</u>	<u>Default</u>	<u>Beschreibung</u>
REQUEST=GetFeature	E		Name des Requests
OUTPUTFORMAT	O	text/xml; subtype= gml/3.1.1	Ausgabeformat
RESULTTYPE	O	results	Angabe, ob die Ergebnisse im Dokument enthalten sein sollen (results) oder ob nur die Anzahl der Treffer zurückgegeben (hits) wird.
PROPERTYNAME	O		Liste von Eigenschaften, welche von den Feature Types abgefragt werden sollen.
FEATUREVERSION= [ALL N]	O		Angabe der Version des Features (sofern eine Versionierung der Features existiert)
MAXFEATURES=N	O		Maximale Anzahl der zurückgegebenen Features einer Abfrage
SRSNAME	O		Angabe des Koordinatenreferenzsystems, wichtig bei geometrischen Features
TYPENAME	O		Liste von Feature Typenamen, die abgefragt werden sollen
FEATUREID	O		Aufzählung von Features, die über die ID angesprochen werden
FILTER (nur mit TYPENAME)	O		Angabe des zu verwendenden Filters
BBOX (nur mit TYPENAME)	O		Angabe einer Bounding Box (rechteckiger Bereich der abgefragt werden soll)
SORTBY	O		Sortierung

Tabelle 6: GetFeature-Parameter ([Mar11] S.44)

- Transaction** Wie bereits erwähnt, ist diese Methode für den Aufbau des Web-GIS von essentieller Bedeutung. Mit einem WFS-T ist es möglich, unabhängig von der Datenquelle (Datenbanksystem) im Hintergrund, die Manipulation von Daten einheitlich durchzuführen. Dies erleichtert vor allem die notwendigen Anpassungen der Web-Mapping-Anwendung, wenn der Fall eintritt, dass ein Austausch der Datenquelle erfolgt. Ein WFS-T stellt drei grundlegende Operationen zur Verfügung. Mit "INSERT" werden neue Daten hinzugefügt, "DELETE" löscht vorhandene Daten und mittels "UPDATE" werden Daten verändert. Die Tabelle 7 gibt eine Übersicht zu den verwendbaren Parametern.

<u>URL-Komponente</u>	<u>E/O</u>	<u>Default</u>	<u>Beschreibung</u>
REQUEST=Transaction	E		Name des Request
OPERATION=UPDATE, DELETE, INSERT	E		Auszuführende Operation
TYPENAME	E		Liste von Feature Typenamen auf denen die Operation ausgeführt wird
RELEASEACTION=[ALL SOME]	E		Rücksetzen der Locks(Sperre) von Features
FEATUREID	O		Aufzählung von Features, die über die ID angesprochen werden
FILTER	O		Angabe des zu verwendenden Filters
BBOX	O		Angabe einer Bounding Box (rechteckiger Bereich der abgefragt werden soll)

Tabelle 7: WFS-T-Parameter ([Mar11] S.45)

- LockFeature und GetGMLObject** Die letzten beiden Methoden werden nur am Rand behandelt, da diese für die aktuelle Implementierung des Web-GIS vorerst nicht von Bedeutung sind. Mit dem LockFeature-Request können Features vor Änderungen gesperrt bzw. freigegeben werden. Dies ist z. Bsp. notwendig, wenn mehrere Nutzer zeitgleich Objekte verändern und so können die Objekte vor anderen Zugriffen geschützt werden. Da das Konzept nicht vorsieht, dass existierende Objekte über die Web-Mapping-Anwendung verändert werden können, sondern nur neu angelegt werden, ist das Sperren und Freigeben von Features nicht notwendig. Wenn im Laufe der Weiterentwicklung des Web-GIS das Verändern von Objekten über die Web-Anwendung implementiert wird, ist dieser Mechanismus zu nutzen, um zu gewährleisten, dass der Mehrfachnutzerbetrieb fehlerfrei von statten geht.
- Mit GetGMLObject können, wie bei GetFeature, Objekte angefordert werden. Der Unterschied liegt darin, dass der Request kürzer ist und das Ergebnis einfacher strukturiert ist.

Analog zu dem WMS werden die zu setzenden Parameter durch die Verwendung eines Web-Mapping-Frameworks dynamisch angepasst. Es sind zwar bei dem Anlegen der Objekte, wie der Karte und den Layern, diverse Parameter zu setzen, die Menge der Parameter ist aber im Vergleich zu einem manuellen Aufbau der Requests wesentlich geringer.

Damit sind die grundlegenden Begriffe und technischen Aspekte geklärt, die für die Realisierung des Web-GIS von Nöten sind. Bevor ein konkreter Entwurf des Systems vorgenommen wird, sind zunächst geeignete Softwarekomponenten zu finden. Zusätzlich sind diverse Kartenportale zu sichten, um einen Überblick über die gängigen Funktionalitäten und deren Aufbau zu erhalten.

3 Analyse web-basierter Kartenportale

Das Internet beinhaltet eine Vielzahl an Kartenportalen. Thematisch bewegen diese sich hauptsächlich im Bereich der Navigationshilfen, um z. Bsp. den aktuellen Standort zu ermitteln und Reiserouten zu planen. Des Weiteren existieren fachspezifische Portale, die sich beispielsweise mit Aspekten der Umwelt befassen. Um einen Überblick über den Leistungsumfang von Kartenportalen zu erhalten und daraus Anforderungen an das eigene System abzuleiten, werden ausgewählte Web-GIS untersucht. Der relevante Aspekt liegt darin, welche Funktionen innerhalb des Web-GIS genutzt werden können. Diese sind für die Gestaltung der Web-Oberfläche des Web-GIS von Bedeutung. Neben den Funktionalitäten ist die Anordnung der Bedienelemente ein wichtiger Punkt und sollte sich nach den gängigen Implementierungen richten, um die Einarbeitungszeit bezüglich der Bedienung zu minimieren.

Die Analyse der Web-GIS erfolgt in zwei Kategorien. Es werden Web-GIS, die nur für die Betrachtung von Geodaten konzipiert sind (Karten-Viewer), untersucht und Kartenportale, die den Fokus auf Crowdsourcing legen und das Eintragen und Editieren von Daten ermöglichen.

3.1 Karten-Viewer

3.1.1 Google Maps

Der wohl bekannteste Vertreter unter den web-basierten Kartenportalen ist "Google Maps". Google Maps ist ein Dienst der Firma "Google Inc." und existiert seit dem Jahr 2005. Es handelt sich um einen weltweit verfügbaren Kartendienst, der es ermöglicht auf Basis von Satellitenbildern und Vektor-basierten Darstellungen Informationen über das Verkehrsnetz, Orte, Sehenswürdigkeiten, Gastronomiebetrieben, Handelsgeschäfte und vieles mehr abzurufen. Neben der Verortung werden weitergehende Informationen zu den dargestellten Elementen angeboten. Mit Google Maps besteht zudem die Möglichkeit eine Route zu planen und 360-Grad-Ansichten und Fotografien von bestimmten Orten einzusehen (Google Street View).

Der für die Analyse wichtige Punkt ist die Benutzeroberfläche. Die Abbildung 2 zeigt einen frei gewählten Kartenausschnitt mit allen Bedienelemente der Web-Anwendung.

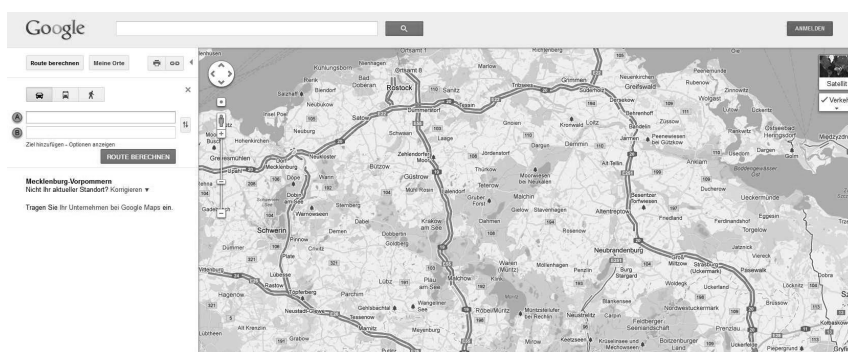


Abbildung 2: Benutzeroberfläche von Google Maps

Die Aufteilung der Oberfläche legt den Fokus auf die Karte. Der linke Bereich beinhaltet ein Werkzeug zur Planung von Routen. Die Bedienelemente sind auf das mindeste reduziert. Oben links kann die Zoom-Stufe festgelegt werden und der Kartenausschnitt in die vier Himmelsrichtungen verschoben werden. Neben der Navigation durch diese Elemente kann diese auch durch Maus erfolgen. Auf der rechten Seite befindet sich die Themenauswahl (Verkehr, Wetter, Gelände...). Über das oben gelegene Eingabefeld kann in der Karte nach diversen Begriffen gesucht werden. Schließlich kann unten links eine Legende eingesehen werden. Damit sind alle benötigten Funktionen übersichtlich integriert. Die Stärke von Google Maps liegt in der Masse an zur Verfügung gestellten Informationen. Auf Basis der Google-Suchmaschine können unzählige Orte und Einrich-

tungen gefunden werden.

Google Maps zeigt als am häufigsten genutzte Web-Mapping-Anwendung, wie einfach und komfortabel georeferenzierte Informationen abgerufen werden können. Das Fehlen von räumlichen Analysefunktionen (Streckenmessung, Flächenmessung, Fangkreise...) begründet sich vor allem auf der Tatsache, dass Google Maps in erster Linie als Marketinginstrument fungiert und diesen Zweck optimal erfüllt. (Vgl. [Goo13])

3.1.2 GAIA-MV

Das Landesamt für Umwelt, Naturschutz und Geologie Mecklenburg Vorpommern (LUNG) stellt mittels GAIA-MV (GeoAccessInternetApplicationMecklenburgVorpommern) Geodaten aus Mecklenburg-Vorpommern der Bevölkerung zur Verfügung. GAIA-MV ist im Rahmen des Aufbaus der GDI-Deutschland entwickelt worden. Das System stützt sich auf die genormten Webservices WMS und WFS. Somit stehen die dargestellten Karten nicht nur über das GAIA-MV-Kartenportal zur Verfügung, sondern können mittels entsprechender Software (GIS mit der Möglichkeit diese Webservices einzubinden) im eigenen System genutzt werden. GAIA-MV besteht aus zwei Web-Mapping-Anwendungen, die sich hinsichtlich der Komplexität unterscheiden.

- **GAIA-MVlight** Bei der Light-Version handelt es sich um eine abgespeckte Web-GIS-Version. Diese eignet sich vor allem für ungeübte GIS-Nutzer. Die Web-Oberfläche gestaltet sich wie folgt (siehe Abbildung 3):

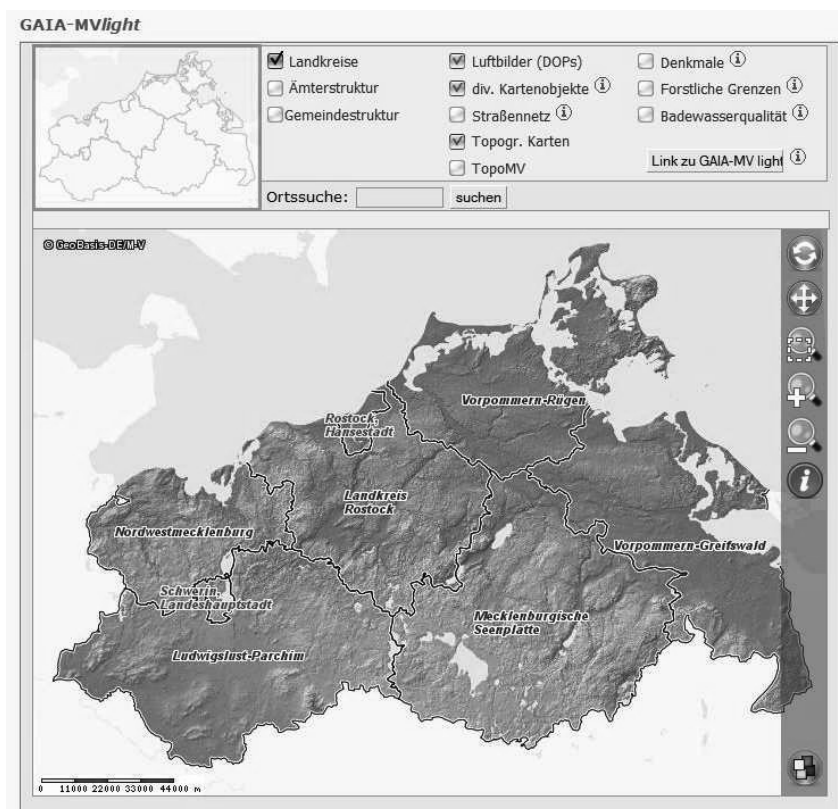


Abbildung 3: Benutzeroberfläche von GAIA-MVlight

Der obere Abschnitt des Web-GIS besteht zum einem aus einer kleinen Übersichtskarte, die es erleichtert bei hohen Zoomstufen eine Orientierung zu finden, und zum anderen aus einer Liste der thematischen Karten. Eine Suchfunktion ist ebenfalls implementiert. Unter diesen Bedienelementen befindet sich die Kartendarstellung inkl. einer Maßstabsanzeige. An der rechten

Seite befinden sich sämtliche Funktionsknöpfe (Neuladen, Rauszoomen auf volle Ausdehnung, Zoombox, Reinzoomen, Rauszoomen und Informationsanzeige).

- **GAIA-MVprofessional** Diese Version erweitert die Light-Version um eine Vielzahl an Funktionen und geht hinsichtlich des Designs in Richtung Desktop-basierter GIS (z. Bsp. Quantum GIS) (Abbildung 4).

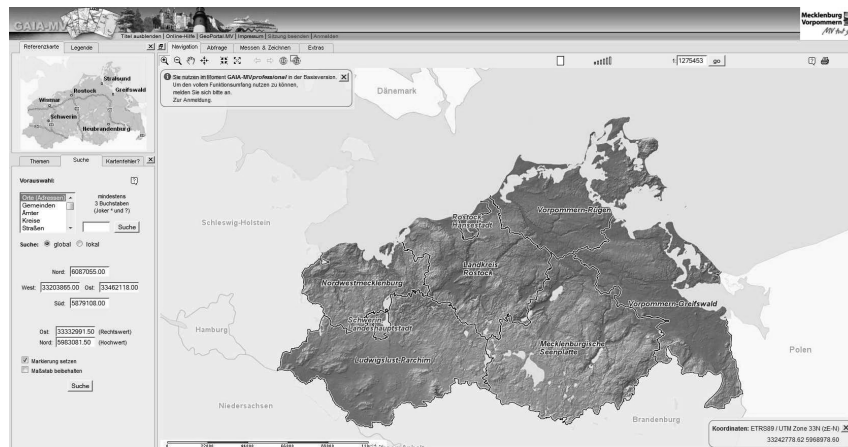


Abbildung 4: Benutzeroberfläche von GAIA-MVprofessional

Die linke Menüleiste stellt Funktionen für die Suche (nach Namen und Koordinaten), die Auswahl der verfügbaren Themen, eine Legende und eine Übersichtskarte zur Verfügung. Der Hauptteil der Anwendung besteht aus der Kartendarstellung. Über die Reiter über der Karte können eine Vielzahl an Funktionen aktiviert werden. Zu diesen Funktionen zählen Navigationselemente sowie Mess- und Zeichenwerkzeuge. In den unteren Abschnitt der Karte kann zudem der Maßstab und die aktuelle Position samt Koordinatenreferenzsystem abgelesen werden. Zusammenfassend stellt GAIA-MV in der Professional-Version ein mächtiges Kartenportal dar, und besticht vor allem durch den massiven Funktionsumfang. (Vgl. [Lan13])

3.2 Kartenportale mit Editierfunktionen

3.2.1 Open Street Map

Open Street Map (OSM) kann als freie Alternative zu Google Maps angesehen werden. Die Inhalte von OSM decken sich mit denen von Google Maps. Das Hauptaugenmerk liegt ebenfalls auf Geodaten aus dem Gebiet der Infrastrukturnetze. Weitere Informationen wie Wasserwege, Schienennetze, Naturgebiete, Sehenswürdigkeiten und Gebäude sind ebenfalls in den Datenbestand integriert. Die Besonderheit von OSM liegt in der Art und Weise der Datenpflege. Crowdsourcing stellt die Grundlage für den stetig wachsenden Datenbestand dar. Die OSM-Community trägt neue Daten ein und überprüft die Korrektheit der Informationen. Die aktiven Nutzergruppen erstrecken sich über den gesamten Globus und bieten zum Teil in sehr dünn besiedelten Gebieten zumindest Informationen über das Verkehrsnetz. Die folgende Abbildung 5 zeigt die Benutzeroberfläche von OSM.

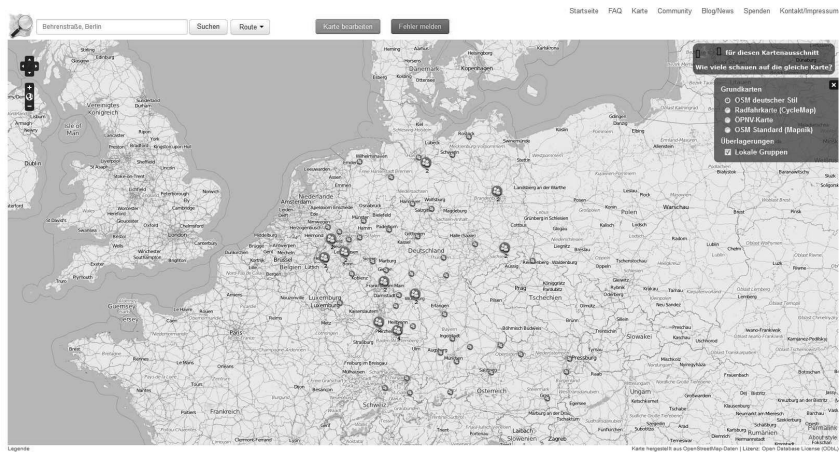


Abbildung 5: Benutzeroberfläche von Open Street Map

Das Design von OSM setzt auf eine klare Struktur und verzichtet weitestgehend auf zusätzliche Funktionen. Die Navigation baut primär auf die Steuerung per Maus/Touchpad auf. Das Zoomen mittels des Mausekzes und Verschieben des Kartenausschnitts durch das Ziehen der Karte ermöglicht eine schnelle Navigation. Alternativ können die oben links eingebauten Bedienelemente genutzt werden. In dem rechten Bereich befindet sich schließlich die Themenauswahl. Damit sind alle Bedienelemente abgedeckt, die für die Navigation benötigt werden. Zusätzlich lassen sich über die obere Menüleiste Orte suchen und eine Route planen. Aufgrund der Fülle an Informationen ist OSM eine ernst zu nehmende Alternative zu Google Maps. Es ist noch zu erwähnen, dass sämtliche OSM-Daten frei verfügbar sind und für den eigenen Datenbestand genutzt werden können. (Vgl. [FOS13])

Das Editieren von OSM-Datensätzen kann über verschiedene Wege erfolgen. Es sind Editoren entwickelt worden, die im Browser laufen (Potlatch) oder als eigenständige Anwendung auf dem eigenen System zu installieren sind. Der zweite Typ ist sowohl für Desktop-PCs (JOSM - Java-OpenStreetMap-Editor) als auch für mobile Endgeräte im Form von Apps (OSMapTuner) verfügbar. JOSM ist einer der am häufigsten verwendeten Editor. Die Benutzeroberfläche richtet sich dem traditionellen Aufbau von Desktop-GIS (siehe Abbildung 6)



Abbildung 6: Benutzeroberfläche von JOSM [Ope13g]

Da davon auszugehen ist, dass primär Personen, die mit dem Umgang mit Geodaten und GIS vertraut sind, die Datenbestände von OSM editieren, stellt die Bedienung von JOSM keine Hürde dar. (Vgl. [Ope13f])

3.2.2 Klarschiff-HRO

"Klarschiff-HRO" ist ein Bürgerportal der Hansestadt Rostock. Dieses ermöglicht den Bürgern Meldungen über illegale Mülldeponien, Schlaglöchern und weiteren Gefahren über ein interaktives Kartenportal durchzuführen. Die Kartenbasis bilden die OSM-Datenbestände. Das Portal ist im Aufbau sehr übersichtlich. Der gesamte Bildschirm ist mit der Kartendarstellung gefüllt. Die Navigation entspricht der von OSM (Maus/Touchpad/Bedienelemente oben links). Am rechten oberen Rand sind die für die Meldungen benötigten Menüpunkte über die Karte gelegt. Diese umfassen eine Suchfunktion, ein Interface für die Erstellung neuer Meldungen und die Auswahl bereits gemeldeter Probleme (Abbildung 7).



Abbildung 7: Benutzeroberfläche von Klarschiff-HRO

Um eine neue Meldung einzutragen, ist als erstes der entsprechende Ort in der Karte auszuwählen. Anschließend wählt man über den Menüpunkt "neue Meldung" entweder "Problem melden" oder "Idee melden". Ist dies durchgeführt worden, erscheint eine Eingabemaske.

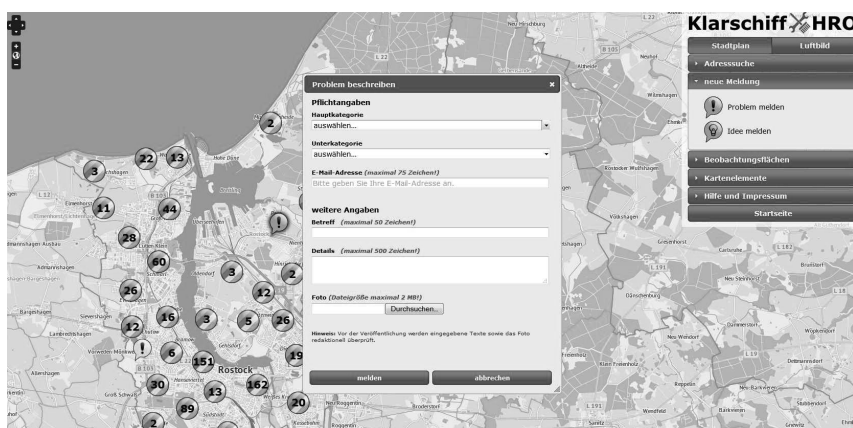


Abbildung 8: Eingabemaske für die Meldung neuer Probleme

An dieser Stelle spezifiziert man das Problem. Es werden vorgefertigte Probleme angeboten, die über eine ausklappbare Liste ausgewählt werden. Eine weitere Angabe, die für die Meldung zu tätigen ist, ist die Angabe einer Email-Adresse. Um das Problem genauer zu beschreiben, können Fotos hochgeladen werden und textuelle Beschreibungen eingefügt werden. Sofern alle Angaben eingetragen worden sind, kann der Inhalt übermittelt werden und wird der Karte hinzugefügt. Die

Handhabung des Portal ist sehr simpel und ermöglicht auch ungeübten Nutzer von Web-GIS ein leichten Einstieg. (Vgl. [Han13])

3.3 Schlussfolgerungen für das Design des Portals

Der Umgang mit interaktiven Kartenportalen stellt vor allem Anforderungen an das Design und die Handhabung. Dem Nutzer sollten nur die Funktionen zur Verfügung gestellt werden, die für den Zweck des Portals benötigt werden. Zusätzlicher Ballast, wie Funktionen und Eingabemasken, die für die eigentliche Aufgabe des Portals nicht relevant sind, können für Verwirrung sorgen und u. U. zu einer fehlerhaften Bedienung der Anwendung führen. Für das zu erstellende Kartenportal heißt das, dass zwar alle gängigen Funktionen theoretisch eingebaut werden können, aber dem Provider des Web-GIS die Wahl gelassen werden muss, welche Funktionalitäten integriert werden sollen. Auf Basis der Analyse von diversen Kartenportale sind folgende Funktionen zur Verfügung zu stellen:

1. Navigation (Zoomen, Verschieben)
2. Objektanfragen
3. Messinstrumente (Linien, Flächen)
4. Übersichtskarte
5. Themenauswahl
6. Koordinatenanzeige
7. Anlegen neuer Objekte inkl. setzen der Attribute

Für die Erfassung von neuen Daten bietet es sich an, vorgefertigte Eingabemasken zu verwenden. So können unpassende bzw. fehlerhafte Eingaben durch den Nutzer abgefangen werden und im selben Zuge die Überprüfung der neu angelegten Daten erleichtert werden.

Hinsichtlich der Anordnung der Bedienelemente sollte dies wie folgt gestaltet werden:

- Die Kartendarstellung ist das zentrale Element der Anwendung und sollte den Großteil der Fläche des Kartenportals einnehmen.
- Werkzeuge (Navigation, Messung, Anfragen, Anlegen neuer Objekte) sollten innerhalb einer Leiste angeordnet werden.
- Alle weiteren Elemente (Übersichtskarte, Themenauswahl...) sollten links der Karte angezeigt werden und bei Bedarf ausgeblendet werden können, damit eine größere Darstellung der Karte möglich ist.

Unter Berücksichtigung dieser Vorgaben, wird die Einarbeitungszeit für die Benutzung des Kartenportals signifikant reduziert und das Portal kann auch durch ungeübte Nutzer von GIS bedient werden.

4 Analyse und Bewertung ausgewählter Softwarekomponenten

Dieses Kapitel befasst sich mit der Auswahl geeigneter Softwarekomponenten für den Aufbau des Web-GIS. Unabhängig von der Architektur des Systems werden drei Komponenten benötigt. Diese umfassen eine Datenbank, einen Kartenserver und die Web-Mapping-Anwendung. Der Markt bietet mittlerweile eine Vielzahl an fertigen Softwareprodukten aus dem Bereich der Geoinformationen an. Insbesondere die Organisation "OSGeo" administriert und organisiert mehrere Projekte, mit dem Ziel unterschiedlichste Programme für die Verarbeitung von Geodaten der Öffentlichkeit zur Verfügung zu stellen. Neben den von der OSGeo verwalteten Projekten gibt es noch weitere Unternehmen und Organisationen, die derartige Produkte entwickeln. Im Folgenden werden ausgewählte Produkte vorgestellt und hinsichtlich der Verwendung für das Web-GIS analysiert.

4.1 Bewertungskriterien

Um eine aussagekräftige Analyse durchzuführen, sind im Vorfeld die Kriterien für eine Bewertung der Softwarekomponenten festzulegen. Ziel ist eine möglichst objektive Einschätzung für die Verwendung der Komponenten zu erarbeiten. Die Kosten für eine Softwarekomponente sind nicht relevant, da nur freie Produkte in Frage kommen. Kriterien wie die Hardwareanforderungen werden nicht betrachtet. Dies begründet sich auf den im Vorfeld durchgeführten Recherchen. Diese ergaben, dass bei keinem der untersuchten Komponenten konkrete Angaben zu den minimalen Hardwareanforderungen gemacht wurden. Die aktuellen und letzten Computergenerationen sollten alle in der Lage sein, die Softwareprodukte zu verwenden. Je nach Art der Komponente werden die folgenden Kriterien untersucht:

1. Geodatenbanken

- Unterstützte Betriebssysteme
- Installationsaufwand
- Konfigurationsaufwand
- Unterstützte Geometrien
- Funktionsumfang für räumliche Daten

2. Mapserver

- Unterstützte Betriebssysteme
- Installationsaufwand
- Konfigurationsaufwand
- Unterstützte Standards und Normen
- Unterstützte Datenformate und Datenbanken

3. Web-Mapping-Werkzeuge

- Installationsaufwand
- Flexibilität bei der Entwicklung
- Features
- Komplexität und Programmieraufwand

4.2 Geodatenbanken

Die erste Softwarekomponente, die die Grundlage für die GeodatenSpeicherung bildet, ist die Geodatenbank. Eine Geodatenbank, ist ein auf räumliche Daten spezialisiertes Datenbanksystem. Die Verwendung von räumliche Daten wird häufig durch Erweiterungen für ein bestehendes Datenbanksystem realisiert. Diese Erweiterungen integrieren u. a. geometrische Datentypen und räumliche Funktionen. Die räumlichen Informationen werden in Tabellen (Spalten und Zeilen) angelegt (relationales Modell). Das Design der Tabellen kann mit unterschiedlichen Datentypen erfolgen und bietet dem Nutzer ausreichend Möglichkeiten die Daten flexibel zu modellieren. Durch den relationalen Ansatz können zudem die Tabellen untereinander in Beziehung gesetzt werden. Dies ermöglicht dem Nutzer komplexe Datenmodelle zu erstellen. Weitere Vorteile, die mit der Verwendung von Datenbanksystemen einhergehen, sind Prüfungen auf Konsistenz der Daten, Freiheiten von Redundanzen und ein Mehrfachnutzerbetrieb. Aufgrund der starken Verbreitung von räumlichen Datenbanksystemen bieten eine Vielzahl an Geoinformationssystemen (Desktop-GIS und webbasierte Systeme) Schnittstellen an, um direkt auf die Daten zuzugreifen. Sofern die Datenbank im Kontext eines Webservers läuft, kann auf die lokalen Daten auch über das Internet zugegriffen werden. Im Folgenden werden ausgewählte Datenbanksysteme analysiert, um einen geeigneten Kandidaten für das Web-GIS zu finden. (Vgl. [Tyl08], S. 312)

4.2.1 PostgreSQL/PostGIS

Ein bekanntes Datenbanksystem aus dem Bereich der freien Software ist PostgreSQL mit der Erweiterung PostGIS. Dieses objekt-relationale Datenbanksystem hat eine langjährige Entwicklung durchlaufen. Eine hohe Stabilität und ein großer Funktionsumfang machen PostgreSQL zu einer sehr guten Alternative zu einer kommerziellen Oracle-Datenbank. Mit dem Aufsatz PostGIS können zusätzlich räumliche Daten mit PostgreSQL verwaltet werden. PostGIS integriert neben den geometrischen Datentypen eine Vielzahl an Funktionen für die Manipulation und Analyse räumlicher Daten.

- **Unterstützte Betriebssysteme:**

- FreeBSD
- Linux
- Mac OS X
- Solaris
- Windows

- **Installationsaufwand:** Die Installation von PostgreSQL/PostGIS gestaltet sich einfach. Unter Windows erfolgt die Installation über ein Setup-Programm, das von der Homepage der Entwickler heruntergeladen werden kann. Unter Linux kann man die interne Paketverwaltung nutzen um eine kompilierte Version von PostgreSQL/PostGIS zu beziehen.

- **Konfigurationsaufwand:** Prinzipiell kann PostgreSQL sofort verwendet werden. Es empfiehlt sich aber zumindest einen Nutzer inklusive Passwort zu setzen, um die Datenbank vor ungewollten Zugriffen zu schützen. Weitere Konfigurationen (Verbindungsparameter) können entweder direkt in einer Konfigurationsdatei eingestellt werden oder man nutzt Verwaltungsprogramme wie den "PGAdmin3". Diese Werkzeuge stellen eine GUI zur Verfügung, mit denen die Datenbank konfiguriert und die Daten verwaltet werden können.

- **Unterstützte Geometrien:** Die von PostGIS unterstützten Geometrien basieren auf den OGC-Standard "Simple Feature Specification for SQL (SFS)". Zu den unterstützten Geometrien zählen:

- Punkte (Point)
- Linien (LineString)
- Polygone (Polygon)
- Mehrere Punkte (MultiPoint)
- Mehrere Linien (MultiLineString)
- Mehrere Polygone (MultiPolygon)
- Sammlung von Geometrien (GeometryCollection)

Die neueste Entwicklung zielt auf die Norm "ISO 13249-3 - SQL/MM - Spatial" ab. Im Prinzip basiert die Norm auf dem Simple Feature Modell (Klassenmodell für die Abbildung von Geometrien), erweitert dieses aber um weitere Geometrien. Da die neuen Geometrietypen noch nicht vollständig implementiert sind, werden zum aktuellen Zeitpunkt nur die der SFS unterstützt.

- **Funktionsumfang für räumliche Daten:** Die Analyse und Bearbeitung der Daten erfolgt durch SQL-Befehle. Für die räumlichen Daten steht ein umfangreicher Satz an PostGIS-Funktionen zur Verfügung. Ohne auf jede Funktion im Detail einzugehen, sind die folgenden grundlegenden Operationen möglich:

- Erzeugung, Löschung und Veränderung von Geometrien
- Räumliche Funktionen (Flächenberechnung, Pufferberechnung, Distanzberechnung, Verschnidungen)
- Operatoren für die Analyse topologischer Beziehungen zwischen Geometrien
- Konvertierung der Geometrien in diverse Geodatenformate (WKB, WKT, GML, SVG, KML)
- Transformation von Koordinaten
- Räumliche Indizierung der geometrischen Objekte

Seit der Integration von "SQL/MM - Spatial" richten sich die Bezeichnungen und Aufgaben der Funktionen und Operatoren nach dieser Norm. Sofern es sich um räumliche Funktion handelt, befindet sich das Präfix "ST_" vor jeder Funktionsbezeichnung. "ST" steht für Spatial Type und vereinfacht die Identifikation räumlicher Funktionen. Zudem findet diese Norm auch in anderen SQL-basierten Datenbanksystemen für räumlichen Daten Anwendung und erleichtert somit den Umgang mit anderen Systemen.

Das Datenbanksystem PostgreSQL mit der Erweiterung PostGIS stellt den größten Konkurrenten von Oracle Spatial dar. Der flexible Einsatz, große Funktionsumfang und die Anbindungsmöglichkeiten an GIS-basierten Produkten offeriert dem Nutzer ein Werkzeug zur effizienten Verwaltung räumlicher Daten. (Vgl. [Tyl08], S.311 ff), (Vgl. [The13a])

4.2.2 SQLite / SpatiaLite

Eine Einordnung von SQLite/SpatiaLite in die Kategorie der Datenbanksysteme ist nicht ganz korrekt. Im Prinzip handelt es sich um ein Datei-basiertes Format, verwaltet aber intern die Daten wie eine relationale Datenbank. Die Steuerung der Datenbank erfolgt über die Kommandozeile. Es können aber auch GUI-basierte Verwaltungswerkzeuge genutzt werden. Aufgrund der Architektur ist eine SQLite-Datenbank robust und zeichnet sich durch einer hohen Portabilität aus. Während bei den meisten Datenbanksystemen die Daten exportiert werden müssen, genügt es einfach die SQLite-Datei zu kopieren. Für die Nutzung räumlicher Daten wird das selbe Prinzip wie bei PostgreSQL genutzt. SpatiaLite ist ebenfalls eine räumliche Erweiterung für SQLite.

- **Unterstützte Betriebssysteme:** SQLite und SpatiaLite basieren auf der Programmiersprache C. Da neben kompilierten Binärdateien auch der Quellcode bezogen werden kann, ist es prinzipiell möglich, die Datenbank auf jeden System zu verwenden, wenn der Quellcode auf dem Betriebssystem kompiliert wird. Als kompilierte Binärdateien stehen diese für die folgenden Betriebssysteme zum Download zur Verfügung:
 - Windows
 - Linux
 - Mac OS X
- **Installationsaufwand:** Für die Nutzung von SQLite/SpatiaLite wird nur die SQL-Datenbank-Engine benötigt. Wie in der Einführung bereits erwähnt, nutzt man entweder die kompilierten Binärdateien oder man kompiliert den Quellcode. Die Binärdateien sind dann im System zu hinterlegen. Wo die Dateien zu integrieren sind, hängt vom Betriebssystem ab.
- **Konfigurationsaufwand:** Im Prinzip wird keine Konfiguration benötigt, da es sich nur eine Programmbibliothek handelt. Das Anlegen einer neuer Datenbank und dessen Verwaltung erfolgt über SQL-Befehle. Diese werden entweder über die Kommandozeilen ausgeführt oder man nutzt vorgefertigte Benutzeroberflächen oder eigene Skripte.
- **Unterstützte Geometrien:** Der Umfang der unterstützten Geometrien stützt sich auf den im Kapitel 4.2.1 genannten Simple Feature Modell. Somit werden die Geometrien Punkt, Linie, Polygon und deren Multi-Typen und Kompositionen unterstützt.
- **Funktionsumfang für räumliche Daten:** Der Funktionsumfang gleicht sich ebenfalls dem von PostgreSQL/PostGIS an. Die Norm "SQL/MM - Spatial" gibt den Rahmen der Bezeichnungen und Aufgaben der Funktionen vor und bietet alle für die Analyse für Geodaten benötigten Operationen an.

Auch wenn SQLite/SpatiaLite noch ein relativ neues Datenformat ist, so stellt dies eine leichtgewichtete Alternative zu komplexen Datenbanksystemen dar. Das Format vereinigt die Vorteile von relationalen Datenbanken mit der Portabilität von Datei-basierten Vektorformaten wie den ES-RI Shapefiles. (Vgl. [Tus13], [The13b])

4.2.3 MySQL Spatial

MySQL Spatial ist der letzte zu untersuchende Kandidat unter den Geodatenbanken. Die Basis bildet das Datenbanksystem MySQL. Dieses System existiert bereits seit 1994 und hat sich im Bereich der relationalen Datenbanksysteme etabliert. Eine verbreitete Verwendung von MySQL ist die Anwendung des Systems im Backend von web-basierten Anwendungen. Zudem existieren eine Vielzahl an Frontends für MySQL, um das Datenbanksystem zu verwalten. Die Erweiterung MySQL Spatial integriert die Nutzung räumlicher Daten und Funktionen.

- **Unterstützte Betriebssysteme:**
 - Windows
 - Linux
 - Mac OS X
 - Unix

- **Installationsaufwand:** Aufgrund der breiten Unterstützung der unterschiedlichen Betriebssysteme stehen entweder vorgefertigte Binärdateien oder der Quellcode zur Verfügung. Es wird empfohlen die Binärdateien zu verwenden, da diese bereits vorkonfiguriert sind. Die Binärdateien führen eine automatische Integration von MySQL in das System aus. Die Erweiterung MySQL Spatial ist bereits Bestandteil der Installation von MySQL.
- **Konfigurationsaufwand:** Um die Datenbank nutzen zu können, sollte die Nutzer- und Rechteverwaltung bearbeitet werden. Während unter Windows bereits die erforderlichen Einstellungen im Rahmen der Installation vorgenommen werden, sind unter Linux/UNIX, sofern keine GUI-Werkzeuge verwendet werden, auf der Kommandozeile mehrere SQL-Befehle auszuführen. Die Dokumentation auf der Homepage von MySQL erklärt sehr detailliert, welche Schritte für die Konfiguration auszuführen sind.
- **Unterstützte Geometrien:** Auch hier findet die "Simple Feature Specification for SQL" Anwendung und bietet die identischen Geometrietypen analog zu PostGIS und SpatiaLite an.
- **Funktionsumfang für räumliche Daten:** Der Umfang der Funktionen umfasst die folgenden Punkte:
 - Konvertierung in die Formate WKT und WKB
 - Topologische Operationen
 - Erzeugen, Löschen und Manipulieren von Geometrien
 - Berechnung von u. a. Flächen und Distanzen
 - Transformationen

Zusammenfassend lässt sich sagen, dass mit MySQL Spatial räumliche Daten ebenso gut verwaltet können, wie mit den zwei anderen Systemen. Die Menge an unterstützten Geometrien und die räumlichen Funktionen und Operatoren sind in ihrem Umfang äquivalent. (Vgl. [Sun13], [Tus13])

4.2.4 Auswertung der Geodatenbanken

Die Untersuchung der drei Geodatenbanksysteme zeigt, dass die Unterschiede auf funktionaler Ebene minimal sind. Die Grundlage für die Verarbeitung von Geometrien wird durch die Verwendung von Normen und Standards geregelt. Die größten Unterschiede liegen in dem Umfang des Datenbanksystems und der Art und Weise wie die Datenbank konfiguriert wird. Während SpatiaLite ohne ein umfangreiches Managementsystem auskommt, bestehen PostgreSQL/PostGIS und MySQL aus einer Vielzahl an Modulen und Komponenten. Bezüglich der einzelnen Bewertungskriterien ergeben sich folgende Schlussfolgerungen für das aufzubauende System.

- **Unterstützte Betriebssysteme:** Es sind keine Unterschiede hinsichtlich der unterstützten Betriebssysteme festzustellen. Die zwei am weitesten verbreiteten Systeme Windows und diverse Linux-Distributionen werden alle unterstützt. Somit ist dieses Kriterium für die Auswahl einer Geodatenbank nicht relevant.
- **Installationsaufwand:** Hier zeigen sich ebenfalls minimale Unterschiede. Es können für alle Geodatenbanken kompilierte Versionen genutzt werden. Mit Hilfe von Installationsprogrammen

für Windows und der internen Paketverwaltung von Linux sind die Programme schnell und komfortabel installiert.

- **Konfigurationsaufwand:** Der Aufwand für die Einrichtung der Datenbank ist bei SpatialLite am geringsten. Im Prinzip ist keine Konfiguration der Datenbank notwendig. Bei den anderen beiden Produkten können die Datenbanken nach der Installation zwar sofort genutzt werden, die Rechte- und Sicherheitseinstellungen sollten jedoch überprüft werden. Unter der Verwendung von GUI-Werkzeugen kann dies auch von Personen durchgeführt werden, die wenig Erfahrungen mit Datenbanken besitzen. Ist man aber darauf angewiesen, sämtliche Konfigurationen über die Kommandozeile, SQL-Befehlen und dem Editieren von Dateien durchzuführen, sind weitergehende Kenntnisse zwingend erforderlich. Sollten dennoch gravierende Unklarheiten auftreten, können die Dokumentationen der Entwickler genutzt werden. Diese geben einen ausführlichen Leitfaden für die Einrichtung und zeigen an Hand von vielen Beispielen, wie die räumlichen Funktionen anzuwenden sind.
- **Unterstützte Geometrien:** Es sind keine Unterschiede festzustellen. Alle drei Systeme unterstützen die selben Geometrien, die auf den implementierten Standard "Simple Feature Specification for SQL" beruhen.
- **Funktionsumfang für räumliche Daten:** Analog zu den vorhergehenden Kriterien können auch hier keine Unterschiede festgestellt werden. Mit allen Systemen können räumliche Objekte angelegt, gelöscht, manipuliert und analysiert werden.

Aufgrund der Ergebnisse der einzelnen Kriterien kann kein Kandidat als die optimale Lösung angesehen werden. Im Leistungsumfang sind die Systeme identisch. Gemäß den Anforderungen ist ein leicht-gewichtetes System zu entwickeln, das sich zudem leicht konfigurieren lassen soll. In Anbetracht dieser Rahmenbedingungen eignet sich SQLite/SpatialLite am besten. Da hier auf eine Konfiguration verzichtet werden kann und eigene Skripte für die Erstellung von Datenbanken genutzt werden können, kann mit diesem Produkt den Anforderungen Rechnung getragen werden. Sollte SpatialLite kompatibel zu dem Kartenserver sein, ist dies zu wählen. Andernfalls ist gemäß den Anforderungen des Kartenservers, das passende Produkt zu nehmen. Auf Basis der Analyse der Datenbanksysteme und deren geringen Unterschieden untereinander, stellt die Wahl einer Alternative aber keinen gravierenden Nachteil dar.

4.3 Kartenserver

4.3.1 MapServer/TinyOWS

MapServer ist eine freie Entwicklungsumgebung für die Erzeugung statischer und dynamischer Karten. Die erste Veröffentlichung geht auf das Jahr 1994 zurück. Die University of Minnesota (UMN) initiierte das Projekt und entwickelte MapServer stetig weiter. Mittlerweile ist MapServer durch die OSGEO übernommen worden und findet durch die freie Verfügbarkeit eine weite Verbreitung. Aufgrund der langen Entwicklungszeit stellt MapServer einen der stabilsten und schnellsten Kartenserver dar. Da MapServer zwar über einen WFS verfügt aber keine Transaktionen ermöglicht, haben die Entwickler ein zusätzliches Programm entwickelt. TinyOWS ist ein reiner WFS-T und als Ergänzung zu MapServer zur Verfügung gestellt. MapServer und TinyOWS arbeiten dabei unabhängig voneinander, sodass untereinander keine Abhängigkeiten entstehen.

MapServer arbeitet hinter einem Webserver. Der Webserver nimmt eine Kartenanfrage entgegen, leitet diese an MapServer weiter und schließlich gibt MapServer den angeforderten Kartenausschnitt über den Webserver an den Client zurück. Die Arbeitsweise stellt sich folgendermaßen dar

(Abbildung 9):

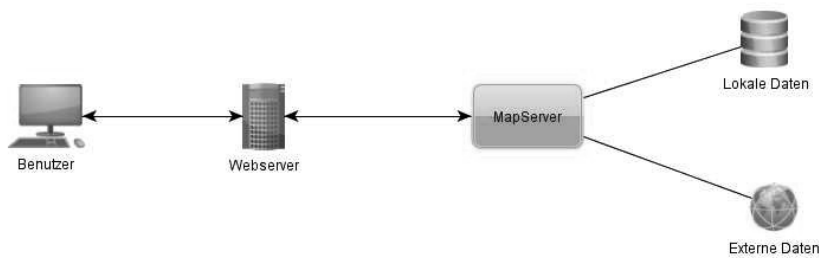


Abbildung 9: Funktionsweise von MapServer

Die Bestandteile von MapServer umfassen mehrere Komponenten. Zu den Wichtigsten zählen die MapServer-Ausführungsdatei und die Mapdatei. Die Ausführungsdatei wird als ausführbare CGI-Anwendung auf dem Webserver hinterlegt. CGI steht für Common Gateway Interface und ist eine standardisierte Schnittstelle für die Kommunikation zwischen einem Webserver und dritter Software. MapServer arbeitet als Vermittler zwischen dem Webserver und den Datenquellen, die für die Kartenerzeugung benötigt werden. Die zweite Komponente, die Mapdatei, dient der Konfiguration der Karten. Es handelt sich bei dieser Datei um eine Textdatei, in der mittels diverser Parameter das Aussehen der Karte und die zu verwendenden Datenquellen spezifiziert werden. TinyOWS ist in der Lage mit den gleichen Mapfiles zu arbeiten (sofern die Konfiguration eines WFS vorliegt). Bei dem Aufruf der MapServer-Ausführungsdatei erfolgt ein Verweis auf ein Mapfile und die Karte wird gemäß der Parameter generiert (Vgl. [Tyl08], S. 43 ff). Im Folgenden werden die Kriterien für die Bewertung analysiert.

- **Unterstützte Betriebssysteme** Eine große Stärke von MapServer ist die Lauffähigkeit auf einer Vielzahl an Betriebssystemen. Neben diversen Linux-Distributionen, wie Debian, Ubuntu, SuSE usw., werden auch sämtliche Windows-Betriebssysteme und MAC OS X unterstützt. Damit sind alle bedeutenden Betriebssysteme abgedeckt und ermöglicht einen flexiblen Einsatz von MapServer im Rahmen dessen Integration in eine bestehende Server-Landschaft.
- **Installationsaufwand** Der Aufwand der Installation variiert abhängig von dem verwendeten Betriebssystem. Unter Linux sind vor der Installation zusätzliche Programmbibliotheken zu installieren. Diese werden benötigt, um zu gewährleisten, dass MapServer u. a. die Geodaten verarbeiten kann. Zu den Programmen zählen z. Bsp OGR (Verarbeitung von Vektordaten) und GDAL (Verarbeitung von Rasterdaten). Dies ermöglicht, auf Basis der Anforderungen an den Kartenserver eine maßgeschneiderte Umgebung aufzusetzen. Nachdem die Bibliotheken installiert worden sind, ist der Quellcode des MapServer unter Verwendung diverser Parameter zu kompilieren und schließlich ist die erstellte Anwendung auf dem Webserver zu hinterlegen. Der gesamte Vorgang kann unter Umständen sehr lange dauern und zu Fehlern führen, falls die einzelnen Bibliotheken nicht korrekt installiert wurden. Unter Windows gestaltet sich die Installation wesentlich komfortabler. Es stehen diverse GUI-basierte Installationsprogramme zur Verfügung. Dabei werden die fehlenden Bibliotheken automatisch installiert. Unter Linux kann man alternativ die interne Paketverwaltung nutzen, um MapServer zu installieren. Der Vorteil liegt darin, dass automatisch die zusätzlich benötigten Pakete integriert werden. Auch wenn man bei dieser Variante keine Kontrolle auf die Konfiguration hat, ist diese Installation so ausgelegt, dass alle von Haus aus unterstützten Funktionen und Formate aktiviert sind.
- **Konfigurationsaufwand** Die Konfiguration der MapServer-Ausführungsdatei erfolgt bereits im Rahmen des Kompilierens durch die Parameter. Die Konfiguration der Services, wie WMS und

WFS, und die Gestaltung der Karten erfolgt in den Mapfiles. Der große Vorteil liegt darin, dass bei der Konfiguration der Karten die Ausführungsdatei nicht verändert werden muss. Der Aufwand ein Mapfile zu erstellen schwankt sehr stark. Je spezifischer die Karte definiert wird, desto mehr Parameter werden benötigt. Eine minimale Konfiguration eines Mapfiles kann sich wie folgt gestalten:

```
MAP
    NAME "Beispiel"
    STATUS ON
    SIZE 600 600
    SYMBOLSET "../symbols.txt"
    EXTENT -180 -90 180 90
    UNITS DD
    SHAPEPATH "../daten"
    IMAGECOLOR 255 255 255
    FONTSET "../fonts.txt"

    #
    # Start of web interface definition
    #
    WEB
        IMAGEPATH "//tmp/ms_tmp/"
        IMAGEURL "/ms_tmp/"
    END

    #
    # Start of layer definitions
    #
    LAYER
        NAME 'Mein Layer'
        TYPE RASTER
        STATUS DEFAULT
        DATA karte.tiff
    END
END
```

Gemäß des Beispiels wird ein Mapfiles mit dem Parameter "MAP" eingeleitet und dieser und jeder weitere Parameter wird durch "END" beendet. Es ist darauf zu achten, dass die ein- und ausleitenden Tags korrekt zu verschachteln sind. Ein Überlappen von Tags führt zu einen fehlerhaften Mapfile. Innerhalb der MAP-Umgebung werden an erster Stelle die globalen Parameter wie Name, Größe der Karte, Symbole, Ausdehnung der Karte, Einheiten und die Datenquellen angegeben. Die WEB-Umgebung wird benötigt, um MapServer mitzuteilen, an welcher Stelle im Webserver die generierten Karten abgelegt werden sollen (Zwischenspeicher für temporäre Karten). Schließlich erfolgt in der LAYER-Umgebung die Definition der darzustellenden Kartenebene. Neben diesen Basisparametern existieren noch dutzende weitere Parameter. Ob diese verwendet werden müssen, hängt stets von der jeweiligen Anwendung ab. Um den Aufwand für die Erstellung von Mapfiles zu minimieren, gibt es zwei Möglichkeiten. Entweder kann man mit GIS, wie Quantum GIS, automatisch Mapfiles erzeugen oder man kann mit Mustern arbeiten. Die automatische Erzeugung mittels GIS bringt aber den Nachteil mit sich, dass man weniger Kontrolle über den Inhalt des Mapfiles hat und ggf. noch Änderungen vornehmen muss. Die Arbeit mit Mustern hingegen, gibt dem Entwickler mehr Kontrolle. Prinzipiell gestaltet sich die Konfiguration der Mapfiles als sehr überschaubar, da es sich stets um eine einzelne lesbare

Textdatei handelt, die einfach modifiziert werden kann.

- **Unterstützte Standards und Normen** Ein wichtiger Punkt sind Standards und Normen. Standardisierte Webservices wie WMS und WFS bilden in der heutigen Geodaten-Landschaft einen essentiellen Stützpfiler, für den Aufbau einer interoperablen Umgebung für den Austausch von Geodaten. MapServer unterstützt u. a. in der aktuellen Version 6.2.1 die folgenden OGC-konformen Standards bzw. ISO-Normen:

- WMS
- WFS
- GML (Geography Markup Language - Auszeichnungssprache zum Austausch raumbezogener Objekte)
- SLD (Styled Layer Descriptor - Vorgaben für die Ausgestaltung von Rasterkarten)
- FES (Filter Encoding - Vorgaben für die Filterung von Informationen auf Basis von XML)

Da die Standards und Normen in regelmäßigen Abständen aktualisiert werden, kann es z. T. dazu führen, dass die neuesten Versionen der Standards von MapServer nicht unterstützt werden. Dennoch ist die breite Unterstützung der OGC-Standards gegeben und ermöglicht einen hohen Grad an Interoperabilität der internen und externen Geodatenbestände.

- **Unterstützte Datenformate und Datenbanken** Die unterstützten Formate richten sich nicht nach der MapServer-Ausführungsdatei. Im Rahmen der Installation sind diverse Programmbibliotheken zu installieren, die es ermöglichen unterschiedliche Raster- und Vektorformate als Quelle zu nutzen. Unter Verwendung der aktuellsten Versionen von GDAL (Rasterdaten, Version) und OGR (Vektordaten, Version) können die folgenden Formate von MapServer genutzt werden (Auswahl):

- Rasterformate: BMP, GIF, JPEG, GeoTIFF, WMS, PostGIS Raster, GeoRaster
- Vektorformate: ArcSDE, Shapefile, GML, KML, MySQL, Oracle Spatial, PostgreSQL/PostGIS, WFS, SpatialLite, GeoJSON

Es ist zu beachten, dass unter Verwendung von TinyOWS nur eine PostgreSQL/PostGIS-Datenbank verwendet werden kann. Da es sich bei diesem Datenbanksystem aber um eine stabile und leistungsstarke Lösung handelt, kann dieser Nachteil hinsichtlich der Flexibilität bei der Auswahl einer Datenbank kompensiert werden. Somit ist bei der Wahl von TinyOWS als WFS-T-Implementierung zwingend eine PostgreSQL-Datenbank im Backend zu nutzen.

Die komplette Übersicht aller unterstützten Formate kann unter [Ope13b] und [Uni13] eingesehen werden. In Anbetracht der Tatsache, dass jederzeit die Möglichkeit besteht zusätzliche Treiber für nicht-unterstützte Formate zu installieren, ist MapServer in der Lage alle gängigen Geodatenformate zu nutzen. Mit MapServer in Verbindung mit TinyOWS steht eine Kartenserver-Lösung zur Verfügung, die durch den hohen Grad an Flexibilität und der Möglichkeit mit Konfigurationsmustern zu arbeiten besticht(Vgl. [Uni13]).

4.3.2 degree

Deegree ist ein auf java-basierendes Framework für den Aufbau einer Geodateninfrastruktur. Deegree besteht aus mehreren Werkzeugen für das Datenmanagement, den Datenzugriff, die Visualisierung und Sicherheitsprogrammen. Der Fokus von deegree liegt auf den offenen Standards und Normen wie WMS und WFS, die die Grundlage für diesen Kartenserver bilden. Analog zum MapServer läuft deegree ebenfalls im Kontext eines Webservers. Dieser muss in der Lage sein, in Java

erstellte Programme auszuführen. Seitens der Entwickler empfiehlt es sich den Apache Tomcat zu verwenden, da sich die Entwicklung von deegree auf diesen stützt. Es können auch andere Java-Server verwendet werden, ob deegree unter diesen aber fehlerfrei läuft, kann nicht gewährleistet werden.

- **Unterstützte Betriebssysteme:** Es werden Windows, Linux, Mac OS X und Solaris unterstützt. Somit sind alle gängigen Betriebssysteme abgedeckt.
- **Installationsaufwand:** Vor der Integration von deegree ist an erster Stelle das "Java Development Kit (JDK)" in den Versionen 6 oder 7 zu installieren. Zusätzlich benötigt man die aktuelle Java Runtime Environment (JRE). Unabhängig vom Betriebssystem ist dieser Schritt schnell abgearbeitet, da unter Windows Setup-Dateien genutzt werden können und unter Linux die interne Paketverwaltung für die Installation genutzt werden kann. Anschließend ist der Webserver (Java-Server) auf dem Betriebssystem aufzusetzen. Um deegree in den Webserver zu integrieren werden sog. WAR-Archive genutzt. Das Dateiformat WAR steht für "Web Application Archive" und beschreibt die Art und Weise wie eine Web-Anwendung nach der Java-Servlet-Spezifikation zu packen ist (ähnlich den JAR-Dateien). Nach dem Bezug des WAR-Archivs (über die Entwickler-Webseite [lat13]) ist das Paket in einem bestimmten Ordner innerhalb des Webserver zu hinterlegen. Das Paket wird dann automatisch entpackt und deegree kann über die entsprechende URL mit einem Webbrowser aufgerufen werden.
- **Konfigurationsaufwand:** Für die Konfiguration von deegree steht ein Web-Interface zur Verfügung. Dieses besteht aus einer Vielzahl an Eingabemasken für die Konfiguration der Datenquellen, Webservices und Sicherheitseinstellungen. Die Abbildung 10 zeigt das Interface in der aktuellen Version 3.2.2.

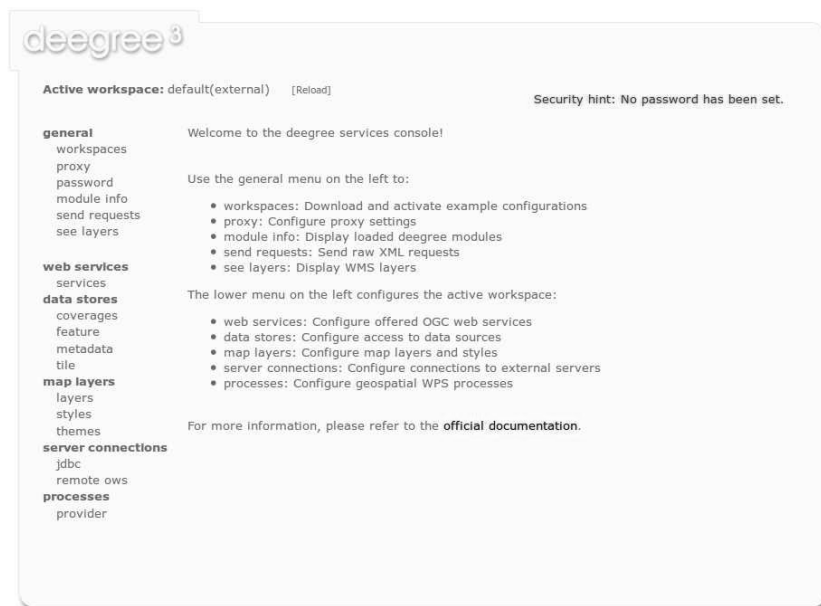


Abbildung 10: Web-Interface von deegree

Durch Anklicken eines Menüpunktes gelangt man zu den entsprechenden Konfigurationen. Diese Konfiguration basiert auf 2 Prinzipien. Entweder es existiert eine übersichtliche Eingabemaske, in der nur die Parameter gesetzt werden müssen oder es wird ein XML-Editor geöffnet. Der zweite Weg kann sich als schwierig erweisen. Es werden zwar XML-Muster zur Verfügung gestellt, welche die für die Konfiguration benötigten Tags setzen, der Inhalt ist aber manuell

einzutragen. Je nach Komplexität der Konfiguration kann die Erstellung einen erhöhten Zeitaufwand bedeuten und zusätzliche Recherchen notwendig machen. Kenntnisse über XML und den implementierten Standards und Normen aus dem Bereich der Geoinformationen sind unumgänglich. Zudem ist XML-Code zwar lesbar, dessen Interpretation hingegen kann ohne ausreichende Kenntnisse nur schwer erfolgen. Mittlerweile steht eine sehr ausführliche Dokumentation zur Verfügung, die einen detaillierten Überblick zu allen Aspekten der Konfiguration gibt.

- **Unterstützte Standards und Normen:** Da deegree auf den Standards und Normen für Geoinformationen aufbaut, liegt eine optimale Unterstützung dieser vor. Es werden die folgenden Standards und Normen unterstützt (aktuelle Version 3.2.2):

- WMS
- WFS
- CSW (Catalogue Service for the Web - Vorschrift für die Veröffentlichung von Geodaten)
- WPS (Web Processing Service - Mechanismus für die Verarbeitung von Geodaten über das Internet)

Zusätzlich stellt deegree eine auf INSPIRE basierende Sicht zur Verfügung. Dies ermöglicht die Bereitstellung und Integration von Daten, die gemäß der INSPIRE-Richtlinie zu harmonisieren sind. Dieser Fakt kann besonders für Institutionen von Bedeutung sein, die nach dieser Richtlinie die eigenen Daten zur Verfügung stellen müssen. Zusammenfassend stellt deegree eine optimale Basis dar, um die Interoperabilität der Geodaten zu gewährleisten.

- **Unterstützte Datenformate und Datenbanken:** deegree ermöglicht es ebenfalls eine Vielzahl an unterschiedlichen Raster- und Vektorformaten zu integrieren. Zu den unterstützten Formaten zählen:

- Vektorformate: ESRI Shapefile, PostgreSQL/PostGIS, Oracle Spatial/Locator, MIF, ArcSDE, GML, KML und alle relationalen Datenbanken mit JDBC-Unterstützung
- Rasterformate: PNG, GIF, JPEG, BMP, TIFF, GeoTIFF, ECW und Oracle GeoRaster

Deegree stellt ein relativ neues und OGC/ISO-konformes Softwareprodukt dar und zeichnet sich durch die INSPIRE-Sicht aus. Dieser Fakt macht deegree gerade für Ämter und Vermessungseinrichtungen interessant und kann bei dem Aufbau einer Geodateninfrastruktur enorm viel Arbeit abnehmen. Zusätzlich ist zu erwähnen, dass aufgrund der Tatsache, dass deegree von einem Unternehmen (lat/lon) entwickelt wird, ein professioneller Support genutzt werden kann. (Vgl. [lat13])

4.3.3 Geoserver

Das letzte Produkt, das vorgestellt wird, ist "GeoServer" und baut ebenfalls auf Standards und Normen auf. GeoServer wird von der OSGEO administriert und wurde im Jahr 2001 erstmals veröffentlicht. Die Entwicklung von GeoServer liegt vollständig in den Händen der Community, die aus Privatpersonen und Unternehmen besteht. GeoServer gilt zudem als Referenzimplementierung des OGC WFS, WMS und WCS und wird häufig als Kernkomponente von Geodateninfrastrukturen verwendet.

- **Unterstützte Betriebssysteme:** GeoServer zeigt sich bei der Unterstützung von Betriebssystemen sehr flexibel. Neben Windows und MAC OS X werden sämtliche Linux-Distributionen unterstützt. Somit wird es in der Regel nicht notwendig sein, ein gesondertes Betriebssystem aufzusetzen, um GeoServer in eine bestehende IT-Landschaft einzubinden.

- Installationsaufwand:** Bei der Installation wird, da es sich um auf Java basierendes Programm handelt, die Java Runtime Environment (JRE) benötigt. Eine Installation des Java Development Kit (JDK) wird nicht benötigt. Da GeoServer nur im Kontext eines Java-Webserver läuft, ist ein entsprechender Server Grundvoraussetzung. Die Entwicklung von GeoServer basiert auf dem Apache Tomcat und sollte als Webserver verwendet werden. Es gibt zwei unterschiedliche Möglichkeiten GeoServer auf einem System aufzusetzen. Entweder man nutzt ein WAR-Archiv, das auf einem bestehenden Java-Server abgelegt wird und automatisch entpackt wird oder man nutzt ein Gesamtpaket, bestehend aus einem vorkonfigurierten Apache Tomcat und GeoServer. Unter Windows nutzt man Setup-Programme um den Server inkl. GeoServer zu installieren. Dies gestaltet sich sehr einfach. Unter Linux kann über die interne Paketverwaltung der Apache Tomcat installiert werden. GeoServer, in Form eines WAR-Archivs, ist von dessen Webseite herunterzuladen und in das entsprechende Verzeichnis des Webserver zu hinterlegen. Damit ist die Installation abgeschlossen.
- Konfigurationsaufwand:** Die Administration von GeoServer erfolgt vollständig über eine webbasierte Benutzeroberfläche. An dieser Stelle liegt eine der Stärken von GeoServer. Sämtliche Konfigurationen werden mittels übersichtlichen Eingabemasken vorgenommen. Zusätzlich gibt es eine Fülle an Hilfestellungen, welche die Funktionalitäten erläutern. Auch wenn auf den ersten Blick die Konfigurationsparameter zahlreich sind und so die Komplexität steigt, eröffnet sich nach einer gewissen Einarbeitungszeit ein sehr komfortabler Weg, GeoServer zu konfigurieren. Die Abbildung 11 zeigt die Startseite in der alle relevanten Einstellungen über die linke Menüleiste erreicht werden können.



Abbildung 11: Benutzeroberfläche von GeoServer

- Unterstützte Standards und Normen:** Wie bereits in der Einführung zu GeoServer erwähnt, stützt sich die gesamte Entwicklung von GeoServer auf OGC-Konformität. Dadurch ist eine umfassende Unterstützung der Normen und Standards aus dem Bereich des Geoinformationswesens gewährleistet. Die folgende Übersicht listet alle implementierten OGC-Standards bzw. ISO-Normen auf und beziehen sich auf die aktuelle GeoServer-Version 2.3.0:

- WMS
- WFS
- WCS
- WPS

- **Unterstützte Datenformate und Datenbanken:** Wie bei MapServer und deegree wird ein breites Spektrum an verschiedensten Raster- und Vektorformaten unterstützt. Zu den unterstützten Formaten zählen:

- Rasterformate: GeoTIFF, GTOPO30, ArcGrid, WorldImages, ImageMosaics und Image Pyramids
- Vektorformate: PostgreSQL/PostGIS, ESRI Shapefile, ArcSDE, DB2, MySQL, GML, GeoJSON und Oracle Spatial

Sollte ein benötigtes Format nicht von GeoServer verarbeitet werden können, besteht die Möglichkeit zusätzliche Treiber und Programmbibliotheken (GDAL/OGR) zu nutzen. Alternativ können zusätzliche Module in GeoServer integriert werden, die eine erweiterte Formatunterstützung ermöglichen.

GeoServer stellt eine weitere, etablierte Kartenserver-Lösung zur Verfügung. Insbesondere die Konfiguration über das Web-Interface zeichnet dieses Produkt aus. (Vgl. [Ope13c])

4.3.4 Auswertung der Analyse der Kartenserver:

Prinzipiell sind die Unterschiede zwischen MapServer, GeoServer und deegree eher auf technischer Ebene und der Komplexität zu finden. Jeder der drei Kartenserver optimiert die Datenintegration in eine Geodateninfrastruktur durch die Verwendung von Normen und Standards. Die Frage nach den Versionen dieser Normen und Standards ist als Entscheidungskriterium irrelevant, da stets eine Abwärtskompatibilität der Standards und Normen gegeben ist. Die folgende Auswertung fasst alle Produkte zusammen, um die optimale Lösung für den Aufbau des Web-GIS zu finden.

- **Unterstützte Betriebssysteme** Dieses Kriterium hat keine Auswirkung auf die Wahl des Kartenservers. Alle Kartenserver bieten eine breite Unterstützung der gängigen Betriebssysteme.
- **Installationsaufwand** Hier gibt es große Unterschiede zwischen MapServer und deegree/GeoServer. Während bei MapServer eine Vielzahl an zusätzlichen Komponenten benötigt wird, gestaltet sich die Integration der Java-basierten Kartenserver wesentlich einfacher. Unter Berücksichtigung dieses Kriteriums sollte deegree oder GeoServer gewählt werden.
- **Konfigurationsaufwand** Hier zeigen sich signifikante Unterschiede in der Handhabung der Kartenserver. Während bei deegree und GeoServer sehr viele Einstellungen zur Verfügung stehen und die Komplexität dadurch stark ansteigt, werden erweiterte Kenntnisse im Umgang mit diesen Produkten benötigt. Eine automatisierte Konfiguration, wie sie gemäß der Anforderungen möglich sein soll, ist somit kaum zu realisieren. MapServer hingegen stellt durch die Konfiguration über Mapfiles eine optimale Lösung dar. Da es sich stets um eine strukturierte Text-basierte Datei handelt, können diese mit einfachen Mitteln auf Basis von Mustern und eigenen Skripten erstellt werden.
- **Unterstützte Standards und Normen** Da alle Lösungen sowohl einen WMS als auch eine WFS implementieren, kann gemäß dieses Kriteriums jeder der vorgestellten Kartenserver verwendet werden.

- **Unterstützte Datenformate und Datenbanken** Auch hier zeigen sich nur sehr geringe Unterschiede zwischen den Kartenservern. Alle gängigen Lösungen verfügen über Kartenserver, um Raster- und Vektor-basierten Formate zu verarbeiten.

Nach dieser Analyse würde sich MapServer in Verbindung mit TinyOWS am besten eignen. Das ausschlaggebende Kriterium, das diese Entscheidung begründet, liegt in der Art und Weise wie der Kartenserver konfiguriert wird.

4.4 Web-Mapping-Werkzeuge

Die Grundlagen für den Aufbau einer Infrastruktur sind durch eine Datenbank und einen Kartenserver gegeben. Schließlich wird noch eine Web-Anwendung benötigt, die es ermöglicht die Karten zu betrachten und diverse Operationen über diesen auszuführen. Um eine sog. Web-Mapping-Anwendung zu erstellen, stehen diverse Frameworks zur Verfügung. Mit den Frameworks kann man mit relativ geringem Programmieraufwand eigene Web-Mapping-Anwendungen erstellen. In der Regel sind eine Vielzahl an grundlegenden Komponenten, wie eine Kartendarstellung, Navigations- und Analysewerkzeuge und vorgefertigte GUI-Elemente in den Frameworks implementiert. Bei der Erstellung gilt es die benötigten Komponenten untereinander in Verbindung zu bringen und die einzelnen Bedienelemente anzuordnen. Die Grundlage für die Frameworks bilden häufig JavaScript-Bibliotheken. Im Folgenden werden freie Frameworks untersucht, die als Basis für die eigene Web-Mapping-Anwendung potentiell genutzt werden können.

4.4.1 OpenLayers

Ein sehr weit verbreitetes Web-Mapping-Framework ist OpenLayers (OL). Es ist eine reine JavaScript-Bibliothek, die in den meisten modernen Webbrowsern (Mozilla Firefox, Microsoft Internet Explorer) genutzt werden kann. Die OL-API ist hinsichtlich des Funktionsumfang vergleichbar mit der Google Maps-API (Framework für die Erstellung von Web-Mapping-Anwendung), mit dem Unterschied, dass OL frei ist und um weitere Funktionen erweitert und verändert werden kann und keine Nutzungsbeschränkungen und Lizenzgebühren vorliegen.

- **Installationsaufwand:** Um OL für die eigene Web-Mapping-Anwendung nutzen zu können, ist nur das Framework von der OL-Homepage herunterzuladen und auf dem eigenen Webserver abzulegen. Bei der Programmierung einer Web-Anwendung ist entsprechend auf die Bibliothek ein Verweis zu setzen und OL kann im vollem Umfang genutzt werden.
- **Flexibilität bei der Entwicklung:** Allgemein kann man eine Web-Anwendung nach seinen eigenen Vorstellungen gestalten. Dennoch gibt es einige Einschränkungen bei dem Design der Web-Oberfläche. Einige Bedienelemente der Karte können nur an vordefinierten Stellen positioniert werden. Des Weiteren ist das Aussehen der Elemente (Symbole, Farbgebung) vorgegeben. Dies führt dazu, dass eine auf OL-basierende Web-Mapping-Anwendung im Design anderen OL-Anwendungen sehr ähnlich ist. Auf der anderen Seite kann man mit sehr wenig Aufwand eine Web-Mapping-Anwendung erstellen.
- **Features:** Im Folgenden werden die wichtigsten Features der OL-API aufgelistet. Neben den Bedienelementen sind vor allem die Wege für die Einbindung von Karten von Interesse. Es muss die Möglichkeit bestehen über die standardisierten Webservices WMS und WFS auf Datenbestände zugreifen zu können.

- Navigationselemente für die Karte: Zoomen, Verschieben, Layerauswahl
- Zugriff auf Geodaten/Webservices: WMS, WFS, WFS-T, MapServer-Mapfiles und GML
- Analysewerkzeuge: Objektabfragen, Strecken- und Flächenmessung, Selektion von Features
- Weitere Komponenten: Objekt-Highlighting, Zeichenwerkzeuge, Hover-Funktion, On-The-Fly-Koordinatentransformation, Kachelung von Karten (Tiles)

Bei den gelisteten Features handelt es sich nur um eine Auswahl der elementarsten Werkzeuge, die für eine Web-Mapping-Anwendung benötigt werden. Zusammenfassend ist der Funktionsumfang von OL sehr groß und nimmt dem Entwickler viel Arbeit ab.

- **Komplexität und Programmieraufwand:** Sofern ausführliche Kenntnisse in der Programmierung mit JavaScript vorliegen, stellt die Erstellung einer OL-Anwendung keine Probleme dar. Die API ist sehr übersichtlich gestaltet und es existieren viele Beispiel-Anwendungen. Wie aufwendig und komplex die Web-Mapping-Anwendung dann tatsächlich ist, hängt von den Anforderungen ab. Für den Fall, dass nur eine simple Kartendarstellungen mit Navigationswerkzeugen benötigt wird, ist der Grad der Komplexität sehr gering. Nach einer gewissen Einarbeitungszeit stellen weitere Funktionen wie das Editieren von Kartenobjekten und die Einbindung unterschiedlicher Quellen keine Hürde mehr da. Eine Schwierigkeit, die mit allen JavaScript-Bibliotheken einhergeht, ist der interne Aufbau der Anwendung. Die Referenzierung von Objekten nach dem "Document Objekt Modell (DOM)" bedarf einer gut überlegten Struktur der Anwendung.

OpenLayers ist gerade aufgrund der Vielzahl an vorgefertigten Komponenten eine sehr gute Lösung, um Web-Mapping-Anwendungen zu erstellen. (Vgl. [Ope13e])

4.4.2 GeoExt

GeoExt ist ein weiteres Web-Mapping-Framework auf der Basis von JavaScript. GeoExt fungiert als Erweiterung zu ExtJS und OpenLayers. ExtJS ist ebenfalls ein JavaScript-Framework, das den Fokus auf interaktive web-basierte Anwendungen legt und aus einer Vielzahl an Bedienelementen besteht. Es ist zu beachten, dass ExtJS in erster Linie unter einer kommerziellen Lizenz läuft, es aber auch eine freie Version (GPL) gibt. Diese bietet zwar nicht den Funktionsumfang wie die kommerzielle Version, ermöglicht aber dennoch eine flexible Gestaltung von Web-Anwendungen. Die freie Erweiterung GeoExt nutzt das bereits genannte ExtJS-Framework und erweitert dies um Funktionen für die Verarbeitung von Geodaten. Des Weiteren wird OpenLayers für GeoExt benötigt. Es werden viele Funktionen von OL erweitert und in vollem Umfang genutzt. GeoExt beseitigt somit die Schwächen von OL im Bereich der Gestaltung von Web-Oberflächen und Anwendung eigener Designs.

- **Installationsaufwand:** Die Einrichtung von GeoExt gleicht der von OL. Es sind nur die JavaScript-Bibliotheken von GeoExt, Ext (in der GPL-Variante) und OL auf einem Webserver zu hinterlegen. Dabei ist die Wahl der jeweiligen Versionen der drei Frameworks entscheidend. Auf der Homepage von GeoExt ist explizit erwähnt, welche Versionen untereinander kompatibel sind. Die Wahl einer falschen Version führt zu Fehlern, da die Frameworks untereinander abgestimmt sein müssen.
- **Flexibilität bei der Entwicklung:** Durch die Erweiterung um Kontrollelemente (Formulare, Buttons...) und Funktionalitäten für die Verarbeitung von Geodaten stehen mehr Freiheiten bei der Entwicklung zur Verfügung als bei OL. Es können sog. Rich-Web-Mapping-Applikationen erstellt

werden, die in ihrem Aufbau und Funktionsumfang traditionellen Desktop-GIS in nichts nachstehen.

- **Features:** Neben den Features von OL, die in vollem Umfang genutzt werden können, implementiert die GeoExt-Bibliothek die folgenden Elemente:
 - Formulare für die Speicherung und Verarbeitung WFS-basierter Geodaten
 - Verschachtelte Layer-Bäume
 - Mappanel für eine flexiblere Gestaltung der Kartendarstellung und Steuerelemente
 - Druckfunktionen für Karten
 - Capabilities-Stores für die Verarbeitung von WMS- und WFS-Capabilities-Dokumenten
- **Komplexität und Programmieraufwand:** Der Aufbau und die Strukturierung einer GeoExt-Anwendung ist wesentlich komplexer als die von OL. Die Vereinigung der drei JS-Bibliotheken sorgt zum einem für eine sehr großen Satz an Funktionen und zum anderen ist stets die Frage zu beantworten, welche Bibliothek für eine bestimmte Aufgabe zu nehmen ist. Um einen Großteil der Unklarheiten zu beseitigen, stehen auf der GeoExt-Homepage Beispiele für alle Elemente zur Verfügung. Nach einer gewissen Einarbeitungszeit, weiß man, wie die Funktionen miteinander zu kombinieren sind und man kann den hohen Grad an Flexibilität bei der Gestaltung der Web-Anwendung voll ausschöpfen.

GeoExt, das Bindeglied zwischen den JS-Frameworks Ext und OL, eignet sich insbesondere, wenn die Gestaltungsmöglichkeiten von OL für eine Web-Mapping-Anwendung unzureichend sind. Abgesehen von der Komplexität, die durch den Funktionsumfang entsteht, ist die Integration der Bibliotheken, wie bei OL, sehr einfach. (Vgl. [Sen13], [Geo13])

4.4.3 MapFish

MapFish stellt in der aktuellen Version ein komplexes und umfangreiches Framework für die Erstellung von Rich-Web-Mapping-Anwendungen dar. Unter Nutzung der bereits untersuchten Frameworks OL und GeoExt werden neben diesen Client-seitigen Technologien auch Server-seitige Operationen ermöglicht, die auf der Basis von Python realisiert werden.

- **Installationsaufwand:** Im Gegensatz zu OL und GeoExt genügt es bei MapFish nicht nur die Bibliothek auf einem Server abzulegen. Im Vorfeld ist Python zu installieren. Hier ist auf die passende Version zu achten. Um eine einfache Integration des Frameworks zu ermöglichen, kann ein Installationsskript genutzt werden (über die MapFish-Homepage verfügbar). Ist diese abgeschlossen, ist zunächst zu überprüfen, ob dieses korrekt ausgeführt wurde, indem mittels der über das Skript installierten virtuellen Python-Umgebung bestimmte Befehle ausgeführt werden.
- **Flexibilität bei der Entwicklung:** Der Grad an Freiheiten für die Entwicklung wird wiederum, neben der Nutzung von OL, GeoExt und Ext, erhöht. Mit Python können Operationen, die auf dem Server ausgeführt werden, gesteuert werden. Des Weiteren verfügt MapFish über eine direkte Anbindung an eine räumliche Datenbank. Mittels SQL-Befehlen können Daten abgerufen und manipuliert werden. Für einen fehlerfreien Betrieb dieser Funktionalitäten sind mehrere Konfigurationsdateien zu editieren. Leider ist die Dokumentation an einigen Stellen lückenhaft, sodass sich der Zweck und die Konfiguration bestimmter Dateien nicht sofort erschließt.

- **Features:** Der Aufbau des Frameworks kann der Abbildung 12 entnommen werden.

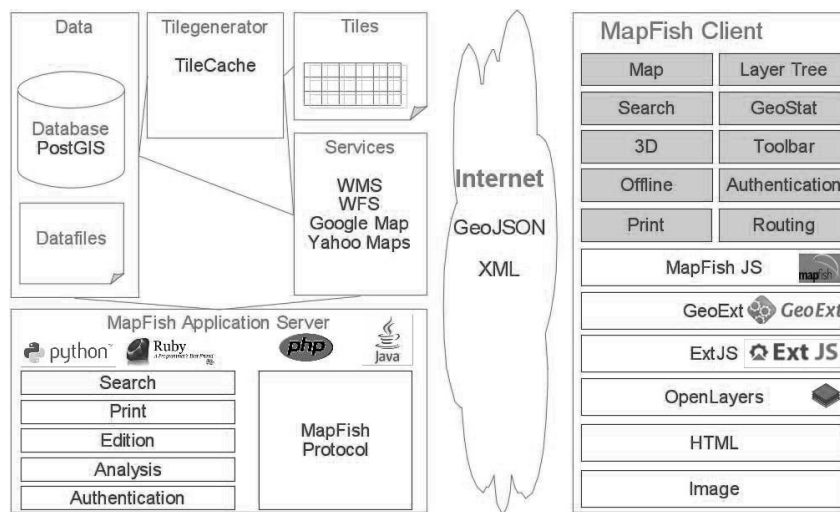


Abbildung 12: MapFish-Framework [Ope13d]

- **MapFish Client:** Der MapFish-Client stellt die für die Erstellung des Web-Mapping-Clients benötigten Komponenten zur Verfügung. Dabei handelt es sich primär um JavaScript-Bibliotheken die auf Web-Mapping spezialisiert sind (Open Layers, GeoExt, MapFish-JS) bzw. Basisklassen für die Erstellung von Webseiten (ExtJS, HTML).
 - **Internet:** Der Datenaustausch zwischen dem Client und dem Server erfolgt über XML und im Falle des Austauschs von Geodaten mittels GeoJSON (Kodierungsformat für Geodaten nach dem Simple Feature Modell).
 - **MapFish Application Server:** Die Server-seitige Bearbeitung kann über verschiedene Programmiersprachen erfolgen. Es werden Python, Ruby, PHP und Java unterstützt. Der Server stellt neben Funktionen wie das Suchen, Drucken, Editieren und Analysieren von Daten, auch einen Authentifizierungsmechanismus zur Verfügung. Der Applikation-Server agiert als Vermittler zwischen dem MapFish-Client und den folgenden Komponenten. Diese sind allerdings nur indirekt teil des Frameworks und werden von diesem nur genutzt.
 - **Services:** Die Services umfassen die nutzbaren Schnittstellen/Services für den Austausch von Geodaten, wie dem Web Feature Service und Web Map Service.
 - **Tiles/Tilegenerator:** MapFish ist in der Lage mit gekachelten Karten (Tiles) umzugehen. Die Kachelung ist ein Mechanismus zur Zerlegung großer Karten in kleinere Teile. Dies kommt insbesondere der Performance bei dem Abruf von Karten zu Gute.
 - **Data:** In welcher Form die Daten vorliegen (Datei-basiert/Datenbank) spielt für deren Nutzung für MapFish keine Rolle. Wie die Daten aufbereitet werden, obliegt den Services wie WMS.
- **Komplexität und Programmieraufwand:** Durch die zusätzliche Verwendung von Python können neben den Client-seitigen Operationen, auch auf dem Server diverse Operationen ausgelagert werden. Bei aufwendigen Analysen von Geodaten ist dies u. U. sogar notwendig, da die Leistung von Webbrowsern eingeschränkt ist. Sollten Server-seitige Funktionalitäten nicht benötigt werden, kann man auch nur auf die JS-Bibliotheken OL, Ext und GeoExt zugreifen. Der Aufwand für die Erstellung einer Web-Mapping-Anwendung ist somit stark abhängig, in wie weit der Server

involviert wird.

MapFish eignet sich vor allem für hoch-komplexe und leistungsstarke Web-Mapping-Anwendungen. Es ist abzuwägen, ob die Vorteile einer Server-seitigen Prozessierung wirklich benötigt werden, da die Einarbeitungszeit durch den Umfang des Frameworks und der lückenhaften Dokumentation lange dauern kann. (Vgl. [Ope13d])

4.4.4 Mapbender

Mapbender hat einen langjährigen Entwicklungszyklus durchlaufen und findet eine weite Verbreitung. Genau betrachtet ist Mapbender ein Content Management System für Web-Mapping-Anwendungen und Geodiensten (WMS, WFS). Die Implementierung erfolgt auf Basis von PHP und JavaScript. Des Weiteren nutzt Mapbender Technologien wie XML und JQuery. Ein Vorteil liegt darin, dass Mapbender mit vorkonfigurierten Applikationen für die Verarbeitung von Geodaten ausgeliefert wird. Die Vorlagen können nach belieben verändert werden, um so die Anwendung den eigenen Vorstellungen anzupassen.

- **Installationsaufwand:** Vor der Installation ist PHP in der Version 5+ aufzusetzen. Des Weiteren wird ein Apache HTTP Server benötigt. Sind diese Voraussetzungen erfüllt, ist Mapbender von der Homepage herunterzuladen. Mapbender ist auf den Server zu hinterlegen und anschließend ist der Server so zu konfigurieren, dass Mapbender erkannt wird. Die Konfiguration erfordert zusätzliche Eingaben auf Kommandozeilenebene und das Editieren von Dateien. Diese dienen primär der Anbindung an eine Datenbank und der Rechteverwaltung. Sind diese Schritte abgeschlossen, werden die weiteren Einstellungen über ein Web-Interface vorgenommen.

- **Flexibilität bei der Entwicklung:** Da Mapbender von Haus aus mit vorkonfigurierten Web-Anwendungen ausgeliefert wird und diese vollständig anpassbar sind, kann eine maßgeschneiderte Entwicklung der Web-Anwendung erfolgen. Des Weiteren ist die Einbindung selbst erstellter PHP-Skripte realisierbar. Auf diesem Weg kann die Web-Anwendung erweitert werden und um nicht implementierte Funktionen ergänzt werden.

- **Features:** Zu den Kernfunktionalitäten von Mapbender zählen u a.:
 - Integration von OGC-Webservices wie WMS und WFS
 - Datenmodell für die Navigation, Anzeige und Abfrage OGC-kompatibler Webservices /Geodaten
 - Dienste für die Authentifizierung
 - integrierter Proxy
 - Benutzerschnittstelle für die Administration von Nutzern und Diensten

- **Komplexität und Programmieraufwand:** Durch die Vielzahl an Verwaltungselementen bedarf es einer gewissen Einarbeitungszeit in Mapbender. Insbesondere die Einrichtung des Basissystems benötigt Kenntnisse im Servermanagement. Sofern Mapbender ordnungsgemäß integriert worden ist, erleichtert das Web-Interface weitere Einstellungen. In Anbetracht der Tatsache, dass Mapbender mehr als nur der Erstellung von Web-Mapping-Anwendungen dient, ist zwar der Aufwand größer als bei OL und GeoExt, minimiert aber den Aufwand zusätzliche und oft benötigte Sicherheitsmechanismen selbst zu implementieren.

Mapbender hat sich speziell in Behörden etabliert, da hier ein großer Fokus auf Sicherheit gelegt wird. Wenn man Mapbender als eine All-in-one-Lösung für Web-Mapping-Anwendungen ansieht, können viele Aufgaben, die neben der reinen Web-Anwendung zu tätigen sind, mit dem Framework abgedeckt werden. (Vgl. [Uni13])

4.4.5 Auswertung der Analyse der Web-Mapping-Frameworks:

- **Installationsaufwand** Die Installation von Mapbender und MapFish ist im Gegensatz zu der von Open Layers und GeoExt aufwändiger. Da aber die für Mapbender und MapFish benötigten zusätzlichen Komponenten mit geringen Aufwand installiert werden können, zeigt der Vergleich keine großen Unterschiede auf.
- **Flexibilität bei der Entwicklung** Die Möglichkeit eine eigene maßgeschneiderte Web-Mapping-Anwendung zu erstellen ist bei allen Frameworks gegeben. Allerdings haben OpenLayers und GeoExt den Vorteil, dass keine Abhängigkeit von Seiten des Servers entstehen, da für die JavaScript-Anwendung weder PHP noch Python benötigt wird.
- **Features** Hinsichtlich des Funktionsumfang sind mit allen Frameworks alle benötigten Komponenten der Web-Mapping-Anwendung realisierbar. Ebenso verfügen alle über eine WMS- und WFS-Schnittstelle.
- **Komplexität und Programmieraufwand** Hier gibt es die größten Unterschiede. Während Open Layers und GeoExt nur auf JavaScript aufbauen und es im Prinzip genügt eine HTML-Datei mit dem entsprechenden JS-Code anzulegen, ist bei MapFish und Mapbender der Aufwand signifikant höher. Es sind diverse Anpassung vorzunehmen, damit MapFish und Mapbender ordnungsgemäß in das System integriert werden können. Auch wenn diese beiden Frameworks die Option bieten in verschiedenen Sprachen zu programmieren (was häufig unumgänglich ist), führt dies zu einer steigenden Komplexität. Des Weiteren ist zu beachten, dass es möglichst einfach sein soll die Web-Mapping-Anwendung anzupassen (Funktionen, Layout), das im Falle von MapFish und Mapbender nicht ohne Weiteres möglich ist, da eine Anwendung aus mehreren Dateien besteht.

Gemäß der Auswertung potentieller Web-Mapping-Frameworks ist anzuraten sich für GeoExt in Verbindung mit OpenLayers und Ext zu entscheiden. Diese Frameworks bieten das richtige Maß an Funktionsumfang und Komplexität, um den Anforderungen an die Web-Mapping-Anwendung gerecht zu werden. Zudem lassen sich nutzerbezogene Aspekte einfacher einarbeiten. So kann beispielsweise eine gesonderte JS-Konfigurationsdatei genutzt werden, um die Anwendung zu konfigurieren.

4.4.6 Betriebssystem des Servers

Laut den Rahmenbedingungen ist ein freies Betriebssystem zu wählen und da alle zuvor untersuchten Komponenten (Datenbank, Kartenserver, Web-Mapping-Framework) eine Vielzahl an Betriebssystemen unterstützen, schränken diese die Wahl nicht ein. Unter den freien Betriebssystemen stehen primär Linux-Distributionen zur Auswahl. Diese haben sich bereits etabliert und bilden häufig die Grundlage für Server-Systeme, da hier keine Lizenzkosten, wie bei Windows anfallen und zahlreiche Linux-Versionen für Server-Architekturen ausgelegt sind. Eine sehr populäre Distribution ist "Ubuntu Server 12 LTS", die häufig eingesetzt wird und somit vielen Nutzer vertraut ist. Auch wenn im aktuellen Ranking die Distribution "Linux Mint" an erster Stelle und Ubuntu an

zweiter Stelle steht, gibt es für Mint (noch) keine Server-Variante, die für das Projekt benötigt wird.
(Vgl. [Dis13])
Somit fällt die Wahl auf Ubuntu Server in der aktuellen Version 12 LTS.

5 Konzeptioneller Entwurf des Systems

Dieses Kapitel befasst sich mit der Modellierung und Konzeption des Web-GIS. Die zentralen Aspekte, die zu klären sind, befassen sich mit der grundlegenden Architektur, unabhängig von der Software-seitigen Implementierung.

5.1 Grundgedanke der Architektur - Service-orientierte Architektur (SOA)

Die erste Fragestellung, die es zu beantworten gilt, ist die Wahl einer Architektur. Gemäß den Anforderungen, ist eine Architektur zu wählen, die eine lose Kopplung der Komponenten ermöglicht. Die Vorteile, die sich daraus ergeben, sind eine flexible Einbindung und einfache Substitution der Komponenten.

Die aktuellen Web-GIS basieren auf der sogenannten Service-orientierten Architektur (engl. service oriented architecture), kurz SOA. Diese Art eine IT-Infrastruktur aufzubauen, ermöglicht einen sehr hohen Grad an Flexibilität. Diese wird vor allem aufgrund der schnellen Entwicklung von Softwareprodukten benötigt. Wenn eine neue Softwareversion veröffentlicht wird oder ein neues geeigneteres Produkt auf den Markt kommt, ist die Aktualisierung oder der Austausch der bestehenden Komponenten unumgänglich, um die Kompatibilität zu anderen Teilnehmern des Netzwerks weiterhin zu gewährleisten.

Die SOA zeichnet sich durch die bereits erwähnte Flexibilität und Anpassungsfähigkeit aus. Des Weiteren ist die Architektur auf eine verteilte Datenverarbeitung ausgelegt. Die innerhalb der Architektur ablaufenden Prozesse, die integrierten Systemkomponenten und die Verbindungen zwischen ihnen werden mittels Webservices realisiert. Man spricht dabei von einer Modularisierung und die Webservices repräsentieren in diesem Kontext Funktionen. Dies ermöglicht es einzelne Module anzupassen, ohne andere Bestandteile in ihrer Funktionsweise zu verändern. Da die Webservices häufig Standards unterliegen, wie denen der ISO und OGC in Geodateninfrastrukturen (siehe Kapitel 2.4), müssen keine Hersteller- und Plattform-spezifischen Einschränkungen in Kauf genommen werden. Wie sich zeigt, bietet die SOA viele Vorteile bei dem Aufbau eines web-basierten GIS. (Vgl. [DAT13])

5.2 Komponenten

Die Ergebnisse der Analyse der Softwarekomponenten in Kapitel 4 legt fest, welche Software für die jeweiligen Aufgaben zum Einsatz kommen. Die Auswahl der geeigneten Komponenten stützt sich auf zwei Faktoren. Auf der einen Seite soll das System mit so wenig Ressourcen wie möglich auskommen und auf der anderen Seite soll die Konfiguration der Datenbank, des Kartenservers und der Web-Mapping-Anwendung mit eigens erstellten Programmen möglich sein. Unter Berücksichtigung dieser Maßgaben soll eine Integration des Web-GIS schnell und unkompliziert gestaltet werden.

Jene Softwarekomponenten, die diesen Anforderungen standhalten, sind:

- **Betriebssystem:** Ubuntu Server 12 LTS
- **Datenbank:** PostgreSQL mit PostGIS
- **Kartenserver:** Mapserver und TinyOWS
- **Web-Mapping-Framework:** GeoExt mit OpenLayers und ExtJS

Gemäß der ausgewählten Komponenten ist als erstes die Frage zu klären, wie diese untereinander anzuordnen sind. Die Abbildung 13 stellt die Architektur dar.

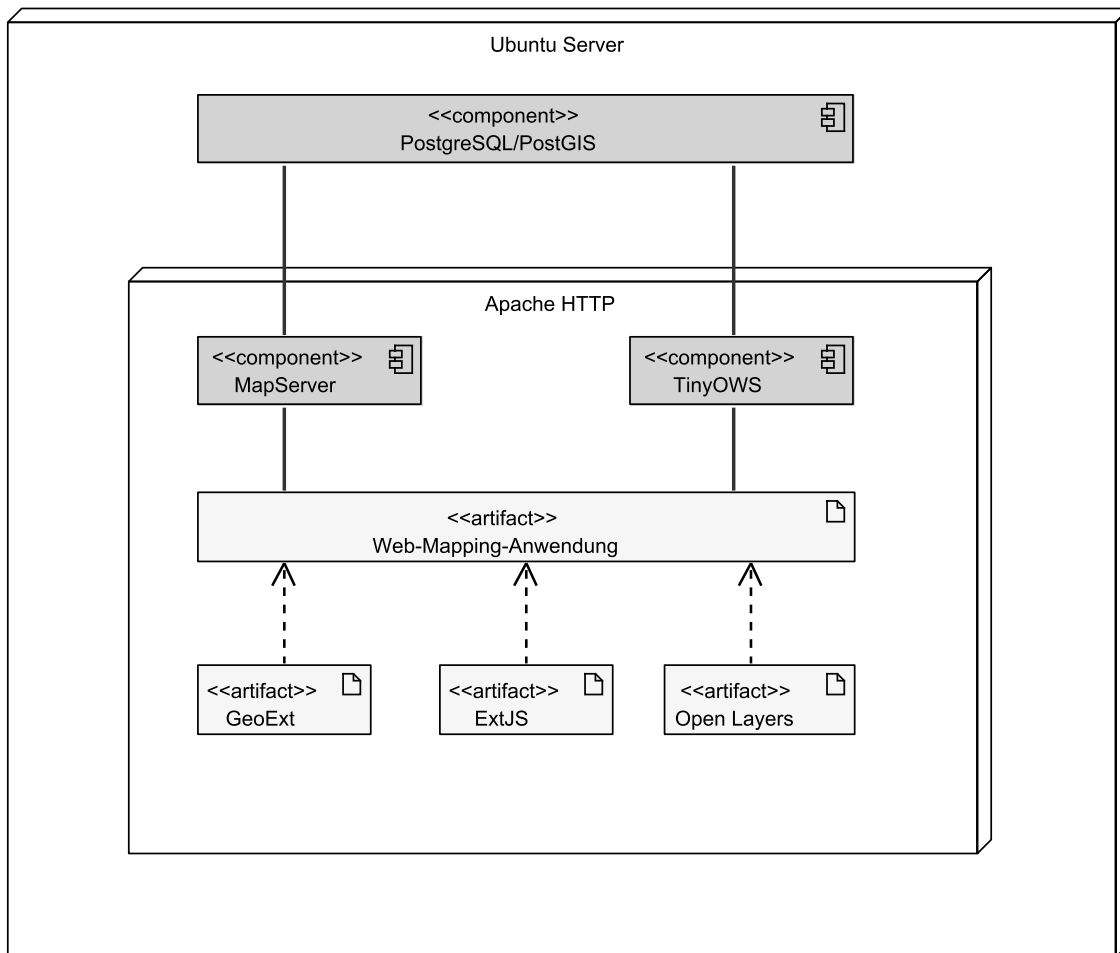


Abbildung 13: Systemarchitektur

Ein großer Vorteil ist, dass nur ein Webserver benötigt wird, um das gesamte System zum Laufen zu bringen. Die Datenbank wird nur lokal auf dem Server verfügbar sein und der Kartenserver sowie die Web-Mapping-Anwendung können beide über den HTTP-Server (Apache) veröffentlicht werden. Falls die Wahl auf GeoServer oder deegree gefallen wäre, würde ein weiterer Webserver benötigt werden, der Java-basierte Anwendungen verarbeiten kann. Damit ist in erster Linie die Bedingung erfüllt, dass das System mit geringen Ressourcen auskommen soll und so wenig zusätzliche Komponenten wie möglich benötigt werden. Nachdem nun geklärt ist, wie die Komponenten anzuordnen sind, wird im Folgenden festgelegt, über welche Schnittstellen die Kommunikation erfolgt.

5.3 Kommunikation und Schnittstellen

Die bisherigen Ausarbeitungen haben gezeigt, dass die Nutzung von Normen und Standards hinsichtlich der Schnittstellendefinition viele Vorteile mit sich bringt. Der WMS und WFS sind von zentraler Bedeutung bei der Festlegung der Kommunikationswege. Die Abbildung 14 zeigt an welchen Verbindungen diese Schnittstellen angewandt werden und wie die Kommunikation mit den anderen Komponenten erfolgt. Neben der WMS- und WFS-Schnittstelle wird zusätzlich PL/SQL (Programmiersprache für die Verbindung von SQL und prozeduralen Programmiersprachen wie C) für die Kommunikation zwischen MapServer/TinyOWS mit der Datenbank benötigt. Der Zugriff durch den Client wird über HTTP realisiert.

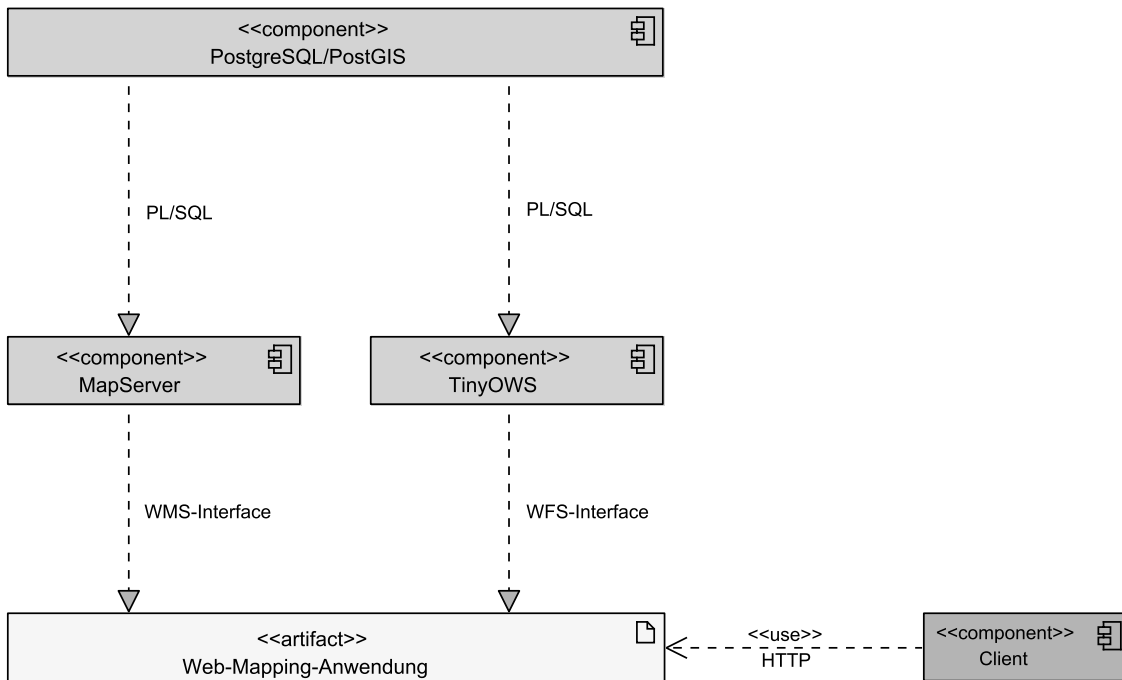


Abbildung 14: Schnittstellen des Web-GIS

Um ein besseren Überblick über die internen Vorgänge bei WMS- und WFS-Requests zu bekommen, zeigen die Abbildungen 15 und 16 die für die Web-Mapping-Anwendung relevanten Abläufe bei der Ausführung der Anforderungen.

■

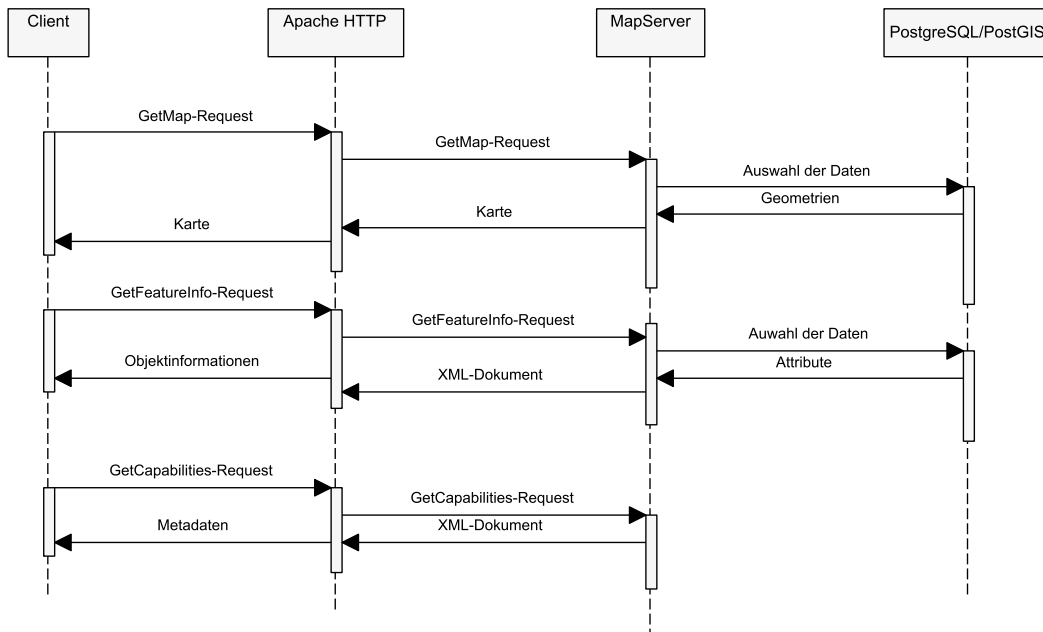


Abbildung 15: Abläufe der WMS-Requests

□

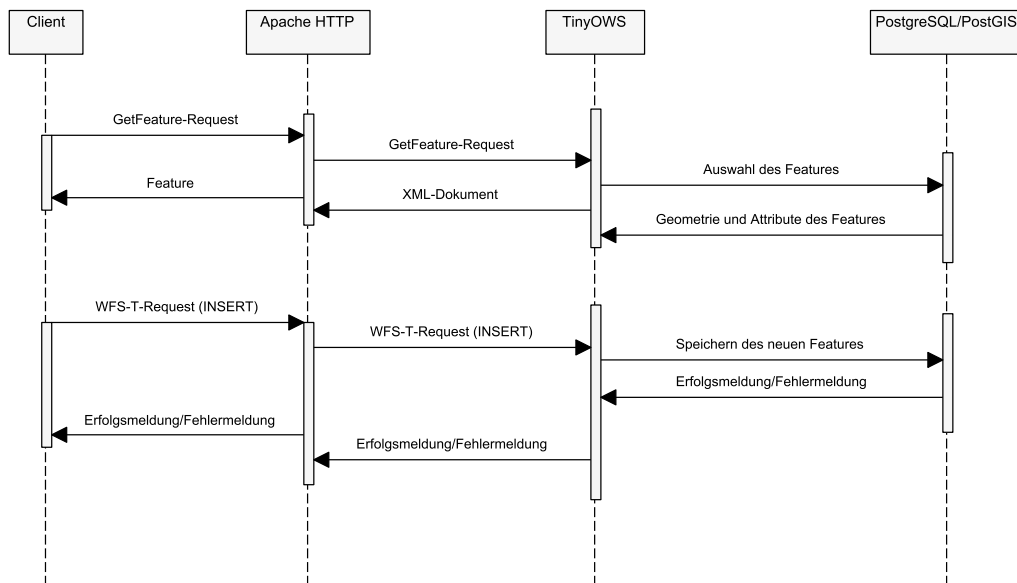


Abbildung 16: Abläufe der WFS-Requests

5.4 Konfigurationswerkzeuge

Das Ziel, das Web-GIS automatisiert und nutzerspezifisch einzurichten, bedarf zusätzlichen Programmen. Diese werden eigens für diesen Zweck erstellt und sollen die gesamten Vorgänge, die für das Aufsetzen des Web-GIS nötig sind, automatisch ausführen. Es ist zu beachten, dass aufgrund der Verwendung eines Server-basierten Betriebssystems eine Benutzeroberfläche nicht zur Verfügung steht. Somit sind die Programme so zu konzipieren, dass diese über die Kommandozeile bzw. dem Terminal/Konsole ausführbar sind. Da es sich um ein Linux-Betriebssystem handelt, bietet es sich an, Shell-Skripte für die Erstellung der Programme zu verwenden. Da eine Vielzahl an Parametern für die Konfiguration benötigt werden, können die Programme mittels Übergabeparameter die notwendigen Informationen erhalten. Des Weiteren sollten die Programme in der Form gekapselt werden, dass jedes Skript nur eine klar definierte Aufgabe übernimmt. Bevor im Kapitel 6 auf die Umsetzung eingegangen wird, ist zu untersuchen, welche freien Parameter für eine nutzerbezogene Einrichtung des Web-GIS benötigt werden. Je nach Softwarekomponente variieren diese.

5.4.1 Datenbankkonfiguration

Die Konfiguration der Datenbank benötigt zum einen die Verbindungsparameter, bestehend aus Angaben über den Nutzernamen und der Host-Adresse der Datenbank und zum anderen werden für das Anlegen einer Tabelle, die für die einzutragenden Daten im Web-GIS benötigt werden, der Tabellename, der Geometrietyp (Punkt, Linie, Fläche), und die Attribute inklusive deren Datentypen benötigt. Dabei ist zu beachten, dass für die aktuelle Entwicklung nur eine Tabelle für die eigenen Daten genutzt werden kann.

Das Datenmodell gestaltet sich, aufgrund der Möglichkeit nur eine einzelnen Tabelle für die eigenen Daten zu nutzen, sehr einfach. Unabhängig von dem Thema, das zu modellieren ist, existiert eine Tabelle, bestehend aus den folgenden Attributen:

- Primärschlüssel (wird automatisch fortlaufend generiert)
- Geometriespalte (Punkt, Linie oder Polygon)
- beliebige Anzahl an weiteren Attributen

Die simple Struktur des Modells begründet sich in erster Linie auf das Konzept der automatischen Einrichtung. Das Anlegen eines Modells, bestehend aus mehreren in Beziehung zueinander stehenden Tabellen, ist mittels eines Skripts nur schwer realisierbar. Sollte dennoch eine umfangreichere Modellierung benötigt werden, ist die Datenbankstruktur manuell mit SQL-Befehlen aufzubauen. Für die Weiterentwicklung des Web-GIS ist anzustreben, dass Automatismen entwickelt werden, um komplexe Datenmodelle über ein Skript zu realisieren.

5.4.2 Konfiguration von Mapserver und TinyOWS

Die Konfiguration von MapServer erfolgt über die im Kapitel 4.3.1 erläuterten Mapfiles. Die benötigten Parameter dienen nur einer Minimalkonfiguration der Mapfiles, reichen aber aus, um eigene Karten auf Basis der eigenen Geodaten aus der PostgreSQL-Datenbank zu erzeugen. Das Mapfile wird konform zur WMS-Schnittstelle konfiguriert, da ausschließlich diese Schnittstelle für den Abruf von Rasterkarten genutzt werden soll. Zu diesen Parametern zählen:

- Name des Mapfiles
- Titel des WMS
- URL von MapServer
- Name des Layers
- Geometrietyt
- Datenbanknutzer
- Datenbankpasswort
- Name der Datenbank
- Name der Tabelle
- Name der Geometriespalte

Für die Konfiguration des WFS-T (TinyOWS) werden zusätzlich die folgenden Parameter benötigt:

- Host-Adresse von TinyOWS
- Titel des WFS-T-Services

5.4.3 Konfiguration der Web-Mapping-Anwendung

Die Web-Mapping-Anwendung soll es ermöglichen, dem Nutzer die Wahl zu überlassen, welche Bedienelemente zur Verfügung stehen sollen. Um dies zu realisieren, wird jedes Bedienelement durch einen Parameter repräsentiert. Der Parameterwert ist entweder "TRUE", wenn das Element vorhanden sein soll oder "FALSE" für den Fall, dass dieses Element nicht integriert werden soll. Zu den Elementen gemäß den Ergebnissen des Kapitels 3 zählen:

- Navigation: Verschieben, Zoomen, Aktion rückgängig/vorwärts, auf volle Ausdehnung der Karte zoomen
- Messinstrumente für Strecken und Flächen
- Objektanfragen

- Einfügen neuer Objekte
- Übersichtskarte
- Themenauswahl (Layer)
- Koordinatenanzeige
- Legende

Mit diesen Funktionen sind alle gängigen Elemente abgedeckt und ermöglichen bei der Nutzung aller Interaktionen den Aufbau einer umfangreichen Web-Mapping-Anwendung.

5.5 Abläufe und Struktur der Web-Mapping-Anwendung

Ein weiterer Aspekt der Modellierung umfasst die Gestaltung der Web-Mapping-Anwendung. Im Vorfeld sind alle Anwendungsfälle zu untersuchen (Abbildung 17).

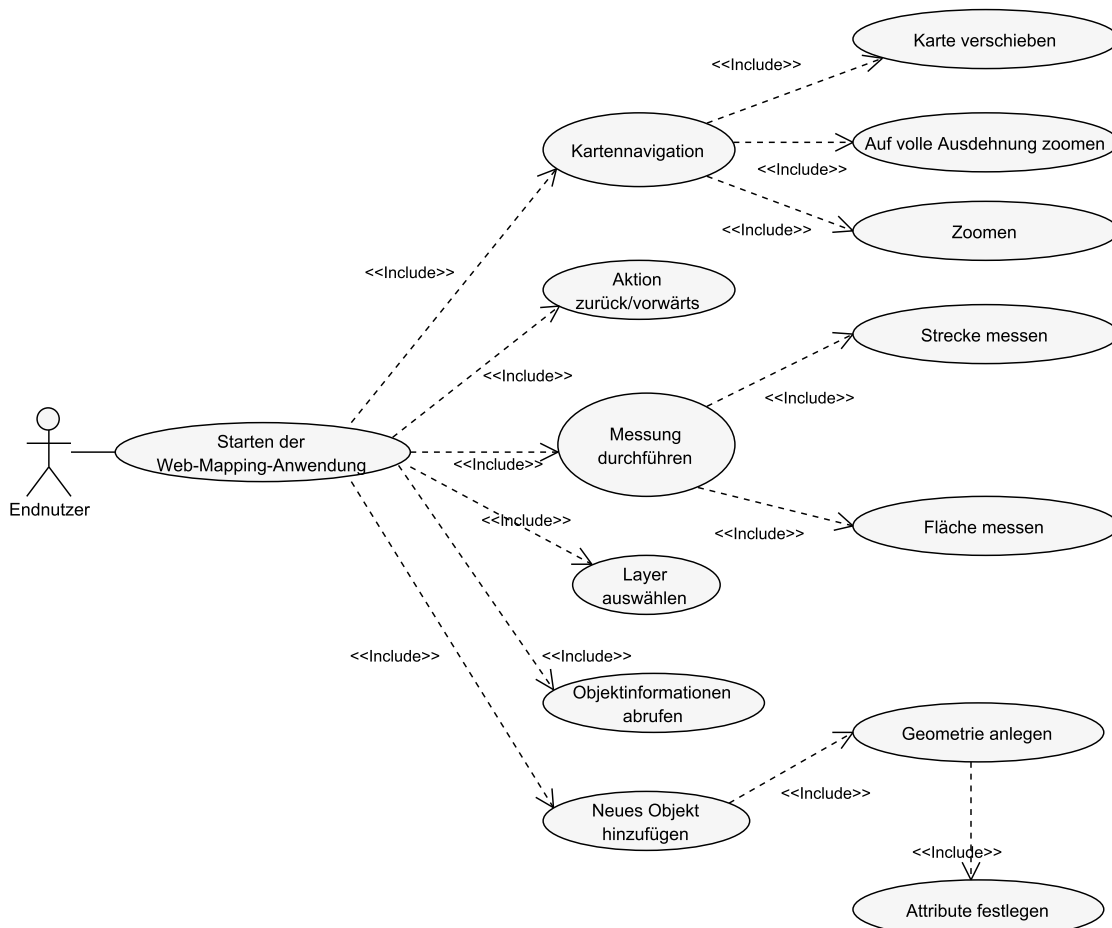


Abbildung 17: Anwendungsfälle in der Web-Mapping-Anwendung

Das Editieren von Karteninhalten der Web-Mapping-Anwendung bedarf besonderer Beachtung. Es sind mehrere Vorgaben einzuhalten, um zum einem die Benutzerführung zu optimieren und zum anderen bestimmte Sicherheitsaspekte zu berücksichtigen. Zu diesen Vorgaben zählen:

- Es ist nur möglich neue Objekte anzulegen.
- Das Verändern und Entfernen bereits eingetragener Objekte ist ausgeschlossen.

- Beim Anlegen eines neuen Objekts kann dieses optional in seiner Position korrigiert werden.
- Das Anlegen eines neuen Objekts kann abgebrochen werden.
- Beim Anlegen eines neuen Objekts sollen, nachdem die Position des Objekts festgelegt wurde, die Attribute eingetragen werden können.

Unter Berücksichtigung dieser Bedingungen ist das Editieren von Kartenobjekten wie folgt zu gestalten (Abbildung 18):

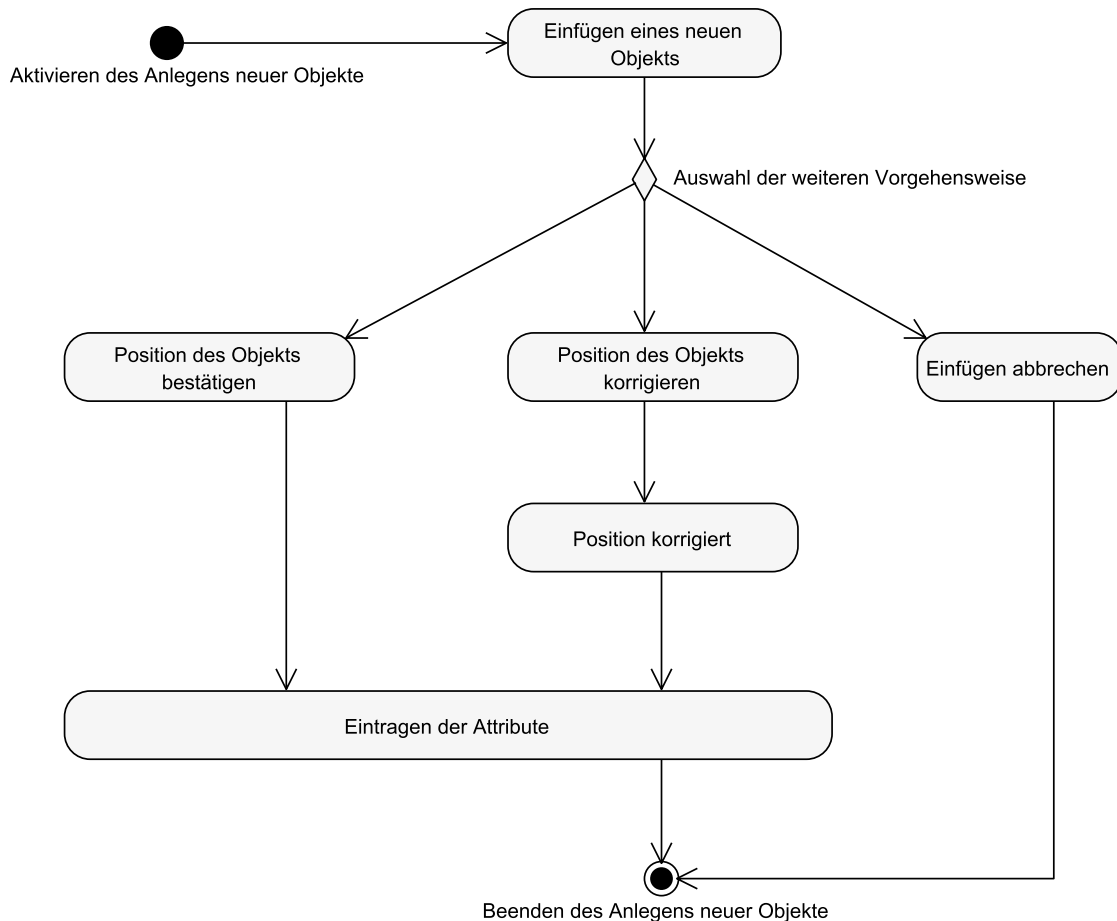


Abbildung 18: Ablauf bei dem Einfügen neuer Objekte

Damit diese Anforderungen umgesetzt werden können, ist ein Modell der Web-Mapping-Anwendung zu entwerfen. Für diese Zwecke wird ein Klassendiagramm genutzt. Es ist aber zu erwähnen, dass sich diese primär nur für objektorientierte Programmiersprachen eignen, was bei der Skriptsprache JavaScript somit nicht der Fall ist, da Strukturen wie Klassen, Methoden und das objektorientierte Verwalten von Variablen (Sichtbarkeit) nicht vorhanden sind. Dennoch lassen sich unter den folgenden Vorgaben, JS-Programme mittels Klassendiagrammen abstrahieren, indem für Klassendiagramme spezifische Elemente substituiert werden:

- Klassen werden durch ganze JavaScript-Dateien ersetzt
- Attribute entsprechen den Variablen
- Methoden werden durch JS-Funktionen ersetzt (inkl. Eingabe- und Rückgabewerte)

Die Abbildung 19 zeigt den Aufbau der Web-Mapping-Anwendung.

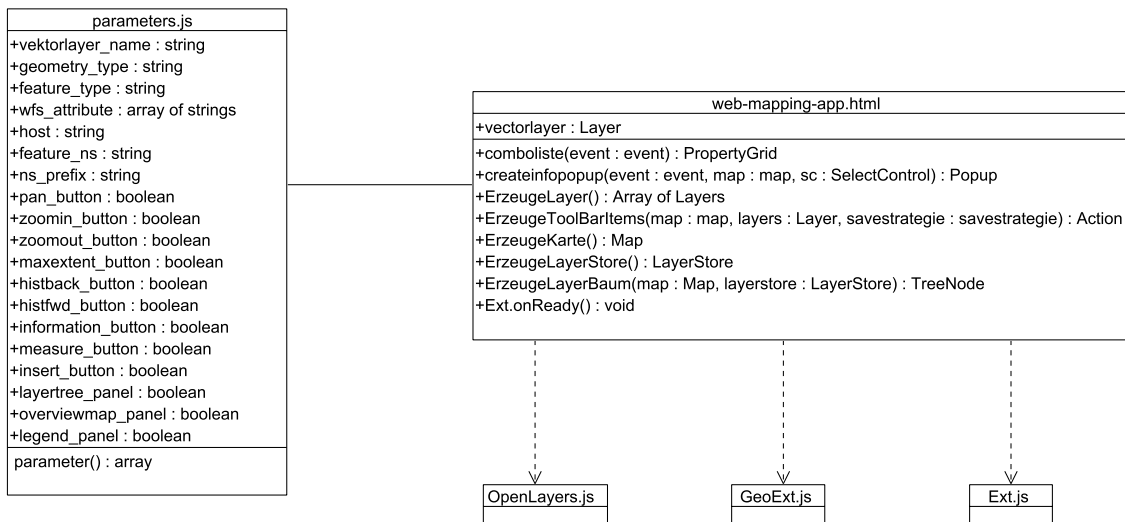


Abbildung 19: Aufbau der Web-Mapping-Anwendung

Im Folgenden werden die JavaScript-Dateien und deren Elemente erläutert.

- **parameters.js** Diese JS-Datei besteht aus allen Parametern, die im Rahmen der nutzerbezogenen Anpassungen der Anwendung benötigt werden (siehe Kapitel 5.4). Durch die Auslagerung aller freien Parameter können Veränderungen einfacher durchgeführt werden, ohne das Hauptprogramm zu verändern. Alle Parameter werden einem Array hinzugefügt und dieses wird schließlich durch den Aufruf der Funktion `parameter()` in dem Hauptprogramm `web-mapping-app.html` geladen.
- **web-mapping-app.html** Hierbei handelt es sich um die eigentliche Web-Mapping-Anwendung. Auf Basis der Datei `parameters.js` erfolgt die Ausführung der Anwendung. Das Programm ist in der Form strukturiert, dass jede der angegebenen Funktionen genau eine Aufgabe erfüllt.

Hinweis: Bei den in dem Modell genannten Datentypen handelt es sich um vorgefertigte Komponenten von `OpenLayers`, `GeoExt` und `Ext`. Unter den Webseiten [Sen13], [Geo13] und [Ope13e] können detaillierte Informationen zu den verwendeten Funktionen und Datenstrukturen abgerufen werden.

 - **comboliste** erzeugt auf Basis eines Klick-Events ein Formular. Der Inhalt des Formulars wird in Form eines Gitters (Grid) dargestellt, das aus zwei Spalten besteht. Die erste Spalte gibt die Bezeichnung des Attributs an und die Zweite dient der Eintragung des zugehörigen Wertes. Die Funktion wird im Rahmen des Anlegens neuer Objekte genutzt.
 - **createinfopoup** wird im Zusammenhang mit der Funktion für den Abruf von bereits existierenden Features genutzt. Sofern in der Anwendung die Info-Funktion aktiviert ist und ein Objekt angeklickt wird, erzeugt diese Funktion ein Popup-Fenster, das eine Übersicht zu den Attributen des ausgewählten Features beinhaltet. Das Fenster positioniert sich am Ort des Features und kann anschließend verschoben, maximiert, minimiert und geschlossen werden.
 - **ErzeugeLayer** In dieser Funktion erfolgt die Angabe der in der Anwendung verfügbaren Layer/Themen. Alle Layer werden einem Array hinzugefügt, welches bei der ersten Ausführung der Anwendung der Karte übergeben wird.
 - **ErzeugeToolBarItems** stellt alle aktivierten Komponenten der Anwendung zur Verfügung. Je nach dem welche Parameter der Datei `parameters.js` bezüglich der Anwendung (Varia-

blen mit der Bezeichnung `*_button` und `*_panel`) auf TRUE bzw. FALSE gesetzt sind, wird die Komponente aktiviert.

- **ErzeugeKarte** erzeugt das Kartenobjekt unter Angabe von diversen Parametern, wie dem Koordinatenreferenzsystem und den verfügbaren Zoomstufen.
 - **ErzeugeLayerStore** ist eine Hilfsstruktur für die Verwaltung der verschiedenen Layer. Die in der Funktion `ErzeugeLayer` generierten Themen werden einem sog. `LayerStore` hinzugefügt und können über diese Struktur verwaltet werden.
 - **ErzeugeLayerBaum** generiert einen Layer-Baum. Der Baum dient der besseren Übersicht und Kategorisierung der Layer. Die Struktur des Baums basiert auf Knoten und Blättern. Die Aufteilung der Themen erfolgt in zwei Kategorien. Der erste Kategorie besteht aus Basislayern (Geobasisdaten), die als Referenz dienen. Die Zweite beinhaltet sog. `Overlays`. Dabei handelt es sich um Themen, die den Basislayern überlagert werden und zusätzliche Informationen (Geofachdaten) darstellen.
 - **Ext.onReady** ist der Initialisierungspunkt der Anwendung. Sobald alle zu importierenden JavaScript-Komponenten geladen wurden, erfolgt die Ausführung der Anwendung.
- **OpenLayer.js/GeoExt.js/Ext.js** Alle drei JS-Dateien dienen dem Import der Funktionen und Objekten der verwendeten Web-Mapping-Frameworks. Die Dateien werden durch das Hauptprogramm eingebunden.

Als letztes ist noch zu klären, wie die Ausführung der Anwendung erfolgt. Das Schema 20 zeigt den Ablauf, der bei dem Start der Web-Mapping-Anwendung ausgeführt wird.

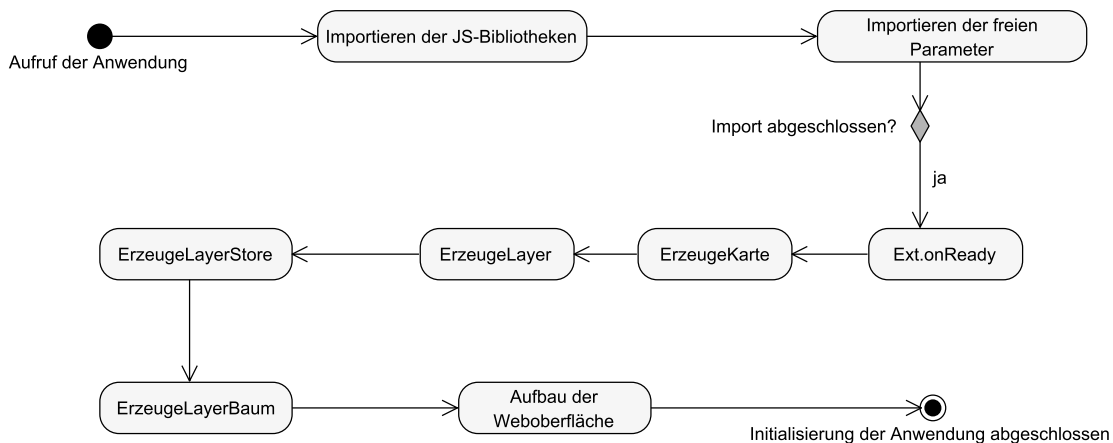


Abbildung 20: Initialisierung der Web-Mapping-Anwendung

5.6 Einrichtung des Web-GIS

Um den Anwender der Programme ein möglichst großes Maß an Komfort zu bieten, ist für die Einrichtung ein gesondertes Programm zu erstellen, in dem alle Parameter erfasst sind und die einzelnen Programme für die Einrichtung der Komponenten aufgerufen werden. Zudem werden einige Parameter gemeinschaftlich genutzt und es erübrigt sich die gemeinsamen Parameter mehrfach anzugeben. Das Programm kann als `Setup-Datei` angesehen werden.

6 Implementierung des Systems

6.1 Grundlegende Herangehensweise

Auf Basis des Konzepts gilt es nun alle Komponenten so zu integrieren, dass dem Nutzer so viel Arbeit wie möglich abgenommen wird. Um dies zu realisieren werden Shell-Skripte genutzt. Je nach zu erfüllender Aufgabe werden einzelne Skripte erstellt (Installation, Konfiguration der einzelnen Komponenten, Routine für das Aufsetzen des Web-GIS), damit im Falle eines Fehlers die Anpassung der Skripte überschaubarer ist. Zudem können die Konfigurationsprogramme über diesen Weg wiederverwendet werden, wenn es zum Beispiel notwendig ist, eine Komponente neu zu konfigurieren.

Da es sich bei der Datenbank, dem Webserver und dem Kartenserver, um fertige Programme handelt, werden diese im Detail nicht erläutert. Bei der Web-Mapping-Anwendung gestaltet sich dies anders. Diese ist die zentrale Komponente des Systems und bildet die Benutzerschnittstelle. Auch wenn die drei Frameworks OpenLayers, GeoExt und Ext genutzt werden, die eine Fülle an vorgefertigten Elementen mit sich bringen, ist die Web-Mapping-Anwendung von Grund auf neu zu erstellen. Die zugehörigen Fragestellungen, wie die Anordnung der Komponenten der Web-Mapping-Anwendung umzusetzen ist und wie die Anwendung zu strukturieren ist, und somit die Benutzervorgaben umgesetzt werden können (Auswahl der verfügbaren Funktionalitäten), ist in den Modellierungen festgelegt worden.

Des Weiteren werden die einzelnen Skripte abgehandelt, welche die Komponenten installieren und auf Basis der vom Nutzer festgelegten Parameter konfigurieren. Am Ende wird noch eine Routine benötigt, die als Einstiegspunkt für die Integration des Web-GIS genutzt wird.

6.2 Vorbemerkung zu der Implementierung

Es sind ausschließlich Shell-Skripte für die Einrichtung und Konfiguration des Web-GIS genutzt worden. Bei dem verwendeten Kommandozeileninterpreter handelt es sich um "Bash", da dieser auf einer Vielzahl von Linux-Systemen standardmäßig genutzt wird. Um die Skripte ausführen zu können, werden ROOT-Rechte benötigt, da Veränderungen an dem System vorgenommen werden. Die Einrichtung des Systems ist in erster Linie darauf ausgerichtet, dass über die Web-Mapping-Anwendung neue Objekte (Geometrien und Attribute) zu einer Karte hinzugefügt werden können. Auch wenn prinzipiell diese Funktionalität deaktiviert werden kann, sind die Skripte dafür ausgelegt (Übergabeparameter).

Die Verwendung einer grafischen Oberfläche (GUI) ist fürs Erste nicht vorgesehen, da unter dem verwendeten Betriebssystem nur eine Konsole verfügbar ist. Eine detaillierte Anleitung zu der Installation des System erfolgt in dem Kapitel 7.

6.3 Installation des Betriebssystems

Die Analyse der Softwarekomponenten ergab, dass als Server-Betriebssystem Ubuntu Server in der Version 12 (LTS) zu verwenden ist. Da es sich um ein freies Betriebssystem handelt, kann dieses ohne Restriktionen bezogen und genutzt werden. Über die Webseite <http://www.ubuntu.com> kann Ubuntu als ISO-Datei heruntergeladen werden. Die Downloadgröße liegt bei ungefähr 700 MB und sollte mit einer Breitbandverbindung in einer akzeptablen Zeit herunterzuladen sein. Ist der Download abgeschlossen stehen zwei Optionen zur Verfügung, wie das Betriebssystem aufgesetzt werden kann:

1. **Nutzung physikalischer Hardware** Sollte Ubuntu auf einen PC als alleiniges Betriebssystem installiert werden, ist zunächst die ISO-Datei auf einen Datenträger (CD/DVD) zu brennen. Alternativ kann ein bootfähiger portabler Datenträger (USB-Stick) verwendet werden. Der Datenträger ist beim Boot-Vorgang einzulegen und es ist sämtlichen Instruktionen des Installati-

onsassistenten zu folgen. Der Installationsprozess ist weitgehend automatisiert und bedarf nur wenigen Eingaben.

2. **Virtualisierung** Alternativ kann Ubuntu virtualisiert werden. Diese Methodik findet vor allem in großen Serverumgebungen Anwendung. Anstatt für jeden Server extra Hardware zur Verfügung zu stellen, nutzt man spezielle Software, um die Server zu virtualisieren. Außerdem kann man über diesen Weg auf jeden Rechner, der nicht innerhalb einer Serverumgebung stationiert ist, unterschiedliche Betriebssysteme installieren (z. Bsp. auf einem Heim-PC).

Häufig findet sich der VMWare-Player, VMWare-Server oder die VMWare-Workstation der Firma "VMWare Inc." im Einsatz. Abhängig von der Wahl der Version kann dieses Softwareprodukt kostenlos genutzt werden. Unter der Adresse <http://www.vmware.com/de/> kann das Programm bezogen werden. Als Host-System für die virtuelle Maschine kann Linux und Windows verwendet werden.

Nachdem das gewünschte Produkt (für die Nutzung am Heimrechner kann der VMWare-Player verwendet werden, für Serverfunktionalitäten der VMWare-Server) heruntergeladen und installiert ist, kann der Installationsvorgang des Betriebssystems erfolgen. Es stehen diverse Werkzeuge zur Verfügung, mit denen sehr einfach Betriebssysteme virtualisiert werden können. Im Prinzip ist im Rahmen des Anlegens einer neuer virtuellen Maschine der Pfad zu der ISO-Datei anzugeben und Angaben über den Nutzer (Name, Passwort) zu treffen. Der restliche Vorgang läuft vollkommen automatisch ab und liefert am Ende ein fertiges Betriebssystem.

6.4 Installationskript

Um das Web-GIS mit all seinen Komponenten einzurichten, wird ein Skript benötigt, das weitgehend automatisch alle Softwarekomponenten installiert. Da Ubuntu über eine interne Paketverwaltung (APT) verfügt, können über diese die Programme installiert werden. Der Vorteil liegt vor allem darin, dass die Programme bei der Nutzung von APT an die korrekten Stellen im Dateisystem hinterlegt und konfiguriert werden. Da aber nicht alle Komponenten über APT verfügbar sind, ist es teilweise notwendig den entsprechenden Quellcode eines Programms zu kompilieren. Diese Vorgänge sind im Rahmen des Skripts zu automatisieren.

Das Skript mit dem Namen "install_all.sh" installiert alle Komponenten. Unter Verwendung der Paketverwaltung ist das Kommando "sudo apt-get install [Name des Pakets]" zu nutzen. Mittels "sudo" erhält man ROOT-Rechte (unter Eingabe des Passworts des Nutzer mit den entsprechenden Rechten) und kann das System verändern. Für den Fall, dass eine Komponente kompiliert werden muss, ist wie folgt vorzugehen:

1. Wechsel in das Verzeichnis des zu kompilierenden Programms mittels "cd"
2. "./configure" für die Konfiguration des Programms
3. "make" für das Erstellen des Programms
4. "sudo make install" für die Integration des Programms in das Betriebssystem

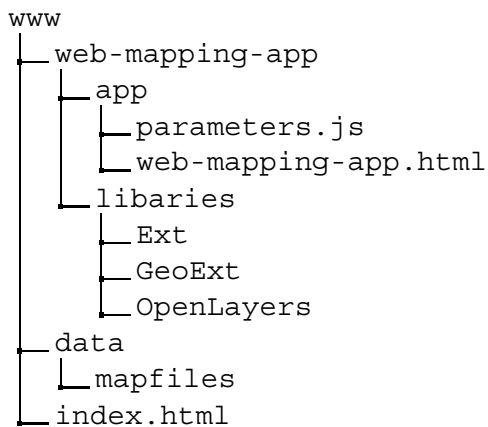
Es ist zu beachten, dass der aktuelle Nutzer Schreib- und Leserechte für das Verzeichnis des Programms besitzen muss.

Selbst erstellte Komponenten wie die Web-Mapping-Anwendung und den zugehörigen Frameworks werden an die entsprechenden Stellen im System hinterlegt. Nach diesem Schema erfolgt die komplette Installation aller für das Web-GIS relevanten Komponenten. Die Reihenfolge der Installationen gestaltet sich wie folgt:

1. Apache Webserver (APT)

2. PostgreSQL mit der Erweiterung PostGIS (APT)
3. MapServer (APT)
4. TinyOWS (Kompilieren)
5. Web-Mapping-Anwendung, dessen Startseite und zugehörigen Frameworks (Kopieren)

Im Zuge des Kopierens der Web-Mapping-Anwendung wird die Ordnerstruktur des Apache Webserver angelegt. Diese ist gemäß der Abbildung strukturiert und wird in dem Verzeichnis des Apache (/var/www/) angelegt.



Nachdem das Skript ausgeführt ist, können die einzelnen Komponenten konfiguriert werden.

6.5 Konfigurationsskripte

Jede der Komponenten (Datenbank, MapServer/TinyOWS, Web-Mapping-Anwendung) sind an den Vorgaben des Nutzers anzupassen. Wie dies realisiert ist, wird im Folgenden erläutert.

6.5.1 Datenbankkonfiguration

Das Skript "config_db.sh" dient der Einrichtung der Datenbank. Der Ablauf des Skripts ist wie folgt:

1. Für die Nutzung räumlicher Daten wird ein Muster benötigt, das bei der Erzeugung neuer Datenbanken genutzt wird, um die Datentypen und Funktionen für die Verarbeitung von Geodaten zu integrieren. Man spricht dabei von dem "postgis_template". Um dieses zu generieren wird mittels PL/SQL auf die Datenbank zugegriffen. Des Weiteren werden zwei SQL-Skripte benötigt, die im Rahmen des PostGIS-Installation mitgeliefert werden. Das Template wird in Form einer Tabelle angelegt, die zwei SQL-Skripte werden auf dieser angewandt und integrieren die räumlichen Funktionen und Datentypen. Schließlich werden die Rechte für diese Tabelle gesetzt.
2. Da das Passwort für die Datenbank noch zu setzen ist, wird automatisch auf die Datenbank gewechselt. Unter Verwendung des Kommandozeilenwerkzeugs für PostgreSQL (psql) wird dann ein Passwort vergeben. Damit der Nutzer erfährt, welche Eingaben zu tätigen sind, erfolgen diverse Anweisungen bei der Ausführung des Skripts.
3. Dieser Schritt umfasst das Anlegen einer neuen Datenbank. Die Datenbank dient den Informationen, die innerhalb der Web-Mapping-Anwendung visualisiert werden und editiert werden können (Anlegen neuer Datensätze).

4. Der nächste Schritt ist optional. Sofern gewollt, werden automatisch OSM-Datensätze (Region Mecklenburg-Vorpommern) in die Datenbank eingespielt. Dies kann ggf. von Interesse sein, falls die OSM-Daten für die eigene Datenverarbeitung benötigt werden.

Das Skript "config_db.sh" bedarf keiner Parameter, da sämtliche Vorgänge für eine nutzerbezogene Einrichtung nicht relevant sind.

Das zweite Skript, das im Rahmen der Einrichtung der Datenbank benötigt wird, trägt die Bezeichnung "create_table.sh". Mittels dieses Skripts wird eine neue räumliche Tabelle für die eigenen Daten angelegt. Damit das Skript funktioniert sind mehrere Übergabeparameter zu setzen. Zu diesen Parametern zählen (Reihenfolge ist gemäß den folgenden Angaben zu wählen):

```
bash create_table.sh [DB-User] [Tabellenname] [Geometrietyp(
  POINT|LINESTRING|POLYGON)] [Attribut 1] [Datentyp 1] [
  Attribut 2] [Datentyp 2]...
```

Neben dem Datenbanknutzer (DB-USER) und dem Namen der Tabelle ist zusätzlich der Geometrietyp zu wählen. Pro Tabelle kann nur ein Typ verwendet werden. Die folgenden Parameter legen fest, welche Attribute in der Tabelle angelegt werden sollen. Die Attribute sind nach dem Schema "Name-des-Attributs Datentyp" dem Skript-Aufruf anzuhängen. Es kann eine beliebige Anzahl an Attributen verwendet werden.

Nach der Ausführung der zwei Skripte ist die Datenbank vollständig eingerichtet und kann genutzt werden.

6.5.2 Konfiguration von MapServer/TinyOWS

Um die Funktionalität für das Hinzufügen neuer Daten über die Web-Mapping-Anwendung zu realisieren, wird ein WFS-T benötigt. Auch wenn eine Konfiguration des WFS-T über Mapfiles realisiert werden kann, gibt es eine effizientere Methode. TinyOWS kann auch über eine XML-Datei gesteuert werden. Die XML-Datei (Ort der Datei: /etc/tinyows.xml) kann mit wenigen Handgriffen angepasst werden und ist wesentlich übersichtlicher als ein Mapfile, da weniger Parameter benötigt werden. Bevor auf den Inhalt dieser Datei eingegangen wird, wird erläutert, wie das Skript (config_wfs_t.sh) zu nutzen ist.

```
bash config_wfs_t.sh [Host-Adresse] [DB-User] [DB-Passwort] [DB
  -Name] [WFS-T-Titel] [Tabellenname lt. DB] [Layer-Titel]
```

Die Parameter bestehen aus Angaben zu der eigenen Datenbank und drei Parametern für die Namensgebung des WFS-T. Ist das Skript ausgeführt, wird die XML-Datei erzeugt. Der Inhalt dieser Datei sieht wie folgt aus:

```

<tinyows online_resource="\http://host/cgi-bin/tinyows"
schema_dir="/usr/local/share/tinyows/schema/">

<pg host="localhost" user="User" password="Passwort"
dbname="Test_DB" port="5432"/>

<metadata name="TinyOWS_Server" title="WFS-T" />
<layer retrievable="1"
      writable="1"
      ns_prefix="tows"
      ns_uri="www.tinyows.org"
      name="Tabelle"
      title="WFS-Titel" />
</tinyows>

```

Die entscheidenden Parameter für einen funktionsfähigen WFS-T sind die für die Verbindung zu der Datenbank (pg-Tag) und die Angaben zu dem Layer (layer-Tag). Der Parameter "name" innerhalb des Layer-Tags muss exakt der Bezeichnung der Tabelle entsprechen, die für den WFS-T genutzt werden soll (Case-Sensitiv).

In der aktuellen Version des Skripts ist es vorerst nur möglich einen Layer im WFS-T zu nutzen. Es besteht aber die Möglichkeit, manuell die Datei anzupassen. Falls ein weiterer Layer für den WFS-T verfügbar sein soll, ist nur der Inhalt des layer-Tag zu kopieren und anzupassen. Da aber in der Web-Mapping-Anwendung auch nur ein Layer editiert werden kann, ist dies nicht weiter notwendig. Für spätere Anpassungen des Systems kann das Skript in der Form modifiziert werden, dass mehrere Layer/Tabellen editierbar sind.

6.5.3 Konfiguration der Web-Mapping-Anwendung

Das letzte Skript, das im Rahmen der Einrichtung benötigt wird, trägt die Bezeichnung "config_web_app.sh" und wird genutzt, um die Web-Mapping-Anwendung anzupassen. Das Skript erzeugt die JavaScript-Datei "parameters.js", die für die nutzerspezifischen Anpassungen der Anwendung genutzt werden. Damit die Anpassungen realisiert werden können, werden eine Vielzahl an Parametern benötigt.

```

bash config\_web\_app.sh [Host des WFS-T] [Anzeigename des WFS-T
-Layers] [Geometrietyp des WFS-T-Layers] [Name der Tabelle
des WFS-T] [Verschieben] [Zoomin] [Zoomout] [Maximale
Ausdehnung] [Schritt zurueck] [Schritt vorwaerts] [Info] [
Messung] [Einfuegen] [Layerbaum] [Uebersichtskarte] [Legende]
[Attribut 1] [Attribut 2] ...

```

Die ersten vier Parameter konfigurieren den WFS-T-Layer in der Anwendung. Die folgenden 12 Parameter legen fest, welche Bedienelemente in der Anwendung aktiviert werden sollen. Die Werte sind für eine Aktivierung auf TRUE zu setzen bzw. auf FALSE, falls dieses zu deaktivieren sind. Die Anzahl der letzten Parameter ist variabel. Sie werden genutzt um anzugeben, welche Attribute des WFS-T-Layers anzuzeigen sind. Dies hat Auswirkungen auf zwei Funktionen. Zum einen auf die Info-Funktion, bei dessen Aufruf ein Pop-Fenster erzeugt wird, das die Eigenschaften des angewählten Objekts darstellt und zum anderen legen diese Parameter fest, welche Attribute durch den Nutzer bei dem Anlegen eines neuen Objekts editiert werden können. Es ist zu beachten, dass es nicht zwingend notwendig ist, alle in der Tabelle vorhandenen Attribute zu nutzen. Damit die Attribute erkannt werden, ist auf deren exakte Schreibweise zu achten.

Der nächste Schritt befasst sich mit der Konfiguration eines Proxy. Damit der WFS-T durch die Anwendung genutzt werden kann, muss die "Same-Origin-Policy" von JavaScript umgangen werden. Diese verhindert, dass XML-HTTP-Requests (wie sie bei dem Anlegen und Abfragen von

Objekten durch die Anwendung genutzt werden) zwischen der Web-Anwendung und dem WFS-T ausgeführt werden können, sofern diese zwei Komponenten eine unterschiedliche IP/Port haben. Da dies bei der Verwendung des Web-GIS stets der Fall ist, muss dieser Mechanismus umgangen werden. Open Layers beinhaltet ein CGI-Programm, um genau dieses Problem zu umgehen. Das Programm mit dem Namen "proxy.cgi" wird automatisch in das CGI-Verzeichnis des Webservers kopiert. Anschließend ist das Programm zu editieren. Die Programm-Datei wird durch das Skript in einem Editor geöffnet. Die Datei ist dann gemäß der Hilfestellung des Skripts zu ändern. Nachdem die Konfiguration abgeschlossen ist, wird die erzeugte JavaScript-Datei (parameters.js) in das Verzeichnis der Web-Mapping-Anwendung kopiert.

6.6 Skript für die Ersteinrichtung des Web-GIS

Wie zu sehen ist, sind für die Einrichtung des Systems viele Parameter zu setzen und mehrere Skripte auszuführen. Auch wenn jederzeit die Möglichkeit besteht, jedes Skript einzeln aufzurufen, bietet es sich an ein extra Skript zu nutzen. Dieses fungiert als ein Setup und ist wesentlich benutzerfreundlicher. Das Skript trägt den Namen "setup.sh" und bedarf keiner Übergabe von Parametern. Allerdings ist die Datei vor dessen Ausführung zu editieren. Die Anpassungen beziehen sich auf die Parameter, die an die einzelnen Skripte zu übermitteln sind. Der Anfang des Skripts ist wie folgt aufgebaut.

```
#Parameter fuer die Einrichtung einer raeumlichen Datenbank
datenbanknutzer=postgres
tabellenname=hotspots
geometriety=POINT

#An dieser Stelle koennen Attribute/Datentypen hinzugefuegt
  werden

#Beispiel
attribut1=bezeichnung
datentyp1=VARCHAR

#Parameter fuer WFS-T (TinyOWS)
hostadresse=localhost
datenbankpasswort=postgres
datenbankname=hotspots
titel=hotspots
layertitel=hotspots

#Parameter fuer Web-Mapping-App
pan=true
zoomin=true
zoomout=true
maxextent=true
histback=true
histfwd=true
information=true
measure=true
insert=true
layertree=true
overviewmap=true
legend_panel=true
```

Wie zu sehen ist, sind dies alle bisher benötigten Parameter der einzelnen Skripte. Zudem werden einige Parameter (Datenbank) gemeinschaftlich genutzt. Sämtliche Variablen sind den eigenen Bedürfnissen anzupassen. Es ist zu beachten, dass unter dem Punkt an dem die Attribute und deren Datentypen angegeben sind, die Liste der Variablen anzupassen ist (siehe Beispiel).

Abhängig von der Anzahl der Attribute und zugehörigen Datentypen sind die Aufrufe der Skripte "create_table.sh" und "config_web_app.sh" minimal zu modifizieren. Wie dies zu tätigen ist, kann in der vorhergehenden Erläuterungen zu den jeweiligen Skript nachgelesen werden.

Nachdem alle Änderungen am Skript eingearbeitet sind, kann dieses mittels

```
bash setup.sh
```

aufgerufen werden und es erfolgt die vollständige Einrichtung des Web-GIS.

6.7 Zusätzliche Skripte für Anpassungen des Web-GIS

Zu den bereits erwähnten Skripten sind noch drei weitere Programme erstellt worden, mit deren Hilfe Shapefiles in die Datenbank überführt und Mapfiles generiert werden können.

Das Skript **insert_data.sh** (Shapefile-Import) ist folgendermaßen zu nutzen:

```
bash insert_data.sh [Shapefile ohne Dateierdung] [Datenbank] [
  Tabellename] [DB-User]
```

Das Skript generiert auf Basis des Shapefiles eine SQL-Datei. Das SQL-Skript wird dann gemäß der übergebenen Datenbankparameter ausgeführt und integriert die Daten.

Um MapServer/TinyOWS zu konfigurieren, können alternativ die Skripte "create_wms_mapfile.sh" und "create_wfs_mapfile.sh" genutzt werden.

create_wms_mapfile.sh und **create_wfs_mapfile.sh** erzeugen jeweils ein Mapfile und werden genutzt, um Daten aus der Datenbank über einen WMS bzw. WFS zu visualisieren. Die Mapfiles entsprechen einer Minimalkonfiguration. Die Skripte sind mit den folgenden Parametern aufzurufen:

```
bash create_wms_mapfile.sh [MapFile-Name] [WMS-Titel] [Server-
  URL] [Layer-Name] [Geometrietyp] [DB-User] [ DB-Passwort] [DB
  -Name] [Tabellename] [Name der Geometriespalte]
```

```
bash create_wfs_mapfile [MapFile-Name] [WFS-Titel] [Server-URL]
  [Layer-Name] [Geometrietyp] [DB-User] [ DB-Passwort] [DB-Name
  ] [Tabellename] [Name der Geometriespalte]
```

Nachdem die Skripte entsprechend dieser Vorgaben aufgerufen wurden, wird ein Mapfile erzeugt, welches automatisch auf dem Webserver hinterlegt wird. Diese beiden Skripte werden in der Installationsroutine nicht genutzt, können aber einzeln ausgeführt werden, um schnell eigene Mapfiles zu erzeugen. Des Weiteren müssen die Layer in der Web-Mapping-Anwendung manuell ergänzt werden, die auf Basis der erzeugten Mapfiles angezeigt werden sollen.

6.8 Web-Mapping-Anwendung

Die Modellierung der Web-Mapping-Anwendung im Kapitel 5.5 zeigt, wie die Anwendung aufzubauen ist. Im Prinzip basiert die Anwendung, neben den genutzten Web-Mapping-Frameworks, auf zwei Dateien. Die Datei "parameters.js" ist eine reine JavaScript-Datei und besteht aus zwei Komponenten. Zum einem sind alle Parameter für die nutzerbedingten Anpassungen enthalten und zum anderen eine Funktion, die alle Parameter in Form eines Arrays an die Hauptanwendung übermittelt. Die Hauptanwendung ist in der Datei "web-mapping-app.html" implementiert. Die

HTML-Datei besteht hauptsächlich aus JavaScript-Code. Dieser Code ist entsprechend des Modells strukturiert und kapselt die Funktionen in der Form, dass jede nur eine bestimmte Funktion umsetzt.

An dieser Stelle wird auf weitere Details zu der Umsetzung verzichtet. Unter Verwendung des Quelltextes der zwei Dateien und den Modellierungen aus Kapitel 5.5 kann man sich genauer über die konkreten Inhalte der Funktionen informieren.

Da die Anwendung im Rahmen der Entwicklung testweise aufgesetzt ist, wird die Web-Oberfläche des Web-GIS erläutert. Die Anwendung ist mit allen verfügbaren Bedienelementen eingerichtet (Abbildung 21).

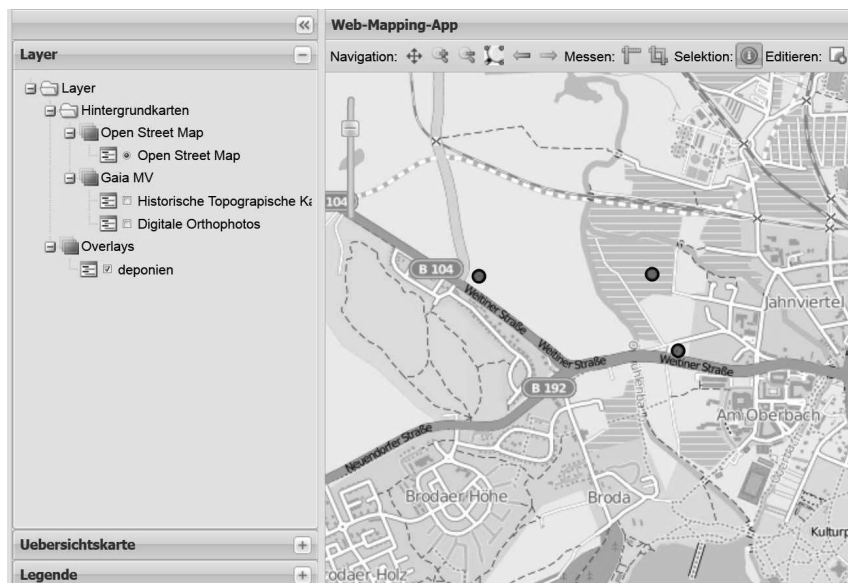


Abbildung 21: Benutzeroberfläche der Web-Mapping-Anwendung

Die Werkzeugleiste baut sich in der Maximalkonfiguration gemäß der Abbildung 22 auf.

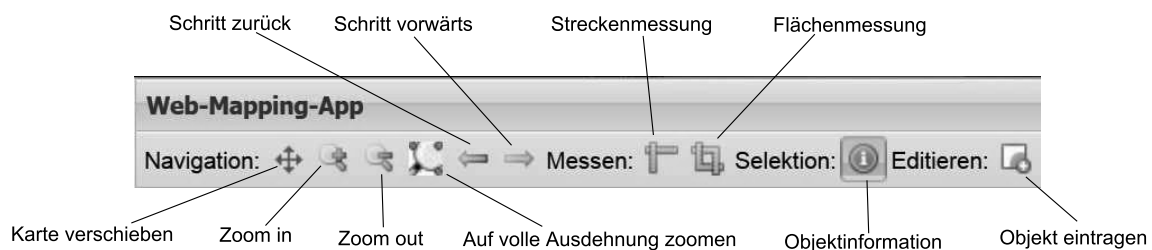


Abbildung 22: Werkzeugleiste der Web-Mapping-Anwendung

Das Seiten-Panel (Abbildung 23) besteht aus drei Elementen. Zu diesen zählen, die Themenauswahl, die Übersichtskarte und die Legende. Es kann immer nur jeweils eines der drei Elemente eingeblendet sein. Das Einblenden erfolgt über den Knopf mit dem "+". Zusätzlich kann das gesamte Panel komplett ein- und ausgeblendet werden, indem der Button mit dem Pfeil gedrückt wird.



Abbildung 23: Elemente des Seite-Panels

6.9 Konfiguration des Webservers

Eine automatisierte Einrichtung des Webservers ist in dieser Version des Web-GIS nicht möglich. Im Kapitel 7 werden kurz die Schritte erklärt, die notwendig sind, um den Webserver einzurichten.

6.10 Aufbau und Inhalte des Installationspakets

Alle in diesem Kapitel genannten Skripte und Programme sind im Form eines Installationspakets zusammengefasst. Das Paket ist wie folgt strukturiert:

```

Installationspaket
├── programme
│   ├── libxml2
│   ├── tinyows
│   └── proxy.cgi
├── skripte
│   ├── config_db.sh
│   ├── config_web_app.sh
│   ├── config_wfs_t.sh
│   ├── create_table.sh
│   ├── create_wfs_mapfile.sh
│   ├── create_wms_mapfile.sh
│   ├── insert_data.sh
│   └── install_all.sh
├── web-mapping-app
│   ├── app
│   │   └── web-mapping-app.html
│   └── libraries
│       ├── Ext
│       ├── GeoExt
│       └── OpenLayers
└── setup.sh
  
```

Das Paket ist auf dem Betriebssystem zu hinterlegen. Das Verzeichnis kann frei gewählt werden. Es empfiehlt sich aber das Home-Verzeichnis zu verwenden, da für die Ausführung der Skripte Ausführungs-, Lese- und Schreibrechte benötigt werden.

Damit sind alle relevanten Erläuterungen zu der Umsetzung des Konzepts abgehandelt. Um zu zeigen, wie das System unter Verwendung der implementierten Skripte funktioniert, wird im folgenden Kapitel eine exemplarische Einrichtung des Web-GIS vorgenommen.

7 Exemplarische Umsetzungen eines Web-GIS

Dieses Kapitel befasst sich mit der nutzerbezogenen Einrichtung des Web-GIS. Mittels der im vorhergehenden Kapitel erläuterten Shell-Skripte werden zwei exemplarische Umsetzungen vorgenommen. Die Umsetzungen befassen sich jeweils mit einer spezifischen Problemstellung. Es soll gezeigt werden, wie flexibel und einfach das System für die eigenen Zwecke angepasst werden kann.

7.1 Problemstellungen für die Umsetzungen des Web-GIS

Die erste Umsetzung hat das Ziel, ein ähnliches System wie "Klarschiff-HRO" (siehe Kapitel 3.2.2) zu implementieren. Die Web-Mapping-Anwendung soll der Erfassung von illegalen Mülldeponien dienen. Diese werden durch Punkte (Geometrietyp: POINT) dargestellt. Um die illegale Mülldeponie genauer zu beschreiben, werden die folgenden Eigenschaften verwendet:

- **Datenbank- und Tabellenname:** deponien
- **Attribut 1:** Abfallart (Restmüll, Sperrmüll, Kunststoffe...) → Repräsentiert als Zeichenkette (VARCHAR)
- **Attribut 2:** Fläche in m² → Repräsentiert als ganze Zahl (INTEGER)

Die Web-Mapping-Anwendung soll aus allen verfügbaren Bedienelemente bestehen.

Die zweite Umsetzung befasst sich mit einer Umweltthematik. Es sind Waldgebiete über die Web-Mapping-Anwendung zu erfassen. Die Gebiete werden durch geschlossene Flächen (Geometrietyp: POLYGON) abgebildet. Ein Waldgebiet ist durch die folgenden Eigenschaften spezifiziert:

- **Datenbank- und Tabellenname:** waldgebiete
- **Attribut 1:** Waldart (Laub-, Misch- und Nadelwald) → Repräsentiert als Zeichenkette (VARCHAR)
- **Attribut 2:** Zustand (Note 1 (gesund) bis 6 (krank)) → Repräsentiert als ganze Zahl (INTEGER)

Die folgenden Bedienelemente sind in dieser Web-Mapping-Anwendung zu aktivieren:

- Verschieben der Karte (pan)
- Zoomen (zoomin/zoomout)
- Anzeigen von Objektinformation (information)
- Eintragen von Objekten (insert)
- Layerauswahl (layertree)

Neben den o. g. Parametern werden zusätzliche Angaben benötigt. Diese dienen primär der Konfiguration der Datenbank, des Webservers und des Web Feature Service. Es werden die folgenden Angaben verwendet:

- **Webserver** Die Standardkonfiguration des Apache-Webserver sorgt dafür, dass die IP dynamisch vergeben wird. Bevor dies geändert wird, ist sich eine eigene Konfiguration zurecht zu legen, da die IP als Parameter bei dem Setup mit übergeben und für die Kommunikation benötigt

wird. Die exakte Konfiguration kann im Kapitel 7.5 eingesehen werden. An dieser Stelle genügt es, eine IP festzulegen. Es wird die IP "192.168.1.100" verwendet. Diese ist aus der IP-Klasse, die für den privaten Gebrauch reserviert sind, und stehen somit in keinem Konflikt zu öffentlichen Netzwerken.

Achtung: Es wird eine bestehende Internetverbindung benötigt, um die Pakete über die Paketverwaltung zu beziehen!

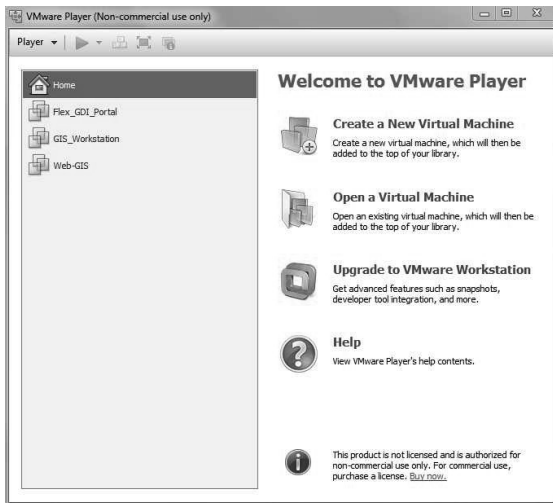
- **Datenbank** Es wird der Standard-Nutzer mit der Bezeichnung "postgres" verwendet. Das Passwort sollte im Vorfeld festgelegt werden, da dieses als Parameter in der Setup-Datei benötigt wird und im Rahmen der Einrichtung der Datenbank festzulegen ist. Die Tabellenbezeichnung und Attribute/Datentypen richten sich nach den Anforderungen der jeweiligen Aufgabenstellung.
- **WFS-T-Server** Für die Einstellungen des WFS werden die folgenden Vorgaben getroffen. Die Host-Adresse des WFS richtet sich nach den Angaben der Webserverkonfiguration. Gemäß der Vorgaben ist dies die IP "192.168.1.100". Das Passwort für die Datenbank richtet sich danach, welches bei der Konfiguration der Datenbank angegeben wird. Des Weiteren ist der Name der Datenbank anzugeben, auf die der WFS zugreifen soll.
- **Web-Mapping-Anwendung** Die zu aktivierenden Elemente sind der konkreten Aufgabenstellung zu entnehmen.

Gemäß dieser Vorgaben ist jeweils eine Implementierung vorzunehmen. Es werden alle relevanten Schritte für die Einrichtung behandelt. Die Kapitel 7.2, 7.3 7.5 und 7.7 sind bei beiden Umsetzungen identisch. Die spezifische Einrichtung abhängig von der Aufgabenstellung wird im Kapitel 7.4 behandelt.

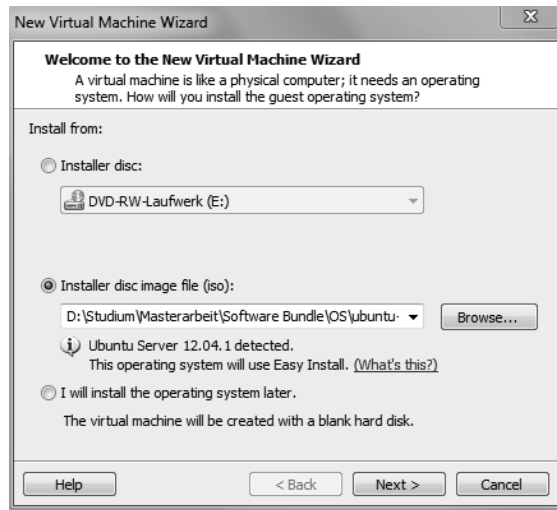
7.2 Installation des Betriebssystems

Als Basis für das Web-GIS wird mit einer virtuellen Maschine gearbeitet. Unter der Adresse my.vmware.com/de/ ist der kostenlose "VMware Player for Windows 32-bit and 64-bit" in der Version 5 herunter zu laden. Als Host-Betriebssystem wird Windows 7 genutzt. Nachdem die Datei vollständig heruntergeladen ist, ist diese auszuführen. Das Setup-Programm leitet den Nutzer durch die Installation. Ist dieser Vorgang abgeschlossen kann der VMWare Player gestartet werden. Sofern noch keine ISO-Datei bzw. ein Datenträger mit "Ubuntu Server 12 LTS" vorliegt, ist dieser über die Adresse <http://www.ubuntu.com/download/server> zu beziehen.

Die Installation des Betriebssystems gestaltet sich sehr komfortabel. Nachdem der VMWare Player gestartet wurde, erscheinen die folgenden Benutzeroberfläche entsprechend der angegebenen Reihenfolge:

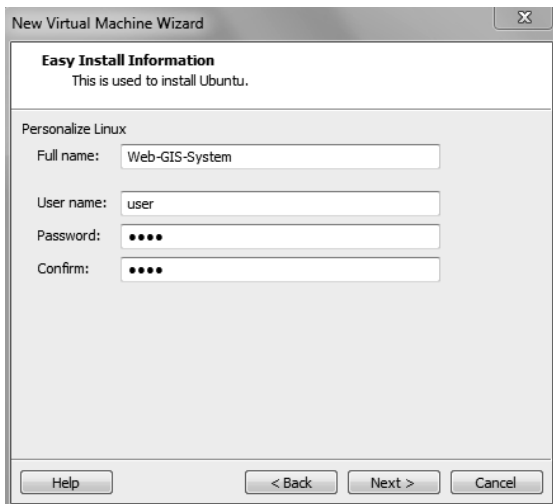


(a) Startoberfläche des VMWare Players

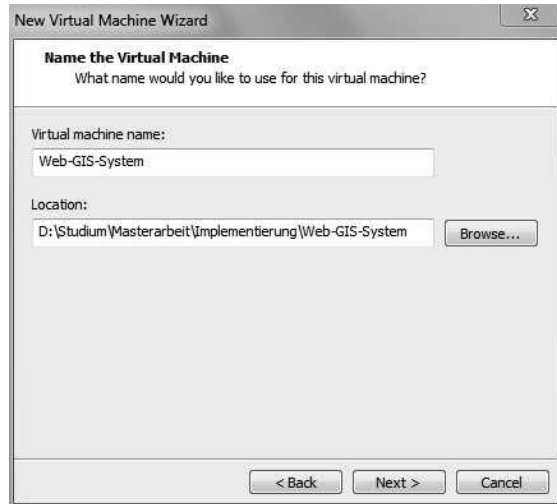


(b) Auswahl der Datenquelle

Abbildung 24: Anlegen einer neuen virtuellen Maschine Schritte 1-2

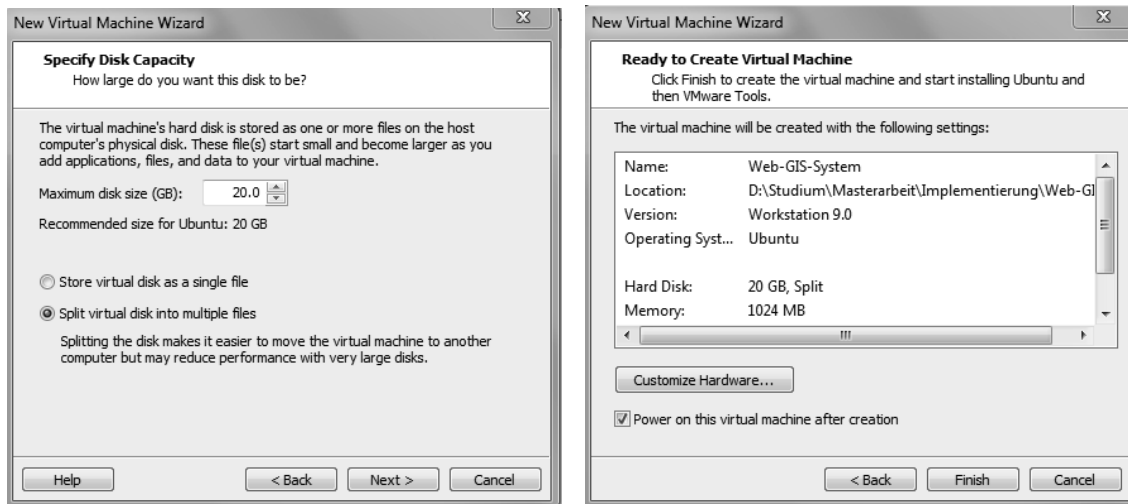


(a) Personalisierung von Linux



(b) Bezeichnung der virtuellen Maschine

Abbildung 25: Anlegen einer neuen virtuellen Maschine Schritte 3-4



(a) Festplattenkapazität festlegen

(b) Anpassen der virtuellen Hardware

Abbildung 26: Anlegen einer neuen virtuellen Maschine Schritte 5-6

- **24(a)** Unter Auswahl des Menüpunktes "Create a New Virtual Machine" wird eine neue virtuelle Maschine erzeugt.
- **24(b)** An dieser Stelle ist die Quelle des zu installierenden Betriebssystems auszuwählen. Es kann ein Datenträger (CD/DVD/BD) gewählt werden oder direkt die ISO-Datei genutzt werden. Ist dies geschehen, wird automatisch erkannt, um welches Betriebssystem es sich handelt.
- **25(a)** Diese Angaben beziehen sich auf das Betriebssystem. Es ist der Name des Betriebssystems, der Nutzer und dessen Passwort zu vergeben.
- **25(b)** An dieser Stelle wird der Name der virtuellen Maschine festgelegt und wo diese im Dateisystem des Host abgelegt werden soll.
- **26(a)** Dieser Schritt umfasst die Festlegung der virtuellen Festplattenkapazität. Die Voreinstellung von 20 GB ist für das Web-GIS ausreichend.
- **26(b)** Am Schluss kann noch die virtuelle Hardware konfiguriert werden. Dieser Schritt kann aber übersprungen werden. Sollten die Einstellungen im Nachhinein verändert werden müssen, kann dies jederzeit erfolgen. Durch Betätigung des "Finish-Buttons" wird die Installation des Betriebssystems ausgeführt. Diese ist vollkommen automatisiert und nach deren Abschluss wird das Betriebssystem gestartet und kann im vollen Umfang genutzt werden.

Hinweis: Da es sich um die US-Version von Ubuntu Server handelt, sollte das Tastatur-Layout angepasst werden. Um dieses für deutsche Tastaturen einzurichten, wird das folgende Kommando benötigt:

```
sudo dpkg-reconfigure keyboard-configuration
```

Anschließend sind folgenden Optionen auszuwählen:

```
Generic 105-kex (Intl) PC --> German --> German --> The default
for the keyboard layout --> No compose key
```

7.3 Übertragen des Installationspakets

Um das Installationspaket auf den Server zu übertragen, ist als erstes die SSH-Funktionalität auf diesen zu installieren.

```
sudo apt-get install openssh-server
```

Für den nächsten Schritt wird ein Programm benötigt, mit dem man unter Windows über einen SSH-Client die Verbindung zum Server aufbauen kann. Eine Option ist das Programm "SSH Secure Shell". Nach dessen Installation unter dem Host-System (Windows 7) ist die Anwendung "SSH Secure File Transfer Client" zu starten (Abbildungen 27).

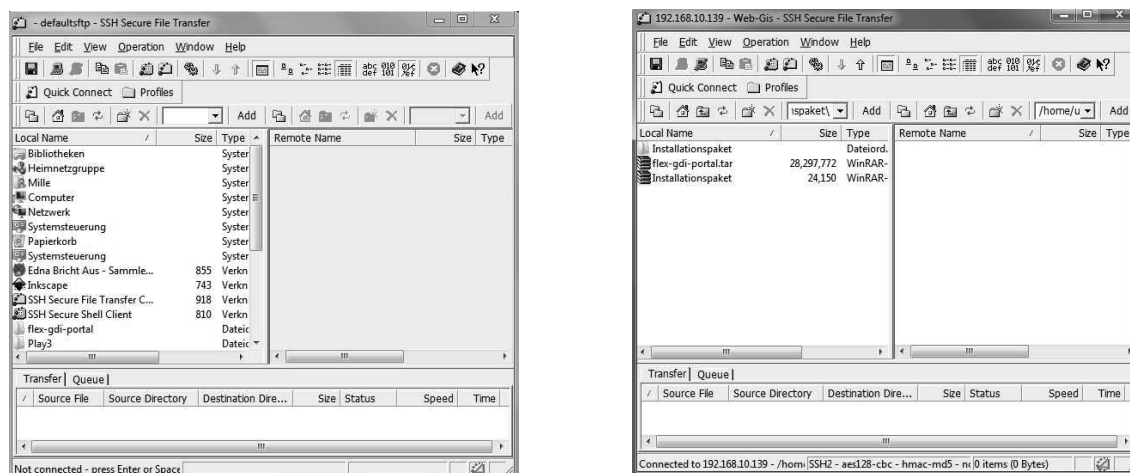


Abbildung 27: Benutzeroberfläche des SSH Secure File Transfer Client

Über den Knopf "Quick Connect" wird die Verbindung zu Ubuntu-Server hergestellt. Unter Angabe der IP des Servers und des Nutzernamens erfolgt der Verbindungsaufbau. Sollte die IP des Servers noch unbekannt sein, kann mittels des Kommandos

```
ifconfig
```

diese ermittelt werden.

Nachdem die Verbindung hergestellt wurde, kann die Übertragung des Installationspakets erfolgen. Auf der linken Seite ist das Dateisystem des Hosts abgebildet. In diesem Fenster ist zu dem Installationspaket zu navigieren. Auf der rechten Seite befindet sich das Dateisystem des Servers. Es wird automatisch das Home-Verzeichnis des Nutzer, mit dem man sich angemeldet hat, angezeigt. Um das Paket zu übertragen, ist dieses per Drag-and-Drop vom linken in das rechte Fenster zu ziehen. Somit stehen die Daten auf dem Server zur Verfügung.

7.4 Konfiguration und Ausführung des Setup-Programms

Vor der Ausführung der Installation des Web-GIS ist die Datei "setup.sh" anzupassen. Je nach den eigene Vorlieben können z. Bsp. nano oder vi verwendet werden. Nutzer, die hauptsächlich mit Windows Erfahrungen haben, sollten nano verwenden, da dieser in der Bedienung näher an

Windows-Texteditoren angelehnt ist. Mit dem Kommando "cd Pfad/zu/setup.sh" wird in an das Verzeichnis gewechselt. Anschließend ist die Datei folgendermaßen zu öffnen:

```
nano setup.sh
```

Gemäß der Vorgaben aus 7.1 sind die Konfigurationsparameter festzulegen. Für die erste Aufgabenstellung bauen sich die Parameter wie folgt auf:

```
datenbanknutzer=postgres
tabellenname=deponien
geometrietyp=POINT

attribut1=abfallart
datentyp1=VARCHAR
attribut2=flaeche
datentyp2=INTEGER

hostadresse=192.168.1.100
datenbankpasswort=postgres
datenbankname=deponien
layertitel=deponien
titel=TinyOWS

pan=true
zoomin=true
zoomout=true
maxextent=true
histback=true
histfwd=true
information=true
measure=true
insert=true
layertree=true
overviewmap=true
legend_panel=true
```

Die Parameter der zweiten Implementierung (Waldgebiete) gestalten sich wie folgt:

```
datenbanknutzer=postgres
tabellenname=waldgebiete
geometrietyp=POLYGON

attribut1=waldart
datentyp1=VARCHAR
attribut2=zustand
datentyp2=INTEGER

hostadresse=192.168.1.100
datenbankpasswort=postgres
datenbankname=waldgebiete
layertitel=waldgebiete
titel=TinyOWS

pan=true
zoomin=true
zoomout=true
maxextent=false
histback=false
histfwd=false
information=true
measure=false
insert=true
layertree=true
overviewmap=false
legend_panel=false
```

Anschließend ist zu überprüfen, ob die Parameter, welche die Attribute und Datentypen festlegen, korrekt übergeben werden (siehe Vorgaben im Kapitel 6.5). Gemäß der Vorgaben bauen sich die Aufrufe der betroffenen Skripte folgendermaßen auf:

```
sudo bash create_table.sh $datenbanknutzer $tabellenname
    $geometrietyp $attribut1 $datentyp1 $attribut2 $datentyp2
sudo bash config_web_app.sh $hostadresse $layertitel
    $geometrietyp $tabellenname $pan $zoomin $zoomout $maxextent
    $histback $histfwd $information $measure $insert $layertree
    $overviewmap $legend_panel $attribut1 $attribut2$
```

Sofern alle Änderungen übernommen sind, ist das Skript zu speichern und auszuführen.

```
sudo bash setup.sh
```

Im Laufe der Ausführung des Skripts sind durch den Nutzer diverse Eingaben zu tätigen. Sofern über die Paketverwaltung (APT) gefragt wird, ob die Installation erfolgen soll, ist mit "y" zu antworten. Bei anderen Eingabeaufforderungen ist den entsprechenden Anweisungen des Skripts zu folgen. Sollte es zu unerwarteten Fehlern kommen, kann das Skript mit der Tastenkombination "Strg+C" gestoppt werden.

Nachdem das Skript alle Komponenten eingerichtet hat, erfolgt die Konfiguration des Apache Webservers.

7.5 Konfiguration des Webservers

Um den Webserver zu konfigurieren, ist als erstes die Datei "/etc/network/interfaces" zu editieren.

```
sudo nano /etc/network/interfaces
```

Die Beispielkonfiguration (statische IP) sieht wie folgt aus:

```
auto eth0
iface eth0 inet static
    address 192.168.1.100
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.168.1.1
```

Im Anschluss ist der Netzwerkadapter und der Apache neu zu starten.

```
sudo /etc/init.d/networking restart
sudo /etc/init.d/apache2 restart
```

Damit die virtuelle Maschine von außen erreichbar ist, ist der virtuelle Netzwerkadapter anzupassen.

Menüleiste Player → Manage → Virtual Machine Settings → Reiter Hardware → Network Adapter. Die Einstellung ist auf "Bridged" zu setzen. Ist dies erfolgt, ist mit "OK" zu bestätigen.

Als nächstes ist zu überprüfen, ob alle Komponenten korrekt installiert sind.

7.6 Ergebnisse der Umsetzungen

Der einfachste Weg die Installation zu überprüfen, kann durchgeführt werden, indem das Web-GIS in einem Browser aufgerufen wird. Die Adresse richtet sich nach der vergebenen IP. In diesem Beispiel lautet die Adresse:

```
http://192.168.1.100
```

Man gelangt bei dem Aufruf zu der folgenden Startseite (Abbildung 28):



Abbildung 28: Startseite bei dem Aufruf des Servers

Über den Button "STARTE WEB-MAPPING-APP" wird die Web-Mapping-Anwendung gestartet. Wenn alles ordnungsgemäß funktioniert, muss die Web-Mapping-Anwendung erscheinen. Je nach Umsetzung variieren der Funktionsumfang der Anwendung und die Art der eintragbaren Informationen.

Für die erste Aufgabenstellung sind alle Bedienelemente aktiviert und es können Punkte (Attribute: Abfallart, Fläche) eingetragen werden. In der Abbildung 29 sind bereits Deponien eingetragen

worden. Der Screenshot zeigt wie eines der Objekte abgefragt wird.

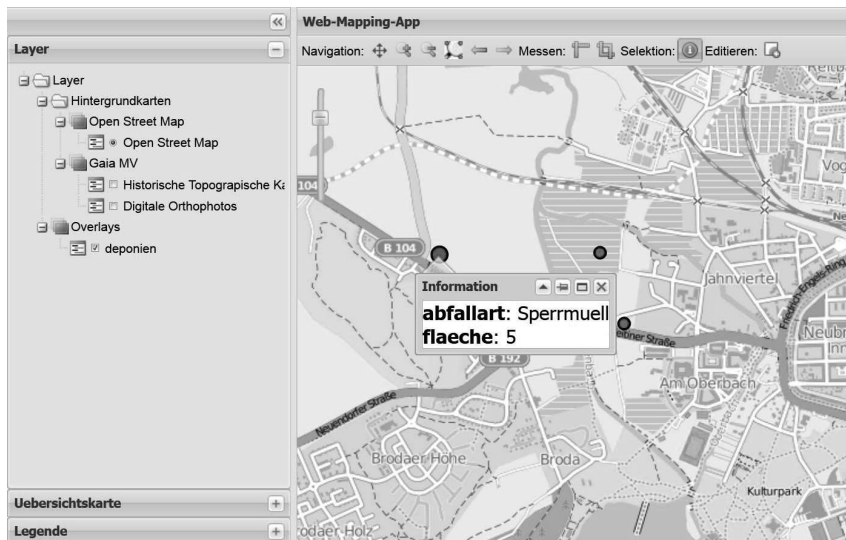


Abbildung 29: Web-Mapping-Anwendung "Mülldeponien"

Die zweite Implementierung stellt Waldgebiete (Polygone) dar. Es sind ebenfalls Beispiele für Waldgebiete eingetragen und eine Objektanfrage ausgeführt worden (Abbildung 30).

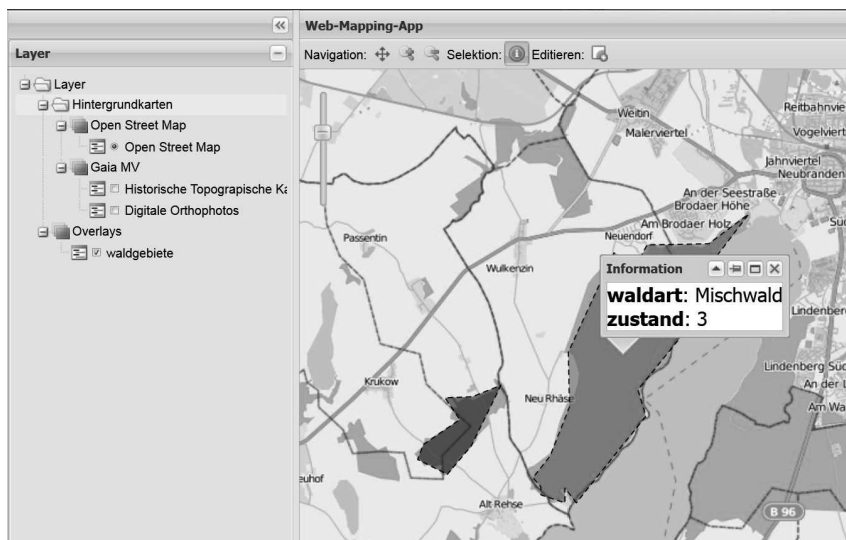


Abbildung 30: Web-Mapping-Anwendung "Waldgebiete"

Im Prinzip sind sich die zwei Umsetzungen relativ ähnlich. Die großen Unterschiede, die für eine nutzerbezogene Implementierung von Bedeutung sind, liegen in den verfügbaren Bedienelementen der Web-Mapping-Anwendung und den Informationen, die eingetragen bzw. abgefragt werden können.

7.7 Testen der Installationen

Sollte die Webseite nicht aufrufbar sein, können alle installierten Komponenten einzeln getestet werden.

- **Apache Webserver:** Wenn Fehler im Webserver auftreten, werden diese in einer Log-Datei protokolliert. Die Datei befindet sich in dem Verzeichnis "/var/log/apache2/error.log". Mit einem Texteditor (nano, vi) kann die Datei geöffnet werden. Wenn bei einzelnen Logs ein "[error]" vorhanden ist, ist dies ein Hinweis auf eine Fehlfunktion des Apache Webservers.
- **Datenbank:** Um die bei der Installation angelegte Datenbank zu überprüfen, sind im Terminal des Servers folgenden Eingaben zu tätigen:

```
#Wechsel zum User der Datenbank
sudo -u postgres psql
#Datenbanken anzeigen
\l
#Beenden
\q
```

Des Weiteren kann man sich direkt mit einer Datenbank verbinden und alle vorhandenen Tabellen ausgeben lassen:

```
#Wechsel zum User der Datenbank
sudo -u postgres psql [Name der Datenbank]
#Alle Tabellen der Datenbank anzeigen
\d
#Beenden
\q
```

Als letztes kann man alle Attribute einer bestimmten Tabelle ermitteln:

```
#Wechsel zum User der Datenbank
sudo -u postgres psql [Name der Datenbank]
#Alle Tabellen der Datenbank anzeigen
\d [Name der Tabelle]
#Beenden
\q
```

- **MapServer** Um die Funktionalität von MapServer zu überprüfen, stehen zwei Möglichkeiten zur Verfügung. Die erste Option ist, mittels eines Webbrowsers die URL zum MapServer aufzurufen. Sofern keine Änderungen am Webserver vorgenommen wurden, lautet die Adresse `http://Adresse_des_Servers/cgi-bin/mapserv?`. Sofern im Browser die Aufschrift "No query information to decode. QUERY_STRING is set, but empty." erscheint, ist MapServer korrekt über den Webserver erreichbar. Alternativ kann im Terminal des Webservers das folgende Kommando genutzt werden, um die Konfiguration zu überprüfen.

```
/usr/lib/cgi-bin/mapserv -v
```

Als Ergebnis erhält man eine Übersicht zu der Konfiguration von MapServer bzw. eine Ausgabe des vorliegenden Fehlers.

- **TinyOWS** TinyOWS kann ebenfalls über zwei Wegen überprüft werden. Im Terminal kann verifiziert werden, ob TinyOWS korrekt konfiguriert ist und eine Verbindung zu der Datenbank herstellen kann.

```
/usr/lib/cgi-bin/tinyows --check
```

Sofern eine fehlerhafte Konfiguration vorliegt, erfolgt eine Ausgabe des Fehlers. Alternativ kann der Webbrowser genutzt werden. Bei dem Aufruf der URL `http://Adresse_des_Servers/cgi-bin/tinyows?service=wfs&request=DescribeFeatureType&version=1.1.0&typename=tows:[Name_der_Tabelle]` werden die Inhalte des WFS ermittelt. Als Ergebnis wird ein XML-Dokument zurückgegeben, indem eine Beschreibung der Struktur der Tabelle bzw. Features enthalten ist (Abbildung 31).

Mit dieser XML-Datei sind anscheinend keine Style-Informationen verknüpft. Nachfolgend wird die Baum-Ansicht des Dokuments angezeigt.

```
-<xs:schema targetNamespace="www.tinyows.org" elementFormDefault="qualified" version="1.1">
  <xs:import namespace="http://www.opengis.net/gml" schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
  <xs:element name="hotspots" type="tows:hotspotsType" substitutionGroup="gml:_Feature"/>
-<xs:complexType name="hotspotsType">
  -<xs:complexContent>
    -<xs:extension base="gml:AbstractFeatureType">
      -<xs:sequence>
        <xs:element name="bezeichnung" type="string" nillable="true" minOccurs="0" maxOccurs="1"/>
        <xs:element name="information" type="string" nillable="true" minOccurs="0" maxOccurs="1"/>
        <xs:element name="geom" type="gml:PointPropertyType" nillable="true" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:schema>
```

Abbildung 31: Überprüfung der TinyOWS-Konfiguration über den Webbrowser

- **Web-Mapping-Anwendung** Eine Überprüfung der Anwendung erfolgt in erster Linie über die Datei "parameters.js". Diese befindet sich in dem Verzeichnis `/var/www/web-mapping-app/app`. Sollte eine Veränderung der Parameter keine Auswirkung haben, besteht nur noch die Möglichkeit den Quellcode der Datei "web-mapping-app.html" zu überprüfen. Der beste Weg einen Fehler festzustellen ist mit Hilfe einer JavaScript-Konsole des Webbrowsers. Nachdem diese aktiviert ist, kann die Web-Mapping-Anwendung aufgerufen werden. Sollten dann Fehler bei der Verarbeitung auftreten, werden diese in der Konsole angezeigt und können als Einstiegspunkt für die Fehlerbehebung genutzt werden.

8 Ergebnisse

In diesem Kapitel wird überprüft, ob alle Ziele erreicht worden sind und die geforderten Rahmenbedingungen eingehalten wurden.

Im ersten Schritt sind diverse web-basierte Geoinformationssysteme gesichtet worden, um einen Überblick über die Funktionalitäten und das Design heutiger Systeme zu erhalten. Es wurde gezeigt, dass alle untersuchten Systeme in ihrem Aufbau relativ ähnlich sind und die implementierten Funktionen, wie die Kartennavigation, zum heutigen Standard von Web-Mapping-Anwendungen gehören. Auf Basis der Analyse ist ermittelt worden, welche Funktionen in dem eigenen System zu berücksichtigen sind und wie das grobe Design zu gestalten ist. Unter Verwendung der momentanen Designtrends ist die Anwendung eines Web-GIS wesentlich effizienter, unter der Voraussetzung, dass der Nutzer bereits Erfahrung von anderen (Web-)GIS hat.

Im Vorfeld war bereits klar, welche Softwarekomponenten für den Aufbau des Systems benötigt werden. Es sind drei Komponenten (Datenbank, Kartenserver, Web-Mapping-Anwendung) nötig. Gemessen an den Anforderungen des Systems, können nur freie Softwareprodukte genutzt werden, die zudem mit einfachen Mitteln konfigurierbar sein sollen. Neben dem muss gewährleistet sein, dass die drei Komponenten auch miteinander arbeiten und kommunizieren können. Um geeignete Produkte zu finden, sind mehrere Implementierungen untersucht worden. Der Markt der freien GIS-bezogenen Produkte offeriert verschiedenste Umsetzung, sodass es in der Regel für jede Anforderung mindestens eine Lösung gibt.

Die Untersuchung der Datenbanksysteme (PostgreSQL/PostGIS, SQLite/Spatialite, MySQL Spatial) zeigte, dass es nur minimale Unterschiede hinsichtlich der Funktionalitäten gibt. Dies ist in erster Linie der Standardisierung zu verdanken (MM/SQL). Ein ähnliches Ergebnis ergab die Analyse der Kartenserver (MapServer/TinyOWS, Deegree, GeoServer). Die Tatsache, dass die Produkte auf technischer Ebene nahezu identisch sind, ist von Vorteil, da der Aspekt der Nutzer-bezogenen und automatisierten Einrichtung der Komponenten signifikant höher gewichtet werden konnte. So ergab die Untersuchung, dass MapServer/TinyOWS sich am besten eignet, da deren Konfiguration am effizientesten vorgenommen werden kann.

Für die Erstellung der Web-Mapping-Anwendung sind diverse Web-Mapping-Frameworks untersucht worden. Es ist ein Kompromiss zwischen Funktionsumfang, Komplexität und der Umsetzbarkeit Nutzer-bezogener Anforderungen getroffen worden. Unter der Verwendung des JavaScript-Frameworks "Open Layers" in Verbindung mit "Ext/GeoExt" für die erweiterte Gestaltung GUI-basierter Elemente, ist ein Framework gefunden worden, das sich am besten als Basis für die Web-Mapping-Anwendung eignet.

Die vorhergehenden Erkenntnisse flossen in das Konzept des Web-GIS ein. Die ausgewählten Komponenten sind konzeptionell in das System eingepasst worden. Der Fokus bei deren Komposition lag insbesondere auf den Schnittstellen. Die angestrebten Schnittstellen wie die des WMS und WFS, sind im System genutzt worden. Der große Vorteil der daraus resultiert, liegt in der Austauschbarkeit der Implementierungen der Komponenten. So ist es beispielsweise möglich den Kartenserver "MapServer" durch "GeoServer" auszutauschen. Da beide auf den selben Spezifikationen aufbauen und die Gestaltung der Schnittstellen identisch ist, ermöglicht das System jederzeit Anpassungen vorzunehmen. Diese stützt sich zudem auf dem Architekturkonzept "SOA", das bei der Gestaltung aktueller Web-GIS immer häufiger Anwendung findet.

Die Analyseergebnisse der Kartenportale sind im eigenen Design der Web-Mapping-Anwendung eingeflossen. Es sind eine Vielzahl an sinnvollen Funktionen eingebunden worden. Des Weiteren ist der interne Aufbau der Anwendung in der Form gestaltet, dass relativ leicht Änderungen vorgenommen werden können. Dies ist für die Weiterentwicklung der Anwendung von zentraler Bedeutung, da auf diesem Weg gravierende Änderungen bis hin zum kompletten Neuaufbau der Anwendung unnötig sind.

Der letzte Aspekt des Konzept umfasste die Frage nach der Art und Weise, wie das System einfach und entsprechend der Anforderungen des Nutzer aufgesetzt werden kann. Es sind mehrere Shell-Skripte konzipiert worden. Diese ermöglichen auf Basis von Parametern, die durch den Nutzer festgelegt werden, eine unkomplizierte Implementierung des Systems. So können auch Personen, die im Umgang mit Web-GIS unerfahren sind, ein solches System selbst aufsetzen.

Die Implementierung des Konzepts ist erfolgreich durchgeführt worden. Die Integration aller Systemkomponenten ist durch die Shell-Skripte realisiert worden. Dem Provider des Web-GIS werden verschiedene Werkzeuge zur Verfügung gestellt, mit dem ein maßgeschneidertes System integriert werden kann. Unter Angabe diverser Parameter wird die Datenbank, der Kartenserver und die Web-Mapping-Anwendung konfiguriert. Um das Maß an Komfort bei der Einrichtung des Systems zu erhöhen, ist zudem ein Setup-Skript entwickelt worden. Dieses beinhaltet alle Konfigurationsparameter und ruft automatisch die anderen Skripte auf. Der Ersteller des Web-GIS muss somit nur dieses Skript in den Parametern anpassen und anschließend dieses ausführen. Unter Verwendung von Anweisungen und Hilfestellungen wird der Nutzer durch den gesamten Vorgang der Installation und Konfiguration geführt.

Auch wenn das System in dieser Version nur für die Debian-basierte Linux-Distribution Ubuntu ausgelegt ist, ist eine Portierung zu anderen Linux-Distributionen möglich. Unter Umständen sind einige Skripte anzupassen. Der Kern bleibt aber unangetastet, da diese Skripte für einen flexiblen Einsatz ausgelegt sind.

Sollte es zu einem späteren Zeitpunkt notwendig sein, das System zu modifizieren (Neue Tabellen anlegen, Kartenserver überarbeiten) können die Skripte auch einzeln genutzt werden.

Ein weitere Vorteil, der aus der Verwendung standardisierter Schnittstellen resultiert, liegt in der Anbindung des Kartenservers in Verbindung mit der Datenbank. Es ist möglich die Daten nicht nur über die Web-Mapping-Anwendung einzusehen und zu bearbeiten, sondern auch mittels anderen GIS, indem der WMS und WFS genutzt wird. So können ggf. komplexe Analysen des Datenbestandes effizient ausgeführt werden, die mit der Web-Mapping-Anwendung aus Gründen des Rechenaufwands nicht durchgeführt werden können.

Der letzte Punkt umfasste die exemplarische Umsetzung des entwickelten Systems. Es ist ein ausführlicher Leitfaden erarbeitet worden, der Schritt für Schritt zeigt, wie das Web-GIS aufgesetzt werden kann. Die Umsetzung ist erfolgreich durchgeführt worden und in dieser Version vollständig einsetzbar. Ob das System lokal, im Intranet oder im Internet verfügbar ist, ist nicht relevant.

Da es unter Umständen zu unerwarteten Fehlern kommt, die beispielsweise durch eine fehlerbehaftet Angabe der Konfigurationsparameter entstehen kann, kommt, sind für alle Komponenten Methoden gezeigt worden, wie diese in ihrer Funktionstüchtigkeit überprüft werden können.

Zusammenfassend ist zu sagen, dass alle Rahmenbedingungen eingehalten wurden. Es ist ein leicht-gewichtetes System auf Basis freier Softwarekomponenten entwickelt worden. Dieses zeichnet sich durch einen hohen Grad an Flexibilität und Erweiterbarkeit aus und verfügt über Mechanismen, die eine simple und Nutzer-bezogene Einrichtung des Systems realisieren. So ist das System auch für Interessengruppen relevant, die im Bereich web-basierter GIS wenig bewandert sind.

9 Fazit

Es wurde gezeigt, dass die Implementierung eines Web-GIS schnell und einfach durchgeführt werden kann. Angefangen bei der Auswahl geeigneter Systemkomponenten, über die Entwicklung eines Konzepts bis hin zu dessen Umsetzung, steht ein Komplettpaket zur Verfügung, mit dem die Integration eines Web-GIS flexibel gestaltet werden kann. Dabei steht stets der Nutzer des Systems im Vordergrund. Auch ohne tiefgreifende Kenntnisse über web-basierte Geoinformationssysteme, Datenbanken und Kartenserver kann das System mit wenigen Handgriffen den Bedürfnissen des Nutzer angepasst werden und auf einfachen Weg implementiert werden.

Das hohe Maß an Flexibilität und Anpassungsfähigkeit ermöglicht es, das Web-GIS für eine Vielzahl an unterschiedlichen Einsatzgebieten zu verwenden. Das System kann z. Bsp. im Rahmen von Projekten für die Erfassung raumbezogener Daten genutzt werden. Da der Nutzer die Entscheidung trägt, welche Informationen erfasst werden sollen (Geometriotyp, Attribute), können verschiedene Objekte erfasst werden. Alternativ ist der Einsatz als reines Kartenbetrachtungswerkzeug für die eigenen Informationen ein potentielles Einsatzgebiet.

Damit ist das große Ziel, ein flexibles und nutzerbezogenes Web-GIS zu entwickeln, erreicht worden. Zudem bietet das System viele sinnvolle Ansätze, um das System zu optimieren und zu erweitern.

10 Ausblick

Das System bietet viel Freiraum für Erweiterungen und Optimierungen. Auch wenn das System in dieser Version vollständig lauffähig ist, können noch andere Ideen umgesetzt werden, um das System auf die nächst höhere Ebene zu führen. Im Folgenden werden Ansätze und Ideen aufgezeigt, wie das System in Zukunft sinnvoll weiterentwickelt werden kann.

- **Konfigurationsskripte** Das Skript für die Installation der Komponenten bedarf keiner Erweiterung. Hinsichtlich der Konfigurationsskripte gestaltet sich dies anders. Da die Konfiguration des Kartenservers und der Web-Mapping-Anwendung auf einen einzelnen selbst definierten Datenlayer beschränkt ist, der im Rahmen der Anwendung visualisiert und editiert werden kann, sollte das Skript in der Form erweitert werden, dass eine beliebige Anzahl an Layern/Tabellen integriert werden kann. Dieses ist zwar jederzeit möglich, setzt aber das direkte Ändern des Quellcodes der Web-Mapping-Anwendung und Konfigurationsdateien des Kartenservers voraus. Diese Funktionalität sollte im Rahmen der Skripte implementiert werden.
- **Web-Mapping-Anwendung** Um den ersten Punkt umzusetzen, sind zusätzliche Änderungen am Quellcode der Anwendung vorzunehmen. Der Fokus liegt in diesem Fall auf der Verwaltung der neuen Layer und zugehörigen Objekten, die ein Editieren in der Anwendung ermöglichen. Des Weiteren können Funktionen für das Verändern und Entfernen bereits eingetragener Objekte umgesetzt werden. Da dies in der aktuellen Version konzeptionell nicht vorgesehen ist, sollte zumindest die Option für deren Verwendung zur Verfügung gestellt werden. Neben dem kann die Anwendung um Funktionen erweitert werden. Einige sinnvolle Funktionen sind, das Exportieren und Drucken von Karten und das Importieren von Daten über die Anwendung.
- **Sicherheitsmechanismen** Ein andere Aspekt, der erarbeitet werden kann, ist die Implementierung von Sicherheitsmechanismen. Beispielsweise kann vor der Nutzung der Web-Mapping-Anwendung eine Authentifizierung eingebaut werden. Dies wäre primär von Bedeutung, wenn bestehende Daten über die Web-Mapping-Anwendung verändert werden können. Da bereits eine Datenbank im Backend verwendet wird, kann diese für die Verwaltung der Nutzer verwendet werden. Auf Seiten des Servers bietet es sich an mit PHP in Verbindung mit SQL zu arbeiten, da sich über diesen Weg sehr effizient Authentifizierungsmechanismen implementieren lassen.
- **Entwicklung einer Web-Mapping-Anwendungs-Version für mobile Endgeräte** Da in der heutigen Zeit die Verbreitung von Smartphones und Tablets voranschreitet, bietet es sich an, eine Version zu entwickeln, die für die Verwendung derartiger Geräte optimiert ist. Es ist zwar möglich mittels eines Webbrowsers die Anwendung mit einem mobilen Gerät zu nutzen, die Handhabung ist aber ausbaufähig. Insbesondere die Steuerung über ein Touchpad stellt besondere Anforderungen an das Design. Es ist anzustreben, dass das ursprüngliche Programm weiterverwendet werden kann. Um dies zu gewährleisten und die Änderungen an der Anwendung zu minimieren, stellt OpenLayers Komponenten zur Verfügung, die für mobile Geräte optimiert sind.
- **Offline-Betrieb** Ein letzter Punkt der implementiert werden kann, bezieht sich auf die Möglichkeit das System für den Offline-Betrieb auszubauen. Wenn z. Bsp. ein Datenlayer um neue Objekte ergänzt werden soll und dies auch vor Ort zu erfolgen hat (z. Bsp. im Gelände), kann es sein, dass keine Internetverbindung verfügbar ist. Somit wäre auch das Web-GIS nicht nutzbar. Um dies aber umzusetzen, sind gravierende Änderungen am Konzept vorzunehmen, da dieses vollständig auf die Verwendung web-basierter Infrastrukturen aufbaut.

Prinzipiell bietet das System vielfältige neue Aufgaben, um das System zu optimieren und dieses um sinnvolle Funktionen zu erweitern. Es ist dabei stets anzustreben, das ursprüngliche Konzept beizubehalten, damit grundlegende Änderungen am System ausbleiben und der Aufwand gering gehalten wird.

Abbildungsverzeichnis

1	Architektur dynamischer Web-Kartografie	5
2	Benutzeroberfläche von Google Maps	14
3	Benutzeroberfläche von GAIA-MVlight	15
4	Benutzeroberfläche von GAIA-MVprofessional	16
5	Benutzeroberfläche von Open Street Map	17
6	Benutzeroberfläche von JOSM [Ope13g]	17
7	Benutzeroberfläche von Klarschiff-HRO	18
8	Eingabemaske für die Meldung neuer Probleme	18
9	Funktionsweise von MapServer	26
10	Web-Interface von deegree	29
11	Benutzeroberfläche von GeoServer	31
12	MapFish-Framework [Ope13d]	36
13	Systemarchitektur	41
14	Schnittstellen des Web-GIS	42
15	Abläufe der WMS-Requests	42
16	Abläufe der WFS-Requests	43
17	Anwendungsfälle in der Web-Mapping-Anwendung	45
18	Ablauf bei dem Einfügen neuer Objekte	46
19	Aufbau der Web-Mapping-Anwendung	47
20	Initialisierung der Web-Mapping-Anwendung	48
21	Benutzeroberfläche der Web-Mapping-Anwendung	56
22	Werkzeugleiste der Web-Mapping-Anwendung	56
23	Elemente des Seite-Panels	57
24	Anlegen einer neuen virtuellen Maschine Schritte 1-2	60
25	Anlegen einer neuen virtuellen Maschine Schritte 3-4	60
26	Anlegen einer neuen virtuellen Maschine Schritte 5-6	61
27	Benutzeroberfläche des SSH Secure File Transfer Client	62
28	Startseite bei dem Aufruf des Servers	65
29	Web-Mapping-Anwendung "Mülldeponien"	66
30	Web-Mapping-Anwendung "Waldgebiete"	66
31	Überprüfung der TinyOWS-Konfiguration über den Webbrowser	68

Tabellenverzeichnis

1	GetCapabilities-Request-Parameter ([Mar11] S. 30)	8
2	GetMap-Request-Parameter ([Mar11] S. 30)	9
3	GetFeatureInfo-Parameter ([Mar11], S. 31)	10
4	GetCapabilities-Request-Parameter ([Mar11] S. 30)	11
5	DescribeFeatureType-Parameter ([Mar11] S.43)	11
6	GetFeature-Parameter ([Mar11] S.44)	12
7	WFS-T-Parameter ([Mar11] S.45)	13

Inhaltsverzeichnis der beigelegten CD

- Hauptverzeichnis
 - Installationspaket
 - programme
 - libxml2
 - tinyows
 - proxy.cgi
 - skripte
 - config_db.sh
 - config_web_app.sh
 - config_wfs_t.sh
 - create_table.sh
 - create_wfs_mapfile.sh
 - create_wms_mapfile.sh
 - insert_data.sh
 - install_all.sh
 - web-mapping-app
 - app
 - web-mapping-app.html
 - libraries
 - Ext
 - GeoExt
 - OpenLayers
 - setup.sh
 - Abschlussarbeit
 - Master-Thesis-Milbratz-2013.pdf

Literatur

- [And05] ANDREAS DONAUBAUER: *Web Feature Service - Geodienst für den Zugriff auf objektrukturierte Geodaten*. In: *Geodateninfrastruktur*. Herbert Wichmann Verlag, 2005, S. 93–100
- [DAT13] DATACOM BUCHVERLAG GMBH: *ITWissen - Das große Online-Lexikon für Informationstechnologie*. März 2013. – Stichwort: Serviceorientierte Architektur
- [Dis13] DISTROWATCH: *Top Ten Distributionen*. <http://distrowatch.com/dwres.php?resource=major&language=DE>. Version: April 2013
- [FOS13] FOSSGIS E.V.: *Open Street Map*. <http://www.openstreetmap.de/>. Version: April 2013
- [Gü05] GÜNTHER PICHLER, Matrin Klopfer: *Spezifikationen und Standardisierungen - OGC, OGC Europe und ISO*. In: *Geodateninfrastruktur*. Herbert Wichmann Verlag, 2005, S. 9–17
- [Geo13] GEOEXT COMMUNITY: *GeoExt: JavaScript Toolkit for Rich Web Mapping Applications*. <http://geoext.org/>. Version: April 2013
- [Goo13] GOOGLE INC.: *Google Maps*. <http://maps.google.de/>. Version: April 2013
- [Han13] HANSESTADT ROSTOCK: *Klarschiff-HRO*. <http://www.klarschiff-hro.de>. Version: April 2013
- [Int13] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO Homepage*. <http://www.iso.org>. Version: März 2013
- [Lan13] LANDESAMT FÜR UMWELT, NATURESCHUTZ UND GEOLOGIE MV: *GAIA-MV*. <http://www.geoportal-mv.de/>. Version: April 2013
- [lat13] LAT/LON GMBH: *deegree*. <http://www.deegree.org/>. Version: April 2013
- [Mar11] MARKUS MILBRATZ (B. ENG.): *Konzeption eines Webportals für den STADTmonitor unter Berücksichtigung der OGC-Spezifikationen*. Neubrandenburg, 2011
- [Ope13a] OPEN GEOSPATIAL CONSORTIUM: *OGC Homepage*. <http://www.opengeospatial.org>. Version: März 2013
- [Ope13b] OPEN SOURCE GEOSPATIAL FOUNDATION: *GDAL*. <http://www.gdal.org/>. Version: April 2013
- [Ope13c] OPEN SOURCE GEOSPATIAL FOUNDATION: *GeoServer*. <http://geoserver.org/>. Version: April 2013
- [Ope13d] OPEN SOURCE GEOSPATIAL FOUNDATION: *MapFish*. <http://mapfish.org/>. Version: April 2013
- [Ope13e] OPEN SOURCE GEOSPATIAL FOUNDATION: *OpenLayers: Free Maps for the Web*. <http://openlayers.org/>. Version: April 2013
- [Ope13f] OPENSTREETMAP FOUNDATION: *Editing*. <http://wiki.openstreetmap.org/wiki/DE:Editing>. Version: April 2013
- [Ope13g] OPENSTREETMAP FOUNDATION: *JOSM*. <http://wiki.openstreetmap.org/wiki/DE:JOSM>. Version: April 2013
- [Pro] PROF. DR.-ING. WOLFGANG KRESSE: *Normen und Standards*. – Vorlesungsunterlagen

- [Pro13] PROFESSUR FÜR GEODÄSIE UND GEOINFORMATIK (GG) AUF UNIVERSITÄT ROSTOCK: *Geoinformatik-Service*. <http://www.geoinformatik.uni-rostock.de>.
Version: März 2013
- [Rei05] REINHARD ERSTLING, Ingo Simonis: Web Map Service. In: *Geodateninfrastruktur*. Herbert Wichmann Verlag, 2005, S. 108–125
- [Sen13] SENCHA INC.: *Sencha Ext JS: JavaScript Framework for Rich Desktop Apps*. <http://www.sencha.com/products/extjs>. Version: April 2013
- [Sun13] SUN MICROSYSTEMS GMBH: *MySQL*. <http://www.mysql.de/>. Version: April 2013
- [The13a] THE POSTGRESQL GLOBAL DEVELOPMENT GROUP: *PostgreSQL*. <http://www.postgresql.org/>. Version: April 2013
- [The13b] THE SQLITE CONSORTIUM: *SQLite*. <http://www.sqlite.org/>. Version: April 2013
- [Tus13] TUSCANY REGION - TERRITORIAL AND ENVIRONMENTAL INFORMATION SYSTEM: *Spatialite*. <https://www.gaia-gis.it/fossil/libspatialite/index>.
Version: April 2013
- [Tyl08] TYLER MITCHELL: *Web-Mapping mit Open Source-GIS-Tools*. Köln : O'Reilley, 2008
- [Uni13] UNIVERSITY OF MINNESOTA: *MapServer*. <http://mapserver.org>. Version: März 2013