



Hochschule Neubrandenburg  
University of Applied Sciences

**Hochschule Neubrandenburg**  
**Studiengang Geoinformatik**

# **Standard-konforme Darstellung von OSM-Daten mittels MapServer**

**Bachelorarbeit**

vorgelegt von: *Ariane Brörmann*

Zum Erlangen des akademischen Grades

**„Bachelor of Engineering“ (B.Eng.)**

Erstprüfer: Prof. Dr.-Ing. Andreas Wehrenpfennig

Zweitprüfer: Dipl.-Geogr. Björn Schilberg

Eingereicht am

31.10.2012

URN: urn:nbn:de:gbv:519-thesis

**Hochschule Neubrandenburg**  
Studiengang Geoinformatik

**Bachelorarbeit**

**Thema:** Stand-konforme Darstellung von OSM-Daten mittels MapServer

für  
Frau Ariane Brörmann

Erstprüfer: Prof. Dr.-Ing. Andreas Wehrenpfennig  
Zweitprüfer: Dipl.-Geogr. Björn Schilberg

Beginn: 31.08.2012  
Abgabe: 31.10.2012

---

Erstprüfer

---

Bachelor

---

eingereicht am:

## **Kurzfassung**

Die Standardisierung von Geodaten und die Gewährleistung einer einheitlichen Verwendung ermöglichen eine globale Geodateninfrastruktur.

Die Darstellung von OSM-Daten mittels Mapnik entspricht jedoch keinem spezifischen Standard, wie beispielsweise dem OGC-Standard und kann daher nicht in verteilten GIS-Systemen eingesetzt werden.

Die vorliegende Arbeit untersucht deshalb einen alternativen Ansatz zur Darstellung von OSM-Daten am Beispiel des MapServers.

Es wird ein neuer Entwurf präsentiert und die erfolgreiche Umsetzung anhand der Erstellung und Auslieferung einer landestypischen Karte illustriert.

OSM-Daten können demzufolge mittels MapServer gerendert und Standardkonform über WMS ausgeliefert werden.

## **Danksagung**

Nach Beendigung meiner Arbeit möchte ich mich bei allen bedanken, die mich stets unterstützt und fachlich beraten haben.

Insbesondere danke ich Prof. Dr.-Ing. Andreas Wehrenpfennig und Dipl.-Geogr. Björn Schilberg, die mich beim Anfertigen der Arbeit betreut haben und mir fachliche Ratschläge gaben, an denen ich mich orientieren konnte.

Des Weiteren bedanke ich mich bei meiner Schwägerin, für das Korrekturlesen meiner Arbeit.

Besonders möchte ich mich bei meiner Familie bedanken, die mich während des Studiums immer wieder motiviert haben, an meinen Zielen fest zu halten.

Abschließend danke ich dem gesamten Team der Intevation, für ein positives Arbeitsklima, was das Anfertigen meiner Arbeit erleichterte.

# Inhaltsverzeichnis

1	Einleitung.....	2
2	Ziel der Bachelorarbeit.....	3
3	Ausgangssituation.....	4
3.1	Kartographie.....	5
3.2	Allgemeine Anforderungen an eine digitale Karte .....	6
3.3	Definition des „deutschlandspezifischen“ Kartenstils.....	7
3.4	OpenStreetMap (OSM).....	9
3.5	Standardisierung von Geodaten.....	10
4	Technische Gegebenheiten.....	13
4.1	WebGIS-System zur Darstellung von OSM-Daten mittels Mapnik.....	15
4.2.1	Import von OSM-Daten in PostgreSQL/PostGIS mittels osm2pgsql.....	16
4.2.2	Rendern der OSM-Daten mit Mapnik und Auslieferung.....	18
5	Analyse der Kartenerstellung mittels Mapnik.....	21
5.1	Aufwand bei der Installation aller benötigten Bestandteile .....	21
5.2	Aufwand beim Umsetzen des „deutschlandspezifischen“ Kartenstils.....	22
5.3	Schlussfolgerung .....	22
6	Entwurf einer neuen Architektur.....	23
6.1	WebGIS-System zur Darstellung von OSM-Daten mittels MapServer.....	23
6.1.1	Import der OSM-Daten in die PostgreSQL/PostGIS-DB mittels Imposm.....	24
6.1.2	Generieren des Mapfiles mithilfe von basemaps .....	24
6.1.3	Rendern der OSM-Daten mit MapServer und Auslieferung der Ergebnisse.....	25
7	Umsetzung.....	27
7.1	Installation der benötigten Komponenten und Import der Beispiel-Daten.....	27
7.2	Konfiguration der basemap-Dateien.....	28
7.2.1	Aufbau und Funktion des Skriptes "generate_style.py".....	30
7.2.2	Konfiguration der inkludierten Dateien.....	31
7.3	Vorgehensweise zur Erstellung des „deutschlandspezifischen“ Kartenstils.....	31
8	Ergebnis.....	33
8.1	Aufwand bei der Installation aller benötigten Bestandteile.....	34
8.2	Aufwand bei der Umsetzung zum „deutschlandspezifischen“ Kartenstils.....	35
8.3	Gegenüberstellung der Darstellungen von OSM-Daten mittels MapServer und Mapnik .....	36
9	Zusammenfassung und Ausblick.....	42
	Glossar.....	43
	Verzeichnisse.....	45
	Quellenverzeichnis.....	45
	Abbildungsverzeichnis.....	46
	Tabellenverzeichnis.....	46
	Listings.....	47
	Abkürzungsverzeichnis.....	47
	Anhang.....	48
	Anhang A – output.png.....	48
	Anhang B – osm.map mit OpenLayers.....	49
	Anhang C – Inhalt der CD.....	49

# 1 Einleitung

Seit historischen Zeiten sind Karten für Menschen nützlich. Während früher analoge Karten eine wichtige Entscheidungsgrundlage für Prozesse in Politik, Wirtschaft, Verwaltung und im privaten Leben darstellten, nehmen heute digitale Karten eine bedeutendere Rolle ein.

Grundlage für die Herstellung digitaler Karten sind Geodaten. Dabei handelt es sich um Daten, die einen direkten oder indirekten Bezug zu einem bestimmten Standort oder geografischen Gebiet aufweisen. Sie setzen sich sowohl aus geometrischen (Form, Größe, Lage), topologischen (Knoten, Kante, Masche) und thematischen (Ortsnamen, PLZ, Einwohnerzahl usw.) als auch zeitlichen Informationen zusammen. Letztere stellen den Zeitpunkt oder Zeitraum für die vorher beschriebenen Eigenschaften fest. Zur Verwaltung von Geodaten, mit den oben genannten Eigenschaften, sind räumliche Datenbanken notwendig. Diese bilden die Basis für Geoinformationssysteme (GIS), welche weiterhin benötigte Hardware, Software und Anwendungen zur Organisation und Präsentation der Geodaten umfassen.

Um folglich eigene Karten oder Kartenanwendungen zu erstellen, müssen zunächst Geodaten bezogen werden. Dabei spielen Vollständigkeit, Genauigkeit und Aktualität eine entscheidende Rolle. Eine Einteilung erfolgt in amtliche und nichtamtliche Geodaten, wobei in beiden Fällen jedoch zum Teil hohe Kosten berücksichtigt werden müssen. Beispielsweise regelt die Kostenordnung für das amtliche Vermessungswesen in den jeweiligen Bundesländern (z. B. [VermKostO Niedersachsen](#)) die Abgabe von Geobasisdaten. Die dabei anfallenden Gebühren richten sich nach dem Bedarf der Geodateninfrastruktur in Europa ([INSPIRE](#)).

Aufgrund dessen sollte die Möglichkeit Geodaten kostengünstig und ohne restriktive Lizenzen zu beziehen näher betrachtet werden.

Das Projekt *OpenStreetMap (OSM)*<sup>1</sup> (s. Kap. 3.4) bietet beispielsweise die sogenannten *OSM-Daten* unter der *Open Database License (OdbL)* für jeden an, ohne eine Gebühr für deren Verwendung zu erheben. Informationen zu dieser Lizenz befinden sich im Kapitel 3.4 und auf den Seiten:

- [http://wiki.openstreetmap.org/wiki/DE:Open\\_Database\\_Licence\\_-\\_Licence\\_Text](http://wiki.openstreetmap.org/wiki/DE:Open_Database_Licence_-_Licence_Text)
- <http://www.openstreetmap.de/lizenzaenderung.html>

Weiterhin stellt das Projekt Software und Anwendungen für die Bearbeitung und Auslieferung der Daten zur Verfügung. Allerdings erfolgt eine Auslieferung der Daten nicht nach einen spezifischen Standard, wie z. B. dem [OGC WMS Standard](#) (s. Kap. 3.5).

Im Rahmen dieser Arbeit soll deshalb untersucht werden, ob es eine Möglichkeit gibt, *OSM-Daten* Standard-konform über einen *Web Map Service (WMS)* auszuliefern. *OGC-konforme WMS-Implementierungen* sind beispielsweise *Deegree*, *GeoServer* und *MapServer* (s. Kap. 6.1.3). Letzterer wird bei dieser Untersuchung eingesetzt, da die Arbeit bei [Intevation](#) entsteht und diese den *MapServer* in vielen Bereichen einsetzt und nun weitere Fähigkeiten herausfinden möchte.

---

1 Eigennamen werden in dieser Arbeit kursiv dargestellt.

## 2 Ziel der Bachelorarbeit

Das Ziel dieser Arbeit ist es, einen alternativen Weg zur Darstellung von *OSM*-Daten zu untersuchen.

Eine Möglichkeit zur Erstellung von Karten aus *OSM*-Daten eröffnet die Verwendung von *Mapnik* (s. Kap. 4.2.2), welcher häufig in Verbindung mit dem *OSM*-Projekt eingesetzt wird. Mit dem Toolkit werden Pixelgrafik-Karten (PNG) gerendert („gezeichnet“), wie bei der Standard-Darstellung auf der [Hauptseite von OSM](#). Diese Ansicht wird *Mapnik*-Darstellung oder auch *Slippy Map* genannt. *Mapnik* wandelt *OSM*-Daten in Kartenkacheln (engl. Tiles) um, welche anschließend vom Webserver ausgeliefert werden. Die Auslieferung der Kartenkacheln erfolgt *Slippy Map*-konform und entspricht keinem spezifischen Standard, wie z. B. dem *OGC-WMS*-Standard.

Die genaue Funktionsweise wird in Kapitel 4.2.2 beschrieben.

Damit *OSM*-Daten über die *Slippy Map*-konforme Auslieferung hinaus Verbreitung finden und auch in Anwendungen bzw. Programmen, welche diese Art der URL basierten Auslieferung nicht unterstützen, angezeigt werden können, ergibt sich die Notwendigkeit nach einer Alternativen zu suchen. Um das primäre Ziel der Verbreitung realisieren zu können, sollte eine möglichst Standard-konforme Auslieferung der Daten erfolgen. Hierzu bietet sich der *OGC-WMS*-Standard an.

In dieser Arbeit soll am Beispiel des *MapServers* untersucht werden, inwieweit dieser geeignet ist, zum einen *OSM*-Daten Standard-konform über *WMS* auszuliefern und andererseits das Toolkit *Mapnik* beim Rendering zu ersetzen.

Des Weiteren wird überprüft, ob sich der „deutschlandspezifische“ Kartenstil [Braun11] auch mithilfe des *MapServers* umsetzen und darstellen lässt.

### 3 Ausgangssituation

Karten können viele verschiedene Situationen und Sachverhalte wiedergeben, wie die folgenden Beispiele (Abb. 1, 2, 3 und 4) zeigen.



Abbildung 1: Beispiel – Zookarte [ZOS]

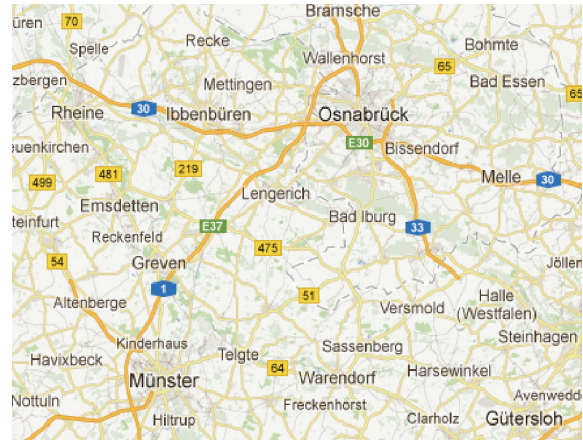


Abbildung 2: Beispiel – Straßenkarte [OSM01]



Abbildung 3: Beispiel – Liegenschaftskarte [VOS]

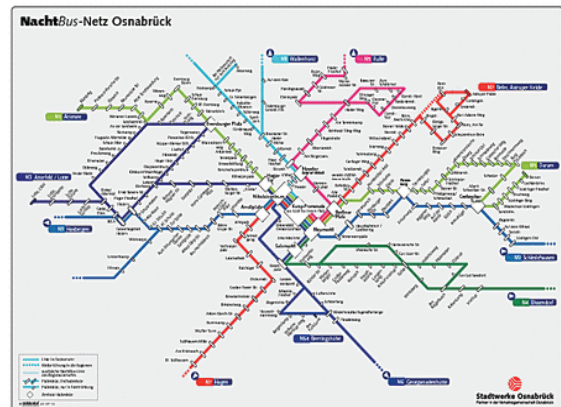


Abbildung 4: Beispiel – Busfahrplan [BOS]

Damit Karten jedoch richtig interpretiert und zweckentsprechend angewendet werden können, bedarf es aus kartographischer Sicht einige Kriterien, die es zu beachten gilt. Um diese zu verdeutlichen, wird in den nächsten beiden Kapiteln 3.1 und 3.2 zum einen das Fachgebiet Kartographie vorgestellt und zum anderen allgemeine Anforderungen an eine digitale Karte erläutert. Die Zielgruppe, welche die Karten nutzen sollen, betrifft deutsche Anwender, daher beschäftigt sich Kapitel 3.3 mit einer landestypischen Farbgebung. Des Weiteren werden in dieser Arbeit digitale Karten betrachtet, die aus OSM-Daten des OSM-Projekts bestehen, welches in Kapitel 3.4 beschrieben wird. Die Darstellung dieser Karten erfolgt nur mittels einer speziellen URL und ist somit für Anwendungen und Programme, welche diese Art der Auslieferung nicht unterstützen, unbrauchbar. Eine Verbreitung über diese Auslieferung hinaus ist nur durch Standard-Konformität möglich, wie Kapitel 3.5 erläutert.



### 3.1 Kartographie

„Die Kartographie ist das Fachgebiet, welches sich dem Sammeln, Verarbeiten, Speichern und Auswerten von Geodaten sowie deren Veranschaulichung widmet.“ [Hake02]

Die Definition beinhaltet bereits die Hauptaufgaben der Kartographie. In Bezug auf die Nutzer von Karten und kartenverwandten Darstellungen besteht die zentrale Aufgabe jedoch darin, dem Anwender die Möglichkeit zu geben, Dargestelltes richtig wahrzunehmen. Nur so ist „eine möglichst zutreffende Vorstellung und Erkenntnis der vergangenen, gegenwärtigen und geplanten Wirklichkeit“ [Hake02] gegeben. Neue Erkenntnisse aus dem eigenen sowie anderen Fachbereichen, wie Topographie, Hydrographie, Photogrammetrie, Fernerkundung und Geoinformatik bewirken eine sich ständig verändernde Wirklichkeit. Auch technische Methoden entwickeln sich ständig weiter. Folglich müssen Geodaten in geeigneten oder erforderlichen Abständen neu gesammelt und verarbeitet werden. Nur so bleibt die Aktualität von Modellen der Erdoberfläche gewährleistet.

Dieser Herausforderung stellen sich Kartographen. Die Erstellung von Karten kann dabei in sieben Schritte unterteilt werden. [Kart12]

1. Sammeln, Zweckbestimmung und Klassifizierung von Informationen
2. Auswahl des Kartennetzentwurfs
3. Festlegung des Kartenmaßstabs
4. Visualisierung der Geodaten mit geeigneten Darstellungsmethoden
5. Festlegung eines konkreten technologischen Ablaufs und Entwicklung eines redaktionellen Gesamtkonzepts
6. Kartenherstellung
7. Veröffentlichung der Karten

Karten helfen bei der Erkundung, Erkennung und Erklärung von Strukturen und Prozessen. Sie dienen der Bildung und Forschung sowie als Arbeitsmittel und Informationsquelle. Weiterhin sind Karten eine Grundlage von Darstellungen in verschiedenen Bereichen (z. B. Orientierung, Verwaltung, Planung und Input für GIS).

Eine Klassifizierung von Karten erfolgt nach ihrer Art oder nach ihrem Typ.

Die Kartenart bezieht sich vorwiegend auf den Inhalt bzw. das Kartenthema. In diesem Bereich wird u. a. zwischen einer *Topographischen* und einer *Thematischen* Karte unterschieden.

In einer *Topographischen* Karte werden beispielsweise bauliche Situationen, Gewässer, Gelände und Straßenverläufe dargestellt. Erscheinungen und Sachverhalte, welche zur Erkenntnis ihrer selbst dienen (z. B. Bevölkerung, Geologie usw.), sind Inhalt von *Thematischen* Karten.

Kartentypen hingegen beziehen sich hauptsächlich auf die Merkmale einer Karte. Die Merkmale betreffen u. a. den Maßstab (groß-, mittel-, kleinmaßstäbig), die Entstehungsart (Grundkarten, Folgekarten), die Herkunft (amtlich, privat), den Raumausschnitt (Erd-, Länder-, Seekarte, Risswerk) und den Nutzungszweck (Orientierungs-, Wanderkarte, Stadtplan).

### 3.2 Allgemeine Anforderungen an eine digitale Karte

Damit Karten ihren jeweiligen Zweck erfüllen können, müssen verschiedene Kriterien eingehalten werden, welche die allgemeinen Anforderungen betreffen.

Da in dieser Arbeit Onlinekarten erstellt werden, zeigt folgende Grafik (Abb. 5), welche Anforderungen speziell für diese gelten.

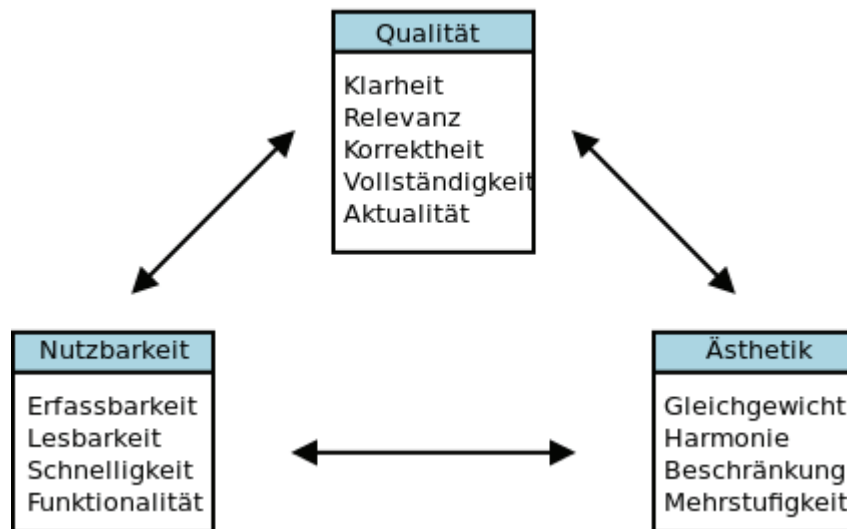


Abbildung 5: Allgemeine Anforderungen an eine Onlinekarte [nach Braun11]

Weitere Kriterien stellen die spezifischen Bedürfnisse von Nutzern dar, welche die Karten anwenden sollen. Da Karten in erster Linie visuell wahrgenommen werden, spielen Farben eine wesentliche Rolle, deren Wirkung die Attraktivität von Karten erheblich beeinflussen. Dabei lassen sich keine universellen Wirkungen jeweiliger Farben festlegen und durchaus unterschiedliche Tendenzen bezüglich ihrer Wahrnehmung ausmachen. Ursache dafür ist u. a. die Herkunft der Menschen, denn unterschiedliche Kulturen, Moden und Erziehungen prägen das Farbempfinden.

Die Tabelle 1 verdeutlicht die Interpretation von Farben im internationalen Vergleich.

	Rot	Gelb	Grün	Blau
Europa	Gefahr	Vorsicht, Achtung	Sicherheit, Sauberkeit	Männlichkeit, süß, ruhig
Japan	Zorn, Gefahr	Würde, Adel, kindisch, freudig	Zukunft, Jugend, Energie	Schande, Niederträchtigkeit
China	Freude, festlicher Anlass	Ehre, königlich	-	-
Arabischer Raum	-	Glück, Wohlstand	Fruchtbarkeit, Kraft	Jugend, Glaube, Wahrheit

*Tabelle 1: Kulturelle Unterschiede in der Farbinterpretation [Braun11]*

Karten werden in verschiedenen Bereichen eingesetzt, u. a. bilden sie den Schlüssel für die Planung von Gewerbegebieten, von Fahrradausflügen und Reiserouten oder stellen die Verteilung der Bevölkerung, z. B. für statistische Zwecke dar. Dabei beeinflussen die unterschiedlichen Funktionen die Art der Darstellung, um Sachverhalte selbsterklärend hervorzuheben. Des Weiteren erleichtert eine farbliche Umsetzung die Dechiffrierung, vorausgesetzt diese wird passend gewählt. Die Farbgestaltung orientiert sich dabei an die Herkunft der möglichen Anwender.

### **3.3 Definition des „deutschlandspezifischen“ Kartenstils**

Die Wurzeln des OSM-Projekts liegen im angelsächsischen Raum und demzufolge richtet sich die Darstellung der OSM-Hauptkarte nach dem Farbempfinden der Engländer. Um diese Karte für den Gebrauch deutscher Nutzer zu adaptieren, ergibt sich die Notwendigkeit einer farblichen Umgestaltung, weshalb nun auch ein „deutschlandspezifischer“ Kartenstil [Braun11] vorhanden ist.

Die darin festgelegten Farbwerte werden in dieser Arbeit, für die Umsetzung eines eigenen Kartenstils herangezogen und daher im Folgenden aufgeführt.

(Anm. des Autors: Die Definitionen und Bilder werden im Original aus der Bachelorarbeit [Braun11] übernommen.)

#### **Farbbetonung**

Je geringer die Ordnung der Straße desto heller wird die Farbe gewählt. Dies geht von rot für die Autobahnen über hellgelb für Kreisstraßen bis hin zu weiß für Straßen in Ortsgebieten.



Abbildung 6: Farbverlauf nach Ordnung der Straßen [Braun11]

## Farbgestaltung der Straßen

Die folgende Tabelle zeigt die abgeleitete Farbgestaltung der Straßen. Vergleichend wird die bisherige Farbgestaltung mit dargestellt.

Schlüssel	Wert	Kommentar	Bisheriger Kartenstil	„deutscher“ Kartenstil
highway	motorway	Autobahn		
highway	trunk	Auto- bahnähnliche Straße		
highway	primary	Bundesstraße		
highway	secondary	Landstraße		
highway	tertiary	Kreisstraße		
highway	residential	Gemeindestraße		

Tabelle 2: Straßenfarben im bisherigen und "deutschen" Stil [Braun11]

## Farbgestaltung von Waldflächen

Auf der weiteren Tabelle ist die dezentere Darstellung der Grünflächen gegenüber den bisherigen Waldflächen abgebildet.



Schlüssel	Wert	Kommentar	Bisheriger Kartenstil	„deutscher“ Kartenstil
landuse	forest	Waldfläche		

Tabelle 3: Waldflächen im bisherigen und "deutschen" Stil [Braun11]

## Sonstige Aspekte

Zusätzlich wird beim Aufbau des „deutschen“ Kartenstils auf die Umgestaltung von weiteren Farben und Symbolen der Karte geachtet. Unübersichtlichkeiten der Karte, wie beispielsweise zu viele gestrichelte Linien von Wald- und Wanderwegen sollen vermieden und verbessert werden.

- „weniger ist mehr“ - stärkere Generalisierung einsetzen.  
Das heißt: informativ - ja, überladen - nein
- Kontrastreiche Darstellung
- Flächen: dezentere Farben verwenden
- Linien: kräftigere Farben verwenden
- Veränderung von Beschilderungen und Symbolen

## 3.4 OpenStreetMap (OSM)

OSM bezeichnet eine *Freie* Weltkarte. Das Projekt wurde 2004 von Steve Coast in Großbritannien gegründet. Weltweit verteilt, sammeln mehr als 500.000 Interessierte, z. T. auch Fachfremde, Geodaten, entweder durch Aufnahmen mit einem GPS-Gerät oder Abzeichnen von Luftbildern (bspw. von *Bing*<sup>2</sup>). Nach der Modellierung mit einem geeigneten Editor (*Potlach*, *JOSM* u. a.) werden die Daten, entweder als Punkte (*nodes*), Linien (*ways*), Flächen (*areas*) oder Relationen (*relations*), mit den zugehörigen Attributen, welche aus einem Schlüssel (*key*) und einem Wert (*value*) bestehen, in die OSM-Datenbank eingefügt.

Einige Kommunen und Organisationen spenden ebenfalls Daten. Auf der Projektseite

<sup>2</sup> „Microsoft erstellt für Bing bis Ende 2012 neue Luftbilder und erlaubt dem OpenStreetMap Projekt das Abzeichnen[sic!] von den Bildern.“ Quelle: <http://blog.openstreetmap.de/2011/07/osm-wochennotiz-nr-52/>

<http://frida.intevation.org/> können z. B. *Freie* Vektordaten der Stadt Osnabrück heruntergeladen werden.

Mittlerweile ist *OSM* die größte *Freie* Geodatenbank. Aus den Daten entstehen Karten für Radfahrer, Wanderer, Segler, Skifahrer und viele weitere Interessengruppen. Die Karten darf jeder bearbeiten, einsetzen, kopieren und weiterverbreiten, auch für kommerzielle Zwecke [OSM08]. Die einzigen Bedingungen betreffen ein Nutzerkonto, mit einer aktuellen Mail-Adresse und das Einhalten der *OdbL*, unter welche die Daten seit 12.09.12 stehen.

Die Lizenz, deren Vereinbarung ausschließlich für den Schutz von Daten entwickelt wurde, erlaubt jedem Nutzer, die Datenbank *Frei* zu verteilen und zu verändern. Alle Daten, welche mit *OSM*-Daten kombiniert werden, fallen unter die selbe Lizenz. Bei Endprodukten ist die Angabe von *OSM*, wie in der *OdbL vorgegeben*, ausreichend, sodass nicht jeder einzelne Urheber genannt werden muss.

Kachel-Server, welche von Anwendungsentwicklern genutzt werden, stehen nicht *Frei* zur Verfügung. *OSM*-Server bestehen ausschließlich aus gespendeten Ressourcen. Dadurch ist die Kapazität sehr begrenzt. Eine intensive Nutzung der Kacheln würde die Möglichkeit der Anwender, Karten zu bearbeiten, erheblich beeinflussen. Des Weiteren ist es ein Missbrauch der Leistung gegenüber den Spendern und Sponsoren, welche Hardware und Bandbreite zur Verfügung stellen. Die Voraussetzungen für die Nutzung der Kacheln enthält die „Tile-Usage-Policy“.

Unabhängig von *OSM* wurde 2006 die *OSM-Foundation* gegründet. Hierbei handelt es sich um eine Non-Profit-Organisation, die durch verschiedene Aktivitäten Gelder und Spenden sammelt, um *OSM* finanziell zu unterstützen.

Zusammenfassend bietet *OSM* die Möglichkeit, *Freie* Kartendaten mittels GPS-Gerät oder Mobiltelefon zum Navigieren und zum Berechnen von Routen zu verwenden, eigene Karten zu erzeugen, sowie vorhandene Kartendarstellungen (z. B. *Slippy Map*) über das Internet zu nutzen (bspw. durch Einbinden in eine eigene Webseite).

### **3.5 Standardisierung von Geodaten**

Eine Auslieferung von *OSM*-Karten erfolgt lediglich *Slippy Map*-konform und verhindert somit eine Anzeige in Programmen und Anwendungen, welche diese Art der Auslieferung nicht unterstützen. Um eine Verwendung in verschiedenen Systemen zu gewährleisten, werden einheitliche Regeln, bezüglich der Konformität benötigt, daher beschäftigen sich verschiedene Organisationen mit der Standardisierung von Geodaten, um die Verbreitung der offenen Standards weiter zu forcieren. Im Bereich der Geoinformatik sind die *International Standardization Organisation (ISO)* und das *Open Geospatial Consortium (OGC)* von großer Bedeutung, dessen Standards die Basis für die Untersuchungen in dieser Arbeit sind.

## OGC

Das OGC wurde 1994 gegründet und besteht mittlerweile aus über 350 Mitgliedern aus Industrie, öffentlicher Verwaltung, Privatwirtschaft, Universitäten und Forschung. Die zentralen Aufgaben sind die globale Entwicklung, Förderung und Harmonisierung von offenen Standards. Die offenen Standards sollen dabei helfen GIS-Systeme für einen einfachen Austausch von Geodaten und Anweisungen aufzubauen. Für die Interoperabilität beim Transfer von Geodaten zwischen Client und Server über das Internet, hat das OGC in den letzten Jahren hauptsächlich Standards für Webservice-Umgebungen (*OGC Web Services (OWS)*) entwickelt. [OGC12]

Zwei dieser spezifizierten Standards sind *Web Map Service (WMS)* und *Web Feature Service (WFS)*. Innerhalb des OGC-Kontextes bilden *WMS* und *WFS* ein Beispiel für die Architektur eines verteilten GIS. Der Zugriff auf die Geodaten (ausschließlich Vektordaten) erfolgt durch *WFS*. Für die Visualisierung der Daten ist der *WMS* zuständig (s. Abb. 7).

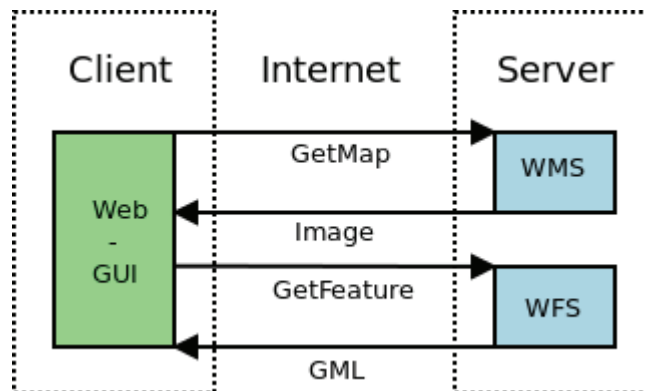


Abbildung 7: Transfer von Geodaten zwischen Client und Server mittels WMS und WFS

Durch den Client wird eine Anfrage, mit entsprechenden Parametern an einen *WMS*- oder *WFS*-Dienst gesendet. Der jeweilige Dienst bereitet die Geodaten anhand der Anfrage-Parameter auf und sendet diese in einem geeigneten Format an den Client zurück. Als Grundlage für den Austausch der Geodaten dient ein HTTP-Protokoll, welches die Methoden "GET"<sup>3</sup> und "POST" unterstützt.

Diese Arbeit befasst sich in erster Linie mit der Visualisierung der Geodaten, weshalb die genaue Funktionsweise mittels der *WMS*-Spezifikation erläutert wird.

<sup>3</sup> Bezeichnungen von Protokollen, Funktionen und Skripten werden in dieser Arbeit durch Hochkommata gekennzeichnet.

## WMS

Die WMS-Spezifikation umfasst Beschreibungen der Anfrage-Parameter, welche von einem Nutzer benannt werden müssen oder können. Weiterhin wird festgelegt, wie der WMS-Dienst aus einer solchen Anfrage Kartenausschnitte und gegebenenfalls die inhaltliche Aufbereitung der Karte erzeugen soll. Dabei ermöglichen drei verschiedenen Funktionen einem Nutzer über "HTTP" mit dem WMS-Dienst zu kommunizieren.

1. "GetCapabilities"
2. "GetMap"
3. "GetFeatureInfo"

Mit "GetCapabilities" wird die Fähigkeit des WMS abgefragt (s. Listing 1). Als Antwort erhält ein Nutzer ein XML-Dokument mit Metainformationen. Diese umfassen allgemeine Angaben zum Anbieter des WMS, unterstützte Ausgabeformate und abfragbare Layer.

```
http://sweden.atlas/cgi-bin/mapserv-gp?map=/home/abroermann/osm-demo/basemaps/osm-german\_style.map&VERSION=1.1.1&SERVICE=WMS&REQUEST=GetCapabilities
```

Listing 1: Beispiel-URL einer "GetCapabilities" - Anfrage

Die Funktion "GetMap" liefert ein georeferenziertes Rasterbild (Karte) vom WMS zurück. Eine Anfrage besteht aus notwendigen und optionalen Parametern. Beispielsweise müssen gewünschte Layer, das zugrunde liegende Koordinatensystem, Größe der Kartenausgabe und Ausgabeformat immer per Anfrage angegeben werden. Die Hintergrundfarbe oder auch das Ausgabeformat eines Fehlerreports sind hingegen nicht notwendig.

Welche Parameter mindestens angegeben werden müssen, zeigt Listing 2.

```
http://sweden.atlas/cgi-bin/mapserv-gp?map=/home/abroermann/osm-demo/basemaps/osm-german\_style.map&VERSION=1.1.1&SERVICE=WMS&REQUEST=GetMap&LAYERS=land8,landuse8,waterarea8,waterways8,railways8,roads8,borders8,places8&STYLES=&SRS=EPSG:3857&BB0X=701959.5496,6601397.5496,1318680.4504,7218118.4504&WIDTH=800&HEIGHT=800&FORMAT=image/png&
```

Listing 2: Beispiel-URL einer "GetMap" - Anfrage

Das zurückgelieferte Bild (s. Abb. 8) sieht folgendermaßen aus.





Abbildung 8: Ergebnis einer "GetMap" - Anfrage

Des Weiteren kann ein WMS freiwillig Angaben zu einer Position im dargestellten Kartenausschnitt beantworten. Dazu wird die optionale Funktion "GetFeatureInfo" verwendet. Das zurückgelieferte Dokument enthält festgelegte, thematische Informationen (s. Kap. 1) der zugrunde liegenden Daten.

Die Anfrage-URL besteht wie bei "GetCapabilities" aus notwendigen und optionalen Parametern. "GetFeatureInfo" wird in dieser Arbeit nicht verwendet und daher nicht weiter erläutert.

## 4 Technische Gegebenheiten

Für die Darstellung von digitalen Karten werden *WebGIS*-Systeme benötigt. Diese bezeichnen im Allgemeinen eine *GIS*-Applikation, welche auf *Webservices* zurückgreifen, die Geodaten ausliefern und mittels einer speziellen *Web*-Applikation oder einem Geobrowser darstellen. Der Transfer von Geodaten über ein Netzwerk erfolgt nach dem Prinzip einer *Client-Server*-Architektur wie in Abb. 9 gezeigt wird.

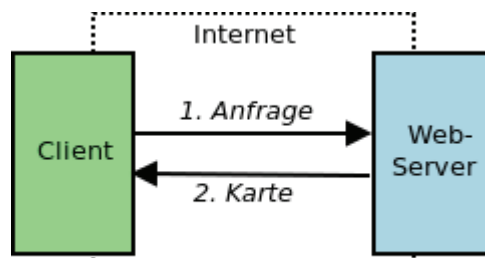


Abbildung 9: Allgemeiner Transfer von Geodaten über das Internet

*WebGIS*-Systeme unterscheiden sich nach ihrer Funktionalität aufseiten des Server-Rechners bzw. Client-Rechners. So enthält beispielsweise ein Desktop-GIS auf dem Client-Rechner alle GIS-Funktionalitäten, einschließlich Geodaten. Wird ein Client-Rechner primär zur Visualisierung und gegebenenfalls für einfache Interaktionen mit einer Karte (z. B. Verschieben der Karten, Zooming, Distanzmessung usw.) verwendet, befinden sich die Hauptfunktionen auf dem Server-Rechner. Der Aufruf zur Darstellung von Karten erfolgt dabei durch einen zeitgemäßen Webbrowser, welcher als Client dient und eine, vom Nutzer gestellte Anfrage an den Server leitet. Dieser verarbeitet die Anfrage und sendet die benötigten Daten in gewünschter Form an den Client zurück.

*WebGIS*-Systeme werden in vielen Bereichen eingesetzt, beispielsweise verwenden Kommunen, Städte und Länder Auskunft- und Informationssysteme mit *WebGIS*-Funktionen in verschiedenen Bereichen, wie z. B. Katasteramt, Umweltverträglichkeitsprüfungen, Standortanalyse und Einsatzsteuerung von Polizei und Militär. Im privaten Gebrauch können *WebGIS*-Systeme den Alltag, u. a. durch Standortsuche oder Routenplanung im Internet erleichtern.

Die verwendeten *WebGIS*-Systeme basieren entweder auf kommerzielle (*ArcView*) oder Freie Software (*degree*, *GeoServer*, *MapServer*).

Das *WebGIS*-System von *OSM*, basiert auf das statische Mapserver-Prinzip, wobei eine Karte erstellt und in Form eines Rasterbildes („static map“) angeboten wird.

Die Grundlage für die Kartenerstellung bilden *OSM*-Daten, welche mit *Mapnik* gerendert werden.

Die genaue Funktionsweise wird im folgenden Kapitel erläutert

## 4.1 WebGIS-System zur Darstellung von OSM-Daten mittels Mapnik

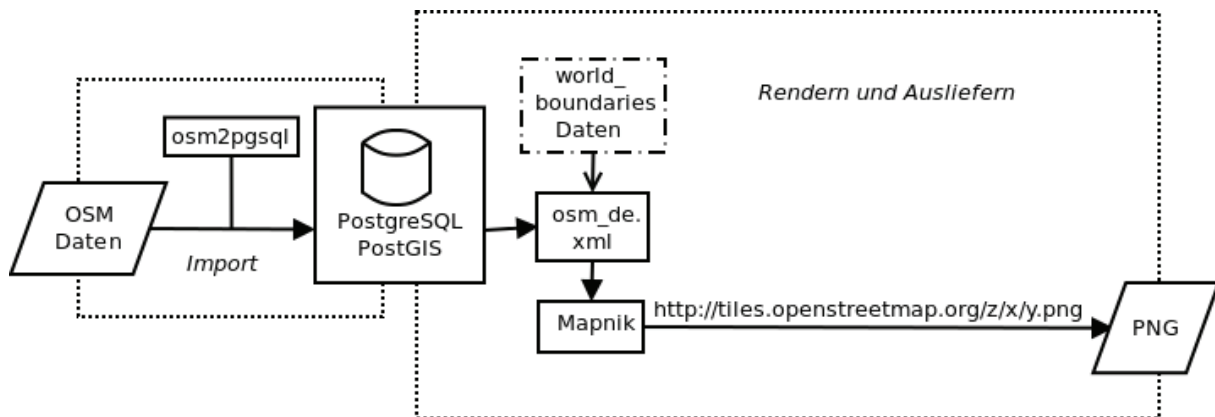


Abbildung 10: Systemarchitektur zur Darstellung von OSM-Daten mittels Mapnik

Wie in Abb. 10 hervorgeht, setzt sich der Gesamtablauf aus zwei wesentlichen Prozessen zusammen.

Der erste Prozess bezieht sich auf den Import der OSM-Daten in eine PostgreSQL/PostGIS-DB, mithilfe des Tools *osm2pgsql*, welches die erforderliche Tabellenstruktur für Mapnik erstellt. Die Bestandteile und ihre jeweilige Funktion werden in Kap. 4.2.1 beschrieben.

Das Rendern der Daten durch Mapnik, auf Grundlage der Datei "osm\_de.xml", welche die Art der Darstellung festlegt und die Slippy Map-konforme Auslieferung, stellen den zweiten Prozess dar. (s. Kap. 4.2.2)

Die "world\_boundaries/"-Daten werden nur zur Darstellung von Ländergrenzen und Küstenlinien in niedrigen Zoombereichen verwendet (s. Kap. 4.2.1) und sind daher nicht unbedingt notwendig.

### 4.2.1 Import von OSM-Daten in PostgreSQL/PostGIS mittels osm2pgsql

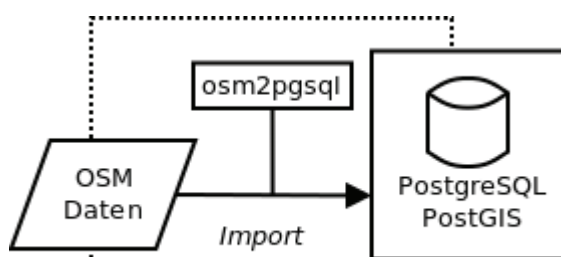


Abbildung 11: Importprozess mithilfe von osm2pgsql

#### OSM-Daten

OSM-Daten werden von verschiedenen Servern zur Verfügung gestellt. Eine Liste dieser befindet sich auf der Seite <http://wiki.openstreetmap.org/wiki/Planet.osm>.

In dieser Arbeit dient der Geodatenserver der Geofabrik [GEO12] als Bezugsquelle.

Die Geofabrik bereitet *Freie* Geodaten von OSM auf und verarbeitet diese. Hergestellt werden Shapefiles, Kartenbilder, Kacheln oder komplette Web-Kartenanwendungen. Je nach eigenem Interesse kann ein geographisches Gebiet frei gewählt werden. Eine Liste aller vorhandenen Datensätze befindet sich auf der Seite <http://download.geofabrik.de/osm/>. Die Dateien werden in regelmäßigen Abständen aus dem „Planet file“ ausgeschnitten, welches einen Gesamtabzug der OSM-DB darstellt.

Mithilfe des Tools *osm history splitter*<sup>4</sup> werden folgende Bereiche ausgeschnitten:

- Kontinente
- Länder nach Bedarf
- für Nordamerika: Census-Regionen
- für Deutschland: Bundesländer; in Bayern, NRW, Ba-Wü auch Regierungsbezirke
- für Großbritannien: England, Schottland, Wales; England in einzelne Counties
- für Frankreich: Regionen

Natürlich kann auch das gesamte „Planet file“ verwendet werden. Das Laden der Daten kann dabei, je nach Leistung des Rechners, viel Zeit in Anspruch nehmen und unter Umständen einige Stunden bis Tage dauern.

Für das Veranschaulichen der Erstellung von Karten mit *Mapnik* und der daraus resultierenden Darstellung, werden OSM-Daten von Niedersachsen verwendet. Der Datensatz „[niedersachsen.osm.pbf](#)“ hat eine Größe von 123M und das Laden in eine DB

<sup>4</sup> Werkzeug um einzelne Bereiche aus zuschneiden. Basiert auf Jochen Topf.  
<https://github.com/MaZderMind/osm-history-splitter/>

beträgt wenige Minuten.

OSM-Daten können nur aus einer *PostgreSQL/PostGIS*-DB gelesen werden, weshalb sich die Notwendigkeit ergibt, diese zu verwenden.

## **PostgreSQL/PostGIS**

*PostgreSQL* ist ein *Freies* objektrelationales Datenbanksystem. Das Projekt *POSTGRES* entstand im Jahr 1980 an der Universität in Berkeley (Kalifornien) im Fachbereich Informatik und wird aktuell von Entwicklern auf der gesamten Welt betreut. Die Funktionsweise entspricht einem Client-Server-Modell. Der Server bietet Dienste an, die von einem Client angefordert werden können. Das Serverprogramm bei *PostgreSQL* ist *postmasters*. Dieses verwaltet Datenbankdateien und clientseitige Verbindungen, mit den gestellten Anfragen. Verschiedene Client-Programme können für die Kommunikation mit dem Server genutzt werden. *PostgreSQL* beinhaltet dazu einen einfachen, aber schnellen Datenbankmonitor *psql*, und als grafischen Client z. B. *PgAccess*. [Post12]

Um die Verwaltung von geographischen Daten zu ermöglichen wird die *PostGIS*-Erweiterung eingesetzt.

Das *PostGis*-Projekt wird von der *Open Source Geospatial Foundation (OSGeo)* betreut und implementiert die *Simple Feature Access-Spezifikation* des *OGC*.

## **osm2pgsql**

Zum Import der *OSM*-Daten muss das Tool *osm2pgsql* verwendet werden, welches von *Debian* („*Squeeze*“) als fertiges Paket vorliegt. Die Installation erfolgt schnell und ein Aufruf des Tools mit dem Parameter `[-help]` oder `[-h]`, listet alle Parameter auf, welche im Zusammenhang mit *osm2pgsql* verwendet werden können. Auf der Wiki Seite [osm2pgsql](#) des *OSM*-Projekts befindet sich eine Beschreibung zum Werkzeug.

Da *Mapnik* unabhängig vom *OSM*-Projekt entstanden ist und sich das Datenformat unterscheidet, wird mithilfe von *osm2pgsql* dieses konvertiert und anschließend in die DB geladen. Dabei entsteht eine Tabellenstruktur mit vier Tabellen, sowie es von *Mapnik* verlangt wird.

- `planet_osm_point` → Sammlung aller punktförmigen Objekte
- `planet_osm_line` → Sammlung aller linienförmigen Objekte
- `planet_osm_polygon` → Sammlung aller flächenförmigen Objekte
- `planet_osm_roads` → Sammlung einiger linienförmigen Objekte für niedrige Zoomstufen

Die *OSM*-Daten beinhalten auch Daten von Ländergrenzen und Küstenlinien, welche jedoch nicht als Polygone und nur mit hohem Detailgehalt vorliegen. Daher eignen sie sich nicht für die Darstellung in niedrigen Zoomstufen (bei großen Ausschnitten), weshalb *Mapnik* zum Rendern von Ländergrenzen und Küstenlinien externe Daten in Form von Shapefiles nutzt. Diese können vom *OSM-Tile-Server* bezogen werden.

## 4.2.2 Rendern der OSM-Daten mit Mapnik und Auslieferung

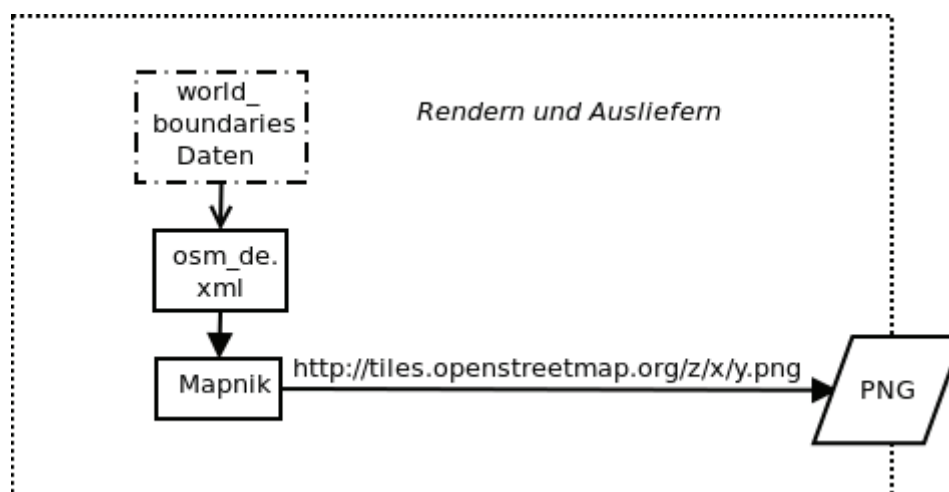


Abbildung 12: Prozess: Rendering mittels Mapnik und Auslieferung

Mit *Mapnik* können Karten aus OSM-Daten gerendert werden. Entwickelt wurde das Programm von Artem Pavlenko in der Programmiersprache C++ mit Bezug zu einer Python-API.

Das Resultat von *Mapnik* sind Rastergrafiken, welche sich gut für die Verwendung am Bildschirm eignen.

Aus den Daten, die alle als Vektoren vorliegen, können theoretisch Karten generiert werden, die jeden beliebigen Ausschnitt der Erde, in jeder beliebigen Auflösung und Projektion, mit unterschiedlichen Stilen anzeigen. Dies erfordert jedoch eine Menge Rechenzeit, aufgrund der Datenmenge, weshalb eine stückweise Vorberechnung der Karten im Web zu Rastergrafiken erfolgt. Die gesamte Welt wird dabei in Kacheln zerlegt und in einzelne quadratische Bereiche unterteilt, welche zusammengesetzt ein Bild ergeben. Bei einer Anfrage wird dann die passende Grafik geliefert.

Die *Slippy Map* auf der Hauptseite vom OSM-Projekt ist auch eine vorgerenderte Karte. Sie stellt eine Abbildung der Erde im Bereich von 85° Nord bis 85° Süd in der Mercator-Projektion dar. Dabei werden Kacheln mit einer Größe von 256 x 256 Pixeln benutzt. Die Gesamtabbildung der Welt wird mit einer Kachel dargestellt und entspricht der niedrigsten Zoomstufe (0). Bei der nächsthöheren Zoomstufe (1) entstehen vier Kacheln mit einer Gesamtgröße von 512 x 512 Pixeln (s. Abb. 13).

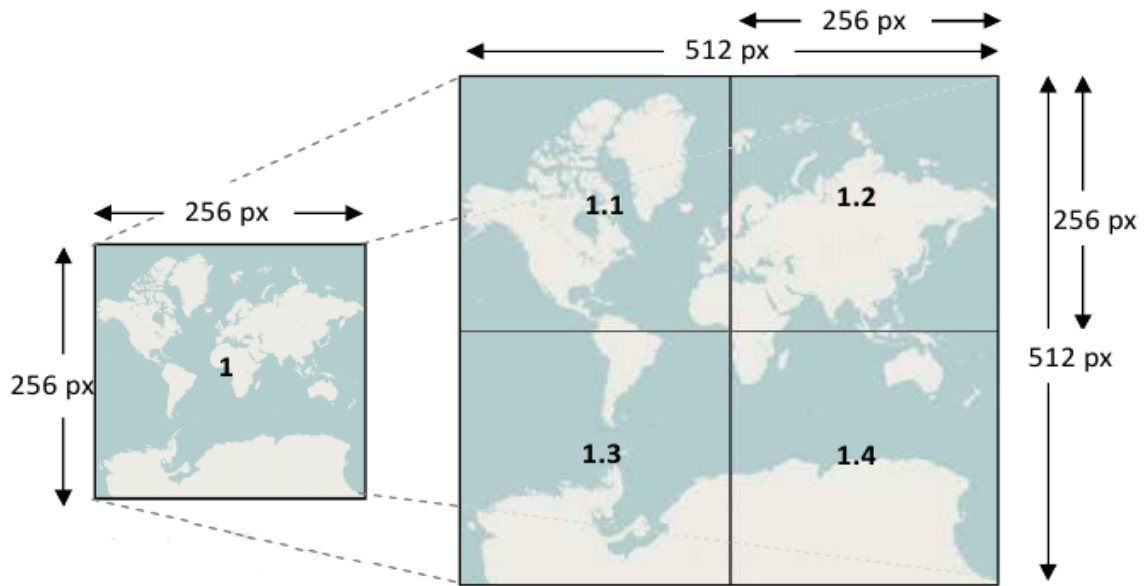


Abbildung 13: Kacheinteilung [Braun11]

Das bedeutet, bei jeder höheren Zoomstufe, wird jede bereits vorhandene Kachel in vier weitere geteilt. Die höchste Stufe (18) besteht dann aus 69 Mrd. Kacheln und die dabei benötigte Speicherkapazität beträgt beachtliche 344 TB.

Die Kartenkacheln können vom *OSM-Tile-Server* mittels spezieller URL-Anfrage (s. Listing 3), welche die Zommstufe ( $z$ ) und die Lagekoordinaten ( $x$ ,  $y$ ) beinhaltet, bezogen werden. Die Auslieferung erfolgt anschließend *Slippy Map-konform*.

```
http://tiles.openstreetmap.org/z/x/y.png
```

Listing 3: URL zur Anforderung von Kartenkacheln vom OSM-Tile-Server

Für das Rendern eines eigenen Kartenstils, benötigt *Mapnik* eine Datei im XML-Format, welche die Definitionen zur Darstellung der Karte beinhaltet.

Der aktuelle *Mapnik*-Stil ist in der Datei "osm.xml" definiert und kann vom [OSM-Repository](#) heruntergeladen werden [Braun11].

Die Datei besteht aus einem „Style“-Bereich, welcher „Style“-Elemente, mit den Stil-Definitionen für Objekte enthält und aus einem „Layer“-Bereich zur Festlegung, welche Daten aus der Datenbank in den „Style“-Elementen verwendet werden sollen. Der Weiteren beinhaltet der „Layer“-Bereich SQL-Befehle zum Abrufen der entsprechenden Daten aus der DB [Braun11].

Die Abbildung 14 reflektiert den allgemeinen Aufbau der Datei.

```

<?xml version="1.0" encoding="utf-8"?>
<Map bgcolor="#b5d0d0" srs="&srs900913;">
  <Style name="...">
    <Rule>
      ...
    </Rule>
  </Style>
  ...
  <Layer name="..." status="on" srs="...">
    <StyleName>...</StyleName>
    <Datasource>
      ...
    </Datasource>
  </Layer>
  ...
</Map>

```

Abbildung 14: Allgemeiner Aufbau der Datei osm.xml

Die Datei "osm.xml" bildet das Grundgerüst für die Gestaltung des „deutschlandspezifischen“ Kartenstils. Die Änderungen betreffen im Einzelnen die Symbol- und Farbgestaltung (s. Kap. 3.3), sowie die Generalisierung und Veränderung von Details [Braun11].

Die Abbildung 15 zeigt das Kartenbild, welches *Mapnik* auf Grundlage der neu entstandenen "osm\_de.xml" rendert und ausliefert.

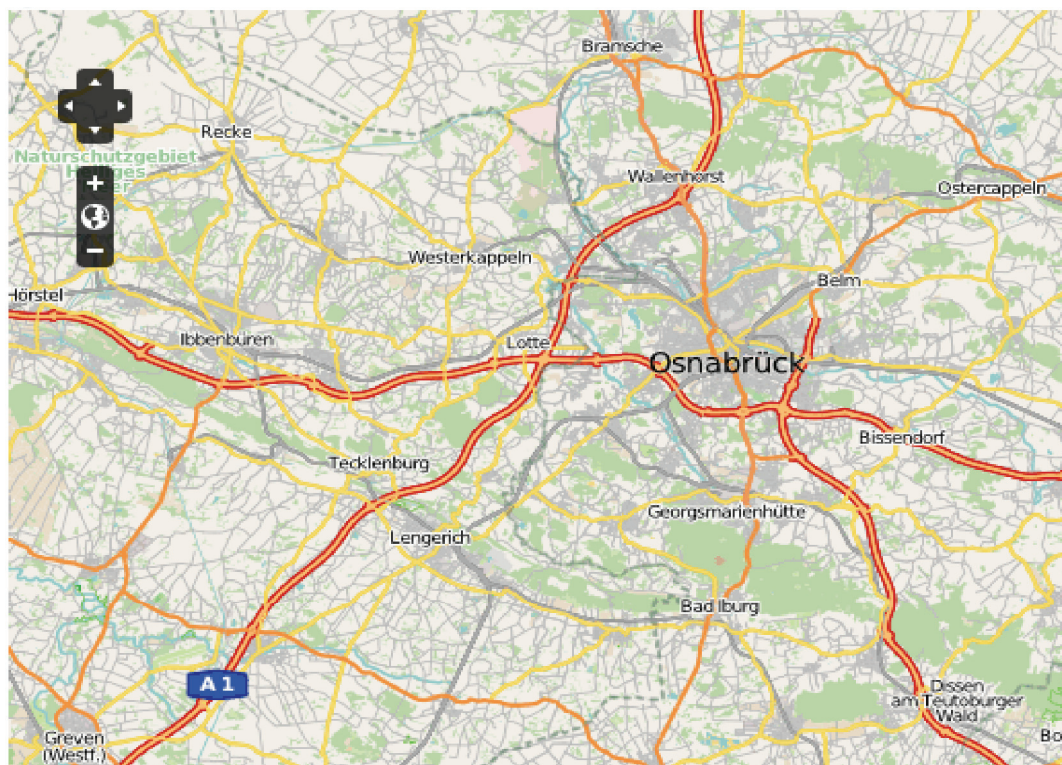


Abbildung 15: OSM deutscher Stil



## 5 Analyse der Kartenerstellung mittels Mapnik

Alle benötigten Bestandteile (s. Tab. 4), zur Darstellung von OSM-Daten mittels *Mapnik* werden in Kap. 5.1 hinsichtlich des Installationsaufwands betrachtet. Eine Einschätzung bezüglich des Aufwands, zur Umsetzung des „deutschlandspezifischen“ Kartenstils erfolgt anschließend in Kap. 5.2.

In Kap. 5.3 werden die Funktionalitäten zusammengefasst, welche die Darstellung von OSM-Daten mittels *Mapnik* bietet und weshalb dieses nicht für verteilte GIS-Systeme geeignet ist.

### 5.1 Aufwand bei der Installation aller benötigten Bestandteile

Bestandteil	Bezugsquelle
OSM-Daten	<a href="http://wiki.openstreetmap.org/wiki/Planet.osm">http://wiki.openstreetmap.org/wiki/Planet.osm</a>
<i>osm2pgsql</i>	Debian Paket
<i>PostgreSQL/PostGIS</i>	Debian Paket
<i>Mapnik</i>	<a href="http://mapnik.org/download/">http://mapnik.org/download/</a> oder <a href="https://github.com/mapnik/mapnik">https://github.com/mapnik/mapnik</a>
<i>osm.xml</i>	<a href="http://svn.openstreetmap.org/applications/rendering/mapnik/osm.xml">http://svn.openstreetmap.org/applications/rendering/mapnik/osm.xml</a>
<i>world_boundaries</i>	<a href="http://tile.openstreetmap.org/world_boundaries-spherical.tgz">http://tile.openstreetmap.org/world_boundaries-spherical.tgz</a> <a href="http://tile.openstreetmap.org/processed_p.tar.bz2">http://tile.openstreetmap.org/processed_p.tar.bz2</a> <a href="http://tile.openstreetmap.org/shoreline_300.tar.bz2">http://tile.openstreetmap.org/shoreline_300.tar.bz2</a> <a href="http://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/cultural/10m-populated-places.zip">http://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/cultural/10m-populated-places.zip</a> <a href="http://www.naturalearthdata.com/http://www.naturalearthdata.com/download/110m/cultural/110m-admin-0-boundary-lines.zip">http://www.naturalearthdata.com/http://www.naturalearthdata.com/download/110m/cultural/110m-admin-0-boundary-lines.zip</a>

Tabelle 4: Bezugsquellen der verwendeten Bestandteile zur Darstellung von OSM-Daten mittels *Mapnik*

Das Laden der OSM-Daten und die Installation der *PostgreSQL/PostGIS*-DB verlaufen problemlos. Das Installieren von *Mapnik* kann allerdings Schwierigkeiten bereiten, da *Debian* kein fertiges Programmpaket mit der aktuellen Version enthält, sodass dieses eigenständig erfolgen muss. Dazu wird zunächst *Python* konfiguriert und ausgeführt, sowie alle zusätzlich benötigten Bibliotheken installiert. Läuft *Mapnik* fehlerfrei, kann die Datei "om.xml" einfach aus dem Repository heruntergeladen und im *Mapnik*-Verzeichnis gespeichert werden. Das Beziehen und Entpacken der "world\_boundaries"-Daten, sowie das Verschieben der Inhalte in einen Ordner ist sehr zeitaufwendig.

## 5.2 Aufwand beim Umsetzen des „deutschlandspezifischen“ Kartenstils

Die Karte musste nicht von Grund auf neu gestaltet werden, weshalb bei der Umsetzung des neuen Stils keine Schwierigkeiten entstehen. Die vorhandene "osm.xml" Datei kann bequem an den entsprechenden Stellen verändert werden. Einzige Voraussetzung dazu, ist die Kenntnis über den genauen Aufbau der Datei und den Zusammenhang mit weiteren, inkludierten Dateien.

## 5.3 Schlussfolgerung

Die Darstellung von OSM-Daten mittels *Mapnik* liefert immer eine Rastergrafik, welche in eine Web-Applikation eingebunden werden kann und somit Anwendern einige Funktionen zur Interaktion mit der Karte ermöglicht.

Allerdings werden nur *Slippy Map*-konforme (s. Kap. 4.2.2) Kartenbilder ausgeliefert, die keinem spezifischen Standard entsprechen und somit nicht für die Anwendung in verteilten GIS-Systemen, welche eine globale Geodateninfrastruktur ermöglichen, geeignet sind,

Im Unterschied zu den bisher betrachteten GIS-Systemen, entspricht der Aufbau bei verteilten GIS-Systemen, nicht mehr einer starren Client-Server-Architektur, wie aus der Abbildung 16 hervorgeht.

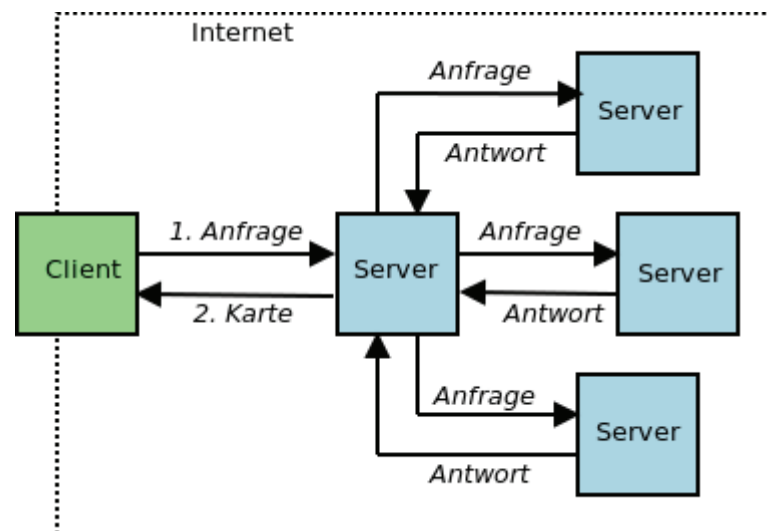


Abbildung 16: Architektur eines verteilten GIS-Systems

Ein Client stellt nun eine Anfrage an einen Server, welcher ebenfalls die Funktion eines Clients übernehmen kann und diese an verschiedene Rechner, mit GIS-Funktionalitäten bzw. Geodaten weiterleitet.

Grundlage für den Austausch von Geodaten zwischen verschiedenen Server-Rechnern im Internet bildet ein Datenformat, auf welches alle bereitgestellten Dienste lesend und schreibend zugreifen können müssen. (s. Kap. 3.5)

Somit ergibt sich die Notwendigkeit nach einer Alternativen zu suchen, um das primäre Ziel der globalen Verbreitung einer Geodateninfrastruktur zu realisieren.

In dieser Arbeit soll daher am Beispiel des *MapServers* untersucht werden, inwieweit dieser geeignet ist, *OSM*-Daten zum einen Standard-konform über *WMS* auszuliefern und andererseits *Mapnik* beim Rendering zu ersetzen.

## 6 Entwurf einer neuen Architektur

### 6.1 WebGIS-System zur Darstellung von OSM-Daten mittels MapServer

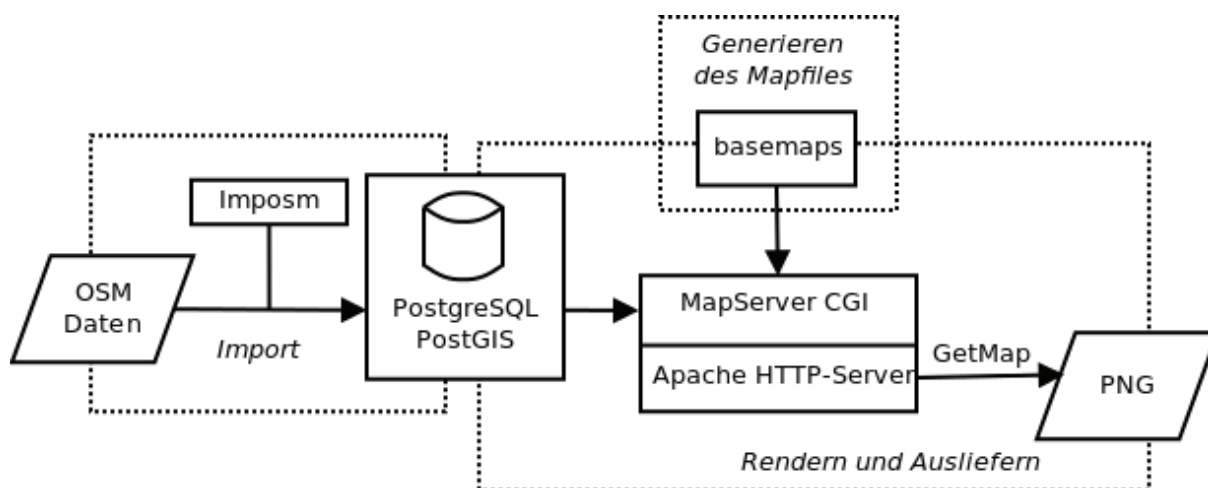


Abbildung 17: WebGIS-System zur Darstellung von OSM-Daten mittels MapServer

In der Abbildung 17 ist zu erkennen, dass sich der Ablauf nun aus drei wesentlichen Prozessen zusammensetzt.

Im ersten Prozess (s. Kap. 6.1.1) werden die *OSM*-Daten nun mithilfe von *Imposm* in die *PostgreSQL/PostGIS*-DB importiert. Diese verwendet *basemaps* im zweiten Prozess (s. Kap. 6.1.2), zum Generieren des Mapfiles, wodurch *MapServer* im dritten Prozess (s. Kap. 6.1.3) die Karte rendern kann und anschließend durch den *Apache* HTTP-Server ausgeliefert wird.

Die genauen Funktionen der einzelnen Bestandteile, innerhalb der Prozesse, werden ebenfalls separat in den folgenden Unterkapiteln betrachtet und erläutert.

### 6.1.1 Import der OSM-Daten in die PostgreSQL/PostGIS-DB mittels *Imposm*

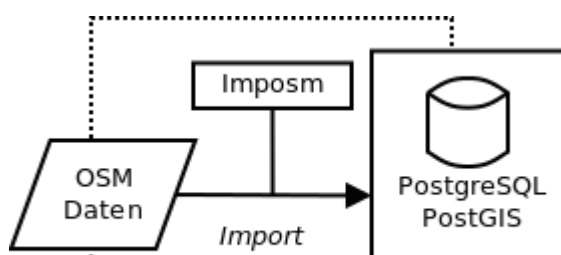


Abbildung 18: Importprozess mithilfe von *Imposm*

Die OSM-Daten (s. Kap. 4.2.1) werden mit *Imposm* in eine *PostgreSQL/PostGIS*-DB (s. Kap. 4.2.2) importiert. Dabei wird eine Datenbankstruktur, mit Tabellen für jeweils unterschiedliche Datentypen angelegt, die vor allem für das Zeichnen von Karten optimiert wurde und somit einen schnelleren Zugriff ermöglicht. [Imp12] Außerdem werden automatisch generalisierte Tabellen für kleine Maßstäbe erstellt, welche zum Abruf von großen Datenmengen (bspw. Straßennetzwerke) erforderlich sind. Ein weiterer Vorteil von *Imposm* ist, dass ausschließlich Daten importiert werden, welche für den Anwendungszweck von Bedeutung sind. Die Bearbeitung der Daten wird dadurch vereinfacht und zeitlich verkürzt. [Omn12]

Hilfen zum Umgang mit *Imposm* befinden sich auf der Seite <http://imposm.org/>.

### 6.1.2 Generieren des Mapfiles mithilfe von *basemaps*

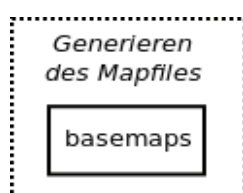


Abbildung 19: Generieren des Mapfiles mithilfe von *basemaps*

Während der Arbeit hat sich herausgestellt, dass die manuelle Erstellung von Mapfiles für die Darstellung und Auslieferung von OSM-Daten sehr aufwendig ist, aber mithilfe des Projekts *basemaps* vereinfacht werden kann und deshalb bei der semi-automatischen Erstellung von Mapfiles eingesetzt wird. [bas12]

Das Projekt *basemaps* befindet sich auf der Seite <https://github.com/mapserver/basemaps>, die jedoch keine Installationsanleitung für die GNU/Linux-Distribution Debian 6.0 („Squeeze“) enthält, welches für die Arbeit verwendet wird. Eine ausführliche Benutzerhilfe ist ebenfalls nicht vorhanden. Daraus ergibt sich die Notwendigkeit, sowohl eine [Installationsanleitung](#) als auch eine detaillierte [Beschreibung](#) zum Umgang mit dem Programm zu verfassen.

Die in Englisch verfassten Dokumente befinden sich in einem persönlich erstellten Git-Repository auf der Seite <https://github.com/arianebroermann/basemaps> und werden somit der Community zur Verfügung gestellt.

Beide Dokumente sind auf der beigefügten CD enthalten.

### 6.1.3 Rendern der OSM-Daten mit MapServer und Auslieferung der Ergebnisse

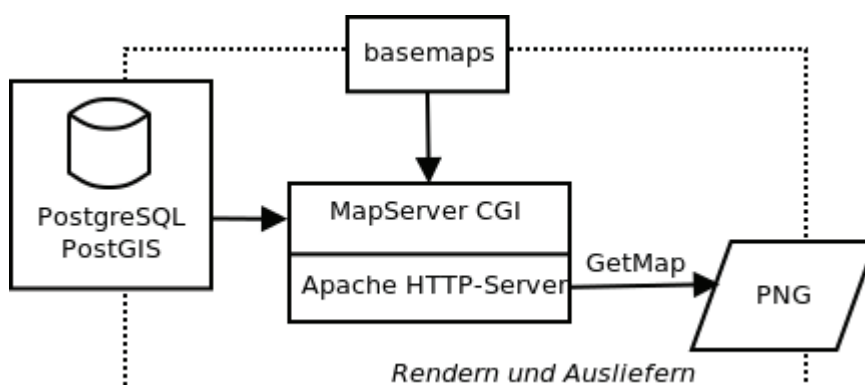


Abbildung 20: Prozess: Rendering mittels MapServer und Auslieferung durch Apache HTTP-Server

Der *MapServer* ist eine in der Programmiersprache C geschriebene *Freie Rendering-Engine* für Geodaten, welcher ursprünglich an der Universität in Minnesota (UMN), mit Unterstützung der NASA entwickelt wurde und seit 2007 ein offizielles Projekt der *OsGeo* ist.

In diesem Prozess (s. Abb. 20) wird der *MapServer* als Kartenserver eingesetzt. Seine Aufgabe besteht darin, dynamische Karten in Echtzeit (On-Demand) zu rendern und über das Internet anzuzeigen. Um die generelle Funktionsweise zu verstehen, wird diese im folgenden erläutert.

Die wichtigsten Bestandteile des *MapServers* bestehen aus einem Mapfile, Geodaten, HTML-Seiten, "Mapserver-CGI" und dem HTTP-Server. Wie diese zusammenhängen, zeigt die Abbildung 21.

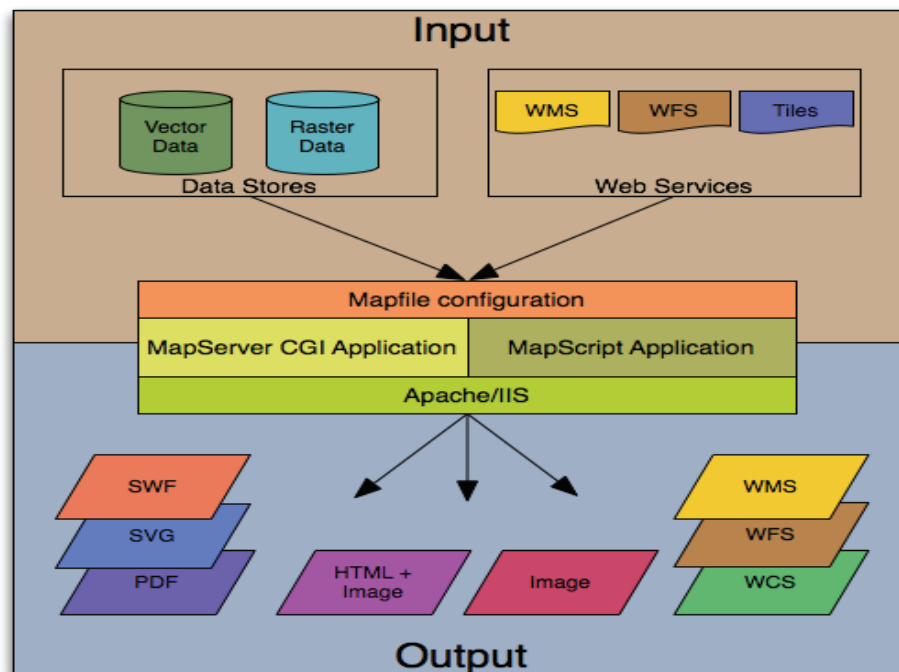


Abbildung 21: MapServer Architektur [Map01]

Ein Client stellt eine Anfrage an einen HTTP-Server (z. B. *Apache*, *Microsoft-IIS*), welcher diese zur GIS-Schnittstelle "MapServer-CGI" weiterleitet, wo eine Auswertung und Interpretation GIS-spezifischer Kommandos und der Datenzugriff erfolgt.

Die Darstellung der Karte und mögliche Interaktionen mit dieser werden durch das Mapfile festgelegt, welches die zentrale, textbasierte Konfigurationsdatei mit strukturiertem Aufbau der *MapServer*-Anwendung ist und dessen Name sich aus der Dateierweiterung "\*.map" ableitet.

Darauf basierend erstellt „MapServer-CGI“ die Karte und liefert diese an den HTTP-Server zurück, welcher die Antwort mittels HTTP-Protokoll fertig stellt und zum Client sendet.

Eine ausführliche Dokumentation sowie die aktuelle *MapServer* Version zum Download sind auf der Seite <http://mapserver.org/> zu finden [Map01].

Im vorliegendem Fall wird der *MapServer* zur Laufzeit von den durch *basemaps* generierten Mapfiles gesteuert und rendert nach einer "GetMap"-Anfrage die Karte, welche anschließend als Antwort vom Apache HTTP-Server in Form eines Kartenbildes ausgeliefert wird.

## 7 Umsetzung

### 7.1 Installation der benötigten Komponenten und Import der Beispiel-Daten

In Vorbereitung auf das Rendern der *OSM-Daten* und Erstellen eines Mapfiles mit *basemaps* müssen zunächst die dafür notwendigen Bibliotheken und Programme installiert werden, deren genauen Befehle zum Herunterladen der [Installationsanleitung](#) entnommen werden können. Dabei ist zu beachten, dass bei einigen Schritten Administrationsrechte vorhanden sein müssen.

Zu Beginn wird *PostgreSQL* und *PostGIS* installiert, wozu *Debian* fertige Pakete in der Version 8.4 bereithält. (Anm.: Zum Zeitpunkt der Erstellung der Installationsanleitung im Juli 2012.) Wenn neuere Versionen vorhanden sind, dann sollten diese benutzt werden.

Alle nachfolgenden Installationen werden im neu angelegten Arbeitsverzeichnis "osm-demo/" durchgeführt.

*Imposm* läuft mit *Python 2.5, 2.6* und *2.7*, deshalb muss vor der Installation überprüft werden, ob eine dieser Versionen vorhanden ist. Des Weiteren werden zusätzliche Bibliotheken benötigt, welche auf der Seite <http://imposm.org/docs/imposm/latest/install.html> dokumentiert sind. Wenn bereits „pythonbasierte“ Programme auf dem Arbeitsrechner laufen, ist es empfehlenswert *Imposm* in einer virtuellen *Python*-Umgebung zu installieren.

Anschließend wird die DB, beispielsweise "osm" angelegt. Dazu wird das Skript "create-db.sh" verwendet, welches alle wichtigen Schritte zum Anlegen einer DB enthält und nach eigenen Bedürfnissen angepasst werden kann.

Zum Auszuführen des Skripts muss zum Benutzer „postgres“ gewechselt werden und nach Beendigung ist ein Neustart von *PostgreSQL* notwendig.

Als Ergebnis entsteht in der DB "osm" eine Liste mit folgenden Relationen. (s. Tab. 5)

Schema	Name	Typ	Eigentuermer
public	geometry_columns	Tabelle	osm
public	spatial_ref_sys	Tabelle	osm

Tabelle 5: Liste der Relationen in der DB "osm"

Nun kann das Projekt *basemaps* installiert werden.

Da es sich bei diesem um ein Git-Projekt handelt, muss sichergestellt sein, dass sich auf dem Arbeitsrechner das dezentrale Verwaltungssystem *Git* in der aktuellen Version befindet und somit *basemaps* aus dem Git-Repository heruntergeladen werden kann.

Danach wird in das Verzeichnis "basemaps/" gewechselt, welches das Pythonskript "imposm-mapping.py", mit den Definitionen zum Aufbau der Tabellen beinhaltet und somit festlegt in welche die *OSM-Daten* importiert werden sollen.

Der Importprozess setzt sich aus mehreren Schritten zusammen. Zu Beginn werden die *OSM-Daten* gelesen, wozu ein direkter Zugriff auf alle Knoten und Wege erforderlich ist, um die Wege und Relationen der Geometrien herzustellen. Dieses wird jedoch nicht vom XML-

oder PBF-Datenformat unterstützt, weshalb *Imposm* einen Zwischenspeicher anlegt, der einen direkten Zugriff auf alle Elemente ermöglicht.

Im zweiten Schritt werden die OSM-Daten in die DB geschrieben. Dazu werden die Merkmale aus dem zwischengespeicherten Daten vom ersten Schritt gelesen, alle Geometrien gebaut und schließlich in die Tabellen importiert.

Der folgende letzte Schritt ist optional und dient zur Optimierung der Tabellen, welche basierend auf dem räumlichen Index zusammengefasst werden.

Das Lesen, Schreiben und Optimieren der Daten kann einzeln oder in einem Schritt erfolgen.

Die Tabelle 6 zeigt die Liste aller neu hinzugekommenen Relationen in der DB "osm".

Schema	Name	Typ	Eigent\xFCmer
public	geometry_columns	Tabelle	osm
public	osm_new_admin	Tabelle	osm
public	osm_new_aeroways	Tabelle	osm
public	osm_new_amenities	Tabelle	osm
public	osm_new_buildings	Tabelle	osm
public	osm_new_landusages	Tabelle	osm
public	osm_new_landusages_gen0	Tabelle	osm
public	osm_new_landusages_gen00	Tabelle	osm
public	osm_new_landusages_gen1	Tabelle	osm
public	osm_new_places	Tabelle	osm
public	osm_new_railways	Tabelle	osm
public	osm_new_railways_gen0	Tabelle	osm
public	osm_new_railways_gen1	Tabelle	osm
public	osm_new_roads	Tabelle	osm
public	osm_new_roads_gen0	Tabelle	osm
public	osm_new_roads_gen1	Tabelle	osm
public	osm_new_transport_areas	Tabelle	osm
public	osm_new_transport_points	Tabelle	osm
public	osm_new_waterareas	Tabelle	osm
public	osm_new_waterareas_gen0	Tabelle	osm
public	osm_new_waterareas_gen1	Tabelle	osm
public	osm_new_waterways	Tabelle	osm
public	osm_new_waterways_gen0	Tabelle	osm
public	osm_new_waterways_gen1	Tabelle	osm
public	spatial_ref_sys	Tabelle	osm

(25 Zeilen)

Tabelle 6: Liste mit vollständigen Relationen in der DB "osm"

## 7.2 Konfiguration der basemap-Dateien

Nach Abschluss des Datenimports wird ein temporärer Ordner angelegt und dem Nutzer alle Rechte für diesen zugewiesen. „MapServer-CGI“ (s. Kap. 6.1.3) benötigt diesen für das spätere Testen mit dem Template von *OpenLayers*.

Als nächstes erfolgt ein Eintrag mit der Pfadangabe zum vorher angelegten, temporären Ordner in der Datei "osmbase.map" (s. Listing 4).



```

$ vi osmbase.map
...
WEB
...
IMAGEPATH "/tmp/ms_tmp/"
IMAGEURL  "/ms_tmp/"
END
...

```

Listing 4: Ausschnitt aus dem Mapfile "osmbase.map"

Anschließend wird das Makefile (s. Listing 5) im Ordner "basemaps/data/" geändert. Die Änderungen beziehen sich auf das *10m-admin-0-boundary-lines-land.zip*<sup>5</sup>, welche in diesem Format nicht mehr aktuell sind und daher entweder namentlich angepasst, oder wie in diesem Fall gelöscht werden, da sie nicht zur Erstellung von Mapfiles mit den Niedersachsen-Daten notwendig sind (s. Kap. 4.2.1).

```

$ cd ~/osm-demo/basemaps/data
$ vi Makefile
...
all: TM_WORLD_BORDERS-0.3 processed_p shoreline_300
...
# delete the following entries
10m_admin_0_boundary_lines_land.shp: 10m-admin-0-boundary-lines-land.zip
unzip 10m-admin-0-boundary-lines-land.zip
touch 10m_admin_0_boundary_lines_land.shp
...
10m-admin-0-boundary-lines-land.zip:
wget
http://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/cultural/10m-admin-0-boundary-lines-land.zip
...

```

Listing 5: Ausschnitt aus dem "Makefile" des Ordners "basemaps/data/"

Nach der Speicherung wird dieses „Makefile“ und anschließend das „Makefile“ im Verzeichnis „basemaps/“ ausgeführt, welches auf Grundlage des Skripts "generate\_style.py" (s. Kap. 7.2.1) das Mapfile generiert.

Im selben Verzeichnis befindet sich nun ein Mapfile mit dem Namen "osm-default.map", welches mit dem Tool *shp2img* getestet werden kann. (s. Listing 6) Dazu wird der Pfad zum Mapfile und der Name der Ausgabedatei als Parameter angegeben und je nach angewandter GD-Bibliothek<sup>6</sup> entsteht eine Karte im Format PNG oder GIF. Bei Problemen zur Erzeugung der Karte, verweist eine Meldung in der Kommandozeile auf die fehlerhafte Zeile im Mapfile.

5 Kartensammlung von Natural Earth, welche in verschiedenen Maßstäben öffentlich zur Verfügung steht.

6 Bibliothek zur dynamischen Erzeugung und Manipulation von Grafiken.

```
$ shp2img -m osm-default.map -o output.png
```

*Listing 6: Kommandozeilenaufruf mit shp2img*

Nach korrektem Ausführen entsteht das Bild „output.png“ (Anhang A).

Um das Mapfile mit *OpenLayers* Template zu testen, wird folgende URL (s. Listing 7) im Browser eingegeben.

```
http://yourserver.tld/cgi-bin/mapserv?map=/path/to/osm-demo/  
basemaps/osm-  
default.map&mode=browse&template=openlayers&layers=all
```

*Listing 7: Beispiel-URL zum Testen mit OpenLayers Template*

Das zugehörige Kartenbild befindet sich im Anhang B.

Die generierten Mapfiles können nun, je nach Anwendungsfall manuell weiter bearbeitet oder durch Anpassungen in den *basemaps*-Konfigurationsdateien verändert werden. Dabei ist allerdings zu beachten, dass durch ein erneutes Ausführen des Skriptes "generate\_style.py", die zuvor erstellten Mapfiles überschrieben werden.

### **7.2.1 Aufbau und Funktion des Skriptes "generate\_style.py"**

Das *Python*-Skript "generate\_style.py" beinhaltet alle konfigurierbaren Parameter zum Erstellen von Kartenstilen für den *MapServer*. Der Aufbau besteht aus verschiedenen Arrays (s. Listing 8), welche im Folgenden erläutert werden.

```
#!/usr/bin/env python  
  
layer_suffixes = {...}  
  
maxscales = {...}  
minscales = {...}  
  
vars= {...}  
  
styles = {...}  
  
style_aliases = {...}
```

*Listing 8: Allgemeiner Aufbau des Skriptes "generate\_style.py"*

Insgesamt besteht die zu generierende Karte aus 18 Layer, welche am Anfang des Skripts im Array [layer\_suffices] einem numerischen Wert zugewiesen werden. Dies ist wichtig, sofern Parameter nur für bestimmte Layer gelten sollen. Des Weiteren wird jedem Layer durch [minscale] und [maxscale] ein Maßstabsbereich zugeordnet. Alle möglichen Parameter zur Beschreibung von Straßen, Wasserflächen, Waldflächen, Landnutzungsflächen und Grenzen befinden sich im Array [vars]. Außerdem sind bereits vordefinierte Kartenstile im Array [styles] enthalten, welche sich auf die geläufigen Karten (*Google*, *Bing* und *Michelin*) im Internet beziehen. Die Namen der Kartenstile werden in das Array [style\_aliases] geschrieben und somit aktiviert.

Zum Generieren von Kartenstilen kann das Skript mit einem beliebigen Editor bearbeitet werden. Die genauen Namen der Parameter und die zugehörigen Standardwerte (z. B. Farbe, Größe usw.) befinden sich im erstellten Git-Repository, auf der Seite [Documentation about tweaking map styles with basemaps](#).

Die Entstehung eines „deutschlandspezifischen“ Kartenstils, wird im Kap. 7.3 beschrieben.

### **7.2.2 Konfiguration der inkludierten Dateien**

Die Hauptinformationen, um ein Mapfile zu generieren, sind im "Makefile" enthalten. Dies betrifft die Datenverbindung zur Datenbank, den Spaltennamen und den Präfix unter dem die Daten abgelegt sind. Des Weiteren wird die Projektion und die zugehörige Maßeinheit zur Darstellung der Daten festgelegt und Projektionen angegeben, welche der WMS-Server unterstützt. Zusätzlich kann die Ausdehnung und der Stil der Karte bestimmt werden. Verweise, zum genutzten Template "osmbase.map" und weitere inkludierte Dateien, werden hier ebenfalls aufgeführt.

Bei den inkludierten Dateien handelt es sich um Kartentemplates mit vordefinierten Stilangaben, zur Erstellung einzelner Layer, bezüglich der Darstellung von Grenzen, Gebäuden, Straßen, Landflächen, Nutzungsflächen und Ortsangaben, sowie zur Festlegung von Koordinatensystemen und Schriftarten, welche verwendet werden können.

Der Ordner "data/" enthält Daten über Ländergrenzen und Küstenlinien zur Darstellung der gesamten Welt.

Weitere Dateien sind in der [Benutzerhilfe](#) aufgeführt und beschrieben.

## **7.3 Vorgehensweise zur Erstellung des „deutschlandspezifischen“ Kartenstils**

Um den landestypischen Stil umzusetzen, muss als erstes ein Name (z. B. 'german') festgelegt werden, welcher im Array [style] des Skriptes "generate\_style.py" geschrieben wird. Im nächsten Schritt werden alle gewünschten Parameter aus dem Array [vars] kopiert, im eigenen Kartenbereich eingefügt und mit den eigenen Werten verändert. Die Werte für Farbe und Größe der Objekte stammen dabei aus der Datei "osm-de.xml" ([Braun11], s. Kap. 3.3).

Abschließend muss nun der eigene Kartenstil im Array [style\_aliases] bekannt gegeben

werden.

Nach dem Speichern kann das Skript geschlossen und im "Makefile" der Stil für die Kartenausgabe angepasst werden. Um direkt einen Ausschnitt der Karte mit den ausgewählten OSM-Daten zu sehen, wird zusätzlich die Kartenausdehnung entsprechend definiert. (s. Listing 9)

```
$ vi Makefile
...
OSM_EXTENT=702345 6661260 1318295 7158256
...
STYLE=german
...
```

*Listing 9: Ausschnitt aus dem "Makefile" im Ordner "basemaps/"*

Nach Ausführen des Skripts befindet sich im Ordner "basemaps/" die Datei "osm-german.map".

Das Ergebnis wird im folgenden Kapitel 8 betrachtet.

## 8 Ergebnis



Die Datei „osm-german.map“ kann mithilfe der Web-Applikation *OpenLayers* (s. Abb. 22) dargestellt werden.

Des Weiteren besteht die Möglichkeit *WMS*-Layer mit einem Desktop-Programm (z. B. *QGIS*) zu beziehen und anzuzeigen, sowie mittels "GetMap"-Request anzufordern.

Eine Einschätzung, bezüglich des Aufwands bei der Installation aller benötigten Bestandteile und bei der Umsetzung zum „deutschlandspezifischen“ Kartenstils, erfolgt in den Kapiteln 8.1 und 8.2. Anschließend wird in Kapitel 8.3 dieser Weg zur Darstellung von *OSM*-Daten dem, mittels *Mapnik* gegenübergestellt.

### **8.1 Aufwand bei der Installation aller benötigten Bestandteile**

Bestandteil	Bezugsquelle
OSM-Daten	<a href="http://wiki.openstreetmap.org/wiki/Planet.osm">http://wiki.openstreetmap.org/wiki/Planet.osm</a>
PostgreSQL/PostGIS	Debian Paket
Imposm	<a href="http://imposm.org/docs/imposm/latest/">http://imposm.org/docs/imposm/latest/</a>
basemaps	<a href="https://github.com/mapserver/basemaps">https://github.com/mapserver/basemaps</a>
MapServer	<a href="http://mapserver.org/download.html">http://mapserver.org/download.html</a>
Apache	Debian Paket

Tabelle 7: Bezugsquellen der verwendeten Bestandteile zur Darstellung von OSM-Daten mittels MapServer

Anmerkungen zum Bezug von OSM-Daten und zur Installation der PostgreSQL/PostGis-DB werden bereits im Kap. 5.2 aufgeführt.

Die Installation von *Imposm* in eine virtuelle *Python*-Umgebung verläuft problemlos und durch das mitgelieferte Skript „create-db.sh“ wird die benötigte Datenbank automatisch erstellt. Das Lesen, Schreiben und Optimieren der Geodaten von Niedersachsen (123M) dauerte allerdings über eine Stunde und es ist nicht möglich weitere Daten im Nachhinein in die bestehenden Tabellen einzufügen, da vorhandenen Daten dadurch überschrieben werden, wenn vor dem Import keine namentliche Änderung erfolgt. Allerdings liegen die Datensätze dann mit unterschiedlichem Präfix vor und können ebenfalls nicht vereint werden. Die Standardstruktur der Tabellen wird dem Skript "default-mapping.py" erstellt. *Basemaps* bezieht sich jedoch auf eine erweiterte Tabellenstruktur und liefert das dafür benötigte Skript "imposm-mapping.py" bereits mit und muss demzufolge vor dem Datenimport installiert werden, wobei ebenfalls keine Schwierigkeiten auftreten, wenn alle notwendigen Voraussetzungen (*MapServer*  $\geq$  6.0.1 und *Imposm*  $\geq$  2.3.0) gegeben sind. Grundlage für *basemaps* ist die *MapServer* Version 6.2.0, welche sich in Bezug auf [CLASS]-Block-Definitionen von einer Version 6.0.\* unterscheiden und daher Probleme bei einzelnen Mapfiles (bspw. "places.map") auftauchen und daher entsprechend angepasst werden müssen. Des Weiteren ist der Verweis zur Bezugsquelle der "world\_boundaries"-Daten im Skript "Makefile" im Verzeichnis "data/" nicht mehr aktuell und sollte manuell geändert, oder wie im Fall dieser Arbeit gelöscht werden. Außerdem ist ein temporärer Ordner erforderlich, welcher erstellt und der in der Template-Datei angegeben werden muss. Alles zusammen kann unter Umständen einige Zeit in Anspruch nehmen, was jedoch aufgrund der schnellen Erstellung eines Mapfiles wieder neutralisiert wird.

Die Installation des *MapServers* ist für Erstanwender mit viel Aufwand und Schwierigkeiten verbunden, da viele Abhängigkeiten notwendig sind, welche z. T. extern bezogen und an der vorgegebenen Stelle entpackt, kompiliert und ausgeführt werden müssen. Die Konfiguration des *MapServers* erfolgt dann zusammen mit den Pfadangaben, zu den Abhängigkeiten und optionalen Bibliotheken. Wenn dabei keine Probleme entstehen, kann der Quellcode kompiliert und die entstehende Binärdatei in das *Apache* "cgi-bin"-Verzeichnis kopiert werden. Die Installation von *Apache* ist wiederum einfach, da Debian dazu das fertige Paket *apache2* bereithält.

## **8.2 Aufwand bei der Umsetzung zum „deutschlandspezifischen“ Kartenstils**

Die Umsetzung des „deutschlandspezifischen“ Kartenstils erfolgt in mehreren Schritten, welche im Folgenden beschrieben und ausgewertet werden.

Zu Beginn wird die Datei "osm\_de.xml" mit einer landesüblichen Gestaltung [Braun11] herangezogen, um daraus die Farbwerte bestimmter Sachverhalte für die eigene Umsetzung auszuwählen. Viele Objekte haben in beiden Dateien, welche die Stil-Definitionen enthalten, die gleiche Bezeichnung, was eine Zuordnung mit den jeweiligen Farben erleichtert, aber dennoch sehr zeitaufwendig ist. Eine Festlegung mit selbst gewählten Farbwerten wäre daher effektiver.

Die Übertragung der ausgewählten Farbwerte in das Skript "generate\_style.py" gestaltet sich, aufgrund des übersichtlichen Aufbaus und des einfachen Konzepts, unproblematisch. Die Angaben von betreffenden Sach- bzw. Straßenobjekten werden dazu aus dem Array [vars] kopiert und mit den neuen Farbwerten im eigenen „Style“-Bereich gespeichert. Die bereits vordefinierten Parameter bezüglich der Außenlinien von Straßenobjekten werden an einigen Stellen angepasst und zusammen mit den sonstigen Standardwerten, für den eigenen Kartenstil übernommen.

Nach Angabe des neuen Kartenstils im "Makefile" und dem Ausführen des bearbeiteten Skripts, kann das Ergebnis mit einem Webbrowser grafisch betrachtet werden.

Offensichtliche Fehler und ungeeignete Darstellungen können anschließend im zugehörigem Mapfile behoben, bzw. neu definiert werden.

Abschließend wird mittels "GetMap"- und "GetFeature"-Request das eigene Mapfile auf OGC-Standard-Konformität überprüft, welche die geforderten Ergebnisse liefern und somit die Unterstützung von OGC-Standards beweisen.

Ein Problem, welches auftreten kann, betrifft die Datei "places.map" im Verzeichnis "basemaps/". Hier werden die Definitionen für Objekte verwendet, wie sie in der Mapfile-Referenz zum *MapServer* ab Version 6.2 angegeben sind. Ab dieser Version muss nicht mehr innerhalb eines [CLASS]-Blocks ein [STYLE]-Block definiert werden. Bei älteren Versionen kommt es daher zu einer Fehlermeldung (z. B. msDrawMap(): Image handling error. Failed to draw layer named 'places8'. msAddLabel(): General error message. msAddLabel error: missing style definition for layer 'places8' ), sobald die gerenderte Karte über einen Browser abgerufen wird. Um eine Fehlermeldung zu vermeiden, muss die Datei "places.map" vorab angepasst werden. Im [Issue Tracker](#) des Git-Projekts befindet sich eine einfache Lösung zu diesem Problem.

### **8.3 Gegenüberstellung der Darstellungen von OSM-Daten mittels MapServer und Mapnik**

Die Gegenüberstellung der Karten beider Darstellungsmethoden von OSM-Daten erfolgt, hinsichtlich der allgemeinen Anforderungen einer digitalen Karte (Qualität, Nutzbarkeit, Ästhetik), wie in Kap. 3.2 aufgeführt. Dazu werden die Karten nebeneinander, jeweils in einer *OpenLayers*-Anwendung, im selben Punkt zentriert, grafisch wiedergegeben und anschließend jede Zoomstufe mit einander verglichen.

Die Qualität bezüglich Korrektheit, Vollständigkeit und Klarheit ist in beiden Karten gleich, da die selben Daten (s. Kap. 4.2.1) verwendet werden. Allerdings gibt es, aufgrund verschiedener Datenquellen, Unterschiede in der Aktualität, denn während die Beispiel-*OpenLayers*-Anwendung für die *Mapnik*-Darstellung an eine *Slippy Map*-Schnittstelle angebunden ist und somit die Karte vom *OSM-Tile-Server* bezieht, welche aus den wöchentlich aktualisierten OSM-Daten des „Planet files“ besteht, greift der *MapServer* auf den einmal importierten Datensatz in der DB zu.

Dafür bietet die *OpenLayers*-Anwendung der *MapServer*-Darstellung, neben den üblichen Funktionen zur Interaktion mit der Karte, die Möglichkeit einzelne Layer nach Bedarf ein- und auszuschalten und dadurch das Rendern der Karte zeitlich zu beeinflussen. Denn dieser Vorgang kann sowohl beim Zooming, als auch beim Verschieben des Ausschnitts, in niedrigen Zoomstufen bis zu 30s dauern. Im Vergleich dazu, erfolgt die *Mapnik*-Darstellung, aufgrund der vorgerenderten Kartenbilder (s. Kap. 4.2.2) ohne längere Wartezeiten.

Für den visuellen Vergleich werden auf den folgenden Seiten Abbildungen desselben Bereichs, aber in vier unterschiedlichen Zoomstufen (8, 13, 15, 17) gezeigt und nach Anforderungen, welche das Erfassen bestimmter Sachverhalte und die Lesbarkeit von Beschriftungen betreffen, sowie ästhetischen Merkmalen eingehend betrachtet.



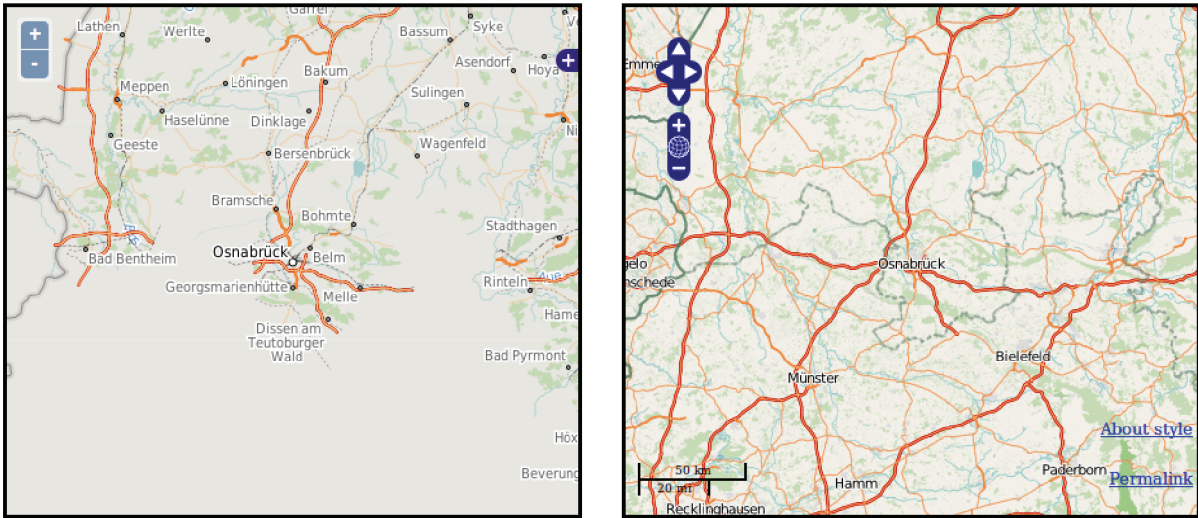


Abbildung 23: Zoomstufe 8

Zoomstufe 8 (Abb. 23)	Rendern mittels MapServer	Rendern mittels Mapnik
<b>Erfassbarkeit</b>	Straßen, Waldflächen, Gewässer (Seen, Flüsse) deutlich zu erkennen	Darstellung von Bundeslandgrenzen
	Abbildung mehrerer Orte und Platzierung mittels Symbol	
<b>Lesbarkeit</b>	Schriftgröße passend gewählt	Schriftgröße könnte größer sein
<b>Beschränkung</b>	Darstellung von Autobahnen und Schnellstraßen Landstraßen werden nur leicht angedeutet	Darstellung enthält nur wenige, größere Städte
<b>Gleichgewicht</b>	Vorhanden	Schwerpunkt liegt auf Straßennetzwerk
<b>Harmonie</b>	Insgesamt harmonisch	Farblich harmonisch Inhaltlich unausgeglichen

Tabelle 8: Vergleich in Zoomstufe 8

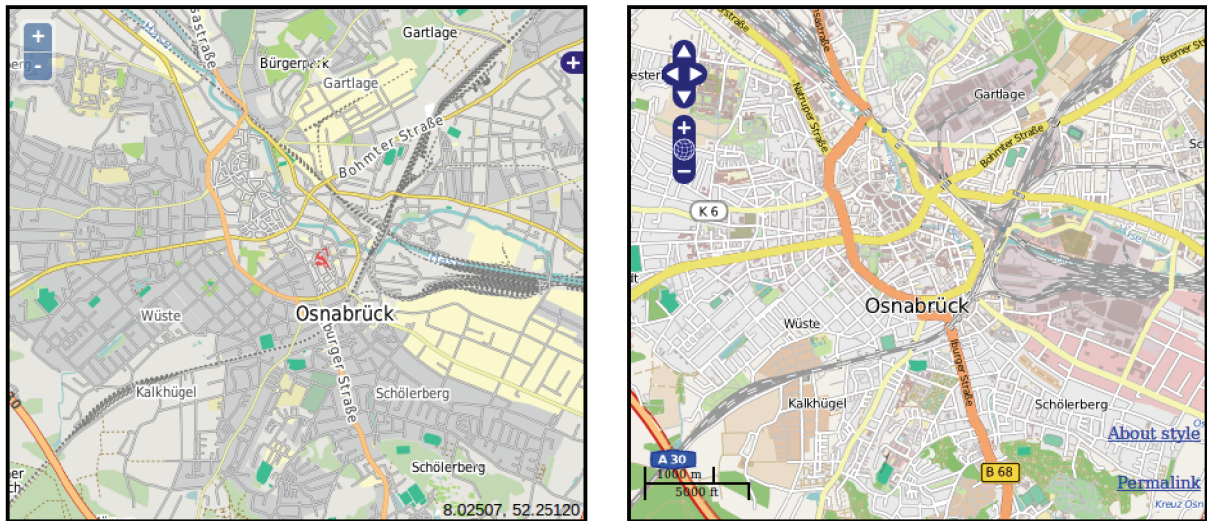


Abbildung 24: Zoomstufe 13

Zoomstufe 13 (Abb. 24)	Rendern mittels MapServer	Rendern mittels Mapnik
<b>Erfassbarkeit</b>	Industrie-, Wohn- und Waldbereiche, sowie Ärztliche Einrichtungen (rot) gut zu erkennen	Alle Sachverhalte und Objekte gut zu erkennen
	Autobahn kaum von Bundesstraße zu unterscheiden	
<b>Lesbarkeit</b>	Autobahn, Bundes-, Land- und Kreisstraße sind zu schmal	Schriftgröße für Ortsangaben passend gewählt
	Fluss kaum zu erkennen	
<b>Beschränkung</b>	Schriftgröße passend gewählt	Schriftgröße für Straßennamen zu klein
	Einteilung der Landnutzungsflächen auf Industrie-, Wohn- und Waldbereiche beschränkt	Vorteil: Darstellung der Schienen und Symbole der Straßen
<b>Gleichgewicht</b>	Schwerpunkt liegt auf Landnutzungsflächen	Zusätzliche Unterteilung innerhalb der Industrie- und Wohnbereiche
<b>Harmonie</b>	Vorhanden	
	Insgesamt harmonisch	

Tabelle 9: Vergleich in Zoomstufe13

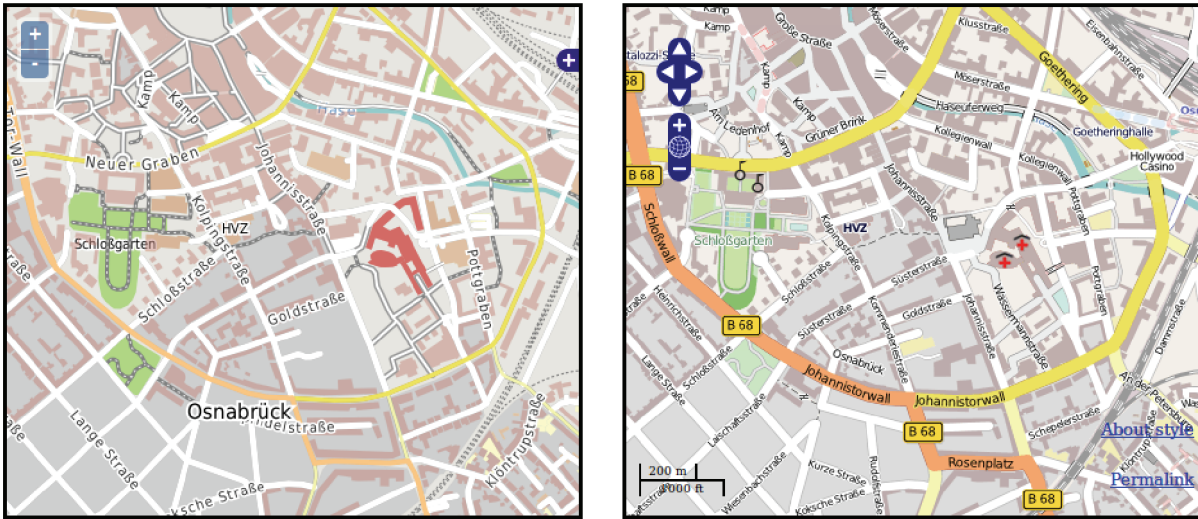


Abbildung 25: Zoomstufe 15

Zoomstufe 15 (Abb. 25)	Rendern mittels MapServer	Rendern mittels Mapnik
<b>Erfassbarkeit</b>	Verschiedene Sachverhalte sowie Standorte von Gebäuden gut zu erkennen	
	Ärztliche Einrichtungen gut zu erkennen	Ausprägung der Bundes- und Landstraßen zu stark
	Bundes- und Landstraßen könnten etwas deutlicher hervorgehoben werden	bessere Darstellung der Schienen
	Kreisstraßen viel zu schmal	Vorteil: Symbol für ärztliche Einrichtungen
<b>Lesbarkeit</b>	Schriftgröße passend gewählt	Beschriftung für Haupt- und Nebenstraßen zu klein
<b>Beschränkung</b>	Nebenstraßen werden nicht beschriftet	Alle Straßen werden beschriftet
<b>Gleichgewicht</b>	Vorhanden	Schwerpunkt liegt auf Beschriftung
<b>Harmonie</b>	Insgesamt harmonisch	Farblich harmonisch Inhaltlich unausgeglichen

Tabelle 10: Vergleich in Zoomstufe 15

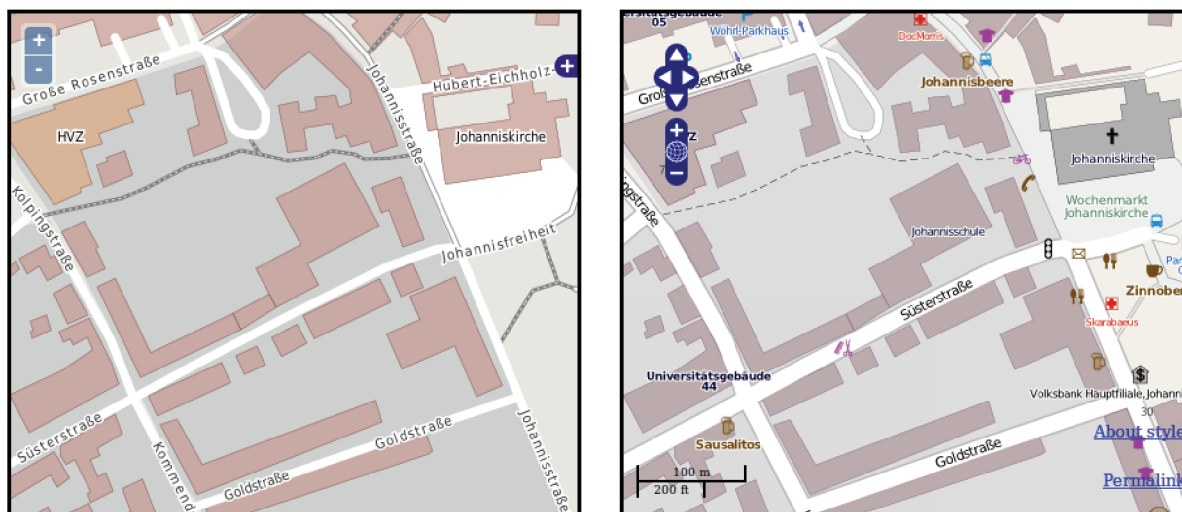


Abbildung 26: Zoomstufe 17

Zoomstufe 17 (Abb. 26)	Rendern mittels MapServer	Rendern mittels Mapnik
Erfassbarkeit	Alle Sachverhalte und Objekte gut zu erkennen	
Lesbarkeit	Schriftgröße passend gewählt	
		Vorteil: viele Symbole
Beschränkung	Keine weiteren Details	Beschriftung vieler Gebäude mit unterschiedlichen Funktionen
Gleichgewicht	Vorhanden	
Harmonie	Insgesamt harmonisch	

Tabelle 11: Vergleich in Zoomstufe 17

Zusammenfassend lässt sich aus den Betrachtungen ableiten, dass beide Kartendarstellungen gut für digitale Anwendungen geeignet sind und nur in einigen Punkten Vor- bzw. Nachteile aufweisen. So ist z. B. die *MapServer*-Darstellung in Zoomstufe 8 eindeutig besser, während die *Mapnik*-Darstellung in Zoomstufe 13 bis 17 viel mehr Details in Form von Symbolen und einer weiteren Einteilung von Sachverhalten innerhalb der Industrie- und Wohnbereiche enthält und somit mehr Informationen zeigt. Zusammen mit der Beschriftung von Straßen und Gebäuden führt dies jedoch zu einem Überladen der Karte, was sich nachteilig auf das Gesamtbild auswirkt und daher die *MapServer*-Darstellung insgesamt harmonischer und ausgeglichener erscheint.

Abschließend werden noch einmal alle benötigten Daten und Programme der jeweiligen Systeme, sowie deren Eignung in der folgenden Tabelle gegenübergestellt.

	<b>OSM-Daten mittels Mapnik</b>	<b>OSM-Daten mittels MapServer</b>
<b>Daten</b>	OSM-Daten ("niedersachsen.osm.pbf")	
<b>Zusätzliche Daten</b>	"world_boundaries/"	-
<b>Datenbank</b>	PostgreSQL/PostGIS	
<b>Datenimport</b>	<i>osm2pgsql</i>	<i>Imposm ≥ 2.3.0</i>
<b>Renderer</b>	<i>Mapnik 2</i>	<i>MapServer ≥ 6.0.1</i>
<b>Dateiformat für die Stildefinitionen</b>	"*.xml"	"*.map"
<b>Systemeignung für Umsetzung des Stils</b>	geeignet	
<b>Standard-konform</b>	-	OGC-Standard-konform
<b>Unterstützte Dienste</b>	-	WMS, WFS
<b>Unterstütztes GIS-System</b>	<i>Map-Server GIS</i>	<i>Verteiltes GIS</i>
<b>Ausgabeformat</b>	Raster-, Vektorgrafik	Raster-, Vektorgrafik über WMS GML über WFS

Tabelle 12: Zusammenfassung der benötigten Bestandteile und Systemeignung

## 9 Zusammenfassung und Ausblick

Das Ziel der vorliegende Arbeit war es, zu untersuchen, inwieweit sich der *MapServer* eignet, zum einen *OSM*-Daten Standard-konform über *WMS* auszuliefern und andererseits *Mapnik* beim Rendering von landestypischen Karten zu ersetzen.

Zu diesem Zweck wurden, ausgehend von der Entstehung und Bedeutung digitaler Karten, sowie deren allgemeinen Anforderungen, insbesondere in Hinsicht einer landesüblichen Farbgestaltung, WebGIS-Systeme betrachtet, welche generell eine Auslieferung von Geodaten ermöglichen.

Eine Möglichkeit zur Erstellung von Karten aus *OSM*-Daten, des *OSM*-Projekts eröffnet die Verwendung von *Mapnik*.

In dieser Arbeit erfolgte daher eine Erläuterung der Funktionsweise und eine Analyse des Ergebnisses.

Dabei ergab sich, dass *Mapnik* ausschließlich Kacheln (engl. Tiles) in Form von Pixelgrafik-Karten rendert, welche nur mittels einer speziellen URL ausgeliefert werden und somit keinem spezifischen Standard, wie beispielsweise dem *OGC-WMS*-Standard entsprechen.

Der *MapServer* unterstützt jedoch *OGC*-Standards und wurde daher bei der Untersuchung eines alternativen Weges zur Darstellung von *OSM*-Daten herangezogen.

Was die Auslieferung der *OSM*-Daten betrifft, konnte anhand einer erfolgreichen Umsetzung des neuen Entwurfs mit dem *MapServer*, gezeigt werden, dass diese nun Standard-konform über *WMS* erfolgt.

Des Weiteren konnte am Beispiel der Erstellung einer „deutschlandspezifischen“ Karte gezeigt werden, dass sich der *MapServer* auch zum Rendern von *OSM*-Daten eignet.

Zwar wurde während der Arbeit festgestellt, dass die manuelle Erstellung von Mapfiles sehr aufwendig ist, aber mithilfe des Projekts *basemaps*, welches bei einer semi-automatischen Erstellung von Mapfiles verwendet wurde, verringert werden konnte.

Zum Vergleich der Darstellungsmethoden von *OSM*-Daten mittels *Mapnik* und *MapServer*, erfolgte eine Gegenüberstellung, wodurch Vor- und Nachteile, in Bezug auf verschiedene Aspekte betrachtet wurden.

Dadurch lässt sich abschließend feststellen, dass sich sowohl *Mapnik* als auch *MapServer* zum Rendern einer landestypischen Karte aus *OSM*-Daten eignen, aber eine Standard-konforme Auslieferung nur mittels *MapServer* möglich ist, weshalb dieser auch in verteilten GIS-Systemen eingesetzt werden kann.

## Glossar

<b>GIF</b>	Grafikformat mit guter verlustfreier Kompression für Bilder mit geringer Farbtiefe
<b>Git</b>	Freie Software zur verteilten Versionsverwaltung von Dateien
<b>GML</b>	Erlaubt die Übermittlung von Objekten mit Attributen, Relationen und Geometrien im Bereich der Geodaten unter Einbeziehung von nicht-konventionellen Daten, wie Sensordaten.
<b>HTML</b>	textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks in Dokumenten
<b>HTTP</b>	Protokoll zum Übertragen von Daten.
<b>INSPIRE</b>	Richtlinie zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft
<b>ISO</b>	Internationale Vereinigung zur Erarbeitung von internationalen Normen
<b>Mercator Projektion</b>	Form der Zylinderprojektion, bei der die Projektion in Richtung der Zylinderachse geeignet verzerrt ist, um eine winkeltreue Abbildung der Erdoberfläche zu erreichen.
<b>Non-Profit-Organisation</b>	gemeinnützige Organisation
<b>OpenLayers</b>	JavaScript-Bibliothek, die es ermöglicht Geodaten im Webbrowser anzuzeigen
<b>OSGeo</b>	Gemeinnützige Organisation mit Hauptsitz in Delaware. Fördert die Entwicklung und Nutzung von Freien und quelloffenen Geoinformationssystemen (GIS).
<b>PBF</b>	kompaktes Datenformat der Rohdaten von OpenStreetMap
<b>Python</b>	Objektorientierte Programmiersprache, plattformunabhängig
<b>Rendering</b>	Zeichnen
<b>Shapefile</b>	Dateiformat für Geodaten, welches von Esri ursprünglich für ArcView entwickelt wurde.
<b>Simple Feature Access-Spezifikation</b>	Definiert die Speicherung und den Zugriff auf Geometriedaten und verschiedene räumliche Operatoren
<b>Slippy Map</b>	Standarddarstellung von Mapnik

<b>Thematische Eigenschaften</b>	Beschreibung von Sachattributen
<b>Topologie</b>	Die Topologie beschäftigt sich mit den nicht-räumlichen und strukturellen Beziehungen beliebiger Elemente in abstrakten Räumen.
<b>URL</b>	Modell zur Bezeichnung und Identifizierung von Dokumenten im World Wide Web
<b>WMS</b>	Schnittstelle zum Abrufen von Auszügen aus Karten über das Internet.
<b>XML</b>	Auszeichnungssprache zur Darstellung von Textdateien mit hierarchisch strukturierten Daten



# Verzeichnisse

## Quellenverzeichnis

- [bas12] <https://github.com/mapserver/basemaps/issues/9> (Zugriff: 10.09.12)
- [BOS] Hauptseite: <http://spassgesellen.blogspot.de/> (Zugriff: 25.10.12)
- [Braun11] Beate Braun: Bachelorarbeit „Deutschlandspezifische Festlegung von Stileigenschaften für OpenStreetMap“, HFT Stuttgart (2011)
- [Hake02] Hake, Günter/ Grünreich, Dietmar/ Meng, Liqia: Kartographie.Berlin; New York: de Gruyter (2002), 8. Auflage
- [Imp12] Hauptseite: <http://imposm.org/> (Zugriff: 28.10.12)
- [Kart12] Hauptseite: <http://www.mr-kartographie.de> (Zugriff: 16.10.12)
- [Map01] Hauptseite von MapServer: <http://mapserver.org/download.html> (Zugriff: 10.09.12)
- [GEO12] <http://download.geofabrik.de/osm/> (Zugriff: 20.10.12)
- [OGC12] <http://www.opengeospatial.org/standards/wms> (Zugriff: 18.09.12)
- [Omn12] Hauptseite: <http://omniscale.de/> (Zugriff: 10.09.12)
- [OSM01] Hauptseite von OpenStreetMap: <http://www.openstreetmap.org/> (Zugriff: 18.09.12)
- [OSM08] Ramm, Frederik/ Topf, Jochen: OpenStreetMap – Die freie Weltkarte nutzen und mitgestalten. Berlin: Lehmanns Media (2008), 1. Auflage
- [Post12] Hauptseite: <http://www.postgresql.org/> (Zugriff: 10.09.12)
- [VOS] Hauptseite von Vermessung Osnabrück:  
<http://www.vermessung-osnabrueck.de> (Zugriff: 25.10.12)
- [ZOS] Hauptseite vom Zoo Osnabrück: <http://www.zoo-osnabrueck.de/> (Zugriff: 25.10.12)

## Abbildungsverzeichnis

Abbildung 1: Beispiel – Zookarte [ZOS].....	4
Abbildung 2: Beispiel – Straßenkarte [OSM01].....	4
Abbildung 3: Beispiel – Liegenschaftskarte [VOS].....	4
Abbildung 4: Beispiel – Busfahrplan [BOS].....	4
Abbildung 5: Allgemeine Anforderungen an eine Onlinekarte [nach Braun11].....	6
Abbildung 6: Farbverlauf nach Ordnung der Straßen [Braun11].....	8
Abbildung 7: Transfer von Geodaten zwischen Client und Server mittels WMS und WFS....	11
Abbildung 8: Ergebnis einer "GetMap" - Anfrage.....	13
Abbildung 9: Allgemeiner Transfer von Geodaten über das Internet.....	14
Abbildung 10: Systemarchitektur zur Darstellung von OSM-Daten mittels Mapnik.....	15
Abbildung 11: Importprozess mithilfe von osm2pgsql.....	15
Abbildung 12: Prozess: Rendering mittels Mapnik und Auslieferung.....	18
Abbildung 13: Kacheinteilung [Braun11].....	19
Abbildung 14: Allgemeiner Aufbau der Datei osm.xml.....	20
Abbildung 15: OSM deutscher Stil.....	20
Abbildung 16: Architektur eines verteilten GIS-Systems.....	22
Abbildung 17: WebGIS-System zur Darstellung von OSM-Daten mittels MapServer.....	23
Abbildung 18: Importprozess mithilfe von Imposm.....	24
Abbildung 19: Generieren des Mapfiles mithilfe von basemaps.....	24
Abbildung 20: Prozess: Rendering mittels MapServer und Auslieferung durch Apache HTTP- Server.....	25
Abbildung 21: MapServer Architektur [Map01].....	26
Abbildung 22: Anzeige von "osm-german.map" mit OpenLayers Template.....	33
Abbildung 23: Zoomstufe 8.....	37
Abbildung 24: Zoomstufe 13.....	38
Abbildung 25: Zoomstufe 15.....	39
Abbildung 26: Zoomstufe 17.....	40

## Tabellenverzeichnis

Tabelle 1: Kulturelle Unterschiede in der Farbinterpretation [Braun11].....	7
Tabelle 2: Straßenfarben im bisherigen und "deutschen" Stil [Braun11].....	8
Tabelle 3: Waldflächen im bisherigen und "deutschen" Stil [Braun11].....	9
Tabelle 4: Bezugsquellen der verwendeten Bestandteile zur Darstellung von OSM-Daten mittels Mapnik.....	21
Tabelle 5: Liste der Relationen in der DB "osm" .....	27
Tabelle 6: Liste mit vollständigen Relationen in der DB "osm" .....	28
Tabelle 7: Bezugsquellen der verwendeten Bestandteile zur Darstellung von OSM-Daten mittels MapServer.....	34
Tabelle 8: Vergleich in Zoomstufe 8.....	37
Tabelle 9: Vergleich in Zoomstufe 13.....	38
Tabelle 10: Vergleich in Zoomstufe 15.....	39
Tabelle 11: Vergleich in Zoomstufe 17.....	40
Tabelle 12: Zusammenfassung der benötigten Bestandteile und Systemeignung.....	41

## Listings

Listing 1: Beispiel-URL einer "GetCapabilities" - Anfrage.....	12
Listing 2: Beispiel-URL einer "GetMap" - Anfrage.....	12
Listing 3: URL zur Anforderung von Kartenkacheln vom OSM-Tile-Server.....	19
Listing 4: Ausschnitt aus dem Mapfile "ombase.map".....	29
Listing 5: Ausschnitt aus dem "Makefile" des Ordners "basemaps/data/".....	29
Listing 6: Kommandozeilenaufruf mit shp2img.....	30
Listing 7: Beispiel-URL zum Testen mit OpenLayers Template.....	30
Listing 8: Allgemeiner Aufbau des Skripts "generate_style.py".....	30
Listing 9: Ausschnitt aus dem "Makefile" im Ordner "basemaps/".....	32

## Abkürzungsverzeichnis

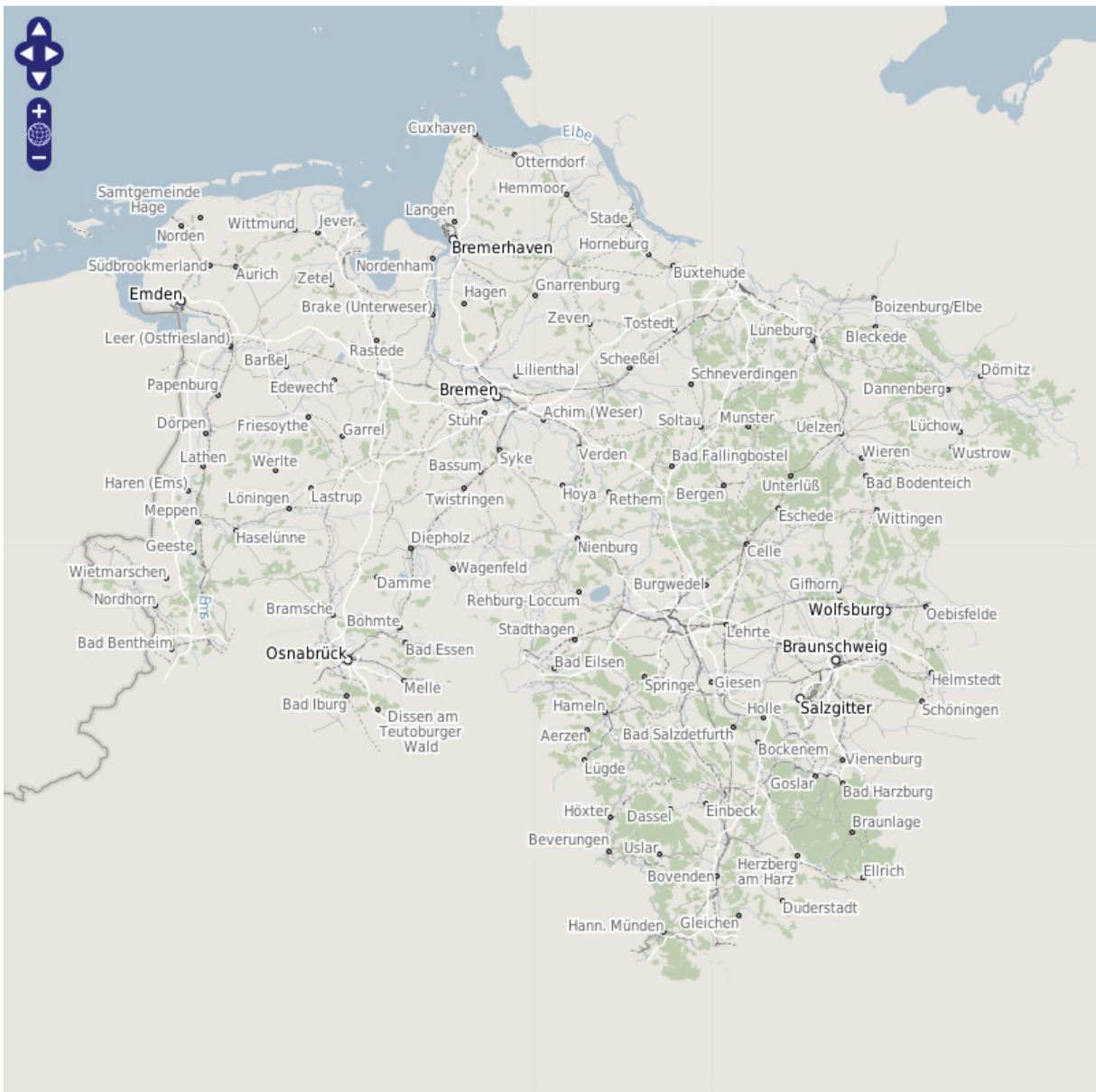
<b>CGI</b>	Common Gateway Interface
<b>DB</b>	Datenbank
<b>GIF</b>	Graphics Interchange Format
<b>GIS</b>	Geoinformationssystem
<b>GML</b>	Geography Markup Language
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>INSPIRE</b>	Infrastructure for Spatial Information
<b>ISO</b>	International Organization for Standardization
<b>NASA</b>	National Aeronautics and Space Administration
<b>ODbL</b>	Open Database Licence
<b>OGC</b>	Open Geospatial Consortium
<b>OsGeo</b>	Open Source Geospatial Foundation
<b>OSM</b>	OpenStreetMap
<b>OWS</b>	OGC Web Services
<b>PBF</b>	Protocolbuffer Binary Format
<b>PNG</b>	Portable Network Graphics
<b>URL</b>	Uniform Resource Locator
<b>WFS</b>	Web Feature Service
<b>WMS</b>	Web Map Service
<b>XML</b>	Extensible Markup Language

# Anhang

## Anhang A – output.png



## Anhang B – osm.map mit OpenLayers



## Anhang C – Inhalt der CD

Arbeit.pdf

RenderingOSMDataDebianSqueeze6.0.md.txt

Documentation\_about\_tweaking\_map\_styles\_with\_basemaps.md.txt

vergleich.html

vergleich.js

## **Eidesstattliche Erklärung**

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

Osnabrück, den 31.10.2012

*Unterschrift*