

Master Thesis

im Rahmen des Universitätslehrganges
„Geographical Information Science & Systems“ (UNIGIS MSc)
am Zentrum für GeoInformatik (Z GIS) der
Paris Lodron-Universität Salzburg

zum Thema

Flurstücke per WFS

Am Beispiel einer WebGIS-Anwendung
für private Waldbesitzer

vorgelegt von
Dipl. Ing. Wald & Forstwirtschaft **Michael Sinner**
U1476, UNIGIS Msc Jahrgang 2010

Zur Erlangung des Grades
„Master of Science Geographical Information Science & Systems“

Gutachter:
Ao. Univ. Prof. Dr. Josef Strobl

Wien, den 01.05.12

Eigenständigkeitserklärung

Ich versichere, diese Master Thesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt wurde. Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Wien, 02. Mai 2012



Michael Sinner

Zusammenfassung

Spätestens seit der Spezifikation von Geowebdiensten durch das Open Geospatial Consortium ist der Interoperabilität von webbasierten Anwendungen der Weg geebnet. Öffentliche Datenanbieter stehen vor der Entscheidung, welche Daten über welchen Dienst und welches Nutzungsmodell bereitzustellen. Während WMS-Dienste eine gute Kontrolle über die Basisdaten ermöglichen, bekommt der Nutzer per WFS-Dienst eine Kopie der Vektordaten übermittelt.

Am Beispiel einer prototypischen WebGIS-Anwendung für Privatwaldbesitzer wird gezeigt, welche Probleme im speziellen durch kostenpflichtige Datenangebote verursacht werden. Zwei Varianten werden besprochen, in denen einmal eine Kopie der Geodaten auf dem Client angefertigt wird, in einem weiteren Fall die Geodaten nur als Kartierungshilfe dienen.

Durch die Untersuchung eines Anwendungsfalles werden Hürden und Lösungen aufgezeigt und unter Sicherheitsaspekten beleuchtet. Empfehlungen im Umgang mit sensiblen Geodaten per WFS-Dienst werden entwickelt und besprochen. Eine Aufstellung von Stärken und Schwächen von WFS-Diensten gegenüber Alternativen sollen Behörden und Datenlieferanten helfen, den geeigneten Dienst für ähnliche Datenprodukte auszuwählen.

Abstract

With the specification of web services through the Open Geo-spatial Consortium, interoperable Web-Applications have been increasingly installed predominately in the public sector. Data Publishers are to decide which data should be published by which service and through which usage model. While WMS-Services maintain the copyrights of underlying vectorial data with the provider, WFS-Services seem to give out copies of features and attributes away without further control.

An example regarding a prototype WebGIS-Application for private forest owners, shows the problems involved with supplying non-free geodata through a WFS-Service. Two alternatives are discussed, where features are copied locally or are merely a support for editing features.

By looking at a possible Use Case, obstacles and solutions are highlighted and

examined under security aspects. Recommendations regarding the handling of valuable spatial data are developed and reviewed. An Assembly of strengths and weaknesses of a WFS-Service as oppose to known alternatives shall give administrations and data providers the requisite know-how to choose the appropriate service for similar data products.

Inhalt

Abbildungen.....	7
Tabellen.....	8
Abkürzungen.....	9
1 Einleitung.....	10
1.1 Ziele.....	11
1.2 Gegenstand.....	13
1.3 Ähnliche WFS-Dienste.....	14
1.4 Ähnliche Projekte.....	15
1.5 Wesentliche Begriffe.....	16
1.6 Methoden.....	17
1.6.1 Anwendungsfall-Diagramme.....	18
1.6.2 Aktivitätsdiagramme.....	18
1.6.3 Systemarchitektur-Diagramme.....	19
1.6.4 Hypothesen.....	19
2 Der WFS-Dienst.....	20
2.1 LVG Bayern.....	20
2.2 Capabilities.....	20
2.3 DescribeFeatureType.....	26
3 Die Anwendung.....	27
3.1 Anforderungen.....	27
3.1.1 Use-Case.....	27
3.1.2 Trennung von Flurstücks- und Waldflächenverwaltung.....	28
3.1.2.1 Flurstücksverwaltung.....	28
3.1.2.2 Waldflächenverwaltung.....	30
3.1.3 Funktionalität.....	30
3.1.4 Genauigkeit.....	31
3.1.5 Datenquellen und Ebenen.....	32
3.2 Architektur.....	32
3.2.1 Allgemeine Grundsätze.....	32
3.2.2 Grundlegende Konzepte.....	33
3.2.2.1 Kommunikation.....	33
3.2.2.2 Serverseitige Komponenten.....	36
3.2.2.3 Clientseitige Komponenten.....	37
3.2.3 Modularer Aufbau.....	38
3.2.4 GIS-Funktionen.....	40
3.2.4.1 Neuen Bestand anlegen.....	40

3.2.4.2 Existierenden Bestand bearbeiten.....	41
3.2.4.3 Bestand Löschen.....	42
3.2.4.4 Bestände vereinigen.....	43
3.2.4.5 Bestände aufteilen.....	44
3.2.5 Datenmodell.....	45
4 Ergebnisse.....	47
4.1 Umsetzung.....	47
4.2 Realisierbarkeit.....	47
4.3 Workflow.....	48
4.4 Speicherung.....	50
4.5 Missbrauch.....	53
5 Diskussion.....	55
5.1 Stärken und Schwächen des WFS-Dienstes.....	55
5.2 Alternativen.....	55
5.3 Empfehlungen.....	57
5.4 Zusätzliche Rollen eines Flurstück-WFS-Dienstes.....	59
5.5 Ausblick.....	60
5.6 Zusammenfassung.....	61
6 Anhang.....	63
6.1 GetCapabilities	63
6.2 HTML-Struktur der Anwendung.....	64
6.3 Datenmodell in GeoDjango.....	65
7 Literatur.....	66

Abbildungen

Abb. 1: WaldInfoPlan.....	15
Abb. 2: BayernViewer agrar.....	16
Abb. 3: Die drei WFS Requests.....	21
Abb. 4: UML-Diagramm der Anwendungsfälle.....	28
Abb. 5: Datenmanagement-Funktionen.....	31
Abb. 6: Client-Server Kommunikation beim Anlegen der Eigentumsgrenzen.....	35
Abb. 7: Client-Server Kommunikation beim Anlegen von Waldbeständen.....	36
Abb. 8: Systemarchitekturskizze - Modularer Aufbau.....	39
Abb. 9: Aktivitätsdiagramm "Bestand anlegen".....	41
Abb. 10: Aktivitätsdiagramm "Bestand bearbeiten".....	42
Abb. 11: Aktivitätsdiagramm "Bestand löschen".....	43
Abb. 12: Aktivitätsdiagramm "Bestände vereinigen".....	44
Abb. 13: Aktivitätsdiagramm "Bestand teilen".....	45
Abb. 14: Datenmodell.....	46
Abb. 15: Screenshot des Prototyps in Google Chrome.....	49
Abb. 16: Flurstücke kopieren.....	51
Abb. 17: Flurstücke als Kartierunterlage.....	52

Tabellen

Tab. 1: Ausgabeformate.....	22
Tab. 2: Räumliche Operatoren.....	23
Tab. 3: Vergleichsoperatoren.....	24
Tab. 4: Arithmetische Operatoren.....	25
Tab. 5: Feature Attribute.....	26
Tab. 6: Kartenthemen.....	32
Tab. 7: Rollen der Flurkarte beim Erstellen von Waldbeständen.....	53

Abkürzungen

AJAX	Asynchronous JavaScript and XML
ALB	Automatisiertes Liegenschaftsbuch
ALK	Automatisierte Liegenschaftskarte
ALKIS	Amtliches Liegenschaftskatasterinformationssystem
BBOX	Bounding Box Filter in WFS-Requests
CC-SA	Creative Commons Share Alike License
DFK	Digitale Flurkarte
DRM	Digital Rights Management
DVD	Digital Versatile Disc
ESRI	Environmental Systems Research Institute, Inc.
FBG	Forstbetriebsgemeinschaft
GDAL	Geospatial Data Abstraction Library
GEOS	Geometry Engine Open Source
GIS	Geographisches Informationssystem
GML	Geography Markup Language
GPL	GNU General Public License
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
LVG	Landesamt für Vermessung und Geoinformation
OGC	Open Geospatial Consortium
OWS	OpenGIS Web Services
SOA	Service Oriented Architecture
SQL	Structured Query Language
UML	Unified Modeling Language
WBV	Waldbesitzervereinigung
WFS	Web Feature Service
WFS-T	Transactional Web Feature Service
WMS	Web Map Service
WOA	Web Oriented Architecture
XML	Extensible Markup Language

1 Einleitung

Mit der Veröffentlichung der ersten WMS- und WFS-Spezifikation im Mai 2005 durch das Open Geospatial Consortium wurden die Weichen für ein interoperables Daten-Management gestellt, wie es seitens der öffentlichen Verwaltung gewünscht wird.

Während der Zugang zu öffentlichen Daten in der Vergangenheit noch durch aufwändige Verfahren behindert wurde, kommen die Vorteile webbasierter Austauschplattformen zunehmend ans Licht.

Die Qualität und Vielfalt der angebotenen Geodaten wachsen, während die Datenerfassung und Datenverarbeitung durch den Vormarsch neuer Technologien stark vereinfacht wird.

Eines der wertvollsten Datenprodukte in Bayern ist das „Amtliche Liegenschaftskataster“ (ALKIS), das vom „Landesamt für Vermessung und Geoinformation“ (LVG) vertrieben wird. Jeder erfasste Punkt hat eine Präzision von weniger als drei Zentimetern und ist Teil eines topologischen Netzwerkes aus Polygonen, die mit amtlichen Daten, wie Eigentümer und Flächengröße hinterlegt sind.

Aktuell beziehen etliche Kommunen Daten vom LVG oder es bestehen Nutzungsverträge. In regelmäßigen Abständen werden neue Daten auf DVD gebrannt und postalisch an die Kunden versandt. Häufig müssen die Geodaten für speziellere Weiterverarbeitungen geographisch transformiert und im Format konvertiert werden. Datennutzer müssen speziell in der Verwendung von Geodaten ausgebildet werden und verbringen oftmals viel Zeit mit der „unproduktiven“ Aufgabe ein GIS „aufzusetzen“.

Für die Mehrzahl der Nutzer von Flurstücksdaten in geographischen Informationssystemen sind Webanwendungen leichter zu benutzen und zu verstehen.

Das LVG in Bayern hat den Trend hin zu einem interoperablen Geodaten austausch früh erkannt und einen WMS-Dienst zusammen mit dem „Bayern Viewer“ veröffentlicht, der sowohl die Nutzung innerhalb eines eigenen webGIS-Portals¹ als auch per Einbindung in Desktop-Anwendungen ermöglicht. Flurstücke können per Suchfunktion gefunden und Attributdaten, wie Flächengröße oder Gemeindezugehörigkeit ermittelt werden.

Obwohl dies eine clevere Lösung für die meisten Anwendungsfälle darstellt, handelt es sich beim Bayer-Viewer um einen Raster-Daten-Dienst mit der Kehrseite, nicht auf die

1 <http://www.geodaten.bayern.de/BayernViewer2.0/index.cgi> (02.04.2012)

Vektordaten zugreifen zu können.

Die Angst, Vektordaten per WFS-Dienst anzubieten, rührt von dem vermeintlichen Kontrollverlust über wertvolle Daten. Ein Nutzer eines Dienstes könne sich lokal eine Kopie des kompletten Datenumfangs des LVG anfertigen, wodurch jegliche Datenhoheit ad absurdum geführt würde. Diese Gefahr mag bestehen, wenn jegliche sicherheitsrelevanten Einstellungen und Barrieren außer Kraft gesetzt sind. Ein Zugriff auf Vektordaten per WebGIS-Applikation bietet hierbei eine beruhigende Limitierung von Geometrien pro Anfrage im Hintergrund einer Benutzer-Schnittstelle.

1.1 Ziele

Das übergeordnete Ziel dieser Arbeit ist, neue Erkenntnisse über Einsatzmöglichkeiten von vektorbasierten Web-Diensten zu gewinnen.

Das spezielle Ziel dieser Untersuchung ist, die Vorteile und Gefahren in der Arbeit mit Katastraldaten und Webdiensten besser zu verstehen. Eine prototypischen Web-Anwendung und deren exemplarischer Anwendungsfall soll umgesetzt werden, um zwei konkrete Möglichkeiten der Verwendung von Flurstücksdaten durch den Anwender zu beleuchten und unter dem Aspekt der Usability zu untersuchen.

Methoden

Als geeignete Methode zur Erreichung dieses Zieles dient die Fallstudie, in der ein konkreter Anwendungsfall untersucht und unter verschiedenen Aspekten bewertet wird. Die zum Fall gehörenden Voraussetzungen und Rahmenbedingungen werden geschildert [KAISER 1983] und zusammengefasst, um ein möglichst holistisches Verständnis des Untersuchungsgegenstandes zu erlangen. Zukünftige technische Entscheidungen für produktive Anwendungsfälle können so durch den Vergleich mit dieser Untersuchung erleichtert werden.

Gegenüber alternativer Methoden, wie einer reinen Literaturstudie, wird hierbei erhofft, durch eine praxisnahe Betrachtung auch zahlreiche technische Aspekte beleuchten zu können.

Unter der Beleuchtung mehrerer Aspekte wie Usability, Sicherheit und technischen

Voraussetzungen werden Vor- und Nachteile der gewählten Herangehensweise gegenüber bestehender Lösungen herausgearbeitet.

Im gewählten Lösungsansatz werden zwei Prototypen entwickelt. Zum einen wird seitens der Bayerischen Vermessungsverwaltung ein Dienst zur Verfügung gestellt, wie er im Produktiv-Betrieb in ähnlicher Weise betrieben werden könnte. Zum anderen wird eine WebGIS-Anwendung programmiert, die diesen Dienst einbindet, um Erkenntnisse im Umgang mit dieser neuen Schnittstelle zu gewinnen.

Der Grund für die Wahl eines WebGIS anstatt beispielsweise eines Desktop-Clients liegt darin, die Philosophie hinter der OGC-Schnittstelle weiterzutragen, in der Daten für einen möglichst großen Anwenderkreis zugänglich gemacht werden sollen. Weiters wird die Annahme getroffen, dass eine erfolgreiche Umsetzung in einem WebGIS mit einer möglichen Verwendung in einer Desktop-Anwendung in Hinblick auf den Datendienst als vergleichbar anzusehen ist.

Neben diesen Methoden werden weiters Hypothesen aufgestellt, die den Einsatz des Dienstes in der Anwendung grundsätzlich rechtfertigen. Folgende Annahmen werden getroffen und in der vorliegenden Arbeit auf ihren Wahrheitsgehalt hin untersucht.

A. Die Anwendung kann gänzlich mit OpenSource-Software umgesetzt werden.

Es müssen sowohl auf dem Server wie auf dem Client keine proprietären Softwarekomponenten eingesetzt werden. Die Anwendung kann gänzlich mit kostenlos bereitgestellter Software, die durch eine uneingeschränkte Lizenz veröffentlicht ist, umgesetzt werden.

B. Der WFS-Dienst unterstützt den Benutzer bei der Erstellung von Forstkarten.

Der Benutzer erfährt durch die Einbindung des WFS-Dienstes keine Einschränkungen gegenüber herkömmlichen Datengrundlagen. Weiters kann ein flüssiger Workflow durch die Verwendung des WFS-Dienstes sichergestellt werden.

C. Es muss keine Kopien der Flurkarte lokal gespeichert

werden.

Zur Verwaltung der Eigentumsgrenzen muss keine dauerhafte lokale Kopie der Flurstücksgeometrie-Daten auf dem Server oder dem Client gespeichert werden.

D. Ein Missbrauch des WFS-Dienstes kann ausgeschlossen werden.

Es entstehen keine Sicherheitslücken in der Anwendung, die es einem böswilligen Anwender ermöglichen würden, sich eine Kopie sämtlicher Flurstücksdaten des Diensteanbieters zu beschaffen oder Flurstücksdaten auf dem Server zu verändern. Ausserdem besteht durch Sicherheitsmechanismen in der Anwendung ein umfassender Schutz von personenbezogenen Daten.

Abschließend werden Empfehlungen für den Umgang mit WFS-Diensten im Zusammenhang mit Vermessungsdaten aufgezeigt und ein Blick in zukünftige Entwicklungen und Möglichkeiten geworfen.

1.2 Gegenstand

Aufgrund der weiten Verbreitung sowie etablierten Standardisierung gegenüber Alternativen wie etwa dem Mapfish-Protokoll² wird für diese Untersuchung der WFS-Dienst als Schnittstelle zwischen Geodaten und Benutzer ausgewählt.

Weiters wurde das Landesvermessungsamt in München als Partner gewonnen, wodurch die Möglichkeit geschaffen wurde, mit reellen Daten und einem potenziellen Dienst-Lieferanten zusammenzuarbeiten.

Der Anwendungsfall stammt aus dem Gebiet der forstlichen Planung, in der Flurstücke das Grundgerüst jeglicher Forstkarten bilden. Während Großwaldbesitzer seit langem ausgefeilte Methoden der geographischen Informationsverarbeitung zur forstlichen Planung und Kartierung verwenden, fehlen den Kleinwaldbesitzern insbesondere die Werkzeuge und Ressourcen, um Karten zu erstellen oder Waldmessdaten auszuwerten. Eine einfache Webanwendung auf der Basis von OpenSource-Software soll nun mit Hilfe eines Flurkarten-Layers dem Waldbesitzer ein Werkzeug in die Hand geben,

2 <http://trac.mapfish.org/trac/mapfish/wiki/MapFishProtocol> (10.04.12)

selbstständig exakte Karten seiner Bestände anzulegen, um nach Eingabe selbst gemessener Daten und deren Auswertung eine eigene mittelfristige Planung in seinem Wald vorzunehmen.

1.3 Ähnliche WFS-Dienste

WFS-Dienste stellen gegenüber WMS-Diensten bei Vermessungsbehörden eine Minderheit dar. Der vorliegende Testservice des LVG, wie er in Abschnitt 2 näher vorgestellt wird, erweitert somit das Spektrum weniger bestehender Vektor-Daten-Dienste.

Gazetteer Service WFS-G

Einer der bekanntesten Dienste derzeit ist der „Gazetteer“, der versucht Informationen wie Hausadressen etc. durch Bereitstellung von Koordinaten bzw. durch zugeordnete Wörter zu verorten [GDI-BY 2006]. Das LVG in Bayern, sowie andere Landesvermessungsämter betreiben diesen Service, der Punktgeometrien und Attribute als Antwort zurückliefert. WFS-G ist eine eigene Teilspezifikation des OGC [OGC 05-035r2] und befindet sich derzeit noch in Arbeit.

GisInfoService (Steine und Erden e.V.)

Auf der Basis von Mapbender wird den Vereinsmitgliedern ein webbasiertes Boden-Auskunftssystem angeboten. Hier kommt neben mehreren WMS-Diensten für zugriffsberechtigte Personen und nur für bestimmte Bundesländer ein WFS-Dienst zur Anzeige und Abfrage der DFK zum Einsatz [DVW BAYERN 2010].

1.4 Ähnliche Projekte

WaldInfoPlan

Waldinfoplan ist eine Web-Plattform, die von der Genossenschaft für Waldwirtschaft und den beteiligten forstlichen Zusammenschlüssen geschaffen wird³. Hauptaufgabengebiet dieser WebGIS-Anwendung ist die Verwaltung von Waldpflegeverträgen innerhalb einer Waldbesitzervereinigung sowohl vom Büro als auch von mobilen Computern. Es stehen Werkzeuge zur Organisation von Holzlagerplätzen, sowie zur Maßnahmenplanung zur Verfügung.

Die Digitale Flurkarte kann per WMS-Dienst eingeblendet werden. Sie unterstützt das Auffinden von Flächen und hilft bei der Überprüfung von Grenzabständen vor Ort.

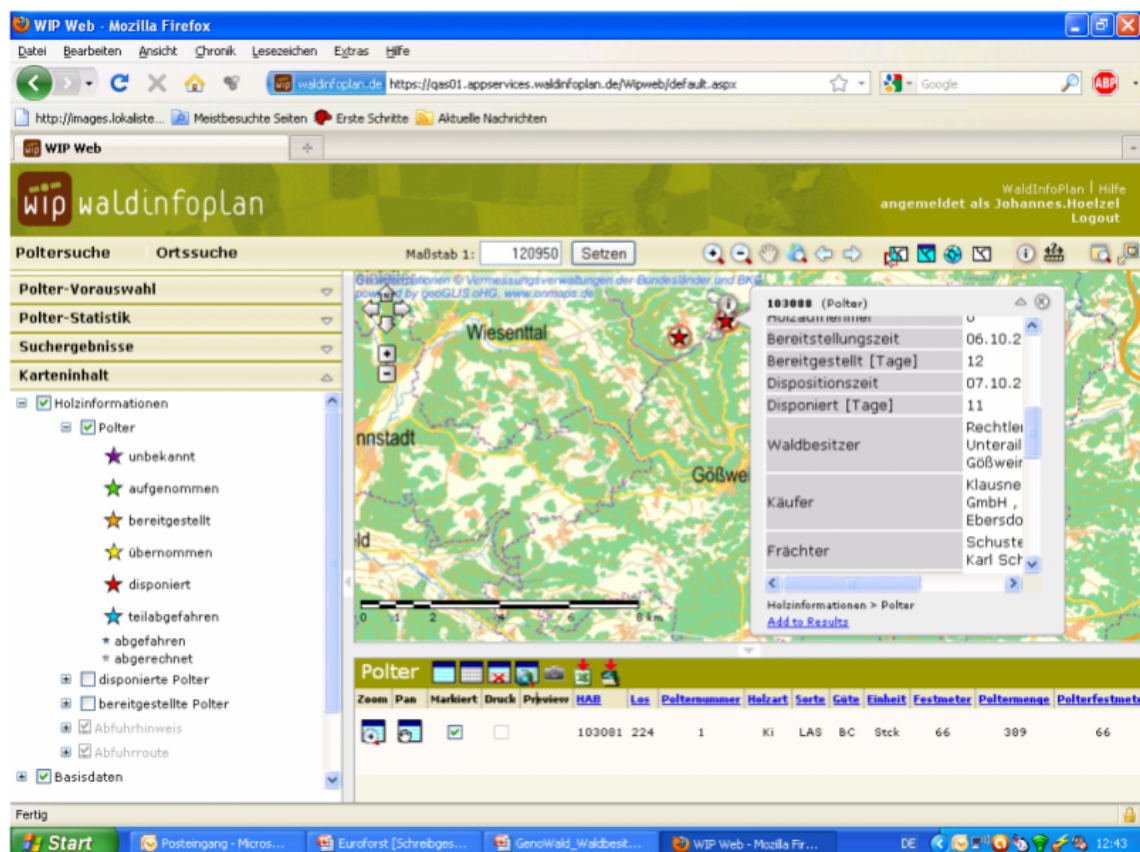


Abb. 1: WaldInfoPlan

³ <http://www.waldinfoplan.de/wip.html> (29.02.2012)

BayernViewer agrar

Diese WebGIS-Anwendung bietet landwirtschaftlichen Betrieben Unterstützung bei der Einreichung von Mehrfachanträgen. Darüber hinaus können die Nutzer des Onlinedienstes weitere Fachinformationen, wie z.B. die Anzeige der Naturschutzgebiete, Bodenschätzungsergebnisse für Erosionskataster und Anbauplanung sowie die neue Fachschale „Tatsächliche Nutzung“ einsehen und nutzen⁴.

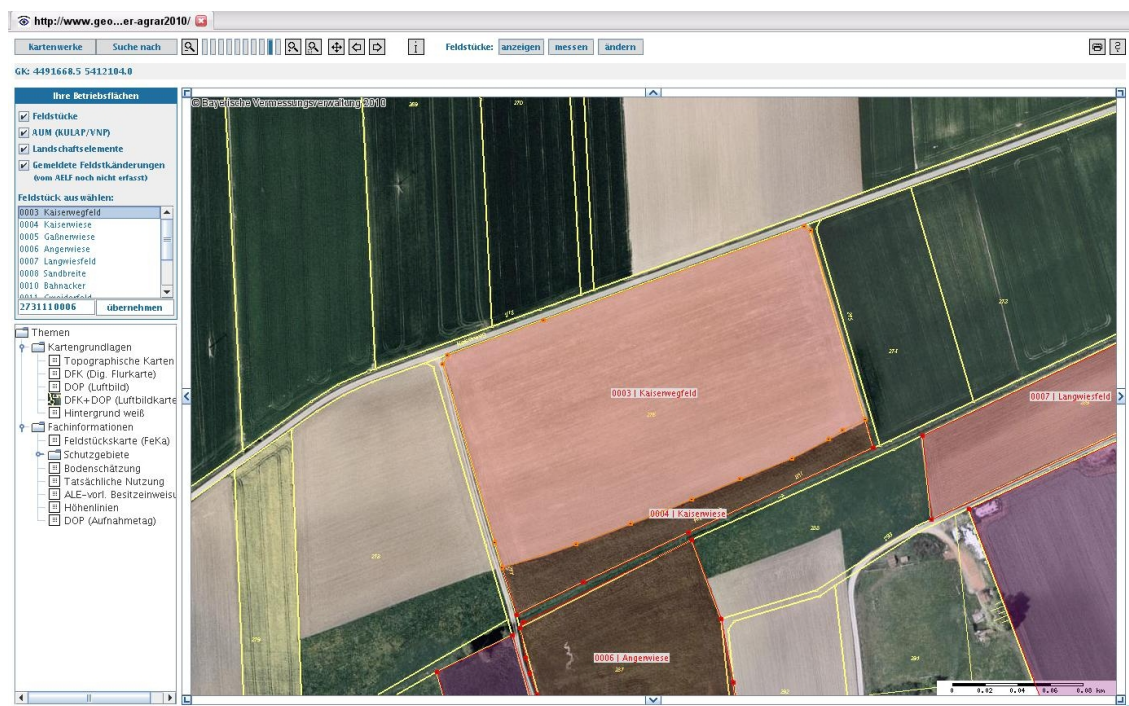


Abb. 2: BayernViewer agrar

Flurstücke werden auch hier per WMS-Dienst eingebunden. Diese Anwendung wird direkt von der „Bayerischen Vermessungsverwaltung“ bereitgestellt und bietet Landwirten einen kostenlosen Zugang.

1.5 Wesentliche Begriffe

WBV / FBG

Waldbesitzervereinigungen (WBV) und Forstbetriebsgemeinschaften (FBG) sind Vereine, deren Mitglieder in der Regel Waldbesitzer mit kleinen bis mittleren Waldflächen sind. Durch den Zusammenschluss mehrerer Klein-Waldflächen sollen die Holzernte und die

⁴ <http://www.vermessungsamt-bayreuth.de/artikel,,302.html> (29.02.2012)

Holzvermarktung verbessert und zentral gesteuert werden. In Bayern werden immer häufiger sogenannte „Waldpflegeverträge“ geschlossen, die die Waldbewirtschaftung an die Vereine als Treuhänder, gegen Entgelt aus dem Holzernteerlös abgeben.

WFS / WMS

Der Web Feature Service (WFS) sowie der Web Map Service (WMS) stellen Web-Dienste dar, mit deren Hilfe Geodaten aus einer Datenbank oder anderer Datenquellen mittels HTTP-Protokoll an einen Client ausgeliefert werden können. Der wesentliche Unterschied zwischen den beiden Diensten ist die Repräsentation von Features durch Raster- bzw. Vektordaten. Während der WMS-Dienst nur Bildkacheln liefert, kann der WFS-Dienst Vektordaten z.B. als GML-Objekte senden und im Falle von WFS-T auch Änderungen an den Server zurücksenden.

ALB / ALK / ALKIS / DFK

Das Automatisierte Liegenschaftsbuch (ALB) konzentriert sämtliche von den Vermessungsbehörden in Deutschland verwaltete Flurstücke zentral in einer Datenbank. Im „Automatisierten Liegenschaftskataster“ (ALK), oft auch „Digitale Flurkarte“ (DFK) genannt, werden die ALB-Daten visualisiert und in einer Karte bereitgestellt. Das „Amtliche Liegenschaftskataster-Informationssystem“ (ALKIS), als Nachfolgesystem, führt das ALB und das ALK in einem GIS zusammen. Bayern plant die sukzessive Einführung von ALKIS an den Vermessungsämtern ab der zweiten Jahreshälfte 2012⁵.

1.6 Methoden

„Modelle sind ein natürlicher Weg um komplexe Vorgänge und Systeme zu verstehen, zu untersuchen und zu planen, um sie dann in die Realität zu transformieren.“

[BORN ET AL. 2004]

Da diese Arbeit Aspekte der Systemtheorie im Allgemeinen und der Softwareentwicklung im Speziellen aufweist, werden verschiedene Methoden, wie die Entwicklung von Modellen und Systemarchitekturskizzen verwendet, um Sachverhalte zu veranschaulichen.

Daneben wird ein Prototyp entwickelt, um die Theorie in der Praxis zu erproben. Wichtige

⁵ http://vermessung.bayern.de/geobasis_lvg/ALKIS.html (29.02.2012)

Aspekte, die während der Implementierung der Anwendung auftreten, werden festgehalten und fließen in die Untersuchung ein.

Am Schluss werden die zu Beginn definierten Thesen aufgrund der Erfahrungen im Entwurf und der Umsetzung der Anwendung auf ihren Wahrheitsgehalt hin überprüft und kritisch hinterfragt.

1.6.1 Anwendungsfall-Diagramme

Zum besseren Verständnis der Zielgruppen und deren Motivation mit der Anwendung zu interagieren, wird ein Use-Case-Diagramm verwendet (s. Seite 28). Mittels UML werden die Akteure und Aktionen als Objekte dargestellt und mit Linien verbunden. Der Kontext der Aktionen wird schnell aufgrund der Lage innerhalb oder außerhalb der Anwendung, die als rechteckiges Feld im Hintergrund dargestellt wird, ersichtlich.

Ein Use-Case-Diagramm soll keine Details über konkrete Vorgehensweisen bei der Lösung eines Problems liefern, sondern vielmehr ein allgemeines Gefühl für die Komponenten und deren Umwelt vermitteln.

Konkretere Planungen und Lösungsstrategien werden auf der Basis von Anwendungsfällen entwickelt und verwenden andere Methoden, wie Aktivitätsdiagramme oder Systemarchitekturskizzen.

1.6.2 Aktivitätsdiagramme

Um die im Anwendungsfall grob skizzierten Aktionen detaillierter betrachten zu können, kommen unter anderem Aktivitätsdiagramme zum Einsatz.

Bei diesem Diagramm-Typ wird eine Aktion in kleinere Aktionen unterteilt und die sich ändernden Zustände, sowie Entscheidungswege festgehalten. Der Anfang der Aktion wird als schwarzer Punkt zuoberst markiert. Der Betrachter kann nun den Pfeilen abwärts folgend sich durch die Unteraktionen bewegen, bis er zum Schluss angelangt ist. Das Ende der Aktion wird durch einen schwarzen Punkt mit einem zusätzlichen Kreis abgebildet.

Diese Art der visuellen Veranschaulichung kann gleichermaßen als Anleitung für die Programmierung, als auch als eine Planungsskizze angesehen werden, die hilft, unvorhergesehene Sonderfälle frühzeitig zu sondieren.

In dieser Arbeit kommen Aktivitätsdiagramme zur Veranschaulichung konkreter Interaktionen des Anwenders mit der Anwendung zum Einsatz, die im Ganzen zu einem

Workflow zusammengefügt werden können (s. Seite 41 ff.).

1.6.3 Systemarchitektur-Diagramme

Eine einheitliche Bezeichnung für diese Art des Diagrammes existiert nicht. Es finden sich jedoch Begriffe, wie „Architekturskizze“⁶ oder „Komponenten-Diagramm“⁷.

Ziel dieses Diagrammes ist es die wesentlichen Komponenten eines Systems zu vereinen und ihr Zusammenspiel zu verdeutlichen. Wie in oben genannten Diagrammtypen werden auch hier Objekte mit Linien verbunden. Hinzu kommen sich überlagernde Objekte, Logos und Beschreibungen der Verbindungspfeile.

So können die einzelnen Komponenten meist besser als in verbaler Form zusammengebracht und auf einen Blick im Gesamtsystem lokalisiert werden.

1.6.4 Hypothesen

„Ich lege meiner Untersuchung immer eine Behauptung zugrunde, die ich für besonders stark halte; und das, von dem ich dann den Eindruck habe, dass es damit in Einklang steht, nenne ich wahr; was dagegen damit nicht in Einklang zu stehen scheint, nenne ich unwahr.“

[Platon]⁸

Um die Ziele dieser Arbeit klar abzustecken und zu quantifizieren, werden Hypothesen (s. Seite 12) aufgestellt, die zugleich den Kern der Motivation des Autors widerspiegeln.

Auf diese Weise ist es möglich das Ergebnis der Untersuchung klar festzuhalten, auch wenn eine Hypothese nur bedingt zutreffen sollte.

6 http://www.jterms-project.de/jterms_Plattform/architekturskizze.html (08.04.2012)

7 <http://de.wikipedia.org/wiki/Komponentendiagramm> (08.04.2012)

8 Platon, Phaidon (100a), in: Sämtliche Werke Bd. II, Hamburg 1994, ISBN 349955562X

2 Der WFS-Dienst

Der in dieser Arbeit verwendete WFS-Dienst liegt ausschließlich als Prototyp vor und wird zu Forschungszwecken unentgeltlich zur Verfügung gestellt. Aus technischer Sicht steht hinter dem Dienst ein Geoserver (s. Seite 37), der auf Basis von Java implementiert ist⁹.

2.1 LVG Bayern

Das LVG mit Sitz in München ist zentraler Ansprechpartner für den gesamten Bereich der amtlichen Vermessung in Bayern¹⁰. Neben der Erfassung von Vermessungsdaten und deren Datenhaltung wickelt das LVG den Vertrieb von Geodaten unter anderem über ein Geoportal ab.

Die Produktpalette reicht von Orthophotos, über Laserscanning-Daten und der Digitalen Flurkarte (DFK) bis hin zu historischen Kartenausschnitten.

Flurstücke als Auszug der DFK werden im SHAPE-Format ab einem Preis von 3,40 € angeboten. Per Web-Dienst werden bisher nur Raster-Daten ausgeliefert, die über einen nutzungsabhängigen Tarif mit und ohne Speicherung abgerechnet werden (LVG Preisliste 02/2012). Die Anzahl aller Flurstücke in Bayern beläuft sich derzeit auf ca. 10,3 Millionen¹¹.

2.2 Capabilities

Da der Dienst nur der Datenabfrage dient, wird der Basis-WFS ohne Transactional Erweiterung (WFS-T) eingesetzt. Datenänderungen obliegen der Hoheit der Vermessungsverwaltungen und werden über andere Schnittstellen abgewickelt.

Mit dem WFS-Dienst sind drei verschiedene Requests möglich – GetCapabilities, DescribeFeatureType und GetFeature (s. Abb. 3).

Mit dem Befehl „GetCapabilities“ kann Auskunft über Schnittstellen, Funktionen, Filter und verschiedene Metadaten des Dienstes erhalten werden. Die zurückgelieferten

9 Herr Ulrich Meier, Landesamt für Vermessung und Geoinformation; Online-Dienste, Basiskomponente GeodatenOnline, Alexandrastraße 4, D-80538 München (Email vom 24.11.2011)

10 <http://vermessung.bayern.de/service/organisation/lvg.html> (28.02.2012)

11 <http://vermessung.bayern.de/service/organisation.html> (28.02.2012)

Werte sind entscheidend für den Umgang des Clients mit dem Dienst an sich und werden im Folgenden vorgestellt.

Der Aufruf von GetCapabilities erfolgt durch Angabe einer URL und Anfügen des Key-Value-Paares „REQUEST=GetCapabilities“:

```
https://geoportal.bayern.de/WfsAccess/query.xml?
REQUEST=GetCapabilities
```

Ausgabeformate

Per Spezifikation muss ein WFS-Dienst standardmäßig das Format GML ausliefern können. Wird kein Format-Parameter übergeben, liefert der Dienst in diesem Fall GML3 zurück.

Für die Anwendung relevant ist das JSON-Datenformat oder auch GeoJSON. Es wird an Stelle von GML verwendet, da es in JavaScript-Syntax geschrieben ist und daher von JavaScript-Anwendungen ohne Parsen direkt ausgelesen werden kann. Insbesondere bei kleinen Datenmengen wird ein Performance-Vorteil gegenüber XML-Daten erwartet¹².

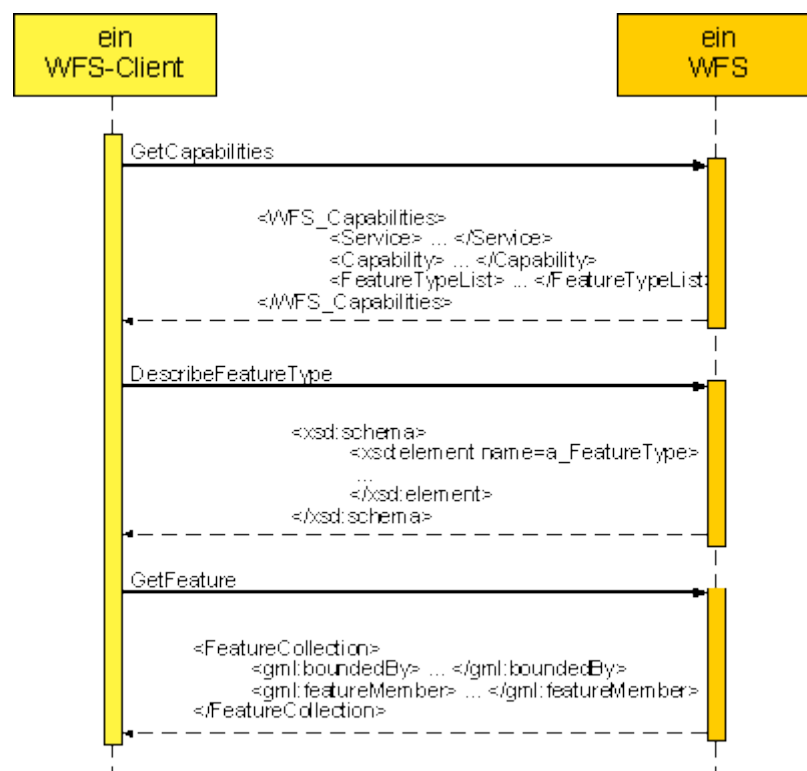


Abb. 3: Die drei WFS Requests.

Quelle: [FIRL, M. 2004]

¹² <http://edwardawebb.com/tips/xml-json> (08.04.2012)

Format	Codierung	WFS-Version	Beschreibung
GML2	Text	1.0.0	XML-Schema für Geodaten in der Version von 2001 [STROBEL UND REINHARDT 2006].
GML2-GZIP	Binär	1.1.0	Wie GML2, aber mittels „GZIP“ komprimiert.
text/xml; subtype=gml/ 2.1.2	Text		XML-Schema für Geodaten in der Version 2.1.2 von 2002 [STROBEL UND REINHARDT 2006].
text/xml; subtype=gml/ 3.1.1	Text	1.1.0	XML-Schema für Geodaten in der Version 3.1.1 von 2005 [STROBEL UND REINHARDT 2006].
GML3	Text	1.0.0	XML-Schema für Geodaten in der Version von 2003 mit Erweiterungen wie Topologie, Zeit und Maßeinheiten [STROBEL UND REINHARDT 2006].
SHAPE-ZIP	Binär	1.0.0	Komprimierte Sammlung von Dateien, die im Ganzen als „Shapefile“ bezeichnet werden. Proprietäres Format der Fa. ESRI Inc.
JSON/ GEOJSON*	Text	1.0.0	Geographische Erweiterung der JavaScript Object Notation.

Tab. 1: Ausgabeformate

* für diese Anwendung relevant

FeatureTypes

Als FeatureType werden die Daten-Ebenen bezeichnet, die jeweils aus einer Datenquelle stammen. In diesem Fall gibt es genau ein FeatureType, der auf dem Server im Namespace „by“ (für „Bayern“) liegt und im Ganzen „by:forchheim“ heißt. Diese Bezeichnung muss beim Anfordern von Features im Request mindestens enthalten sein.

Referenzsystem

Die Daten liegen auf dem Server im geodätischen Referenzsystem „Deutsches Hauptdreiecksnetz Zone 4“ vor, das den EPSG-Code 31468 trägt. Das Koordinatensystem ist Gauß-Krüger und es wird das Bessel Ellipsoid von 1841 verwendet.

Da es sich um metrische Koordinaten handelt, muss bei der simultanen Verwendung mit UTM-basierten Daten, wie WGS84, eine „NTv2“-Transformation durchgeführt werden. NTv2 wie auch „NADCON“ bezeichnet eine gitternetzbasierte Interpolation auf Grundlage von jeweils in beiden Systemen bekannter Festpunktdaten¹³.

Räumliche Operatoren

Die folgenden räumlichen Operatoren entsprechen der offiziellen Filter-Spezifikation [OGC 04-095] sowie der SimpleFeature-Spezifikation für SQL [OGC 99-049].

Operator	Beschreibung
Disjoint	Features berühren und überlappen sich nicht.
Equals	Features sind exakt gleich.
DWithin	Erstes Feature befindet sich innerhalb der definierten Entfernung des zweiten Features.
Beyond	Erstes Feature befindet sich jenseits der definierten Entfernung des zweiten Features.
Intersect	Features überschneiden sich.
Touches	Features berühren sich.
Crosses	Features (Linien) überschneiden sich.
Within	Erstes Feature befindet sich komplett innerhalb des zweiten Features.
Contains*	Erster Feature beinhaltet komplett das zweite Feature.
Overlaps	Features überlappen sich
BBOX*	Features befinden sich innerhalb eines definierten Rechtecks

Tab. 2: Räumliche Operatoren

* für diese Anwendung relevant

Insbesondere für diese Anwendung von Interesse sind **BBOX**, um alle Features innerhalb des sichtbaren Kartenfensters abzufragen und **Contains**, um nur ein Feature zurückzuliefern, das an der Stelle eines Mausklicks liegt.

¹³ http://www.geod.nrcan.gc.ca/tools-outils/ntv2_e.php (06.04.2011)

Vergleichsoperatoren

Es stehen 4 Abfrageoperatoren für Feature-Attribute zur Verfügung, wobei **Simple_Comparisons** mehrere sehr nützliche Operatoren beinhaltet. Für die Suche nach bestimmten Flurstücksnummern oder Gemeindennamen kommt beispielsweise der „=“-Operator zur Anwendung. Daneben ist für die textuelle Suche **Like** sehr nützlich, um auch Werte zurückzuliefern, die keine exakte Übereinstimmung aufweisen.

Operator	Beschreibung
Simple_Comparisons*	Boole'sche Operatoren, wie =,<>,>,<,<=,>=
Between	Zahlenwerte zwischen einem Minimal- und einem Maximalwert
Like*	Die Zeichenkette kommt innerhalb der anderen vor.
NullCheck	Überprüfung ob ein Wert für das Attribut existiert.

Tab. 3: Vergleichsoperatoren

* Für die Anwendung relevant

Arithmetische Operatoren

Die arithmetischen Operatoren erlauben Berechnungen und Analysen der vorhandenen Geometrien und Feature-Attributen. Für diese Anwendung sind sie zunächst nicht relevant, da bestimmte Berechnungen wie **Area** clientseitig mittels OpenLayers abgewickelt werden (s. Tab. 4).

Für zukünftige Aufgaben ist es jedoch womöglich nützlich auf bestimmte serverseitige Berechnungsmethoden zurückgreifen zu können.

N	f	N	f	N	f
1	abs	1	geomLength	2	notEqualTo
1	abs_2	2	getGeometryN	1	numGeometries
1	abs_3	1	getX	1	numInteriorRing
1	abs_4	1	getY	1	numPoints
1	acos	1	getZ	2	overlaps
1	Area	2	greaterEqualThan	1	parseBoolean
1	asin	2	greaterThan	1	parseDouble
1	atan	2	IEEEremainder	1	parseInt
2	atan2	3	if_then_else	2	pointN
3	between	11	in10	2	pow
1	boundary	3	in2	1	PropertyExists
1	boundaryDimension	4	in3	2	Quantile
2	buffer	5	in4	0	random
3	bufferWithSegments	6	in5	2	relate
1	ceil	7	in6	3	relatePattern
1	centroid	8	in7	1	rint
2	classify	9	in8	1	round
1	Collection_Average	10	in9	1	round_2
1	Collection_Bounds	1	int2bbool	1	roundDouble
1	Collection_Count	1	int2ddouble	1	sin
1	Collection_Max	1	interiorPoint	1	sqrt
1	Collection_Median	2	interiorRingN	2	StandardDeviation
1	Collection_Min	2	intersection	1	startPoint
1	Collection_Sum	2	intersects	2	strConcat
1	Collection_Unique	1	isClosed	2	strEndsWith
2	contains	1	isEmpty	2	strEqualsIgnoreCase
1	convexHull	2	isLike	2	strIndexOf
1	cos	1	isNull	2	strLastIndexOf
2	crosses	1	isRing	1	strLength
2	difference	1	isSimple	2	strMatches
1	dimension	1	isValid	4	strReplace
2	disjoint	3	isWithinDistance	2	strStartsWith
2	distance	1	length	3	strSubstring
1	double2bool	2	lessEqualThan	2	strSubstringStart
1	endPoint	2	lessThan	1	strTrim
1	envelope	1	log	2	symDifference
2	EqualInterval	2	max	1	tan
2	equalsExact	2	max_2	1	toDegrees
3	equalsExactTolerance	2	max_3	1	toRadians
2	equalTo	2	max_4	2	touches
1	exp	2	min	1	toWKT
1	exteriorRing	2	min_2	2	union
1	floor	2	min_3	2	UniqueInterval
1	geometryType	2	min_4	2	within
1	geomFromWKT	1	not		

Tab. 4: Arithmetische Operatoren

2.3 DescribeFeatureType

Beim Aufruf von „DescribeFeatureType“ werden die Attributfelder mit zugehörigen Datentypen des FeatureTypes aufgelistet.

Neben einer eindeutigen ID („gml_id“) sind Felder enthalten, die eine Einschränkung der Suchanfrage ermöglichen. Insbesondere wichtig sind hierbei der Gemeindename und die Gemarkung, in der ein Flurstück liegt. Für Anwender, die ihre Flurstücksnummer kennen, steht die Abfrage des Zählers („zae“) und Nenners („nen“) in Kombination mit der Gemarkung („gemarkung“) zur Verfügung.

Eine wichtige Information liefert die „Amtliche Fläche“ („afl“), die als amtlich geltende Flächengröße des Flurstücks von einer Finanzbehörde zur Berechnung von steuerlichen Nutzungssätzen (vgl. Seite 31) verwendet wird. Diese kann in einzelnen Fällen von der in der Anwendung berechneten Flächengröße um wenige Quadratmeter abweichen.

Die Request-URL:

```
https://geoportal.bayern.de/WfsAccess/query.xml?
REQUEST=DescribeFeatureType&typeName=by:forchheim
```

Name	Datentyp	Beispiel	Beschreibung
gml_id*	String	forchheim.9690	Feature ID
gmnr*	Integer	2190	Gemeindennummer
zae*	Integer	782	Zähler
nen*	Integer	3	Nenner
afl	Integer	1008	Amtliche Fläche
gsl*	String	09471145	Gemeindeschlüssel
gemeinde*	String	Hirschaid	Gemeindenname
kreis*	String	Bamberg	Landkreis
gemarkung*	String	Hirschaid	Gemarkung

Tab. 5: Feature Attribute

* Für diese Anwendung relevant

3 Die Anwendung

Im Folgenden wird die Anwendung mit ihren Anforderungen und die daraus resultierende Architektur skizziert und besprochen. Dieser Abschnitt ist zugleich Grundlage zum Verständnis über die Rolle des WFS-Dienstes und Dokumentation des Lösungsweges hin zu den Ergebnissen des darauffolgenden Abschnittes.

3.1 Anforderungen

Die Spezifikation der Anforderungen von Software, auch „Lastenheft“ genannt, ist ein vom „Institute of Electrical and Electronic Engineers“¹⁴ veröffentlichter Standard (IEEE 830-1998¹⁵), der sowohl die Ziele und die benötigten Funktionen benennt, als auch die daraus resultierenden Lösungswege skizziert.

Die in diesem Abschnitt verwendeten Methoden sind diesem Standard entlehnt und spiegeln nicht eine vollständige Anforderungsspezifikation wider. Es werden lediglich die wichtigsten Inhalte dargestellt, um die Nachvollziehbarkeit des Lösungsweges offen zu legen.

3.1.1 Use-Case

UML-Diagramme eignen sich zur stark vereinfachten Darstellung von Anwendungsfällen [LONGLEY ET AL. 2005]. In diesem Fall treten drei Akteure mit der Webanwendung direkt und indirekt in Interaktion (s. Abb. 4).

Ein Waldbesitzer erhebt Bestandsdaten über seinen Wald und gibt diese zusammen mit Ort und Geometrie seiner Bestände in die Webanwendung ein. In der Webanwendung werden die Daten verwaltet und Übersichten generiert. Diese Übersichten dienen schließlich dem Waldbesitzer und Vertretern von Waldbesitzer-Vereinigungen, sowie Holzkäufern als Grundlage für die Maßnahmenplanung und die Holzvermarktung.

¹⁴ <http://de.wikipedia.org/wiki/IEEE> (16.04.2012)

¹⁵ <http://standards.ieee.org/findstds/standard/830-1998.html> (16.04.2012)

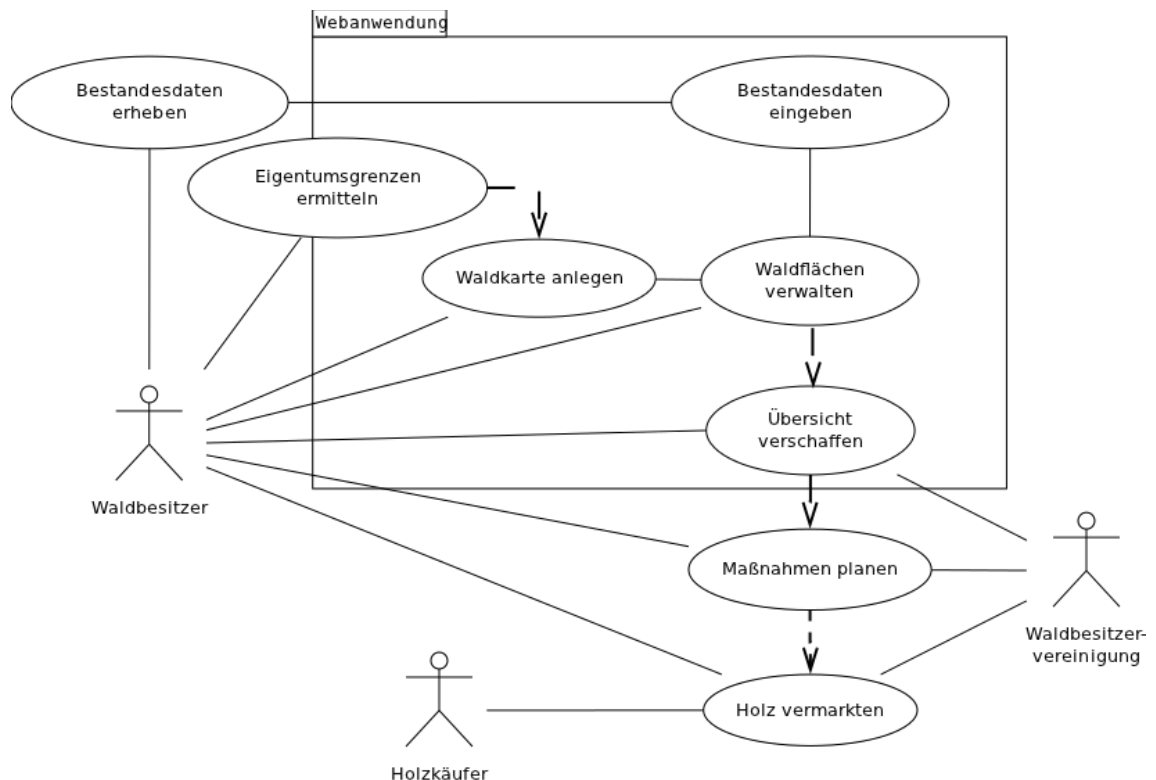


Abb. 4: UML-Diagramm der Anwendungsfälle

Die klassische Abfolge einer einfachen Forstverwaltung von der Inventarisierung über Maßnahmenplanung bis hin zum Holzverkauf findet sich hier wider. Theoretisch ist es möglich, die zuletzt genannten Anwendungsfälle (Maßnahmenplanung u. Holzvermarktung) ebenfalls in einer Webanwendung zu realisieren, was allerdings nicht Gegenstand dieser Untersuchung ist.

3.1.2 Trennung von Flurstücks- und Waldflächenverwaltung

Im Sinne eines typischen Forsteinrichtungs-Workflows, wie er von Forstsachverständigen in mehreren Bundesländern in Deutschland praktiziert wird, beginnt eine Kartierung mit der Bestimmung der Flächen, die sich im Eigentum befinden. Häufig liegt eine Flurstücksliste oder ein älterer Flurkartenausschnitt vor, der diesen Arbeitsschritt erleichtert.

Erst in einem zweiten Schritt werden die Waldflächen in Behandlungseinheiten, den sogenannten Beständen unterteilt und jeweils inventarisiert und beplant.

3.1.2.1 Flurstücksverwaltung

In der Flurstücksverwaltung wird dem Anwender ermöglicht seine im Eigentum

befindlichen Flächen zu ermitteln, zu speichern und ihre Attribute abzufragen.

Dieser erste Arbeitsschritt benötigt eine Schnittstelle zur Digitalen Flurkarte, die durch den WFS-Dienst repräsentiert wird. Aufgrund der erwarteten vielfältigen Herangehensweisen der Anwender wird neben einer textuellen Suche der Flurstücks-Nummer in Kombination mit der Gemarkung auch eine räumliche Suche angeboten.

Aus technischer Sicht wird eine Suchfunktion durch Filterencoding des WFS-Dienstes gelöst. Die Suchparameter des Anwenders werden in maschinenlesbare Form umgewandelt – meist als „Key Value Pairs“ – und per Http-GET Request verschickt. Als Antwort erhält der Client eine Auswahl an Datensätzen, welche die Filterkriterien erfüllen.

Die für diese Anwendung relevanten Filter sind in Tab. 2, Tab. 3 sowie Tab. 4 dunkel hervorgehoben.

Hat der Anwender eines seiner Flurstücke durch textuelle Suche oder Interaktion mit der Karte ermittelt, so fügt er es einer Flurstücksliste hinzu, um eine Übersicht seines Besitzstandes zu bekommen. Eine einfache Auswertung, wie Gesamtfläche oder Gruppierung in Gemarkungen ist nun möglich. Die Flurstücksliste wird auf dem Anwendungsserver zusammen mit den Benutzerdaten abgespeichert und steht bei einem erneuten Login wieder zur Verfügung.

Zur Darstellung der Flurstücke auf der Karte gibt es zwei Möglichkeiten.

Eine erneute Abfrage des WFS-Dienstes

Die Einträge in der Flurstücksliste werden verwendet, um eine erneute Abfrage des WFS-Dienstes abzuschicken. Die Flurstück-ID-Nummern werden an den Dienst übertragen, woraufhin die jeweiligen Geometrien geliefert und auf der Karte angezeigt werden. Diese Variante spart Speicherplatz in der Anwendung und ermöglicht theoretisch eine WFS-Nutzung, in der die Speicherung der Geometrie verboten ist. Ein Nachteil könnte entstehen, wenn der Dienst temporär nicht zur Verfügung steht.

Speicherung der Geometrie auf dem Anwendungsserver

Die am einfachsten zu lösende Art, auf die Flurstücksgeometrien zuzugreifen, ist, die Speicherung der Geometrie direkt in der Flurstücksliste im Zuge der Anlage eines neuen Datensatzes. Hierbei ist ein Nutzungsmodell nötig, das die Vervielfältigung der Geometrie ausdrücklich erlaubt.

Vorteil dieser Variante ist die Unabhängigkeit vom WFS-Dienst, nachdem alle Eigentumsgrenzen gefunden sind. Außerdem bedeutet dies für den WFS-Dienst-Betreiber ein reduziertes aufkommendes Datenvolumen und somit geringere Server-Kosten.

3.1.2.2 Waldflächenverwaltung

Die Waldflächenverwaltung beinhaltet das Anlegen, das Löschen und Verändern sowohl der Bestandesdaten als auch der Geometrien von Waldbeständen. Erst nachdem die Eigentumsgrenzen im ersten Schritt (3.1.2.1) ermittelt wurden, kann die Lage der Bestände korrekt festgehalten werden. Wird direkt in der Waldflächenverwaltung, ohne Kenntnis über die Lage der Eigentumsgrenzen, mit der Kartierung begonnen, muss der Anwender eine genaue Vorstellung der Lage seines Bestandes mitbringen oder Abweichungen in Kauf nehmen, die seine Daten verfälschen könnten.

3.1.3 Funktionalität

Da es sich bei der zu erstellenden Webanwendung um ein Waldflächenverwaltungs-System im Sinne einer datenbankgestützten Anwendung handelt, muss primär das Anlegen, sowie Löschen von Einträgen ermöglicht werden. Das Editieren variabler Attribute durch Dateneingaben und Änderungsmöglichkeiten sowie zusammenfassende und separierende Transformationen erweitern die Anwendung um wertvolle Management-Funktionen. Zuletzt ermöglichen Abfragen die Generierung neuen Wissens über die eingegebenen Daten (vgl. Abb. 5).



Abb. 5: Datenmanagement-Funktionen

3.1.4 Genauigkeit

Die Frage nach dem Kopieren genauer Flurstücksgrenzen gegenüber einer reinen Bestandsbildung per Freihandzeichnung ist eine Frage der erwünschten Genauigkeit. Während freie Zeichnungen für viele Anwendungsfälle ausreichend genaue Flächengrößen liefern, können diese nicht den Anforderungen für die „Festlegung des Steuerlichen Nutzungssatzes“ [§ 34 B Abs. 4 Nr. 1 EStG] genügen, wie es für Waldbesitzgrößen ab ca. 30 ha der Fall ist. Für niedrigere Besitzgrößen ist die Besteuerung auf der Grundlage einer Betriebsplanung nicht vorgeschrieben, kann aber freiwillig anstatt eines pauschalen Nutzungssatzes von 4,5 fm/ha und Jahr [R 34B.2 Abs. 2, EStR 2005] herangezogen werden.

Die Kombination aus der Kopie von vorhandenen Außenlinien und der

Freihandzeichnung von Innengrenzen kann bei mittleren bis größeren Besitzgrößen zu einer Zeitersparnis führen.

Im Falle eines komplexen Nutzungsmodelles der WebGIS-Anwendung (siehe Seite 57) kann die Freihandzeichnung zudem als erste Stufe und das Kopieren von WFS-Geometrien als fortgeschrittene Stufe betrachtet werden.

3.1.5 Datenquellen und Ebenen

Die Karte im Netz wird mit einem Luftbild als Basis-Ebene konfiguriert, worauf die verschiedenen Vektor-Ebenen gelegt sind (s. Tab. 6).

Ebene	Typ	Quelle
Google Map Satellite	Base layer	Google API v3.6
Google Map Hybrid	Base layer	Google API v3.6
Flurstücke	Vektor	WFS-Dienst des LVG
Bestände	Vektor	PostGIS-Datenbank auf dem Applikationsserver

Tab. 6: Kartenthemen

Google Maps wird aufgrund der relativ hohen Auflösung im Nahbereich als Basis-Ebene verwendet, kann aber theoretisch durch einen beliebigen WMS-Dienst ersetzt werden. Die Erweiterung durch zusätzliche Ebenen bzw. Themen ist bei einer Erweiterung des Funktionsumfanges der Anwendung ebenfalls denkbar und leicht zu bewerkstelligen.

3.2 Architektur

Die Architektur der Anwendung ist die Dokumentation der umgesetzten Lösungen aus der Anforderung. Im Folgenden werden die für die Untersuchung relevanten Softwarelösungen vorgestellt.

3.2.1 Allgemeine Grundsätze

Aufgrund limitierter Geldmittel und vorhandener Kenntnis über bestehende Software wird der Prototyp ausschließlich mit Open-Source Software (OSS) entwickelt. Der überwiegende Anteil an gängiger WebGIS-Client-Software sowie Geodatenserver-Software steht ohnehin unter freien Lizenzen im Netz bereit, wodurch die Überlegung hin zur freien Software zusätzlich erleichtert wird.

Der WFS-Dienst an sich sowie eingesetzte Datenübertragungsprotokolle unterliegen internationalen Standards. Um zukünftigen Weiterentwicklungen nicht im Wege zu stehen, soll daher bei der Webanwendung besonderes Augenmerk auf die Einhaltung von Standards sowie gängige Webprogrammierpraxis geachtet werden.

Nach BERNARD ET AL. (2005) sind Standards bei Software Systemen nötig, um einer Abhängigkeiten durch Software vorzubeugen, die gängige Standards nicht verwenden.

Die Webapplikation unterliegt einer Client-Server-Architektur. Damit verbunden sind verschiedene Datenübertragungsebenen mit dazugehörigen Übertragungsprotokollen sowie das Prinzip einer verteilten Architektur, wo verschiedene Ressourcen räumlich getrennt liegen und mittels Fernübertragungstechnik miteinander kommunizieren müssen.

3.2.2 Grundlegende Konzepte

Hier werden die zu verwendenden Technologien erläutert und im Kontext der Anwendung besprochen. Neben der Beschreibung der Hard- und Softwarekomponenten an sich wird auch die Kommunikation untereinander vorgestellt.

3.2.2.1 Kommunikation

HTTP

Die Kommunikation des Clients mit dem Server und umgekehrt findet per HTTP-Protokoll statt, das TCP/IP als Transport-Ebene benutzt und die Befehle POST und GET kennt.

Eine Anfrage an den Server wird „Request“ genannt und beinhaltet den Befehlsnamen – meist GET, eine URL mit oder ohne Parametern, sowie einen Request-Header mit Informationen über den Benutzer und die Clientanwendung.

Wird eine Antwort des Servers zurückgeschickt, spricht man von einer „Response“, welche Daten in Form von HTML, Bildern, JavaScript oder CSS im Request-Body übermittelt. Auch hier wird stets ein HTTP-Header übermittelt. Darin können Informationen für den Client enthalten sein, wie z.B., ob die Daten in einem Pufferspeicher („Browser Cache“) abgelegt werden sollen – d.h. wie lange die Daten gültig sind.

AJAX

Unter AJAX versteht man das asynchrone Absetzen von Requests, um bei einem Daten-Update nicht die komplette Internetseite neu laden zu müssen. Diese Technologie ist eine Grundlage für Karten-Clients wie OpenLayers, die auf eine möglichst flüssige Darstellung von Karteninhalten und ein dynamisches nachladen neuer Hintergrundkacheln angewiesen sind.

In Abb. 6 sowie Abb. 7 werden alle JavaScript-Funktionen aufgelistet, die direkt oder indirekt eine AJAX-Anfrage absetzen und eine HTTP-Kommunikation auslösen.

Bei der Flurstücksverwaltung (Abb. 6) treten zwei Server und ein Client in Aktion. Zunächst wird der WFS-Dienst mittels Filterencoding abgefragt, um eine Auswahl oder ein bestimmtes Flurstück zu erhalten. Danach tritt der Client verstärkt mit dem Applikationsserver in Verbindung und falls keine lokale Speicherung der Geometrie (s. 3.1.2.1, Seite 28) erfolgen soll, wiederholt mit dem WFS-Dienst.

Die wesentlichen Kommunikationsbausteine der Flurstücksverwaltung aus Abb. 6 im einzelnen:

- ① Absetzen der Query zum WFS-Dienst und Antwort als Feature
- ② Speichern des Flurstücks als Liste mit ID-Nummern
- ③ Optionale Speicherung der Geometrie im App-Server
- ④ Holen aller Flurstücke, die im Eigentum sind
- ⑤ Zurücksenden der Flurstücke sofern Geometrien vorhanden sind
- ⑥ Holen der Geometrie vom WFS-Dienst
- ⑦ Hinzufügen der Geometrie zur Flurstücksliste

Die wesentlichen Kommunikationsbausteine der Waldflächenverwaltung aus Abb. 7 im einzelnen:

- ① Abrufen der Kartenebenen beim erstmaligen Initialisieren der Karte inklusive Eigentumsgrenzen und bereits kartierte Waldstücke sofern vorhanden
- ② Abrufen der Eigentumsgrenzen (s. Schritt 4 bis 7 in Abb. 6)
- ③ Feature speichern

Wichtig ist hierbei zu verstehen, dass das Zeichnen eines neuen Features komplett auf dem Client passiert. Erst wenn die Bearbeitung des Features durch Doppelklick oder Bestätigung in einem Dialogfeld abgeschlossen ist, wird der Befehl `saveFeature()` aufgerufen und ein AJAX-Befehl abgesetzt.

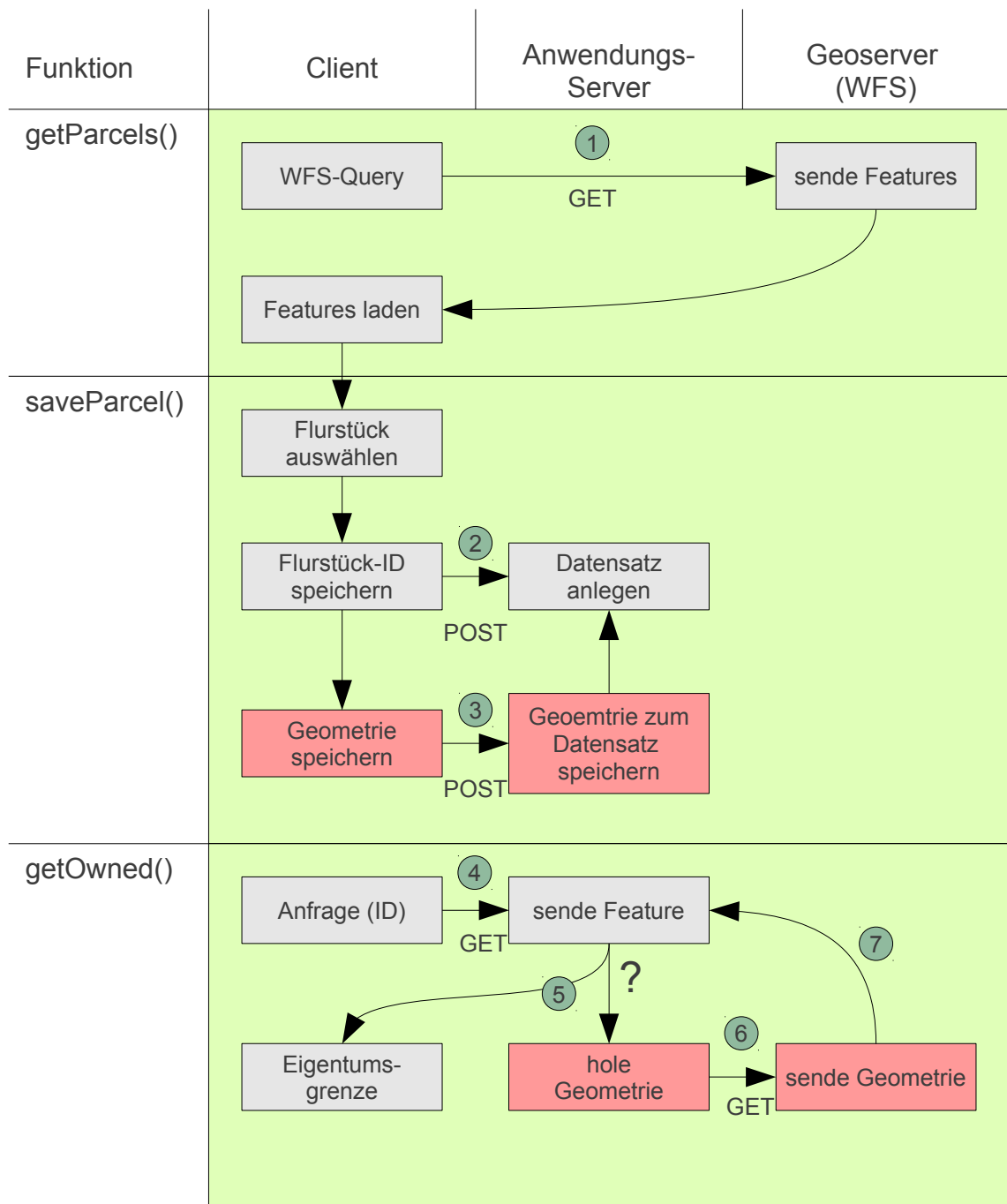


Abb. 6: Client-Server Kommunikation beim Anlegen der Eigentumsgrenzen

Rot markiert sind die Aktionen, die im Falle keiner serverseitigen Geometrie-Speicherung zum Tragen kommen.

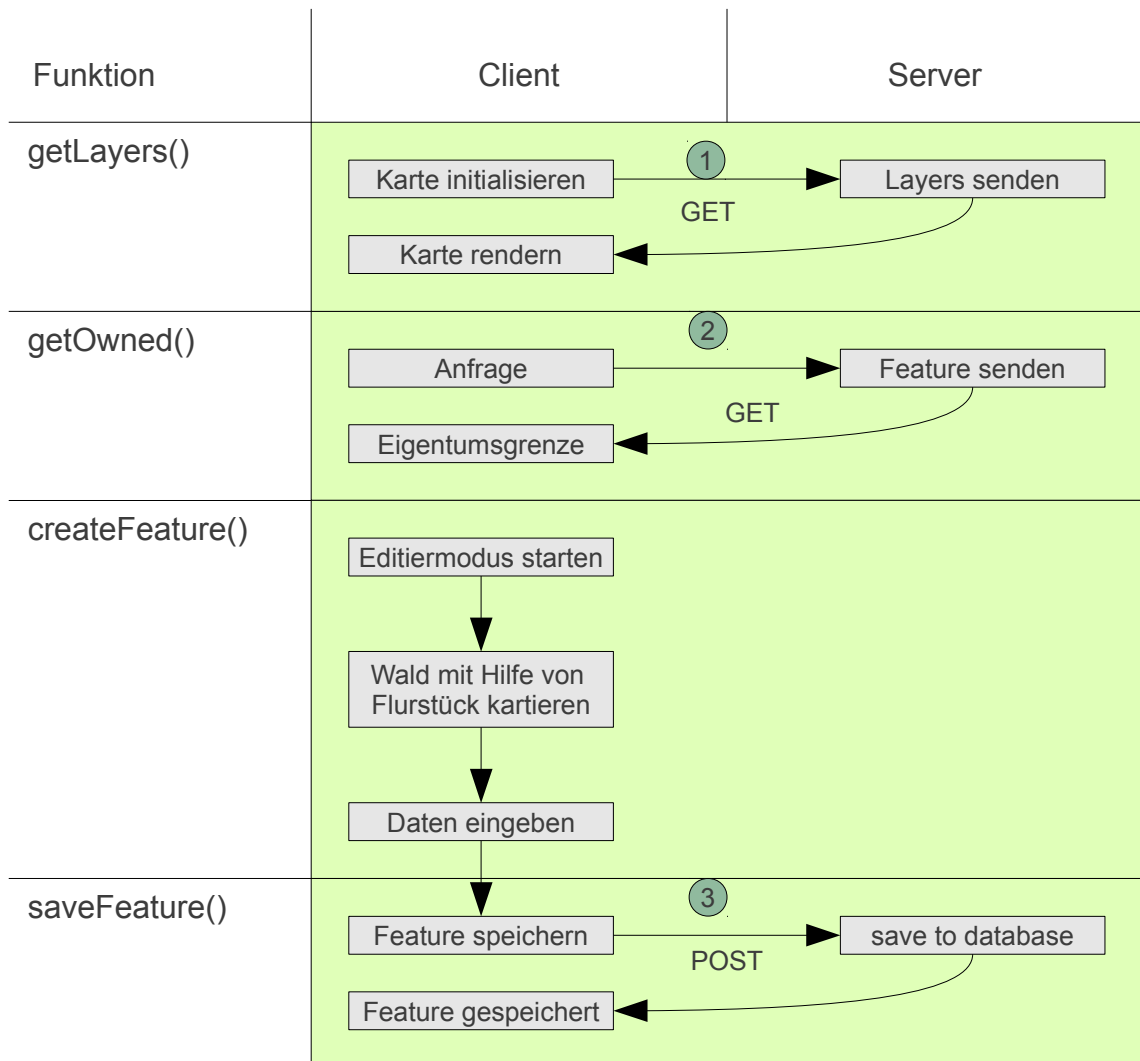


Abb. 7: Client-Server Kommunikation beim Anlegen von Waldbeständen

3.2.2.2 Serverseitige Komponenten

Ubuntu Server 10.04 mit Apache

Als Betriebssystem ist ein Ubuntu Server in der Version 10.04 installiert. Ubuntu wird standardmäßig mit einem Apache Web-Server ausgeliefert, auf dem alle Webdienste gehostet werden. Als Betriebssystem des Clients kommt Ubuntu 11.04 in der Standardversion zum Einsatz. Der Server wird mittels Virtualisierung (in diesem Fall „Oracle VM VirtualBox“¹⁶) innerhalb des Clients ausgeführt, um auf zusätzliche Hardware während der Anwendungsentwicklung verzichten zu können.



¹⁶ <https://www.virtualbox.org/> (20.04.2012)

GeoServer

Der von OpenGeo entwickelte Geoserver¹⁷ bietet verschiedene Geodaten-Dienste an und kann als Proxy-Server Dienste anderer Domains lokal zur Verfügung stellen. Alle gängigen OGC-Dienste, eine große Liste an Dateiformaten sowie Daten-Caching wird unterstützt.

GeoDjango

GeoDjango ist die Bezeichnung für eine geographische Erweiterung von Django, einem auf Python basierenden serverseitigen Web-Framework. Django stellt eine umfangreiche Benutzerverwaltung zur Verfügung und kann die Anwendungsdaten unter anderem in einer PostGIS-Datenbank verwalten, was ideal für WebGIS-Anwendungen geeignet ist, in denen Benutzer ihre persönlichen Geodaten verwalten möchten.

Mittels GeoDjango kann sichergestellt werden, dass jeder Benutzer nur auf die eigenen vertraulichen Daten zugreifen, auf Wunsch jedoch diese auch für andere Benutzer freigeben kann.

GDAL

GDAL¹⁸ steht für „Geospatial Data Abstraction Library“ und unterstützt als Python-Erweiterung in Kombination mit „OGR“ GeoDjango beim Lesen von Simple Features und bei der Konvertierung verschiedenster geographischer Datenformate.

GEOS

GEOS¹⁹ bezeichnet die „Geometry Engine - Open Source“, eine in C++ geschriebene und der „Java Topology Suite“ entlehnte Library. Durch die GEOS API ist es möglich, Geometrien aus verschiedenen Formaten einzulesen, zu verändern und räumlich zu analysieren. Verschiedenste räumliche Operatoren, wie sie auch vom OGC [OGC 99-049 (1999)] spezifiziert sind, kommen zur Anwendung.

3.2.2.3 Clientseitige Komponenten

Folgende Komponenten wurden nach eingehender Recherche und aufgrund ihrer Kompatibilität untereinander ausgewählt und in die Anwendung eingebunden.

¹⁷ <http://geoserver.org> (06.04.2012)

¹⁸ <https://docs.djangoproject.com/en/dev/ref/contrib/gis/gdal/> (07.04.2012)

¹⁹ <https://docs.djangoproject.com/en/dev/ref/contrib/gis/geos/> (07.04.2012)

ExtJS

ExtJS in der Version 3.4.0 ist ein klassisches JavaScript-Framework, das den grundlegenden JavaScript-Umfang um nützliche Funktionen erweitert und hilft, Code zu reduzieren. Die Stärke von ExtJS liegt jedoch in der Möglichkeit, ein Web-basierendes User-Interface aufzubauen, das stark an Desktop-Anwendungen erinnert und damit eine hohe Benutzerfreundlichkeit verspricht.

GeoExt

GeoExt²⁰ in der Version 1.1 erweitert ExtJS um ein Kartenfenster, verschiedene Controls und Schnittstellen zu geographischen Datenformaten. Des weiteren sind bestimmte räumliche Operationen wie Verschneidung und Überlappung auf der Client umgesetzt.

OpenLayers

OpenLayers²¹ in der Version 2.11 bettet allein mit Hilfe clientseitiger Technologien wie HTML, SVG, JavaScript und CSS eine dynamische Karte in den Browser ein. Die Vorteile von OpenLayers gegenüber anderen Kartenrenderern ist die enge Kompatibilität zu OGC-Diensten, sowie die Unabhängigkeit von serverseitigen Technologien.

JSTS

Die JavaScript Topology Suite²² in der Version 0.12.0 ist eine Portierung der Java Topology Suite (JTS) und übernimmt die Geo-processing Operationen, wie das „Splitting“ und „Merging“ von Geometrien auf dem Client, anstatt auf dem Server, wodurch auf den Versand von Geometrie-Informationen per AJAX verzichtet werden kann.

Proj4js

Zur Umrechnung von Koordinaten verschiedener räumlicher Referenzsysteme steht clientseitig die Bibliothek Proj4js²³ zur Verfügung. Proj4js kann leicht in OpenLayers eingebunden werden und ermöglicht eine relativ performante „On the Fly“-Darstellung.

3.2.3 Modularer Aufbau

Die Anwendung kombiniert verschiedenste Komponenten mittels standardisierter Schnittstellen. Die Datenquellen sind dabei unabhängig von der Applikationslogik und der Präsentation in der Anwendung (vgl. Abb. 8). Dadurch bleibt das System erweiterbar

²⁰ <http://geoext.org/> (20.04.2012)

²¹ <http://www.openlayers.org/> (20.04.2012)

²² <https://github.com/bjornharrtell/jsts> (21.04.2012)

²³ <http://trac.osgeo.org/proj4js/> (20.04.2012)

sowie skalierbar.

Im oberen Teil der Grafik neben der Google-API, welche die Basis-Karte beisteuert, stehen die Client-Komponenten mit den JavaScript-Bibliotheken zur Konstruktion der graphischen Benutzeroberfläche samt Karte sowie der Applikationslogik.

In der unteren Hälfte steht die Serverseite, die durch den Geoserver des LVG in München, dem Anwendungsserver mit GeoDjango und der Anwendungsdatenbank und einem zusätzlichen Geoserver repräsentiert wird.

GeoDjango in Kombination mit GDAL und OGR vermag die für die Anwendung benötigten Geodaten selbst aus der PostGIS-Datenbank auszulesen und ohne Hilfe von Geoserver am Client bereitzustellen.

Der zusätzliche Geoserver auf der Anwendungsseite (rechte Seite in Abb. 8) ergänzt die Anwendung um die Möglichkeit, sekundäre Karten aus eingegebenen Benutzerdaten zu generieren, und dient außerdem als Proxy für weitere Kartendienste.

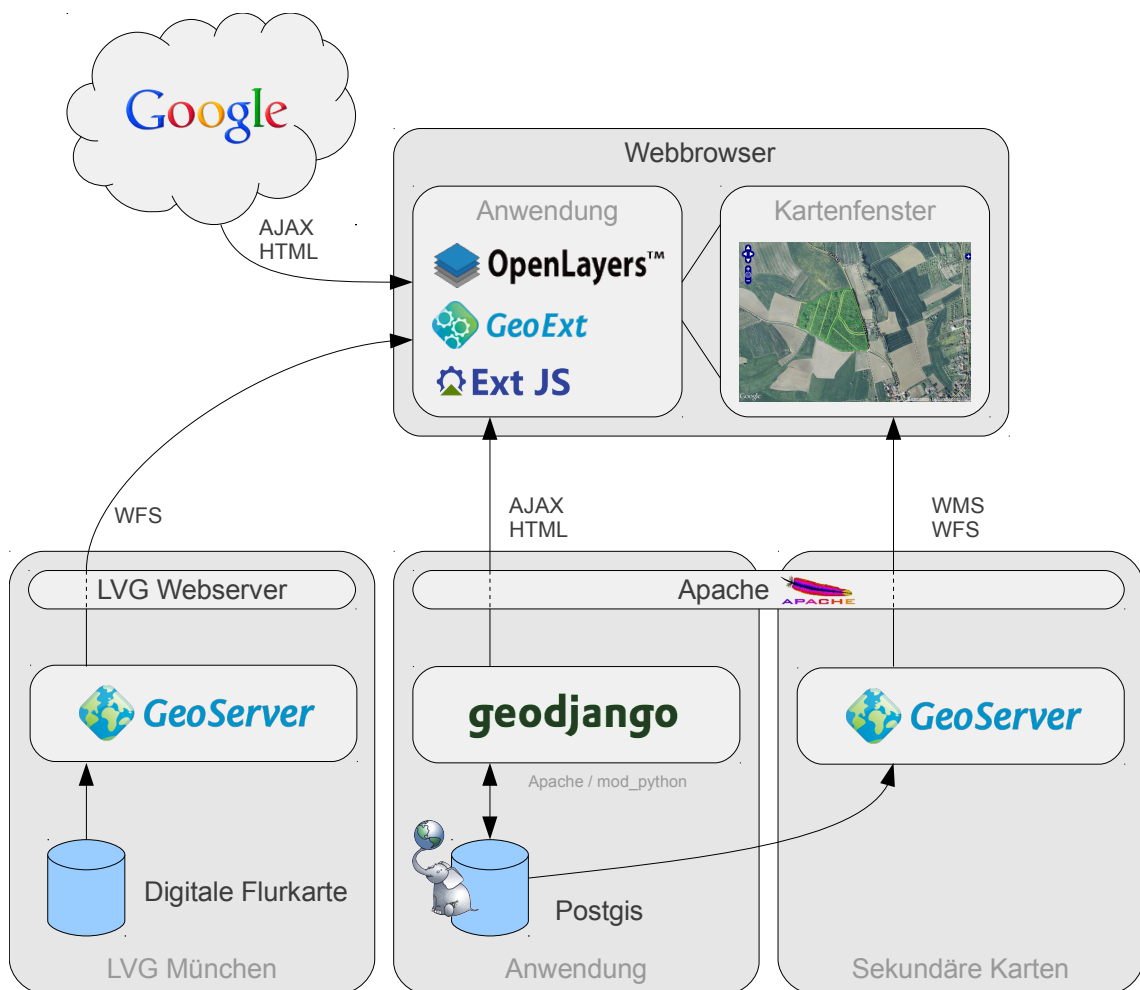


Abb. 8: Systemarchitekturskizze - Modularer Aufbau

3.2.4 GIS-Funktionen

Im Folgenden werden die Funktionen zur Bearbeitung von Waldflächen in der Anwendung vorgestellt. Sonstige Funktionen, die nicht mit der Manipulation von geographischen Daten in Verbindung stehen, werden hier nicht besprochen.

3.2.4.1 Neuen Bestand anlegen

Bestände sind die kleinste Planungseinheit im Wald und werden durch Polygone repräsentiert. Die Anlage eines Bestandes setzt daher die Anlage eines Datensatzes voraus, der sowohl räumliche wie auch nicht-räumliche Werte enthält. Folgender Arbeitsschritt beschreibt die Anlage eines Polygons, zunächst durch die Anfertigung einer Zeichnung oder durch Übernahme aus dem Flurstücks-Layer. Die Eingabe der Daten ist zunächst freiwillig und kann später ergänzt werden (s. Abb. 9).

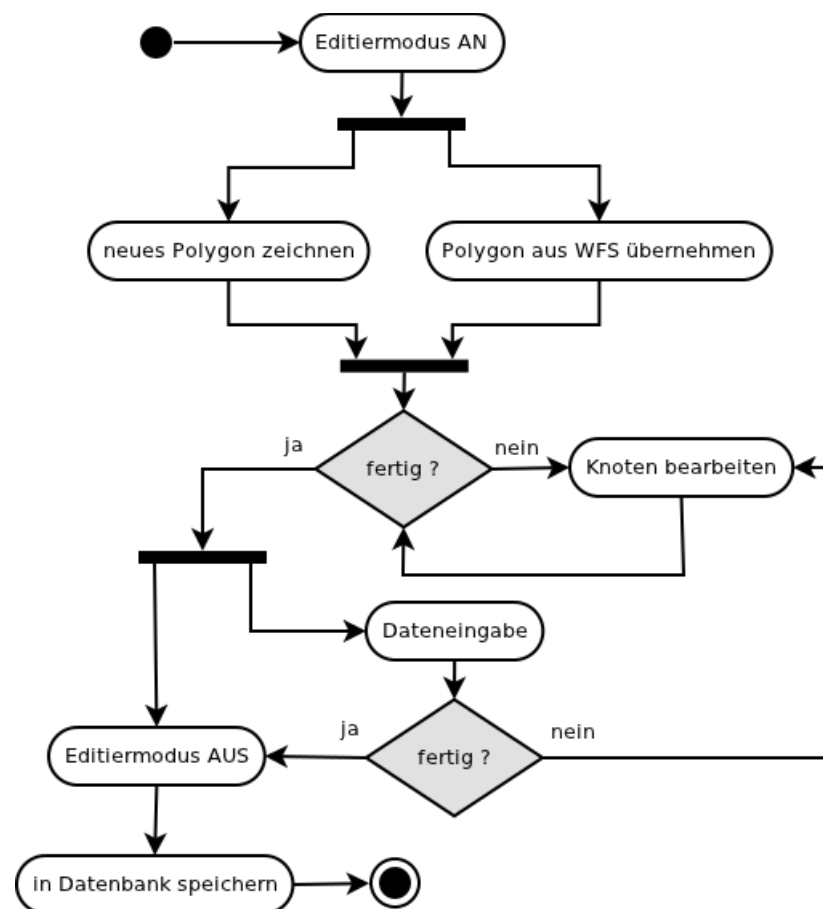


Abb. 9: Aktivitätsdiagramm "Bestand anlegen"

Die Eingaben werden lokal auf dem Client vorgenommen. Die Anlage eines neuen Datensatzes in der Datenbank erfolgt erst ganz am Ende des Arbeitsschrittes, nachdem die Editiersitzung durch Bestätigung in einem Dialogfeld beendet und der Änderung zugestimmt wurde.

3.2.4.2 Existierenden Bestand bearbeiten

Nachdem ein Bestand angelegt ist, können sowohl die Geometrie als auch die Bestandsattribute geändert werden. Dieser Arbeitsschritt ähnelt der Anlage eines neuen Bestandes (s. Abb. 10) mit dem Unterschied, dass die Änderung von Werten und die Änderung der Geometrie optional sind.

Die Änderung von Werten oder Knoten kann so oft wie gewünscht durchgeführt werden. Der Wechsel zwischen der Bearbeitung der Geometrie und von Bestandswerten ist möglich.

Auch hier wird die Änderung erst wirksam, nachdem die Editiersitzung durch Bestätigung

eines Dialogfeldes erfolgt und ein POST-Request an den Server verschickt wurde.

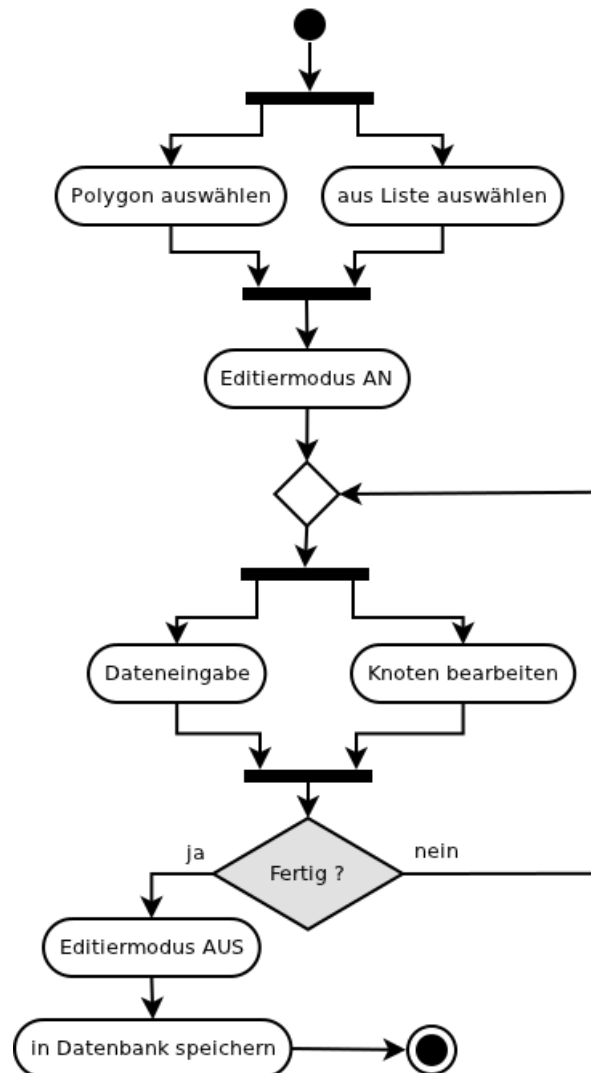


Abb. 10: Aktivitätsdiagramm "Bestand bearbeiten"

3.2.4.3 Bestand Löschen

Eine simple aber notwendige Operation ist das Löschen eines Bestandes, das zunächst eine Auswahl entweder im Kartenfenster oder der Bestandsliste bedarf (Abb. 11).

Der Benutzer bestätigt das endgültige Löschen des Datensatzes in einem Dialogfeld, wodurch eine Mitteilung an den Server zur Entfernung des Datensatzes erfolgt.

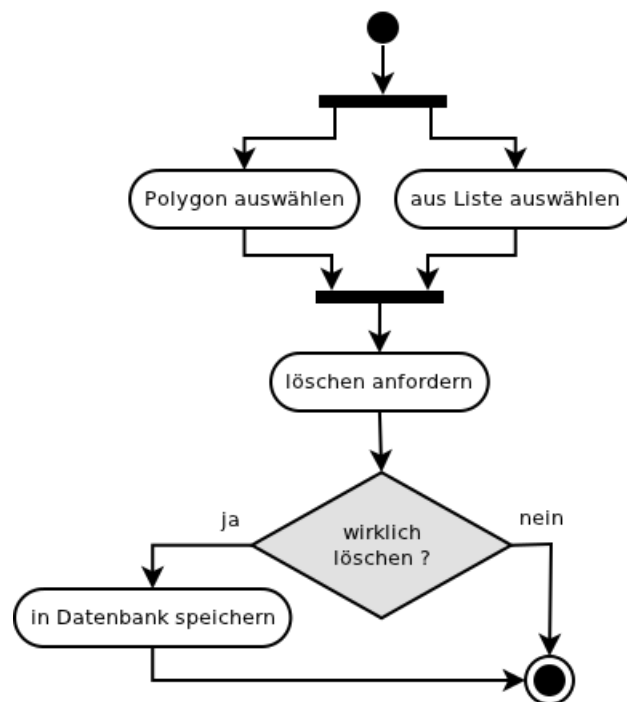


Abb. 11: Aktivitätsdiagramm "Bestand löschen"

3.2.4.4 Bestände vereinigen

Ein erweitertes Planungstool stellt die Möglichkeit der Verschmelzung oder Vereinigung zweier Bestände dar. Hierdurch ist es möglich, mehrere Detailplanungen in einzelne Behandlungseinheiten zusammenzufassen. Dabei wird nicht nur die Geometrie mittels UNION-Operation verschmolzen, es müssen auch die Attributdaten aggregiert und aufsummiert werden.

Bevor das Verschmelzen ausgeführt werden kann, müssen mehrere Plausibilitätsüberprüfungen durchlaufen werden. Dabei muss festgestellt werden, ob die zwei Flächen sich berühren, sich überlappen oder keine von beiden Relationen aufweisen. Bei überlappten Bereichen muss vor der Aufsummierung der Attributwerte eine flächenanteilige Verrechnung durchgeführt werden (s. Abb. 12).

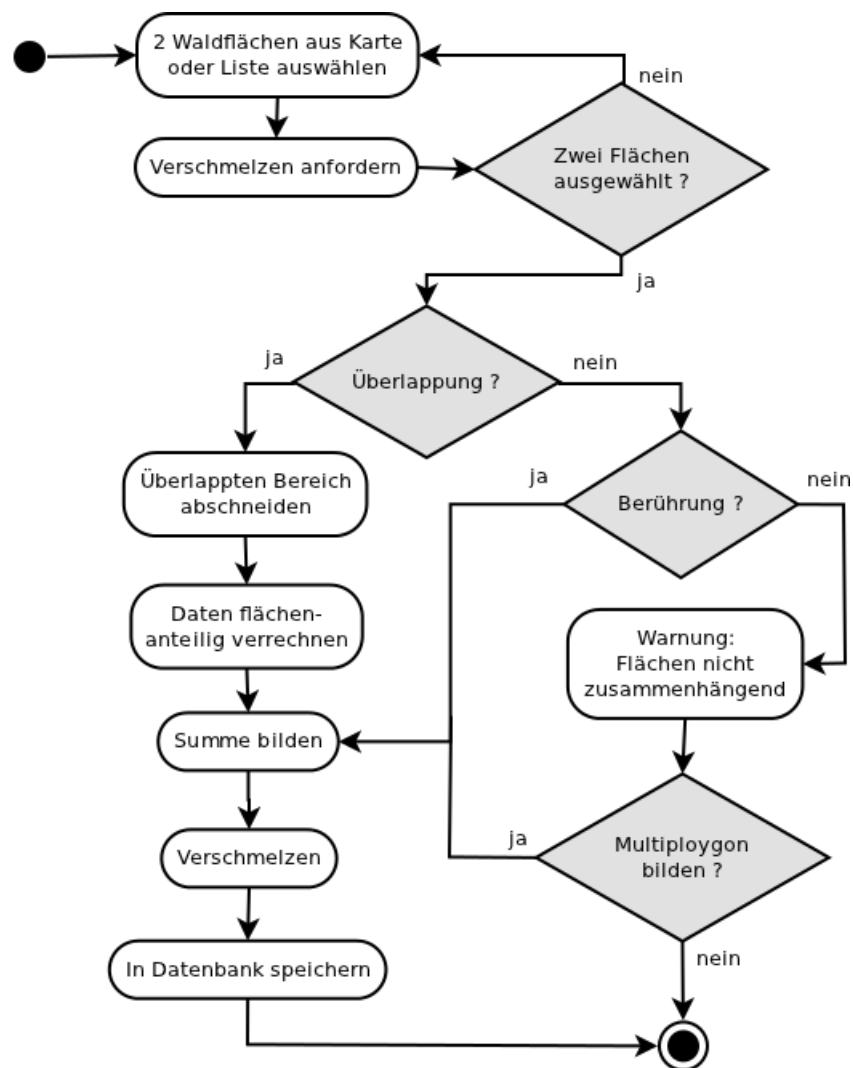


Abb. 12: Aktivitätsdiagramm "Bestände vereinigen"

3.2.4.5 Bestände aufteilen

Die Aufteilung von Beständen in kleinere Behandlungseinheiten ist eine Möglichkeit, von einer Grobplanung zu einer Detailplanung zu wechseln. Dabei können Bestandsattribute zwar ähnlich wie bei der Verschmelzung (s. Abb. 13) verrechnet werden, allerdings sind die Werte im Bestand wahrscheinlich nicht mehr repräsentativ. Lediglich für den gesamten Wald ändern sich die Daten nicht.

Die Aufteilung von Beständen kann entweder durch Sprengen von Multipolygonen oder durch Zeichnen einer Trennlinie erfolgen. Die Änderungen werden erst wirksam nach Bestätigung eines Dialogfeldes und dem Hinweis auf die möglicherweise nicht mehr repräsentativen Werte.

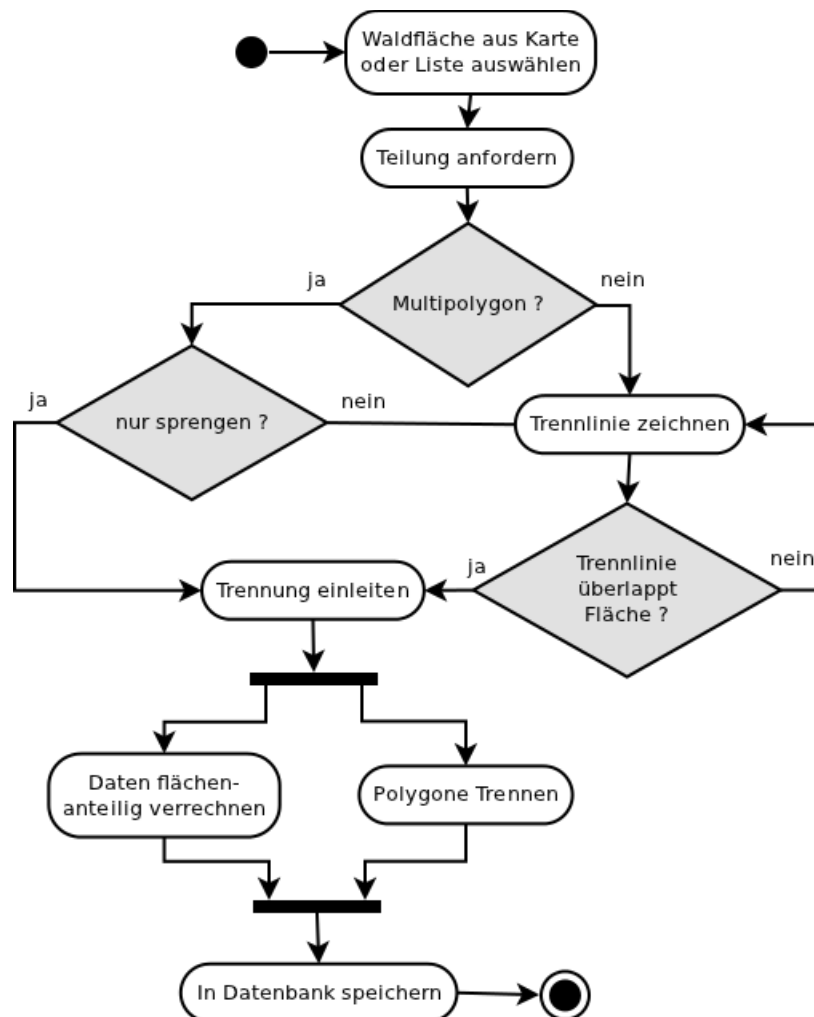


Abb. 13: Aktivitätsdiagramm "Bestand teilen"

3.2.5 Datenmodell

Das Datenmodell ist in der Prototyp-Anwendung sehr einfach gehalten, um zunächst nur die Grundfunktionalitäten zu implementieren. Drei Tabellen werden angelegt.

Eine Benutzerliste für die Zugangsberechtigung auf die Applikation existiert standardmäßig in der Geodjango-Installation und muß lediglich um eigene Felder, wie Adresse oder Name, erweitert werden.

Eine Tabelle speichert alle Waldbestände inklusive einer Auswahl vom Benutzer editierbaren Walddaten sowie deren Geometrie.

Pro Waldbestand können mehrere Aufnahmelisten eingegeben werden – bspw. eine

Messung für jede vorkommende Baumart.

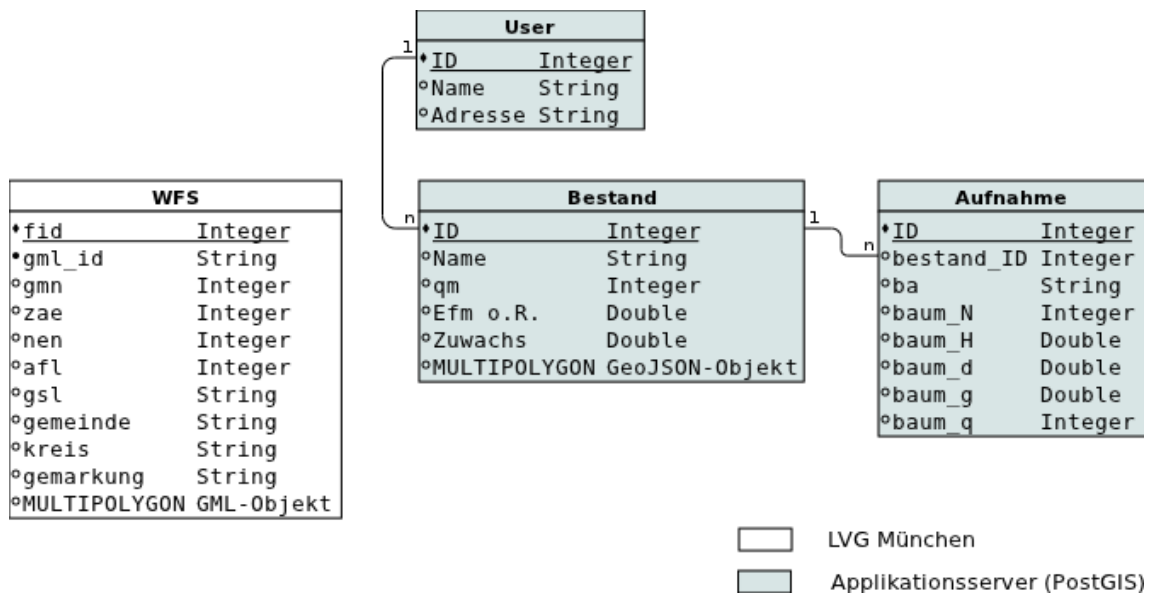


Abb. 14: Datenmodell

Ergänzend ist das Modell des WFS-Dienstes in Abb. 14 gegenübergestellt. Entscheidend hierbei ist, dass keine Verbindung auf Datenbankebene zwischen den Beständen und den Flurstücken existiert, da die Bestände zwar die Geometrie oder ein Teil der Geometrie erben, aber nach deren Anlage nicht mehr logisch verknüpft sind.

Sollte für die Bestände in einem späteren Zeitpunkt ermittelt werden, welche Flurstücke ihnen zugrunde liegen, so kann dies über eine räumliche Verschneidung auf einfache Art client- oder serverseitig gelöst werden.

4 Ergebnisse

Nachdem die Anforderungen und die Architektur der Anwendung spezifiziert worden sind, werden die zu Beginn festgehaltenen Fragestellungen untersucht und die Ergebnisse eingehend erläutert.

4.1 Umsetzung

Der Prototyp wurde in einem Zeitraum von 6 Monaten entwickelt, wovon die hauptsächliche Arbeit darin bestand, die verwendeten Komponenten zu verstehen und kennenzulernen. Die eigentliche prototypische Anwendung konnte schließlich in weniger als 3 Wochen mit den beschriebenen Funktionalitäten fertiggestellt werden.

Schwierigkeiten

Eine Schwierigkeit bei der Visualisierung der Flurstücke aus dem WFS-Dienst war die Unverträglichkeit des Koordinatensystems mit WGS84. Seitens des LVG ist derzeit keine Ntv2-Transformation möglich, da der Geoserver in der jetzigen Version dies noch nicht unterstützt.

Clientseitig war es ebenso mittels Proj4js nicht möglich unter Verwendung einer sogenannten „Gridshift“-Datei eine Umwandlung zu WGS84 zu erzielen. Diese Funktion wird laut dem Hauptentwickler²⁴ in naher Zukunft auch nicht in Proj4js umgesetzt werden. Als Work-Around wurde schließlich eine konstante Verschiebung von 95 Metern nach Osten und 125 nach Norden für die Koordinaten hinzugerechnet.

4.2 Realisierbarkeit

A. Die Webanwendung kann gänzlich mit OpenSource-Software umgesetzt werden.

Diese Hypothese kann gänzlich als wahr eingestuft werden. Sämtlicher von der eingesetzten Software verwendete Code ist im Netz veröffentlicht und entspricht daher gängigen Open-Source Modellen, wie beispielsweise dem „Creative Commons Share Alike“ (CC-SA) oder „Gnu General Public License“ (GPL).

²⁴ Richard Greenwood (www.greenwoodmap.com), Email vom 20.11.2012

Solange die Anwendung auf nicht kommerzieller Ebene betrieben wird und die verwendeten Lizenzen in der Anwendung genannt werden, müssen keine Nutzungsgebühren entrichtet werden. Bis auf zwei Komponenten sind alle freien Lizenzen auch kommerziell ohne Bezahlung nutzbar.

Die erste Lizenz bezieht sich auf den Datendienst von „Google Maps“, der im Falle einer kommerziellen Nutzung je nach Zugriffsfrequenz auf eine kommerzielle Lizenz umgewandelt werden muss.

Die zweite Softwarekomponente, welche kommerzielle Nutzungsmodelle von freien Modellen unterscheidet, ist Ext JS. Diese kann jedoch auch im Falle einer kommerziellen Nutzung kostenlos betrieben werden, wenn der Code der Anwendung mittels „GNU GPL license v3“ veröffentlicht wird.

Da die endgültige Frage, ob die Anwendung mit einer Gewinnabsicht betrieben wird und damit kommerzielle Ziele verfolgt, noch nicht feststeht, muss die Antwort im Einzelfall neu formuliert werden.

4.3 Workflow

B. Der WFS-Dienst unterstützt den Benutzer bei der Erstellung von Forstkarten.

Das „Graphische User Interface“ sowie die Bedienlogik der Anwendung wurde nicht durch andere Personen außer dem Autor getestet, orientiert sich jedoch an bestehenden Desktop-Kartier-Workflows und Webdesign-Standards.

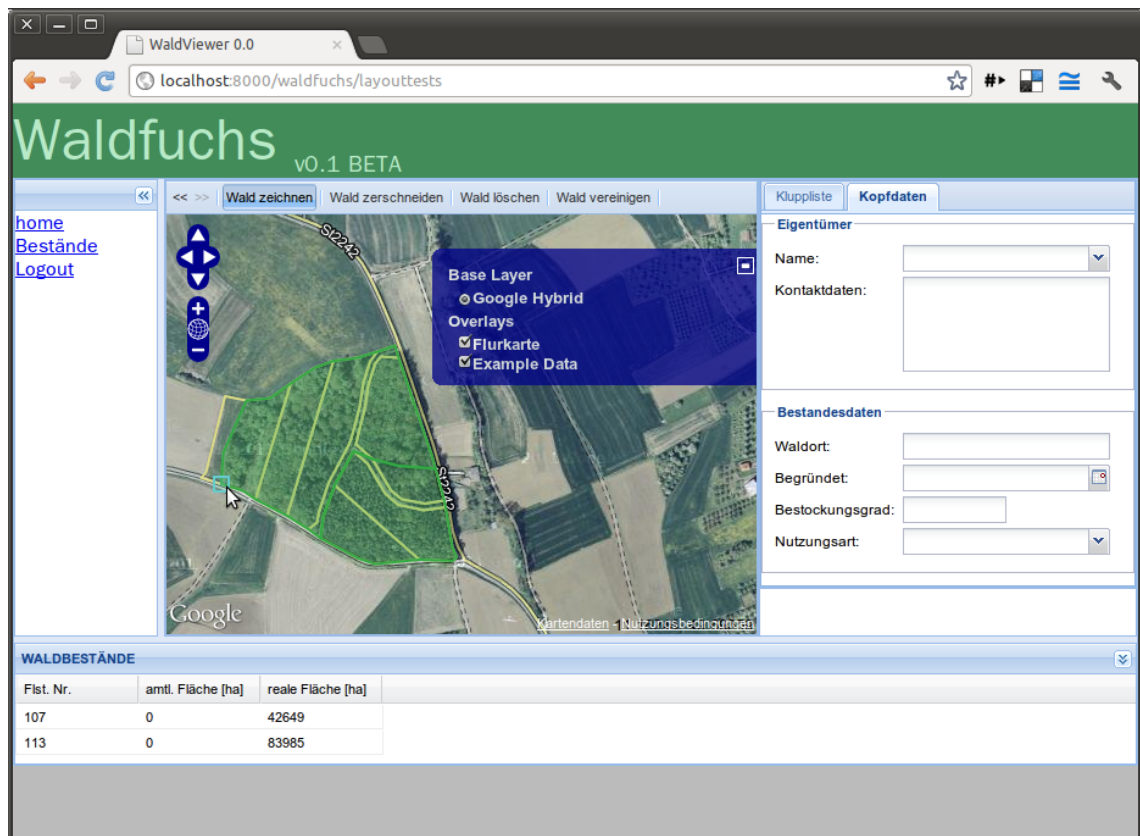


Abb. 15: Screenshot des Prototyps in Google Chrome

Zentrales Element der Anwendung (s. Abb. 15) ist die Karte, die sich dynamisch mit dem Browserfenster verkleinern und vergrößern lässt. Links der Karte befindet sich die ein- und ausblendbare Navigationsleiste. Unterhalb der Karte wurde eine Waldliste platziert, welche die gesamte Browserbreite ausnutzt, um möglichst viele Spalten pro Bestand anzuzeigen. Auf der rechten Seite sind durch Reiter getrennte zusätzliche Angaben pro Bestand einzutragen, wie eine Messliste oder Metadaten.

Direkt oberhalb des Kartenausschnitts sind mehrere Schaltflächen angebracht, welche die in 3.1.3 geforderten Funktionen, wie Zerschneidung, Löschen oder Anlegen von Beständen repräsentieren.

Grundsätzlich erfüllt der Prototyp alle Funktionen aus der Anforderungsspezifikation. Der Ablauf und die Benutzerfreundlichkeit der Arbeitsschritte, die auch als Workflow bezeichnet wird, orientiert sich an dem Prinzip, mit möglichst wenig Aufwand zum gewünschten Ziel zu gelangen.

Folgende Eigenschaften des Benutzerinterfaces, die den Workflow nach Maßgabe der definierten Nutzeranforderungen ermöglichen, können festgehalten werden:

- Snapping-Hilfe beim Editieren in der Karte
- Wechseln zwischen Kartenfenster, Dateneingabefeldern und Tabelle jederzeit möglich
- Automatische Darstellung und Nachladen der Features im sichtbaren Kartenausschnitt
- Ein- und ausblendbare Kartenebenen

Die Frage, ob der WFS-Dienst die Waldkartierung erleichtert, hängt maßgeblich von der Nutzungsbedingung ab. Darf zum Beispiel eine Kopie der Flurstücke in der Anwendungs-Datenbank gespeichert werden, so stellt der Dienst keine Behinderung in einem Workflow dar.

Ist das Kopieren der Geometrie hingegen ausdrücklich untersagt, so muss der Benutzer ohne Snapping versuchen, das Flurstück nachzuzeichnen, was wesentlich längere Zeit beanspruchen würde.

Im vorliegenden Fall besteht keine Nutzungseinschränkung, wodurch der Workflow ohne Einschränkung, wie gefordert, von staten geht.

4.4 Speicherung

C. Es muss keine Kopien der Flurkarte lokal gespeichert werden.

Eine offizielle Nutzungsbedingung für den WFS-Dienst seitens des LVG gibt es noch nicht. Allerdings stellt die Speicherung der Flurstücksgeometrie ähnlich wie in anderen Nutzungsverträgen eine Daten-Vervielfältigung dar. Ist eine Vervielfältigung nicht gewünscht, kann auch auf die Speicherung von Flurkartenelementen verzichtet werden. Im Folgenden werden drei Möglichkeiten dargestellt, in denen eine Vervielfältigung erfolgt bzw. erfolgen kann.

Rolle 1 - „Freihandzeichnung“

Der Waldbestand wird ohne „Snapping“-Funktion mit der Maus gezeichnet und orientiert sich grob an der Flurkarte ohne aber eine genaue Überlappung der Polygonränder zu verursachen. Diese eher ungenaue Variante ist in etwa vergleichbar mit einem GPS-Track, welches der Waldbesitzer durch ablaufen der Grenzsteine aufzeichnet. Hierzu würde ebenso eine Raster-Repäsentation der Flurkarte ausreichen.

Rolle 2 - „Kopie“

Einzelne Flurstücke werden aus dem WFS-Dienst zur Bildung von Polygonen in einem separaten Layer kopiert, der wiederum serverseitig in einer Datenbank abgespeichert wird. Wird die Geometrie nicht verändert, kann man von einer 1:1 Kopie der Geometrie auf dem App-Server sprechen (vgl. Abb. 16).

In der Praxis wird die 1:1-Kopie häufig nur eine temporäre Rolle innerhalb eines Kartier-Workflows spielen, da sich Flurstücke nur dann mit dem Bestand decken, wenn es sich um inselhaft Bestände innerhalb geschlossener Waldgebiete handelt.

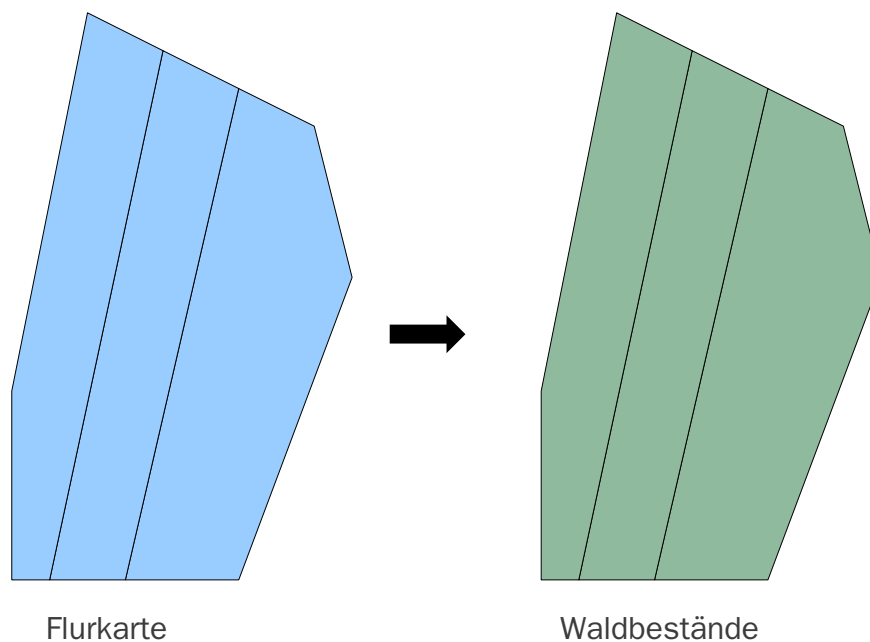


Abb. 16: Flurstücke kopieren

Rolle 3 - „Kartierunterlage“

Der WFS-Dienst wird ähnlich wie beim „Tracing“ nur als Unterlage zur freihändigen

Zeichnung von Polygonen verwendet (vgl. Abb. 17). Da die Kartierunterlage eine echte Vektorgrafik darstellt, kann hier das „Snapping“ eingesetzt werden, um eine genaue Deckung mit der Flurkarte zu ermöglichen. Inwieweit ein Flurstück genau nachgefahren wird, obliegt dem Kartierer.

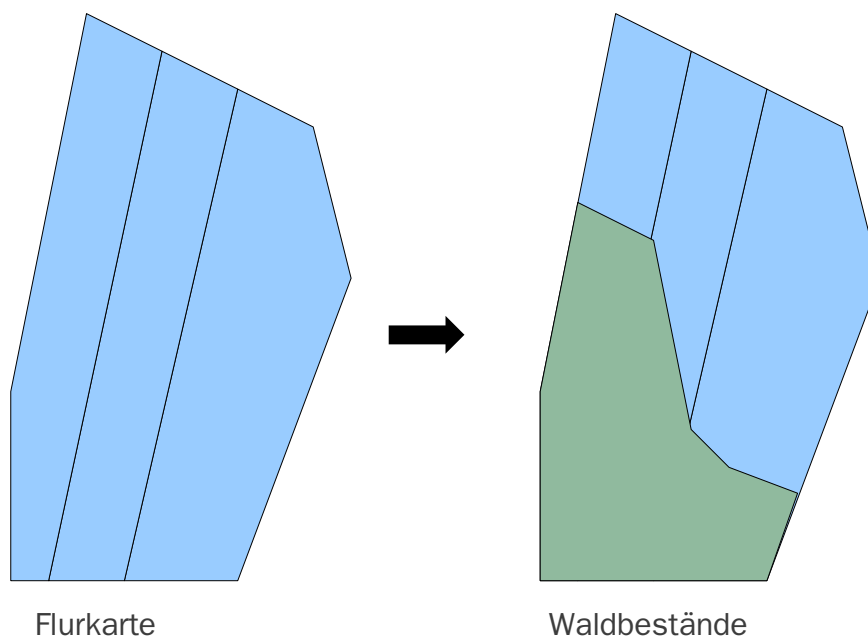


Abb. 17: Flurstücke als Kartierunterlage

Die Bedeutung der vorgestellten Rollen lässt sich vergleichen mit dem Begriff des „Derivative Work“ in der Creative Commons Lizenz²⁵, wie sie bei OpenStreetMap eingesetzt wird.

Solange der Waldbestand in der Anwendung ohne Kartierunterlage kartiert wurde, handelt es sich nicht um eine Kopie und unterliegt nicht der Lizenzvereinbarung.

Gibt es jedoch mindestens eine sich genau überdeckende gemeinsame Linien oder Punkt, dann ist die Rede von einer abgeleiteten Arbeit (Engl.: „Derivative Work“), welche einer Kopie ähnelt und damit der Lizenzvereinbarung unterliegt. Eine detaillierte Gegenüberstellung liefert die Tabelle 7.

²⁵ <http://creativecommons.org/> (03.04.2012)

	Kopie	Ableitung von Original	Abhängig vom Original	Snapping
Rolle 1 Freihandzeichnung	nein	nein	bedingt	nein
Rolle 2 Kopie	ja	ja	ja	nein
Rolle 3 Kartierunterlage	nein	ja	ja	ja

Tab. 7: Rollen der Flurkarte beim Erstellen von Waldbeständen

4.5 Missbrauch

D. Ein Missbrauch des WFS-Dienstes kann ausgeschlossen werden.

Da mit den Flurstücksgrenzen im WFS-Dienst keine personenbezogenen Detailinformationen mitgeliefert werden, besteht zunächst keine Sorge über mögliche Verletzung des persönlichen Datenschutzes seitens des LVG.

Allerdings wird durch die Webanwendung die Möglichkeit gegeben, persönlich relevante Daten einzugeben, die Rückschlüsse über Vermögenswerte oder forstwirtschaftliche Einkünfte zulassen.

Im Internet bestehen mehrere Möglichkeiten der Datenspionage, jedoch auch verschiedene Lösungen, um diese zu unterbinden. Im Folgenden seien drei der wichtigsten Sicherheitsmechanismen im Internet vorgestellt, die auch bei der Webanwendung umgesetzt sind.

- **SSL-Verschlüsselung**

Bereits bei der Datenübermittlung vom und zum Server ist eine Verschlüsselung der Datenpaketinhalte dringend angeraten. „Secure Sockets Layer“ oder auch kurz SSL²⁶ stellt eine im Internet gängige Verschlüsselung bereit, die von Django und allen gängigen Browsern unterstützt wird.

²⁶ http://de.wikipedia.org/wiki/Transport_Layer_Security (01.04.2012)

- **Sessions**

Bei jedem erfolgreichen Login eines Users durch Eingabe von Benutzernamen und Passwort erzeugt der Server eine temporäre Session-ID, die an den Client übermittelt wird. Diese ID wird nun zusammen mit jedem clientseitigen Befehl, wie beispielsweise bei der Übermittlung von Formulareingabedaten, im HTTP-Kopf inkludiert und autorisiert somit den Benutzer zusätzlich am Server²⁷. Um Session-Hijacking zu verhindern, muss auf eine verschlüsselte Übertragung mittels SSL geachtet werden.

In Django werden Sessions durch Einbinden von Session-Middleware-Klassen auf einfache Art ermöglicht. Als Middleware werden Treuhand-Dienste bezeichnet, die sich logisch zwischen Clients und Servern platzieren, um die Kommunikation zwischen diesen weiterzureichen aber auch zu modifizieren oder zu cachen.

- **CSRF**

Um Attacken von Seiten anderer Domains per Cross-Site-Request-Forgery (CSRF)²⁸ zu verhindern bietet Django einen CSRF-Token an, mit dessen Hilfe zusätzlich zur Session-ID sichergestellt ist, dass die Datenübermittlung vom gleichen Rechner stammt, der sich auch per Login angemeldet hat.

Die Einbindung passiert wie auch beim Session Handling per Einbindung sogenannter Middleware-Klassen.

²⁷ [http://de.wikipedia.org/wiki/Sitzung_\(Informatik\)](http://de.wikipedia.org/wiki/Sitzung_(Informatik)) (01.04.2012)

²⁸ <https://docs.djangoproject.com/en/1.4/ref/contrib/csrf/> (01.04.2012)

5 Diskussion

5.1 Stärken und Schwächen des WFS-Dienstes

Der WFS-Dienst ermöglicht in der Theorie den Zugriff auf Vektordaten in sehr hohem Umfang. Für eine Webanwendung macht dies allerdings nur Sinn für eine begrenzte Anzahl Requests, um die Ressourcen im Browser nicht überzustrapazieren. Darum geht die Empfehlung in Richtung einer eher geringen maximalen Feature-Anzahl pro Request (ca. 100 bis 200 Stück) bzw. einer maximalen Flächengröße für „BBOX“-Filter, was wiederum eine hohe Zoomstufe im Kartenfenster erfordert. Eine Begrenzung der Featureanzahl kann sowohl clientseitig, wie auch im Karten-Server eingestellt werden.

Die Geschwindigkeit der Darstellung vieler Features nach einem WFS-Request hängt maßgeblich von der Internetverbindung sowie der Hard- und Software am Client ab. Während der Rechenaufwand eines WMS-Dienstes bei einer großen Featureanzahl eher auf der Serverseite zu suchen ist, dreht sich das Verhältnis bei einem WFS-Dienst eher zur Clientseite um. Dies liegt am Rendering-Mechanismus, der bei einem WMS-Dienst auf Serverseite generiert wird, während Daten aus einem WFS-Dienst am Client erst projiziert und gerendert werden.

Die große Stärke des WFS-Dienstes ist die Bereitstellung von Vektordaten, was die Weiterverarbeitung durch clientseitiges Geoprocessing ermöglicht.

Als Alternative zum dargestellten Rich-Client-Modell²⁹, bei dem die Applikationslogik sehr clientlastig ist, könnte ein WPS-Service angesehen werden, der die clientseitigen Verarbeitungsschritte auf den Server verlagert und ganz nebenbei dem Datenanbieter mehr Kontrolle ermöglicht, was mit seinen Daten geschieht.

5.2 Alternativen

Die vorliegende Anwendung zeigt eine Möglichkeit auf, einen WFS-Dienst einzubinden, um amtliche Geodaten im Vektor-Format mit einzubeziehen.

²⁹ http://de.wikipedia.org/wiki/Fat_Client (02.04.2012)

Es gibt jedoch auch Alternativen zum WFS-Dienst, wie den WMS-Dienst, der zwar keine Vektordaten liefert, dem aber die gleichen Daten zugrunde liegen. Käme ein WMS-Dienst in dieser Anwendung zum Einsatz, so entfielen die genaue Überlagerung von Bestandesgrenzen mit Flurstücksgrenzen. Die Genauigkeit hänge stattdessen viel stärker vom Maßstab der Zeichnung sowie der Genauigkeit und Geschicklichkeit des Anwenders beim Kartieren in der Anwendung ab.

Legt der Anwender keinen Wert auf eine perfekte Genauigkeit, etwa weil er nicht auf die Berechnung des „steuerlichen Nutzungssatzes“ angewiesen ist, so stellt der WMS-Dienst einen nicht zu verleugnenden Ersatz für den WFS-Dienst dar.

Eine weitere Alternative ergibt sich aus der konventionellen Methode, Flurstücksdaten zu beziehen. Der Anwender könnte die Flurkartenausschnitte getrennt von der Anwendung bestellen, sich zuschicken lassen und in die Anwendung hochladen. Für den geübten GIS-Anwender oder forstlichen Sachverständigen stellt dies kein Problem dar. Auch aus Sicht der Anwendungsentwicklung müsste lediglich eine Möglichkeit zum Upload der Daten geschaffen werden.

Ziel der Anwendung ist es aber, wie einleitend erwähnt, dem nicht geübten Anwender ein Werkzeug in die Hand zu geben, ohne Hilfe von Experten seine Bestände in einer Karte einzugeben. Mit dem konventionellen Bezug von Vektordaten wäre dieses Ziel nicht zu erreichen.

Direkt mit WFS-Diensten verwandt sind nicht OGC-konforme Dienste. Der wichtigste Vertreter ist das Mapfish-Protokoll³⁰, welches das Erstellen, Lesen, Ändern und Löschen von Features in einer Datenbank zulässt. Als Format wird wie auch in dieser Anwendung GeoJSON verwendet.

Ein Nachteil dieses Dienstes ist neben einer eingeschränkten Auswahl an Operatoren, die mangelnde Unterstützung des Übertragungsprotokolls in OpenLayers. Während WFS von Haus aus implementiert ist, müsste entweder ein Protokoll-Handler programmiert oder stattdessen der Mapfish-Client eingesetzt werden.

Ein Vorteil wiederum ist die kompakte und leicht verständliche Syntax der Requests, die sehr schnell zu erlernen ist.

Inwieweit Alternativen zum WFS in WebGIS-Anwendungen dieser Art eingesetzt werden

³⁰ <http://trac.mapfish.org/trac/mapfish/wiki/MapFishProtocol> (10.04.12)

können, hängt also maßgeblich von den Anforderungen der Benutzer sowie der zur Verfügung stehenden Entwicklungsressourcen ab.

5.3 Empfehlungen

Auf der Basis der vorliegenden Untersuchung lassen sich Empfehlungen für eine zukünftige Bereitstellung durch Vermessungsbehörden formulieren.

Kostenmodelle „Frei“ vs. „Kostenpflichtig“

Da die Herausgabe von Flurstücksgrenzen mittels WFS, wie die Untersuchung zeigt, ohnehin nur sinnvoll bei geringen Stückzahlen ist, scheint die Frage gerechtfertigt, ob der Aufwand eines nutzungsgenauen Abrechnungssystems sich in der Praxis bezahlt macht.

So könnte eine kostenlose Datenbereitstellung für bestimmte Nutzungspersonen aus der Land- und Forstwirtschaft ohne oder mit nur geringen kommerziellen Absichten eingerichtet werden, damit der Verwaltungsaufwand nicht die potenziellen Einnahmen übersteigen.

Für kleinere Nutzungskreise mit kommerzieller Absicht, wie Planungs- und Ingenieurbüros könnten wiederum verschiedenste Nutzungsmodelle, wie „Pay per View“ oder „Flatrates“ zum tragen kommen.

Die Umsetzung ausgeklügelter Nutzungsmodelle würde ferner eine Digitale Rechteverwaltung (DRM) voraussetzen. Somit wäre es nötig über eine reine Definition von Nutzern und Nutzerrollen hinaus zusätzliche Systeme zu implementieren, die den Datenverkehr überwachen, steuern und gegebenenfalls blockieren. Die Anbringung von Wasserzeichen, wie in der Musik und Filmindustrie üblich, könnte hierbei allerdings nicht angewendet werden, wodurch eine wichtige Sicherheitskomponente ausscheiden müsste.

Ein zwischen dem WFS-Dienst und der Client-Anwendung platziertes DRM-System könnte in Form einer Middleware mit kalkulierbarem Aufwand eingebunden werden. Die Vorteile einer solchen Architektur, wie eine individuelle Abrechnung von Usern, stehen jedoch dem Nachteil einer erhöhten Komplexität und einem erhöhten Personalaufwand gegenüber.

Koordinatensystem und Projektion

Da Webdienste nicht nur im Zusammenspiel mit Desktop-Anwendungen, sondern vielmehr, und in Zukunft immer stärker, mit Webanwendungen zum Einsatz kommen, ist es nötig die geographische Projektion der Daten auf die Anwendung anzupassen. Während eine komplizierte Transformation (inkl. NTV2) eines bundeslandspezifischen Referenzsystems in Desktop-Programmen (wie ArcMap) mühelos möglich ist, bedarf es in Webanwendungen einer Umprojektion durch einen serverseitigen Proxy-Dienst oder einer Umwandlung auf dem Client mittels bestimmter Algorithmen.

Libraries hierfür gibt es im Open-Source-Bereich (z.B. Proj4 & Proj4js), allerdings wird die NTV2-Transformationen noch nicht hinreichend unterstützt. Folgende Voraussetzungen für Vektordaten können also für den Einsatz in Webanwendungen empfohlen werden, um die direkte Verwendbarkeit in Webanwendungen sicher zu stellen:

- Unterstützung für ETRS89 und WGS84
- keine weitere NTV2-Transformation nötig
- Neben der neuesten WFS-Version sollten ältere Protokoll-Version angeboten werden, da sich seit der Version 1.3.0 die Reihenfolge von Latitude und Longitude bzw. Rechts- und Hochwert bei Koordinatenangaben umgekehrt hat.

Passwortschutz und SSL

Die Anwendung wurde zwar für das HTTP-Protokoll entwickelt, kann aber leicht auf HTTPS portiert werden, um die Datenpakete verschlüsselt zu übermitteln. Dadurch können Passwörter sowie private Daten besser vor „Hijacking“ geschützt werden. Seitens des Softwareanbieters besteht die Pflicht, den Server ausreichend gegen Hackerangriffe zu schützen. Die Anforderungen hierfür sind nicht trivial, unterliegen einem ständigen Wandel und verlangen zur Umsetzung fachlich ausgebildetes Personal.

Wird ein passwortgeschützter Dienst in einer Webanwendung verwendet, fragt der Browser bei jeder Session nach Benutzernamen und Passwort, was bei einer hohen Benutzerzahl zu einem erhöhten Aufwand führen würde, da jeder Benutzer vom WFS-Dienstanbieter ein zusätzliches Passwort erhalten müsste. Außerdem wird der Workflow eingeschränkt und der Benutzer muss sich neben dem Login noch ein weiteres

Passwort merken.

Um diesen Fall zu umgehen, gibt es die Möglichkeit den WFS-Dienst durch einen Proxy zu schleusen. Dadurch kann das Passwort für den Client unsichtbar bleiben und der Zugriff wird fortan durch das Login der Anwendung geschützt.

Die wichtigsten sicherheitsrelevanten Aspekte beim Umgang mit Flurstücksdaten sind personenbezogene Daten. Da mit dem WFS-Dienst keinerlei persönliche Daten übermittelt werden, obliegt die Verantwortung bei der Web-Anwendung, sobald durch den Benutzer Personendaten eingegeben werden. Inwieweit diese Verantwortung an den Benutzer übertragen werden kann, muss für den Produktivbetrieb geklärt werden.

Lizenzvereinbarung und Vertragsgestaltung

Gemäß der unterschiedlichen Herangehensweise der Erstellung von Waldflächen aus Flurstücksgrenzen (siehe Abschnitt 4.4), sollte der rechtliche Rahmen für die lokale Speicherung von Kopien oder abgeleiteten Arbeiten klar definiert werden. Vorbild könnte die bereits genannte Creative Commons-Lizenz sein, auch wenn es sich um einen nicht freien Dienst handelt.

Ausserdem sollte die Erlaubnis oder die Einschränkung der Verwendung eines Proxy-Servers in der Nutzungsvereinbarung festgehalten werden.

Logging

Seitens des WFS-Dienst-Anbieters besteht wenig Einfluss in die Programmlogik und damit Einblick in die Folgeprozesse nach einem WFS-Requests. Somit sollte der Datenbereitsteller über alle Requests Buch führen und die Möglichkeit des „Loggings“ verwenden, um diese über einen längeren Zeitraum nachverfolgen zu können. Für kostenpflichtige Nutzungsmodelle muss dies möglicherweise ohnehin geschehen, um eine Abrechnung nach tatsächlich angefallener Nutzung zu ermöglichen.

5.4 Zusätzliche Rollen eines Flurstück-WFS-Dienstes

Die Erfahrungen aus der Entwicklung dieser Anwendung lassen erkennen, dass ein Flurstück-WFS-Dienst nicht nur zur Kartierung von Waldstücken entlang von Eigentumsgrenzen eingesetzt werden kann. Vielmehr kann jede bisherige Nutzung der

Flurkarte auf ihre Verwendbarkeit in einem Webdienst hin überprüft werden. Dabei gilt es abzuwägen, welchen Mehrwert z.B. durch Kosten- oder Zeitersparnis erwartet werden kann.

Im Folgenden werden zwei Beispiele für eine Verwendung von Webdiensten aufgezeigt, die einen Anreiz für die Verwendung des Dienstes in anderen Gebieten schaffen sollen.

Der WFS-Dienst als Berechnungsgrundlage für eine Flurstücksliste

Für bestimmte steuerliche Berechnungen sowie in der Forsteinrichtung, in der die Erstellung von Forstkarten eine wesentliche Rolle spielt, ist es notwendig die tatsächliche Nutzung prozentual pro Flurstück aufzulisten. Dies kann am einfachsten durch Verschneidung zweier räumlicher Ebenen gelöst werden.

Dabei wird eine bestehende Vektorkarte mit den einzelnen Polygonen zugeordneten Nutzungsarten (z.B. Wald, Wiese, Weg, etc.) über eine Flurkarte gelegt und verschnitten. Die entstehenden Polygone werden mit den Attributen aus beiden Ebenen versehen und statistisch anhand der Größe ausgewertet.

Durch die Verwendung eines WFS-Dienstes könnte diese Operation durchgeführt werden, ohne einen Bestellvorgang bei einer Vermessungsbehörde zu tätigen, und helfen viel Zeit einzusparen.

Der WFS-Dienst als Hilfe zur genauen Bestimmung von Grenzen in Verbindung mit GPS

Eine mögliche mobile Anwendung des Dienstes ist bei der Bestimmung bzw. Überprüfung von Eigentumsgrenzen vor Ort denkbar.

Bei der Suche nach den Eigentumsgrenzen könnte die Vektorgeometrie als Navigationshilfe in Kombination mit einem GPS dienen. Neben der visuellen Überprüfung, die durch Rasterdienste längst möglich ist, könnte die Entfernung zur nächsten Grenze an einem mobilen Client „live“ errechnet werden.

Allerdings ist hierzu ein GPS-Empfänger mit einer Genauigkeit von mindestens unter einem Meter nötig, damit die genaue Entfernungsberechnung auch Sinn macht.

5.5 Ausblick

Die neue Version des Web Feature Service 2.0 [OGC 09-025r1] wird seit 2010 vom

OGC erarbeitet und enthält wesentliche Neuerungen.

Durch „Stored Queries“³¹ ist es fortan möglich, wiederkehrende Abfragen serverseitig zu speichern und über eine ID zu einem späteren Zeitpunkt wieder aufzurufen. Dadurch könnte in dem besprochenen Anwendungsfall eine Abfrage bestimmter Flurstücke gespeichert werden, die bereits vom Anwender bezahlt worden sind.

Aus Sicht der Anwendung sind viele Erweiterungen möglich, wie das Teilen der eigenen Waldkarte mit Förstern oder befreundeter Waldbesitzer. Diese Funktion ist in GeoDjango von Haus aus enthalten und müsste lediglich freigeschalten werden.

Der Prototyp kann auch als Grundlage für eine ganze Forstbetriebsmanagement-Suite herangezogen werden, in der neben der eigentlichen Planung auch Werkzeuge des Rechnungswesens, des Controllings und der Holztransportlogistik implementiert werden.

In der geographischen Informationsverarbeitung ist in den letzten Jahren ein stetig steigender Bedarf für WebGIS-Anwendungen zu verzeichnen. Sollte dieser Trend fortwähren, so müsste das Datenangebot auch mehr und mehr an die nachfragenden Anwendungen angepasst werden. Die hier verfassten Empfehlungen für den Umgang mit WFS-Diensten seitens des Datenanbieters könnten sich sehr häufig ändern. Man denke nur an den parallel sich entwickelnden Trend hin zu Mobile Computing mittels Smartphones und Tablet-PCs. Auch hier kommen Datendienste zum Einsatz, die vom Datenanbieter ein gutes Verständnis seiner zugrundeliegenden Technologien abverlangt, um einer wachsenden Nachfrage gerecht zu werden.

5.6 Zusammenfassung

Zu Beginn dieser Arbeit wurde das Ziel definiert, neue Erkenntnisse über die Praxistauglichkeit vektorbasierter Dienste zu gewinnen. Nach einer erfolgreichen Umsetzung eines Prototypen und den daraus gewonnenen Erfahrungen lassen sich sowohl Vorteile gegenüber alternativen Lösungen aber auch Hürden bei der Implementierung festhalten.

Im Kern dieser Arbeit wurde die Problematik der Geometriespeicherung lokalisiert und mittels verschiedener Lösungsansätze analysiert. Eine wichtige Erkenntnis hierbei ist,

³¹ <http://geoserver.org/display/GEOS/GSIP+61++WFS+2.0> (24.03.2012)

die steigende Komplexität einer Anwendung bei einer höheren Komplexität der Nutzungsmodelle.

Das Hinaustragen wertvoller öffentlicher Datenbasen mittels generalisierten Schnittstellen ist erklärtes Ziel der Europäischen Union und in Form der INSPIRE-Richtlinie verbindlich in Papierform festgehalten.

Inwiefern eine Behörde oder ein öffentlicher Eigentümer von Daten von interoperablen Diensten Gebrauch macht, hängt, wie diese Arbeit zeigt, weniger von den bestehenden technischen Möglichkeiten, als von der Entscheidung über die Freiheit oder Nutzungseinschränkung von Daten und deren Vervielfältigung ab.

6 Anhang

6.1 GetCapabilities

(Auszug: Service Information):

```
<WFS_Capabilities
  version="1.0.0"
  xsi:schemaLocation="http://www.opengis.net/wfs
http://geoportal.bayern.de/geoserver/schemas/wfs/1.0.0/WFS-capabilities.xsd"
  xmlns="http://www.opengis.net/wfs"
  xmlns:app="http://geoportal.bayern.de/WfsAccess/app"
  xmlns:dog="http://www.lverma.nrw.de/namespaces/dog"
  xmlns:iso19112="http://www.opengis.net/iso19112"
  xmlns:baymap="http://www.lvg.bayern.de/baymap"
  xmlns:by="http://www.lvg.bayern.de/by"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Service>
  <Name>
    WFS
  </Name>
  <Title>
    Web Feature Service der Bayerischen Vermessungsverwaltung
  </Title>
  <Abstract>
    Der WFS der Bayerischen Vermessungsverwaltung (BVV) ist ein benutzerspezifischer und
    zentraler Zugang zu verschiedenen Objektklassen (FeatureTypes). Nutzer koennen sich
    individuell die einzelnen verfuegbaren Objektklassen fuer ihren Account freischalten
    lassen.
  </Abstract>
  <Keywords>
    Bayerischen Vermessungsverwaltung; BVV; Web Feature Service; WFS; Vektordaten
  </Keywords>
  <OnlineResource>
    http://geoportal.bayern.de/WfsAccess/query.xml
  </OnlineResource>
  <Fees>
    https://geoportal.bayern.de/geodatenonline/inhalte/preise.html
  </Fees>
  <AccessConstraints>
    https://geoportal.bayern.de/geodatenonline/inhalte/nutzungsbedingungen.html
  </AccessConstraints>
</Service>
[...]
```

6.2 HTML-Struktur der Anwendung

```

<!DOCTYPE HTML>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title id='title'>WaldViewer 0.0</title>
    <link rel="stylesheet" type="text/css"
      href="../../js/ext_3.4.0/resources/css/ext-all.css" />
    <link rel="stylesheet" type="text/css"
      href="../../js/GeoExt_1.1/resources/css/geoext-all-debug.css" />
    <link rel="stylesheet" type="text/css"
      href="../../js/prototip_1.3.5.1/css/prototip.css" />
    <link rel="stylesheet" type="text/css"
      href="../../js/OpenLayers_2.11/theme/default/style.css" />
    <link rel="stylesheet" type="text/css"
      href="../../js/OpenLayers_2.11/theme/default/google.css" />
    <link rel="stylesheet" type="text/css" href="../../js/gxp_0.1/src/theme/all.css"/>
    <link rel="stylesheet" type="text/css" href="css/waldfuchs-styles.css" />

    <script src="http://maps.google.com/maps/api/js?v=3.6&sensor=false"></script>
    <script type="text/javascript"
      src="../../js/ext_3.4.0/adaptor/ext/ext-base.js"></script>
    <script type="text/javascript"
      src="../../js/ext_3.4.0/ext-all-debug.js"></script>
    <script type="text/javascript" src="../../js/OpenLayers_2.11/OpenLayers.js"></script>
    <script type="text/javascript" src="../../js/GeoExt_1.1/lib/GeoExt.js"></script>
    <script type="text/javascript"
      src="../../js/gxp_0.1/src/script/loader.js"></script><!-- gxp -->
    <script type="text/javascript" src="../../js/proj4js/lib/proj4js-combined.js"></script>
    <script type="text/javascript" src="../../js/prototype_min_1.7.js"></script>
    <script type="text/javascript"
      src="../../js/scriptaculous_1.9.0/src/scriptaculous.js"></script>
    <script type="text/javascript" src="../../js/prototip_1.3.5.1/js/prototip.js"></script>
    <script type="text/javascript" src="../../js/geofuchs_0.1/geo-fuchs-1.0.js"></script>
    <script type="text/javascript" src="js/geo_config.js"></script>
    <script type="text/javascript" src="js/geo_actions.js"></script>
    <script type="text/javascript" src="js/geo_viewer.js"></script>
    <script type="text/javascript" src="js/geo_forms.js"></script>
  </head>
  <body>
    <table id="layouttable">
      <tbody>
        <tr>
          <td class="wf-head">
            <div id="header">Waldfuchs <sub>v0.1 BETA</sub></div>
          </td>
        </tr>
        <tr>
          <td class="wf-app">
            <div id="app"><!--the app--></div>
          </td>
        </tr>
        <tr>
          <td class="wf-bottom_row">
            <div id="debug"><!--bottom row--></div>
          </td>
        </tr>
      </tbody>
    </table>
  </body>
</html>

```

6.3 Datenmodell in GeoDjango

models.py (Auszug der wichtigsten Klassen)

```
from django.contrib.auth.models import User
from django.contrib.gis.db import models
from decimal import Decimal

class bestand(models.Model):
    owner = models.ForeignKey(User)
    distrikt = models.CharField(max_length=2)
    abteilung = models.CharField(max_length=2)
    nummer = models.PositiveIntegerField(max_length=5)
    qm = models.DecimalField(max_digits=9, decimal_places=1)
    bg = models.DecimalField(max_digits=3, decimal_places=2)
    na = models.CharField(
        max_length=2,
        choices= (
            ('JP', 'Kultur, Pflegebestand'),
            ('JD', 'Stangenholz - bereits mit ersten Vorertraegen'),
            ('AD', 'mittelalter Bestand - noch nicht hiebsreif'),
            ('EN', 'hiebsreifer Bestand'),
        )
    )
    poly = models.MultiPolygonField(
        spatial_index=True, # for fast reference
        geography=True # uses spherical arcs instead of linear distances
    )
    objects = models.GeoManager()
    def ha(self):
        area_ha = self.poly.area * 10000
        return area_ha
    def ort(self):
        return u'%s.%s.%s' % (self.distrikt, self.abteilung, self.nummer)
    def __unicode__(self):
        return u'%s.%s.%s' % (self.distrikt, self.abteilung, self.nummer)
    class Meta:
        ordering = ['distrikt', 'abteilung', 'nummer']

class aufnahme(models.Model):
    bestand = models.ForeignKey(bestand)
    ba = models.ForeignKey('baumart')
    qm_prozent = models.DecimalField(max_digits=5, decimal_places=4)
    v_method = models.OneToOneField('vfm_methode')

    def __unicode__(self):
        return unicode(self.waldort) + ' ' + unicode(self.ba)
```

7 Literatur

AUMANN ET AL. (2003) : *Mobile Liegenschaftsauskunft auf der Basis von OGC Web Services*, Tagungsband 18 "Geodaten- und Geodienste-Infrastrukturen - von der Forschung zur praktischen Anwendung", Reihe IfGIprints, Hrsg. Institut für Geoinformatik der Universität Münster, S. 195 – 207.

BERNARD ET AL. (2005): *Geodateninfrastruktur – Grundlagen und Anwendungen*, Herbert Wichmann Verlag, Heidelberg.

BORN ET AL. (2004) : *Softwareentwicklung mit UML 2*, Addison-Wesley, München, 293 S.

DVW BAYERN (2010): *Mitteilungen des DVW Bayern (1/2010)*, S.149.

Online (28.02.2012) : [www.dvw-](http://www.dvw-bayern.de)

bayern.de/UserFiles/File/Veroeffentlichungen/Mitteilungen/2010_1/aktuelle_notizen_1_2010.pdf

EStG: *Einkommensteuergesetz in der Fassung der Bekanntmachung vom 8. Oktober 2009 (BGBl. I S. 3366, 3862)*, das zuletzt durch Artikel 20 des Gesetzes vom 20. Dezember 2011 (BGBl. I S. 2854) geändert worden ist.

EStR 2005 : *Einkommensteuer-Richtlinien 2005*, Vom 16. Dezember 2005 (BStBl I Sondernummer 1/2005 S. 3) Geändert durch Verwaltungsvorschrift vom 18. Dezember 2008 (BStBl I S. 1017).

EVANS U. SABEL (2012): *Open-Source web-based geographical information system for health exposure assessment*, International Journal of Health Geographics 2012 11:2.

FIELDING, R. T. (2000): *Architectural Styles and the Design of Network-based Software Architectures*, Doctoral dissertation, University of California, Irvine, 2000.

FIRL, M. (2004): *Implementierung eines Gazetteer-Service*, Diplomarbeit, Hochschule für Technik und Wirtschaft Dresden.

Online: http://geoinformatik.htw-dresden.de/abschlussarbeiten/DA_FIRL_2004/ (08.04.2012)

FORGHANI ET AL. (2010): *Web Mapping for the Murray-Darling Basin Authority - The Challenges and Opportunities of Open Access* , Murray-Darling Basin Authority, Canberra, Australia - alan.forghani@mdba.gov.au

GDI-BY (2006) : *Geodateninfrastruktur in Bayern - Ein pragmatisches Konzept*, Landesamt für Vermessung und Geoinformation, Geschäftsstelle GDI-BY , Alexandrastr. 4 , 80538 München., Online (28.02.2012) :
www.gdi.bayern.de/file/pdf/48/GDI_BY_Konzept.pdf

IOSIFESCU-ENESCU, HUGENTOBLE U. HURNI (2010): *Web cartography with open standards – A solution to cartographic challenges of environmental management*, Environmental Modelling and Software, 2010, Vol.25(9), p.988-999 [Peer Reviewed Journal]

KAISER, F. J.: *Die Fallstudien. Theorie und Praxis der Fallstudiendidaktik*. Bad Heilbrunn 1983

LONGLEY ET AL. (2005): *Geographic Information Systems and Science*, John Wiley & Sons Ltd., West Sussex, England.

LVG PREISLISTE (02/2012): *Preisliste - Produkte und Dienste* , Landesamt für Vermessung und Geoinformation, Stand: 01.02.2012 , Online (28.02.2012) :
<https://geoportal.bayern.de/geodatenonline/inhalte/preise.html>

OGC 99-049 (1999): *OpenGIS® Simple Features Specification For SQL* , Version 1.1, OpenGIS Project .
Online (06.04.2012): <http://www.opengeospatial.org/standards/sfs>

OGC 01-047r2 (2001): *Web Map Service Implementation Specification* , Version 1.1.0, Open Geospatial Consortium Inc.
Online (28.02.2012): <http://www.opengeospatial.org/standards/wms>

OGC 04-094 (2005): *Web Feature Service Implementation Specification*, Version 1.1.0,

Open Geospatial Consortium Inc.

Online (28.02.2012): <http://www.opengeospatial.org/standards/wfs>

OGC 04-095 (2004): *Filter Encoding Implementation Specification*, Version 1.1.0, Open Geospatial Consortium Inc.

Online (06.04.2012): <http://www.opengeospatial.org/standards/filter>

OGC 05-035r2 (2006): *Gazetteer Service - Application Profile of the Web Feature Service Implementation Specification*, Version 0.9.3, Open Geospatial Consortium Inc.

Online (03.04.2012) : <http://www.opengeospatial.org/standards/bp>

OGC 09-025r1 (2010): *OpenGIS Web Feature Service 2.0 Interface Standard (also ISO 19142)*, Open Geospatial Consortium Inc.

Online (24.03.2012): <http://www.opengeospatial.org/standards/wfs>

PENG U. ZHANG (2004): *The roles of geography markup language (GML), scalable vector graphics (SVG), and Web feature service (WFS) specifications in the development of Internet geographic information systems (GIS)*, Journal of Geographical Systems, 2004, Vol.6(2), p.95-116 [Peer Reviewed Journal]

SIMÃO A., DENSHAM P.J., HAKLEY M. (2008): *Web-based GIS for collaborative planning and public participation: An application to the strategic planning of wind farm sites*, Journal of Environmental Management, Elsevier Ltd., 90 (2009) 2027–2040

STMELF (12/2009) : *Flyer „Mehrfachantrag-Online/BayernViewer-agrar“*, Hrsg. Bayerisches Staatsministerium für Ernährung, Landwirtschaft und Forsten, Ludwigstraße 2, 80539 München.

STROBEL S. UND REINHARDT W.(2007): *Die Bedeutung von Profilen für den Datenaustausch mittels Geography Markup Language*, In: Mitteilungen des Bundesamtes für Kartographie und Geodäsie, Band 39.

Online: http://137.193.222.80/publikationen/download/strobel_bkg_2007.pdf (06.04.2011)

VOSS, ANGI ET AL. (2004): *Evolution of a participatory GIS*, Computer, Environment and

Urban Systems, Elsevier Ltd. 28 (2004) 635-651

ZHANG, CHUANRONG (2005): *The roles of Web Feature and Web Map Services in real-time geospatial data sharing for time-critical applications*, Cartography and Geographic Information Science, Volume 32, Number 4, October 2005 , pp. 269-283(15)