Webmapping - aber sicher

Silke Reimer

Inhaltsverzeichnis

Abstract

Webmapping, also das Präsentieren von Geodaten in Form von Karten über das Internet, steht oft vor den beiden Herausforderungen, zum einen die Karten in kurzer Zeit zu rendern, zum anderen einen differenzierten Zugang zu den Karten zu gewährleisten. Dieser Vortrag stellt eine Lösung dieses Problem vor, die mit Hilfe des UMN Mapservers und Zope realisiert wurde.

Einführung

80 % aller Daten haben in irgendeiner Form einen räumlichen Bezug. Um die Informationen, die diesen Daten zugrunde liegen, nutzen zu können, kommt der Visualisierung von Geodaten eine entscheidende Rolle zu. Die Möglichkeit des Webmapping, also die Bereitstellung der Karten über HTTP, bietet dabei zusätzliche Vorteile: Die Daten können zentral gehalten werden. Damit wird die Verwaltung der Daten wesentlich einfacher, als wenn jeder seinen eigenen Datensatz halten muss. Dateninkonsistenzen werden vermieden. Darüber hinaus reicht ein gewöhnlicher Internetbrowser aus, um die generierten Karten auf einem Klientenrechner zu visualisieren. Es ist also nicht notwendig, Spezialsoftware auf den Rechnern zu installieren.

Webmapping Anwendungen stehen u.a. vor der Herausforderung, dass der Nutzer erwartet, dass die Karten in einer kurzen Zeitspanne, also innerhalb weniger Sekunden gerendert und dargestellt werden. Dies gilt auch dann, wenn es sich um große und sehr detaillierte Datenbestände handelt.

Rechtliche Aspekte

Ein anderes Problem ergibt sich für den Anbieter von Webmapping Anwendungen daraus, dass die meisten Geodaten nicht frei zugänglich sind. Von den Landesvermessungsämtern werden zwar detaillierte Datensätze digital zur Verfügung gestellt (ALK/ALB-Datensatz, ATKIS, Topographische Karten etc.). Allerdings kosten diese Daten eine Menge Geld und dürfen dann oft nur innerhalb der Behörde oder Firma, die diese Daten gekauft hat, eingesetzt werden. Auch datenschutzrechtliche Gründe spielen bei der Berechtigung des Zugriffs eine Rolle. Dies ist z.B. der Fall bei den ALB-Daten, bei denen Namen und Adressen von Flurstücken enthalten sind.

Eine weit verbreitete Methode für das Bezahlen von Geodaten in Webmapping Anwendungen ist das so genannte pay-per-click. Damit ist gemeint, dass der Geldbetrag, den der Anbieter eines Webmapping Services dem Anbieter der Geodaten zahlen muss, davon abhängig ist, wie oft die Webmapping Anwendung aufgerufen wurde. Da sich die Anzahl der Zugriffe auf eine Webseite schwer im Voraus bestimmen lässt, werden die Kosten für den Anbieter damit unkalkulierbar.

Aus diesen rechtlichen Problemen ergibt sich für den Anbieter von Webmapping die Bedingung, die Anwendungen nur einem ausgewählten Nutzerkreis zur Verfügung zu stellen. Oftmals muss dabei auch zwischen einzelnen Nutzergruppen innerhalb einer Behörde oder Firma unterschieden werden, so dass eine differenzierte Zugangsberechtigung notwendig wird.

In diesem Artikel wird von der Implementation einer Lösung für das skizzierte Problem für eine mittelgroße Behörde in Deutschland berichtet. Zum Einsatz kamen dabei der UMN Mapserver 3.6 sowie Zope, die beide als Freie Software entwickelt werden. Sie werden in diesem Artikel kurz vorgestellt. Danach wird demonstriert, wie sie sich sich zu einer schnellen und sicheren Anwendung vereinigen lassen.

UMN Mapserver

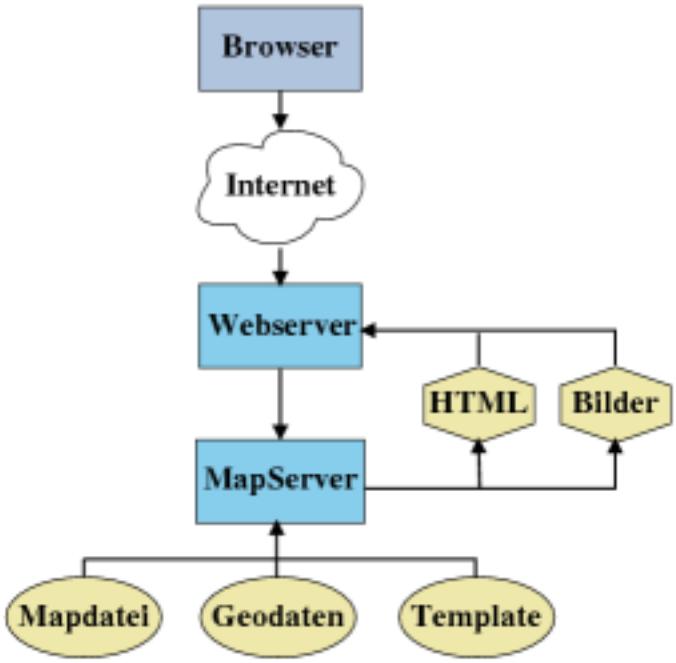
Der UMN Mapserver ist ein freier Mapserver. Er wurde ursprünglich von der University Minnesota entwickelt - daher auch der Name UMN Mapserver. Mit Sicherheit kann man behaupten, dass der UMN Mapserver der am weitesten verbreitete freie Mapserver ist. Der UMN Mapserver steht unter einer X11-ähnlichen Lizenz. Damit werden die 4 Freiheiten im Sinne der Free Software Foundation gewährt: Die Möglichkeit zur Kopie und Änderung der Software sowie das Recht zur Weitergabe von Kopie und Veränderung an beliebige Personen. Allerdings ist es möglich, Änderungen an der Software vorzunehmen und diese als proprietäre Software weiterzugeben, und damit die Freiheit der neuen Variante zu beschränken.

Einige der Stärken des UMN Mapserver sind die hohe Stabilität und die Geschwindigkeit im Rendern der Karten sowie seine Skalierbarkeit. Erreicht wird dies durch verschiedene Mechanismen wie z.B. die Kachelung der Geodaten. Damit müssen zum Rendern von Karten nur diejenigen Daten geladen werden, die innerhalb des gewünschten Kartenausschnitts liegen. Eine Indizierung der Daten beschleunigt zusätzlich das Auffinden der relevanten Geodaten. Nicht zuletzt wegen des Geschwindigkeitsvorteils wird der UMN Mapserver oft proprietären Lösungen vorgezogen.

Der UMN Mapserver ist platformunabhängig und kann verschiedene freie und proprietäre Raster- und Vektordatenformate lesen. Außerdem ist eine Anbindung an Geodatenbanken möglich (PostgreSQL mit PostGIS, Oracle Spatial und ArcSDE). Der Mapserver wird konsequent weiter entwickelt. Besonders aktiv sind hierbei wiederum die Universität Minnesota sowie die kanadische Firma DM Solutions. Innerhalb der letzten Zeit wurde der UMN Mapserver insbesondere um die Spezifikationen der OpenGIS Konsortiums erweitert. So ist es seit der Version 3.6 möglich, den UMN Mapserver als WMS (WebMapping Service) Server und Klienten einzusetzen. In der neuesten Version 3.7 erfüllt der UMN Mapserver auch die WFS (Web Feature Server) Spezifikation.

Funktionsweise

In seiner einfachsten Form wird der UMN Mapserver als CGI-Binary eingesetzt, das - wie in nachstehender Abbilung gezeigt - vom Webserver gestartet wird. Der Mapserver liest dann eine Konfigurationsdatei, die so genannte Mapdatei oder das Mapfile, ein. Dort wird festgelegt, welche Geodaten aus welchen Datenquellen gelesen werden sollen, wie sie dargestellt werden usw. Danach rendert der UMN Mapserver aus den Geodaten die Bilddateien (Karte, Legende, Referenzkarte etc.) und schreibt diese ins Dateisystem. In ein HTML-Template werden die Pfade zu diesen Dateien eingetragen. Dieses Template wird dem Webserver schließlich zur Darstellung übergeben.



Da innerhalb eines Mapfiles alle relevanten Informationen zur Darstellung der Karte festgelegt werden, korrespondiert jeweils eine Mapdatei mit einer Mapserver Anwendung.

In der hier vorgestellten Anwendung wird der Mapserver als CGI-Binary eingesetzt. Trotzdem sei an dieser Stelle der Vollständigkeit halber erwähnt, dass es darüber hinaus möglich ist, den UMN Mapserver über Skriptsprachen (PHP, Perl, Python) anzusprechen.

Sicherheitslücken

Auch wenn der UMN Mapserver eine ganze Reihe von Möglichkeiten in der Konfiguration bietet, so konzentriert sich die Funktionalität im Wesentlichen auf das Rendern von Karten. Das hat den Vorteil, dass der UMN Mapserver darin so gut ist, dass er - wie bereits erwähnt - eine ernst zu nehmende Alternative zu allen proprietären Produkten darstellt, bzw. sie in vielerlei Hinsicht übertrifft. Allerdings verfügt der UMN Mapserver über kein eigenes Sicherheitskonzept. Wenn mehrere Mapserver Anwendungen unter Verwendung des selben Mapserver CGIs in eine Webseite eingebaut werden, dann kann jeder Nutzer, der für eine Anwendung die Zugangsberechtigung zum CGI bekommen hat, damit auch die anderen Mapserver Anwendungen starten, indem er den Namen einer anderen Mapdatei als CGI-Parameter übergibt.

Wenn es darum geht, einzelne Mapserver Anwendungen nur für einige Nutzer zugänglich zu machen, kommt kommt also der Mapdatei eine zentrale Bedeutung zu . Immerhin wird in dieser Datei festgelegt, welche Geodaten zugänglich sind und welches Template an welcher Stelle verwendet werden soll. Es muss daher dafür gesorgt werden, dass das CGI-Binary prinzipiell für alle Nutzer aber nur in Kombination mit einem bestimmten Mapfile zugänglich sein.

Ein weiteres Problem stellen die Bilddateien dar. Sie werden - wie bereits erwähnt - in das Dateissystem geschrieben. Danach sind sie von dort aus über den Webbrowser sichtbar. Um zu garantieren, dass niemand unbefugt Karten von Geodaten sehen kann, zu denen er eigentlich keinen Zugang hat, muss auch bei der Herausgabe der Bilder durch den Webbrowser eine Beschränkung eingeführt werden.

Die Tatsache, dass die Bilddateien in das Dateisystem geschrieben werden, stellt in Bezug auf möglicherweise voll laufenden Plattenplatz ein weiteres Sicherheitsproblem dar. Das Problem kann durch ein Skript gelöst werden, das - abhängig von den Systemanforderungen - die Bilddateien mehr oder weniger häufig abräumt.

Zope

Zope ist ein Applikation-Server, der spezialisiert ist auf Content Managment und Portale. Über Pythonskripte und DTML erlaubt Zope das Entwickeln von dynmischen webbasierten Anwendungen. Zope selbst ist in Python geschrieben. Zahlreiche so genannte Produkte erlauben es, Zope um die verschiedensten Funkionalitäten zu erweitern. Zope steht unter der ZPL (Zope Public Licence), die genau wie die Mapserver Lizenz eine X11-ähnliche Lizenz ist.

Zope ist ein so mächtiges Werkzeug, dass hier gar nicht versucht werden soll, einen vollständigen Überblick zu geben. Ich möchte an dieser Stelle nur auf einen eingeschränkten Bereich des Sicherheitskonzeptes eingehen, um verdeutlichen zu können, wie wir den UMN Mapserver in eine Zope-Anwendung integriert haben.

Sicherheitskonzept

Das Zope Sicherheitskonzept baut auf den drei Säulen Authentifikation, Rechte und Autorisierung auf. Die Authentifikation definiert sich als das Erkennen des Nutzers einer Anwendung. Die Rechte legen fest, welche Aktionen auf einem Objekt von welchen Rollen möglich sind. Die Autorisierung schließlich stellt eine Verbindung zwischen den Nutzern und den Rollen her, die den Zugang zu den Objekten festlegen.

Wie man sieht, haben die Rollen eine wichtige Funktion. Eine Rolle bekommt gewissen Rechte an einem Objekt (z.B. Ausführen eines bestimmten CGI-Binaries, Anschauen einer Seite etc.). Die einzelnen Nutzer bekommen eine oder mehrere Rollen zugewiesen, über die sie dann die Erlaubnis haben, die Aktionen auszuführen, die diesen Rollen erlaubt sind. Die Aufgabe beim Management der Zope Sicherheit liegt also darin, Informationen über die Nutzer und ihre Rollen sowie den Objektrechten und die an sie gebundenen Rollen zu verwalten.

Vordefinierte Rollen

Zope hat die drei vordefinierte Rollen *Anonymous*, *Manager* und *Owner*. Alle Benutzer, einschließlich derjenigen, die sich nicht authentifizieren können, bekommen die Rolle Anonymous zugeteilt. Standardmäßig ist es der Rolle Anonymous erlaubt, Objekte zu sehen ("View") sowie weitere harmlose Aktionen darauf auszuführen.

Die Managerrolle bekommen diejenigen Nutzer zugewiesen, die für das Management der Zope Webseiten oder eines Teilbereiches dafür zuständig sind. Standardmäßig hat ein Nutzer mit der Rolle Manager alle Rechte auf einem Objekt, einschließlich der Modifizierung eines Objektes.

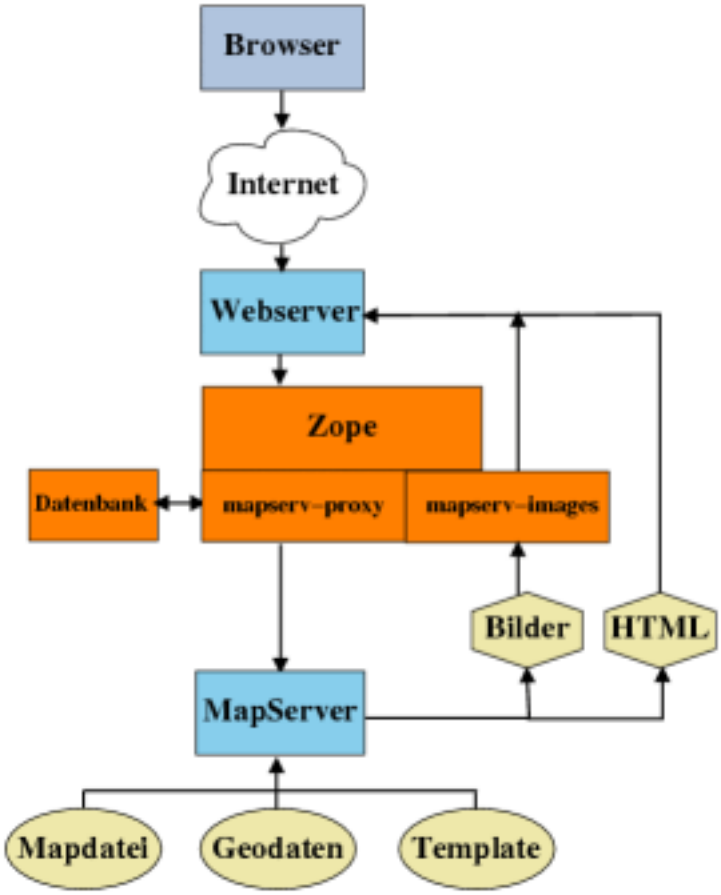
Der Owner ist eine spezielle Rolle. Wenn ein Objekt erzeugt wird, dann bekommt der Nutzer, der es erzeugt hat, die auf das Objekt beschränkt lokale Rolle des Owners. Standardmäßig hat die Ownerrolle dieselben Rechte wie der Manager.

Proxy-Rollen

Manchmal gibt es Objekte, auf die ein Nutzer nicht vollen Zugriff haben soll, die aber von diesem Nutzer verändert werden dürfen. Dies könnte natürlich dadurch erreicht werden, dass dem Objekt eine spezielle Rolle zugewiesen wird, die diese Veränderung des Objektes ohne vollen Zugriff ermöglicht. Eine Alternative ist es, ein Skript zu schreiben, das genau die erlaubte Änderung vornimmt. Das Skript bekommt die dafür notwendige Rolle (z.B. Manager) zugeteilt. Der Nutzer bekommt nur über dieses Skript Zugriff auf das zu verändernde Objekt. Das Skript ist sozusagen der Vertreter (engl. proxy) des Nutzers.

UMN Mapserver und Zope

In diesem Abschnitt soll nun erläutert werden, wie der UMN Mapserver in das Sicherheitskonzept des Zope integriert werden kann. Wie bereits erläutert, sind es die Map- und die Bilddateien, um deren Zugangsbeschränkung wir uns Gedanken machen müssen. Für die Lösung dieser Problemfälle machen wir uns das Prinzip der Proxyrollen zunutze:



Die Mapdatei

Zunächst sorgen wir dafür, dass das Ausführen des Mapserver-CGIs (und damit der Zugang zu jeder Mapdatei) nur für einen Nutzer mit der Rolle Manager möglich ist. Vor den Aufruf des Mapserver Binaries wird das Skript *mapserv-proxy* geschaltet. Ihm wird die Proxyrolle Manager zugewiesen. Damit ist dieses Skript in der Lage, das Mapserver-Binary zu starten. Innerhalb des Skriptes wird jedoch zunächst mit Hilfe von Daten in einer Nutzerdatenbank überprüft, ob die Ausführung des Mapservers mit der angeforderten Mapdatei erlaubt ist. Nur wenn dies der Fall ist, wird die Mapserver Anwendung gestartet.

Bilddateien

Ähnlich verhält es sich mit den Bilddateien. Auch sie sind nur sichtbar für die Rolle Manager. Um sie für einen Benutzer frei geben zu können, wird der Name des Nutzers, der sie erzeugt hat, innerhalb des Namens der Bilddatei eingebaut. Wenn die Bilddateien angefordert werden, prüft das Skript *mapserv-images*, das ebenso wie *mapserv-proxy* die Proxyrolle Manager hat, ob der Name des aktuelle authentifizierten Nutzers und der Nutzername, der im Namen der Bilddatei verwendet wurde, identisch sind. Erst dann wird die Bilddatei frei gegeben.

Rechteverwaltung

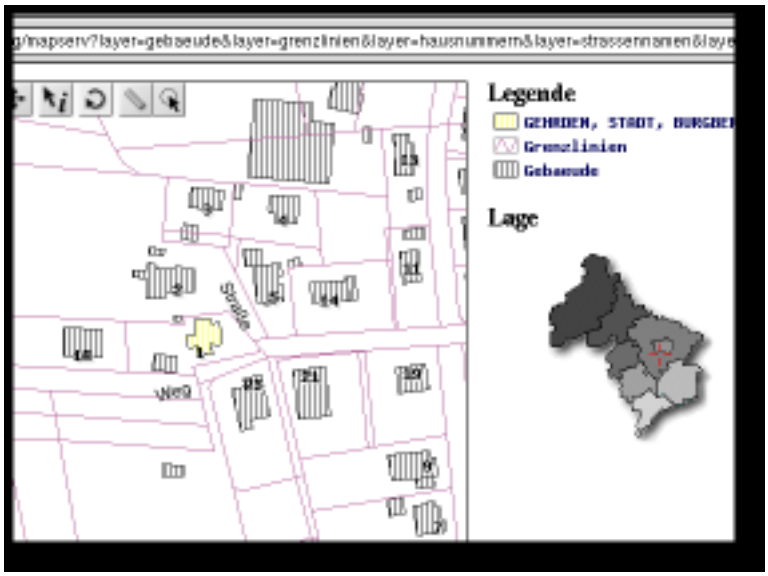
Zur Verwaltung der differenzierten Zugangsberechtigung wird eine Oberfläche bereit gestellt. Es können Nutzer, Gruppen und Rechte angelegt und verwaltet werden. Die Nutzer korrespondieren dabei mit real existierenden Personen. Die Gruppen dienen dazu, mehreren Nutzern gleichzeitig den Zugang zu einer Mapserver Anwendung erteilen zu können. Die Rechte schließlich definieren die Zugangsrechte zu den Mapdateien oder auch einzelnen Kartenebenen innerhalb einer Mapdatei. Diese Zugangsrechte können sowohl einzelnen Nutzern als auch Gruppen erteilt werden.

Wie das Zusammenspiel zwischen Nutzern, Gruppen und Rechten funktioniert, soll nachfolgend an einem Beispiel erläutert werden.

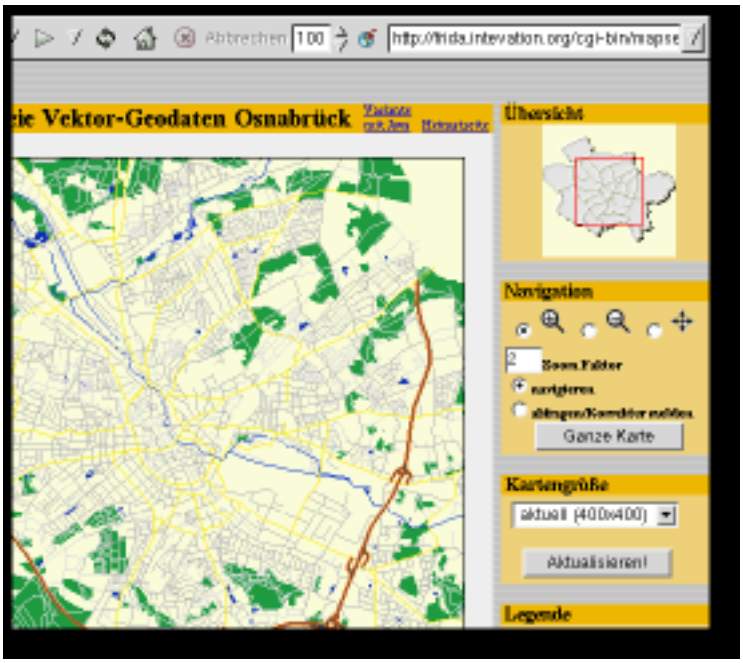
Beispiel

Die Mapserver Anwendungen

Das Beispiel enthält zwei Mapserver Anwendungen. Die erste wurde im Rahmen eines Projektes für die Bezirksregierung Hannover entwickelt. Sie beinhaltet eine Strassensuche auf einem ALB-Datensatz, die es ermöglicht, einzelne Häuser anhand von Gemeinde- und Strassennamen sowie der Hausnummer zu finden. Die Bezirksregierung Hannover hat die ALK/ALB-Daten freundlicherweise zu Präsentationszwecken zur Verfügung gestellt. Ein Beispiel für ein Ergebnis der Strassensuche ist im nachfolgender Abbildung dargestellt.



Die zweite Anwendung ist im Rahmen eines Praktikum bei der Firma Intevation GmbH erstellt worden. Ziel war, einen freien Vektordatansatz der Stadt Osnabrück zu erstellen und die Daten sowie die Möglichkeit einer Fehlereingabe im Internet bereit zu stellen. Ein Ausschnitt aus der Anwendung ist in nachfolgender Abbildung zu sehen.



Einrichtung der Gruppen

Wir richten nun die beiden Gruppen *osnabrueck* und *strassensuche* ein, deren Aufgabe es ist, die Zugangsrechte zu jeweils einer Mapserver Anwendung zu verwalten. Dies geht recht intuitiv über eine Webschnittstelle, die den Administrator nach einer GruppenID (d.h. ein Name, über den die Gruppe eindeutig referenziert wird) sowie einem ausführlichen Gruppennamen fragt. Dieser Gruppenname stellt sozusagen die etwas ausführlichere Beschreibung der Gruppe dar und kann - im Gegensatz zur GruppenID - später jederzeit geändert werden.

Einrichtung der Nutzer

Ähnlich wie die Einrichtung der Gruppen funktioniert auch die Einrichtung der Nutzer. Allerdings können sehr viel mehr Informationen - wie z.B. Adresse und Telefonnummer - zu einem Nutzer gespeichert werden. Dies erleichtert die Identifikation mit der Realperson. Außerdem wird an dieser Stelle ein Passwort vergeben, über das der Nutzer sich später authentifiziert (1. Säule des Zope Sicherheitskonzeptes). Weiterhin können die Nutzer einer oder mehreren Gruppen zugeordnet werden. Wenn man die Gruppen als Rollen definiert (s. auch den nächsten Abschnitt), dann entspricht dies der 3. Säule des Zope Sicherheitskonzeptes, der Autorisierung.

In unserem Beispiel richten wir die drei Nutzer *Silke*, *Ulrike* und *Jan* ein. *Silke* wird in die Gruppe *osnabrueck*, *Jan* in die Gruppe *strassensuche* und *Ulrike* in beide Gruppen eingetragen.

Einrichtung der Rechte

Bisher liegen mit den Gruppen und Nutzern sowie den Mapfiles nur die Grundbausteine für die Rechteverwaltung fest. Nun muss noch eine Verbindung zwischen den Gruppen und den Mapfiles hergestellt werden. In Termini des Sicherheitskonzeptes von Zope stellen die Gruppen sozusagen die Rollen dar, während die Mapfiles die Objekte sind, an denen nun die Rechte vergeben werden müssen (2. Säule des Sicherheitskonzept).

Über eine Webschnittstelle können nun Rechte wie in nachfolgender Tabelle dargestellt eingetragen werden:

Tabelle 1. Rechteverwaltung für Mapserver Anwendungen

Nutzer/Gruppe	*Mapfile*	*Layer*	*Vorgang*
osnabrueck	osnabrueck.map	*	Anzeigen
strassensuche	strassensuche.map	*	Anzeigen
osnabrueck	strassensuche.map	gebaeude	Anzeigen

Die beiden wichtigsten Spalten der Tabelle sind *Nutzer/Gruppe* und *Mapfile*. Damit wird die notwendigen Zuordnung von Gruppen zu den zur Ansicht erlaubten Mapfiles vorgenommen. Die dritte Spalte *Layer* ermöglicht es,

dass nur einzelne Kartenebenen zur Ansicht erlaubt werden, nicht aber die gesamte Anwendung mit allen Informationen. Standardmäßig sind alle Layer bzw. Kartenebenen erlaubt. Dies wird durch den Stern gekennzeichnet. Die Spalte *Vorgang* spielt an dieser Stelle noch keine Rolle, da es zunächst tatsächlich nur darum geht, Mapfiles anzuzeigen. Sie kommt dann ins Spiel, wenn die Digitalisierung hinzugefügt wird.

In vorliegendem Beispiel dürfen alle Nutzer in der Gruppe *osnabrueck* (also *Silke* und *Ulrike*) die Osnabrück Anwendung komplett sehen, von der Strassensuche allerdings nur die Kartenebene *gebaeude*. Alle Nutzer in der Gruppe *strassensuche* (also *Jan* und *Ulrike*) dürfen nur die Anwendung Strassensuche sehen.

Digitalisierfunktion

Neben der reinen Visualisierung gibt es oft der Wunsch zumindest punktuelle Geoinformationen auch über eine Webschnittstelle eintragen also digitalisieren zu können. Als Beispiel sollen hier Sehenswürdigkeiten innerhalb der Stadt Osnabrück über die Webschnittstelle digitalisiert werden können. Für die Umsetzung dieser Anwendung wurde der UMN Mapserver nur zum Rendern der Karte verwendet. Die darüber hinaus notwendigen Schritte wurden mit Hilfe von Python Skripten implementiert und lassen sich wie folgt zusammen fassen:

- Zunächst wird der zu digitalisierende Punkt über die Oberfläche eingegeben.
- Die daraus resultierenden Bildkoordinaten werden in geographische Koordinaten umgerechnet. Zur Überprüfung durch den Nutzer werden diese Daten sofort visualisert. Gleichzeitig wird die Möglichkeit geboten, den Punkt mit Sachinformationen in Form von Attributen zu versehen.
- Gibt der Nutzer sein OK, dann werden die geographischen Koordinaten einschließlich der Attributdaten in einer PostgreSQL Datenbank mit PostGIS Aufsatz gespeichert.
- Von dort aus können sie vom UMN Mapserver direkt visualisert werden.

Einbettung in das Sicherheitskonzept

Zunächst muss dafür gesorgt werden, dass nur autorisierte Nutzer einen Zugang zu der Tabelle bekommen, in der die Punktdaten eingetragen werden. In diesem Fall wurde dazu nicht auf das Konzept der Proxyrollen zurückgegriffen. Es gibt verschiedene kleine Skripte die Lese- oder Schreibzugriff auf die Tabelle haben. Zu Beginn jedes dieser Skripte wird überprüft, ob der autorisierte Nutzer die Berechtigung hat, eine Punkteingabe in diese Tabelle durchzuführen.

An dieser Stelle kommt die Spalte *Vorgang* zum Einsatz, in der nun neben der Aktion *Anschauen* auch die Aktion *Punkteingabe* zur Verfügung gestellt wird. Statt des Mapfiles wird nun eine Tabelle angegeben, so dass es möglich ist, mehrere Anwendungen zur Punkteingabe nebeneinander zu verwalten. Die Sehenswürdigkeiten für Osnabrück werden in der Tabelle *sehenswuerdigkeiten* gespeichert. Um der Gruppe *osnabrueck* die Rechte an der Digitalisierung dieser Sehenswürdigkeiten zu erteilen, muss die Rechtetabelle um den vierten Eintrag in folgender Tabelle erweitert werden.

Tabelle 2. Rechteverwaltung mit Digitalisierung

Nutzer/Gruppe	*Mapfile/Tabelle*	*Layer*	*Vorgang*
osnabrueck	osnabrueck.map	*	Anzeigen
strassensuche	strassensuche.map	*	Anzeigen
osnabrueck	strassensuche.map	gebaeude	Anzeigen
osnabrueck	sehenswuerdigkeiten	*	Punkteingabe

Schlussfolgerungen und Ausblick

Mit Hilfe der Kombination des UMN Mapservers und von Zope ist es gelungen eine gute Lösung für das Problem zu entwickeln, zum einen schnell geographischen Karten für die Bereitstellung im Internet zu erstellen, zum anderen Sicherheitsfragen nicht außer acht zu lassen. Darin zeigt sich eine der Stärken von Freier Software, nämlich dass sich die Gesamtlösung eines Problems oftmals durch die Verwendung von verschiedenen Softwareprodukten entwickeln lässt, die jeweils auf die Lösung eines Teilproblems spezialisiert sind. Freie Software ist für solch eine Kombination von verschiedenen Produkten besonders geeignet, weil evtl. notwendige kleine Anpassungen an den Komponenten problemlos durchgeführt werden können.

Für die Version 3.7 des UMN Mapserver, die kurz vor dem Release steht (bzw. zum Zeitpunkt des Linuxtages evtl. schon herausgekommen ist) gibt es Zope-Produkt namens ZMapserver. Es arbeitet mit der Möglichkeit, den UMN Mapserver über über die Skriptsprache Python ansprechen zu können. Es ist zu prüfen, in wieweit es Sinn macht, die Ansprache des UMN Mapserver im Zope von der reinen CGI-Variante auf die Verwendung von ZMapserver umzustellen. Allerdings bleibt auch damit die Aussage gültig, dass der UMN Mapserver nicht selbst die Sicherheitsüberprüfungen übernimmt, sondern dass diese innerhalb des Zope vorgenommen werden (müssen).

Einige Links zum Thema

- Die Mapserver Homepage [http://mapserver.gis.umn.edu/index.html] bietet neben der Möglichkeit zum Download des UMN Mapserver und einer Reihe von Anwendungsbeispielen eine recht gute Dokumentation für die Entwicklung von eigenen Mapserver Anwendungen.

- Die Zope Homepage [http://www.zope.org/] bietet einen ersten Einstieg in Zope und seine Produkte. Relativ viele Informationen über die Interna von Zope und die Möglichkeiten, Anwendungen mit Zope zu entwickeln finden sich im Zope Developers Guide [http://www.zope.org/Documentation/Books/ZDG/current/contents]. Innerhalb des Zope Documentation Projects [http://zdp.zope.org/] findet sich auch ein Kapitel über das Sicherheitskonzept [http://zdp.zope.org/projects/zbook/book/IV/zopesecurfeatures/Drafts/951802917] von Zope.

- Eine gute Übersicht über Freie Lizenzen [http://www.gnu.org/licenses/license-list.html] und ihre Einordnung findet sich auf den Seiten des GNU Projektes.

- Die im Artikel erwähnte Anwendung Strassensuche wird auf dem Geoserver der Bezirksregierung Hannover [https://www.gis-br-h.niedersachsen.de/gis/] bereit gestellt. Sie ist allerdings nur für autorisierte Benutzer zugänglich, so dass der Link direkt zu der Anwendung hier nicht bereit gestellt werden kann.

- Die Freien Geodaten der Stadt Osnabrück werden im Rahmen von Frida [http://frida.intevation.org/] als Shapefile und im txt-Format zum Download angeboten. Außerdem befindet sich dort auch die im Text erwähnte Anwendung Osnabrück.

- Weitere Information zum Thema freie Geographische Informationssystem und freie Geodaten findet man auf der Heimatseite des FreeGIS [http://freegis.org/index.de.html] Projektes.