



Hochschule Neubrandenburg
University of Applied Sciences

Hochschule Neubrandenburg

Studiengang Geoinformatik

Entwicklung des Webauftritts für das Projekt SWACI-II

urn:nbn:de:gbv:519-thesis2008-0075-8

Bachelorarbeit

vorgelegt von: Torsten Witt

geboren am: 24.11.1982

Zur Erlangung des akademischen Grades

„Bachelor of Engineering“ (B.Eng)

Erstprüfer: Prof. Dr. Andreas Wehrenpfennig

Zweitprüfer: Dipl.-Ing. Mirco Tegler

Bearbeitungszeitraum: 21.07 – 12.09.2008

Eidesstattliche Erklärung

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

Neubrandenburg, den 12.09.2008

.....

Torsten Witt

Kurzfassung

Die Internetpräsenz des SWACI-Projektes erhält durch den Einsatz des Content-Management-Systems TYPO3 neue Funktionalitäten. Spezielle Redakteure können besser und schneller fachliche Informationen verbreiten. Angemeldete Nutzer sind gegenüber dem *EOWEB* authentifiziert und erhalten durch eine One-Stop-Shopping-Funktion die Möglichkeit dieses zu nutzen.

Auf der Basis der bisherigen Schnittstellen wurde der Zugriff zu den verschiedenen Produktgruppen in die neue Version angepasst.

Die Integration des Projektes *SIMONE* verschafft Nutzern die Möglichkeit sich aktiv an dem Projekt zu beteiligen. Die Uploads der *SIMONE*-Nutzer sind Grundlage zur Erzeugung von entsprechenden Plots durch GNUPlot. Diese Plots können auf SWACI durch Nutzer erzeugt und dargestellt werden.

Stichworte

Content-Management-System, TYPO3, One-Stop-Shopping, GNUPlot, SWACI, *SIMONE*

Abstract

The homepage of the SWACI-project get's more functions by using the content-management-system TYPO3. Special editors publishes professional information faster and better than one person. Logged in users are authenticated to the *EOWEB* and are able to use it through an one-stop-shopping-function.

The interfaces to access the product groups are based on the original and are adapted to the new version.

The integration of the project *SIMONE* allows users to engage in the project. The uploads for the *SIMONE*-group are the base for the creation of plots with GNUPlot. These users are able to generate and display the plots.

Keywords

Content-Management-System, TYPO3, One-Stop-Shopping, GNUPlot, SWACI, *SIMONE*

Inhaltsverzeichnis

1 Einleitung.....	5
2 Analyse.....	7
2.1 SWACI.....	7
2.2 EOWEB und DIMS.....	9
2.3 Anforderungen.....	10
3 Content-Management-Systeme.....	16
3.1 Informationen zu Content-Management-Systemen.....	16
3.2 Vergleich verschiedener Web-Content-Management-Systeme.....	18
4 Konzeption SWACI-II.....	23
4.1 Konzept SWACI-II.....	23
4.2 Nutzergruppenstruktur.....	25
4.3 One-Stop-Shopping.....	33
4.4 Produkte.....	34
4.5 Integration SIMONE.....	36
4.6 Logfunktion.....	42
5 Umsetzung.....	43
5.1 Installation des Servers.....	43
5.2 Schnittstelle PHP.....	43
5.3 Nutzer.....	44
5.3.1 Authentifizierung.....	44
5.4 One-Stop-Shopping.....	46
5.5 SWACI-II-Produkte.....	48
5.6 SIMONE.....	51
5.7 Ergebnisse.....	56
6 Tests und Abgleich.....	57
7 Zusammenfassung und Ausblick.....	58
Verzeichnis der Tabellen und Abbildungen.....	59
Verzeichnis der Abkürzungen.....	60
Quellen.....	61
Anhang.....	64

1 Einleitung

Das Deutsche Zentrum für Luft- und Raumfahrt ist ein Forschungszentrum der Bundesrepublik Deutschland. Es beschäftigt sich insbesondere mit Problemstellungen aus der Luft- und Raumfahrt, Werkstofftechnologie, medizinische Technik sowie Mikroelektronik, Softwareentwicklung, Verkehr und Kommunikation. Die Institute des DLR befinden sich an 13 Standorten, wie zum Beispiel in Berlin, Oberpfaffenhofen und Köln. Dem DLR steht ein Etat für die eigenen Forschungs- und Entwicklungsarbeiten sowie Betriebsaufgaben von etwa 450 Millionen Euro zur Verfügung, rund ein Drittel davon wird im Wettbewerb erworben. Daneben verwaltet das DLR das nationale Raumfahrtbudget in Höhe von 846 Millionen Euro [DLR01] [MSN01].

Am Standort Neustrelitz befindet sich unter anderem das nationale Bodensegment des Deutschen Fernerkundungsdaten zentrums (DFD), einige Fachabteilungen des Clusters Angewandte Fernerkundung sowie eine externe Arbeitsgruppe des Institutes für Kommunikation und Navigation (IKN). Es arbeiten dort zurzeit etwa 60 Wissenschaftler, Ingenieure und Angestellte. Eine wesentliche Aufgabe des DFD ist der Empfang, die Verarbeitung und Archivierung von Satellitendaten von mehr als 10 Missionen.

Ein gemeinsames Projekt des IKN und des DFD ist SWACI (Space Weather Application Center – Ionosphere) [DLR02]. Das SWACI-Projekt beschäftigt sich mit den Einflüssen des Weltraumwetters auf Kommunikations- und Navigationssysteme. Für dieses Projekt wurde eine Internetpräsenz geschaffen, die den Besuchern einen Einblick in die Weltraumwetterproblematik geben soll. Dabei wird unter anderem auf die Erscheinungen und die Regionen des Weltraumwetters sowie seine Auswirkungen informiert. SWACI bietet neben allgemeinen Informationen zum Weltraumwetter auch aktuelle Informationen über den Zustand der Atmosphäre vielmehr Ionosphäre. Registrierte Nutzer können einen Zugang zu verschiedenen Produktgruppen erhalten. Diese Produktgruppen heißen *GroundBased*, *RadioOccultation* und *TopsideReconstruction*. Dabei handelt es sich um Plots und Textdateien mit den dazugehörigen Werten. Sie beinhalten unter anderem Angaben zur Elektronendichte der Atmosphäre und Laufzeitverzögerungen von Radiosignalen. Die Daten werden für GroundBased durch bodengestützte Messungen gewonnen. Für die Produktgruppe TopsideReconstruction und RadioOccultation sind Satellitenmissionen die Quelle der Daten.

Ein wesentlicher Gegenstand dieser Arbeit ist das Projekt *SIMONE*. *SIMONE* ist ein weiteres Weltraumwetterprojekt und steht für *Sonnen & Ionosphären MONitoring NETzwerk*. Das Ziel des Projektes ist das Heranführen von Schulen an die Untersuchung solarer und terrestrischer Einflüsse auf die Ausbreitung von Telekommunikationssignalen in der Ionosphäre.

Die beteiligten Schulen bekommen einen so genannten SID – Monitor (Sudden Ionospheric Disturbances) kostenlos zur Verfügung gestellt. Dieses Gerät empfängt Radiowellen-signale mit der Frequenz von 24 kHz des Senders Cutler aus Maine und misst die elektrische Spannung des Signals [SIM01]. Für den Betrieb des SID müssen die Schulen selbständig eine Antenne bauen und aufstellen. Der Empfänger wird mittels eines Analog-Digitalwandlers an einen Computer angeschlossen. Eine Software speichert die Daten und erstellt Tagesdateien. Die Messungen, die in die Tagesdateien eingehen, beinhalten neben einem Offset die gemessene Spannung und die jeweilige Uhrzeit. Die Schulen können über eine Internetpräsenz ihre Daten hochladen und sich für bestimmte Daten von Standorten Grafiken daraus generieren lassen.

2 Analyse

Die Analyse beleuchtet die Elemente und Umgebungen des Projektes. Dabei wird ausführlich auf die bisherige SWACI-Internetpräsenz eingegangen. Weitere Punkte ist das EOWEB und DIMS, welche für das Projekt sehr wichtig sind. Anschließend werden aus der bisherigen Gesamtsituation entsprechende Anforderungen für die Weiterentwicklung der Internetpräsenz abgeleitet. Daraus ergeben sich Anforderungen für die Lösung. Diese sind in einem Fazit dargestellt.

2.1 SWACI

Die Inhalte auf der SWACI-Internetpräsenz werden zweisprachig angeboten, in Englisch als Standardsprache und in Deutsch als Übersetzung. Registrierte Nutzer haben nutzergruppenspezifischen Zugang zu verschiedenen Produktgruppen. Neben den Daten mit den exakten Messwerten im Textformat werden korrespondierende Plots in Form von PNG-Dateien angeboten. Diese sind im Dateisystem abgelegt. Die Produkte von GroundBased werden ständig unter dem selben Dateinamen aktualisiert. Die Aktualisierungsrate beträgt fünf Minuten.

Die Produktgruppe RadioOccultation und TopsideReconstruction werden regelmäßig aktualisiert. Für diese Dateien werden neben den Links zu den Plots und den Textdateien auch einige Metadaten angezeigt. Die Datenhaltung ist so organisiert, dass es für jede Nutzergruppe ein Verzeichnis gibt und dort jeweils ein Unterverzeichnis für die jeweilige Produktgruppe.

Die gesamte Internetseite besteht aus statischen HTML-Seiten. Das hat zur Folge, dass Änderungen immer direkt im Quelltext durchgeführt werden müssen. Der Administrator muss sämtliche Änderungen an der Seite selber vornehmen oder sie in die Hände einer vertrauenswürdigen Person geben, da Zugriff auf sämtliche Seiten besteht. Sollte eine weitere Sprache angeboten werden, so sind alle Seiten ein weiteres Mal in der entsprechenden Sprache anzulegen. Der gleiche Aufwand ist zu betreiben, wenn das Design des Internetauftrittes geändert wird. Hierbei müssen sämtliche HTML- und die CSS-Dateien angepasst werden.

Es hat sich auch als nachteilig erwiesen, dass der Administrator der SWACI-Internetpräsenz zur Zeit keinen Zugriff auf die Logdaten hat, da der physische Server fremd verwaltet wird.

Der Zugriffsschutz auf die Daten erfolgt über *htaccess*. Die Benutzer und Passwörter werden in einer *htgroups*- und *htpasswd*-Datei gespeichert. Das Hinzufügen neuer Nutzer und Gruppen ist relativ aufwendig und kostet Zeit. Daher sind für die bisherige Lösung 4 Standardnutzer die als Gruppen fungierten mit Passwort angelegt und die Daten entsprechend versendet worden. Für diese Nutzer mussten die Dateien lokal vom Administrator mit Hilfe des Programms *htpasswd* angelegt werden. Diese sind dann per FTP auf dem Server kopiert worden, da kein direkter Zugriff auf den Server möglich war. Ein weiterer Nachteil dieser Authentifizierung ist, dass keine allgemeinen Daten über die Nutzer zur Verfügung stehen. Die Registrierung der Nutzer erfolgt

durch Ausfüllen des Anmeldeformulars und Weiterleitung der Daten an den Administrator. Dieser legt dann mit den oben genannten Mitteln die Nutzer an.

SWACI-Gruppenname	Beschreibung	Zugriff auf Produktebene
public	Interessierte Nutzer	public
noncommercial	Hochschulen, Universitäten usw.	public, noncommercial
commercial	Nutzer, die für die Daten zahlen	public, noncommercial, commercial
consortium	Ähnlich Commercial, jedoch mit höheren Privilegien	public, noncommercial, commercial, consortium

Tabelle 2.1.1: bisherige Nutzergruppen und Zugriffsrechte

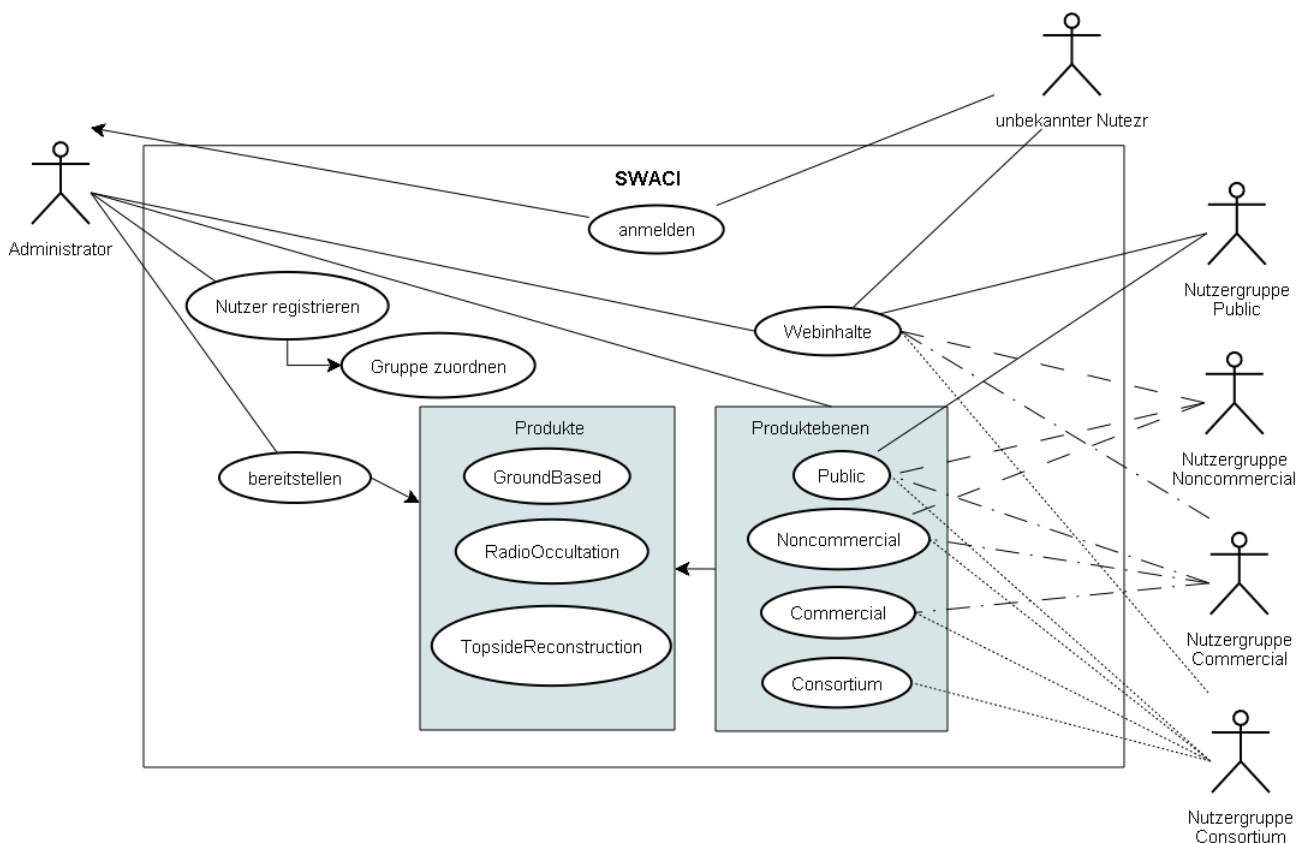


Abbildung 2.1.1: Use-Case-Diagramm SWACI-I

Die Übermittlung von Benutzernamen und Passwörtern zur Anmeldung der Nutzer erfolgte ohne Verschlüsselung durch SSL. Die bisher vorhandenen Nutzergruppen von SWACI sind mit ihren Zugriffsmöglichkeiten in der Tabelle 2.1.1 dargestellt. Die Abbildung 2.1.1 mit einem Use-Case-Diagramm zeigt SWACI und die Interaktion der verschiedenen beteiligten Personen und Nutzergruppen.

2.2 EOWEB und DIMS

EOWEB ist eine Nutzerschnittstelle für den Zugriff auf die Fernerkundungsdatenbibliothek des DFD und steht für *Earth Observation on the WEB*. Ein wesentliches Element dieser Schnittstelle ist der EOS Datenkatalog. Mittels dieses Kataloges können weltweit Fernerkundungsdaten in verteilten Archiven gesucht werden. Diese Daten können direkt aufgerufen oder bestellt werden. Es gibt zu den verschiedenen Produkttypen auch entsprechende Regularien und Preise, die im EOWEB einsehbar sind. Im EOWEB gibt es verschiedene Nutzer und Nutzergruppen, die jeweils Zugriff auf verschiedene Produkte und Funktionen haben [DLR03].

Die Produkte werden durch das *Data Information Management System* (DIMS) zur Verfügung gestellt. Dabei handelt es sich um ein System, das große heterogene Datenbestände mit Raum-Zeit-Bezug verwaltet. Ein Kernmodul des DIMS ist die *Product Library* (PL), zu Deutsch: Produktbibliothek. Es dient als Quelle und Ziel bei der Generierung von Produkten und gewährleistet die Vollständigkeit und Konsistenz dieser [DLR04]. Es arbeitet zu dem als ein Langzeitarchiv, um die Produkte und Rohdaten über sehr große Zeiträume zu erhalten.

2.3 Anforderungen

In den Anforderungen sollen alle notwendigen Neuerungen für die neue SWACI-Internetpräsenz zusammengetragen werden. Die Anordnung erfolgt schwerpunktorientiert. Zur leichteren Verfolgbarkeit im weiteren Prozess erhalten alle Anforderungen eine Kennung in runden Klammern, wie (NI02). Dies bedeutet, dass es sich um eine Nutzeranforderung für die Inhaltselemente mit der laufenden Nummer 02 handelt. Diese Anforderungen sind in Tabelle 2.3.2 zusammengefasst.

2.3.1 Inhaltselemente

Für SWACI-II sind die Inhalte vom Design zu trennen. Das hat den Vorteil einer einfacheren Anpassung des Aussehens von SWACI, ohne mit dem Inhalt in Kontakt zu kommen (NI01). Weiterhin ist die Einführung neuer Sprachen zu erleichtern. Dabei ist nur der Inhalt anzupassen, in konsequenter Loslösung vom Design (NI02).

Die Inhalte von SWACI sollen zur Entlastung des Administrators von Redakteuren mit begrenzten Zugriffsrechten für bestimmte Inhalte gepflegt werden können (NI03). Hierfür muss ein Rechtemanagement die nötigen Zugriffe zu den Inhalten gewährleisten und andere schützen (NI04).

2.3.2 Funktionalität

Im neuen SWACI-Internetauftritt sollen das „look and feel“ der bestehenden Nutzerschnittstellen erhalten bleiben (NF05). Alle bisherigen Produktgruppen und deren Funktionalitäten müssen wie bisher zur Verfügung stehen (NF06). Es ist ein Forum zum Austausch von Informationen und Fragen zwischen Projektpartnern einzurichten (NF07).

Für die SWACI-Nutzer müssen die Funktionalitäten des EOWEB zur Verfügung gestellt werden (NF08). Es ist eine einfache und bequeme Lösung zu schaffen, um den Zugriff auf die Logdaten des Servers zu ermöglichen (NF09). Die Produkte sollen jeweils 31 Tage über SWACI erreichbar sein (NF10). Alle älteren Daten sind über das EOWEB zu finden.

2.3.3 SIMONE

Eine weitere wesentliche Neuerung ist die Integration des Projektes SIMONE. Die Inhalte von SIMONE müssen an geeigneter Stelle platziert werden. Damit die allgemeinen Angaben der Schulen immer aktuell sind, müssen diese eine Möglichkeit haben, ihre Daten zu aktualisieren (NO11).

Da die bisherige Lösung zu SIMONE als Stand-Alone-Lösung entwickelt wurde, muss die vorhandene Datenbank angepasst werden (NO12). SIMONE verfügt über eine Plot-Funktion, die es ermöglicht, aus den einzelnen Messwerten eine Grafik zu erstellen. Dies kann beispielsweise in Form von einem Säulen- und Balkendiagrammen oder in Form eines Graphs geschehen. Die

bestehende Plot-Lösung mittels dem kostenlosen Plotprogramm JGraph¹ ist auf die freie Lösung GNUPlot² umzustellen. Für den Plot ist zu beachten, dass er auch aus mehreren Dateien erzeugt werden kann. Diese entstehen durch Messunterbrechungen (NO13). Ebenso muss die Datenhaltung angepasst werden. Bisher wurden sämtliche Messdaten einzeln in einer Datenbank abgelegt. Diese Messdaten sind als Datei in das Dateisystem abzulegen und zu referenzieren. Die Nutzer müssen in der Lage sein, neue Daten hochzuladen. Der Hochladeprozess ist nur Mitgliedern dieser Gruppe zugänglich zu machen (NO14).

2.3.4 Nutzer

Die bestehenden Gruppen werden angepasst. Die Gruppe *public* wird entfernt und deren Daten für alle zugänglich gemacht. Die Gruppe *noncommercial* ist in *research* zu ändern (NN15).

Für die Integration von SIMONE ist eine weitere Nutzergruppe notwendig. Diese Gruppe hat keinen Zugriff auf andere geschützte Daten (NN16).

Die Tabelle 2.3.1 zeigt die notwendigen Gruppen und ihre Zugriffsrechte, die sich aus den neuen Anforderungen von SWACI ergeben (NN17).

SWACI-Gruppenname	Beschreibung	Beschränkte Zugriffe
research	Hochschulen, Universitäten usw.	Produktgruppe: Research
commercial	Nutzer, die für die Daten zahlen	Produktgruppe: Research, Commercial
consortium	ähnlich <i>commercial</i> , jedoch mit höheren Privilegien	Produktgruppe: Research, Commercial, Consortium
simone	Nutzer des SIMONE-Projekts	SIMONE: Profil, Upload

Tabelle 2.3.1: notwendige Gruppen und Zugriffsrechte

1 <http://www.aditus.nu/jpgraph/>

2 <http://www.gnuplot.info/>

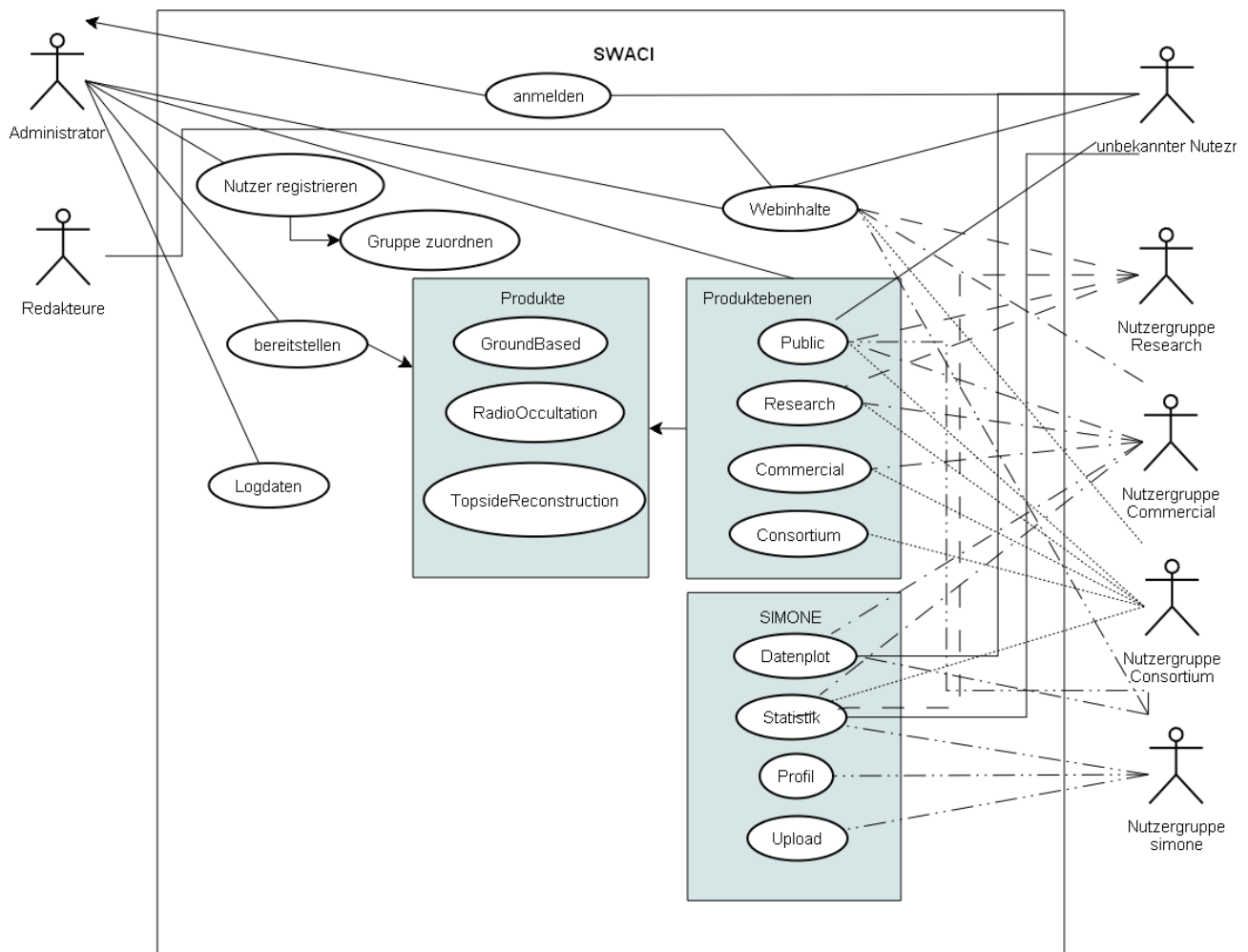


Abbildung 2.3.1: Use-Case-Diagramm SWACI-II

Die Abbildung 2.3.1 stellt die Anwendungsfälle der einzelnen Personen und Gruppen auf der neuen SWACI-Präsenz dar. Es ist zu erkennen, dass die neue Nutzergruppe für SIMONE eingeschränkten Zugriff hat. Ebenso sind die Zugriffe auf SIMONE durch andere Gruppen und nicht authentifizierter Nutzer begrenzt. Für SWACI soll eine andere Form der Authentifizierung gefunden werden, damit auch Daten über die Nutzer zur Verfügung stehen (NN18).

2.3.5 Umsetzung

Die Umsetzung des neuen Internetauftritts ist auf einem System mit SuSE Linux 9.3 Professional durchzuführen. Dabei ist die Installationsumgebung eine virtuelle Maschine (NU18).

2.3.6 Fazit

Die Anforderungen an SWACI-II verlangen die Unabhängigkeit von Inhalt und Design und somit eine moderne Datenhaltung. Die Inhalte können von nicht administrativen Nutzern verfasst und veröffentlicht werden. Die Zugriff auf die Produkte müssen geschützt und für bestimmte Gruppen ermöglicht werden. Die Integration von SIMONE bringt weitere Anforderungen mit sich.

Die Tabelle 2.3.2 mit den Nutzeranforderungen soll zusammenfassend alle oben angeführten Anforderungen für Nutzer auflisten.

Nutzeranforderungen	
Name	Beschreibung
<i>Inhaltselemente</i>	
NI01	Inhalt von Design trennen
NI02	Einführung neuer Sprachen erleichtern
NI03	Redakteure erstellen Inhalte
NI04	Rechtmanagement für Redakteure
<i>Funktionalität</i>	
NF05	Bisherige Schnittstellen erhalten
NF06	Produktzugriff und Funktionalität erhalten
NF07	Forum
NF08	Funktionalität des EOWEB gewährleisten
NF09	Einfacher Zugriff auf Logdaten
NF10	Zugriff auf Daten 31 Tage begrenzen
<i>SIMONE</i>	
NO11	Persönliche Angaben veränderbar gestalten
NO12	Anpassen der Datenbank
NO13	Plot mit GNUPlot umsetzen
NO14	Hochladeprozess für Gruppe SIMONE
<i>Nutzer</i>	
NN15	Nutzergruppen anpassen
NN16	Anlegen einer neuen Nutzergruppe für SIMONE
NN17	Zugriffsrechte einschränken
NN18	Neue Authentifizierung
<i>Umsetzung</i>	
NU19	Installation von SuSE 9.3 Professional auf einer virtuellen Maschine

Tabelle 2.3.2: Nutzeranforderungen

Aus den Nutzeranforderungen ergeben sich für die Umsetzung entsprechende *Systemanforderungen*, die unter Tabelle 2.3.3 zusammengefasst sind.

Systemanforderungen		
Name	Beschreibung	Abgedeckte Nutzeranforderungen
<i>Inhaltselemente</i>		
SI01	Verwaltung von Inhalten	NI01; NI02
SI02	Zentrale Gestaltung des Designs	NI01
SI03	Einführung neuer Sprachen erleichtern	NI02
SI04	Inhalte durch Redakteure erstellbar	NI03
<i>Funktionalität</i>		
SF05	Bisherige Schnittstellen erhalten	NF05
SF06	Integration eines Forums	NF07
SF07	Forum nutzt SWACI-Nutzer	NF07
SF08	One-Stop-Shopping	NF08
SF09	Zugriff auf Logdaten	NF09
SF10	Zugriff auf Daten 31 Tage begrenzen	NF10
<i>SIMONE</i>		
SO11	Zugriff auf SIMONE-Datenbank	NO11, NO13, NO14
SO12	überarbeitete SIMONE-Datenbank	NO12
SO13	Persönliche Angaben veränderbar gestalten	NO11
SO14	Interface zu GNUPlot	NO13
SO15	Upload von Daten	NO14
SO16	Upload auf Gruppe SIMONE begrenzen	NO14
SO17	Sichere Übertragung der Daten durch Archivprozedur	NO14
<i>Nutzer</i>		
SN18	Nutzergruppen anpassen	NN15, NN17
SN19	Nutzergruppe SIMONE anlegen	NN16
SN20	Rechtmanagement für Seiteninhalte	NO14, NF06, NN17
SN21	Nutzergruppen für Redakteure anlegen	NI04
SN22	Rechtmanagement für Redakteure	NI04
SN23	Authentifizierung	NN18
SN24	Nutzerdatenhaltung	NN18
SN25	Authentifizierung über SSL	NN18
<i>Umsetzung</i>		
SU26	Installation von SuSE 9.3 Professional auf einer virtuellen Maschine	NU19
SU27	Anlegen von Nutzern	NU19
SU28	Konfiguration der nötigen Dienste	NU19

Tabelle 2.3.3: Systemanforderungen

Die Tabelle 2.3.3 zeigt die sich ergebenden Systemanforderungen aus den Nutzeranforderungen. Aus der Tabelle wird ersichtlich, dass alle Nutzeranforderungen durch Systemanforderungen abgedeckt sind. Im Anhang [A1] findet sich eine Übersicht mit der Zuordnungen der Nutzer- und Systemanforderungen. Sie soll dem besseren Verständnis und der Übersichtlichkeit dienen.

Um die Systemanforderungen umzusetzen könnte ein *Content-Management-System* (CMS) für einen Großteil geeignet sein. Es kann damit die Authentifizierung, Nutzerdatenhaltung und eine Mehrsprachigkeit umgesetzt werden. Viele CMS-Systeme bieten, auch durch Erweiterungen, die Nutzung eines Forums an. Leider ist es nicht möglich eine One-Stop-Shopping-Funktion, GNUPlot, sowie einige Funktionen für den Zugriff auf die Produkte von SWACI direkt umzusetzen. Dazu gehört der limitierte Zugriff auf die Daten und die Angaben von Metadaten der Produktgruppen RadioOccultation und TopsideReconstruction.

Alternativ könnte eine eigene Lösung auf Basis von PHP oder einer anderen Sprachen entwickelt werden. Für einen solchen Funktionsumfang würde diese Lösung jedoch erheblich mehr Zeit beanspruchen. Zudem werden durch die Entwicklergemeinde der jeweilig genutzten CMS-Lösung ständig Sicherheitslücken des Systems behoben und die Funktionalität erweitert. Eine eigene Lösung führe dazu, selber solche Entwicklungen einpflegen zu müssen. Ein Update des jeweiligen CMS ist dagegen wesentlich einfacher.

Um den SWACI-Nutzern Zugriff auf das EOWEB zu gestatten, müssen diese sich gegenüber dieser Nutzerschnittstelle authentifizieren. Damit dies benutzerfreundlich gelöst wird, muss dieser Prozess für eingeloggte Nutzer im Hintergrund automatisch geschehen. Eine solche Funktion wird auch als *One-Stop-Shopping* bezeichnet.

3 Content-Management-Systeme

Im diesem Kapitel soll auf die Eigenschaften von Content-Management-Systemen eingegangen werden, um eine geeignete Wahl treffen zu können.

3.1 Informationen zu Content-Management-Systemen

Ein Content-Management-System, kurz CMS, ist eine Software, die Inhalte verwaltet. Das heißt, durch sie können Inhalte verschiedener Nutzer erstellt, gelöscht und angeboten werden. Dies geschieht oft mittels einer Datenbank. Ein so genanntes Web-CMS, ist ein Content-Management-System für das Web. Beim Aufruf einer solchen Internetseite werden die Inhalte anhand von bestimmten Vorlagen, so genannten Templates, entsprechend angeordnet und dargestellt.

Bei der Erstellung der Inhalte wird zwischen Autor, Editor und Webmaster unterschieden. Dies wird auch als Publishing-Prozess bezeichnet. Zudem gibt es oft die Möglichkeit eines Content-Life-Cycle-Managements die es ermöglicht, Inhalte zeitlich begrenzt zur Verfügung zu stellen. Das können beispielsweise Nachrichten über anstehende Termine sein, die nach Ablauf des Datums aus der Webseite entfernt werden. Eine weitere wesentliche Funktion eines Web-CMS ist die einfache Umsetzung einer mehrsprachigen Website. Zudem kann eine zielgruppengerechte Darstellung von Inhalten ermöglicht werden, wie z. B. die Barrierefreiheit. Das Stichwort Barrierefreiheit setzt gleichzeitig die Einhaltung von W3C-Standards voraus. Für W3C-kompatible Webseiten muss ein Template angelegt werden, welches den Standards entspricht. Der Inhalt kann dann unabhängig davon geändert werden. Die nachfolgende Tabelle fasst die wichtigsten Eigenschaften von Content-Management-Systemen zusammen.

Zusammenfassung häufiger Eigenschaften

- unterstützt Publishing-Prozess,
- Content-Life-Cycle-Management,
- nutzen Datenbanken,
- Nutzergruppen für Autoren,
- Nutzergruppen für Webseitenbesucher,
- Trennung von Inhalt, Datenstruktur, Layout,
- Einbindung externer Datenquellen und Informationsdienste,
- Vorlagenbereitstellung,
- modulare Erweiterbarkeit

Ein Nachteil bei der Arbeit mit einem CMS ist, dass das Erstellen der ersten Seiten wesentlich mehr Aufwand erfordert. Ein weiterer Nachteil ist, dass für den Server höhere Anforderungen an die Software gestellt werden. Neben einer üblichen Unterstützung von PHP³ kommen zusätzliche Module hinzu, wie eine Datenbank und eventuell zusätzlich benötigte Software wie beispielsweise die Grafikbibliothek GDlib⁴ zum Erzeugen von Grafiken. Oder es wird das Programm ImageMagick⁵ benötigt. Dieses dient ebenfalls dazu Bilder zu generieren und sie anzuzeigen. Bei kommerziellen Webservice-Anbietern hat das oft einen Umstieg auf einen höherwertigen und damit teureren Vertrag zur Folge. Da dies vom DLR selbst realisiert wird, stellt es keinen Nachteil dar. Jedoch muss das CMS und die installierte Software stets aktuell gehalten werden, um mögliche Sicherheitslücken zu schließen. Solche Sicherheitsupdates müssen auch für verwendete Erweiterungen durchgeführt werden. Nicht zu unterschätzen sind die gestiegenen Anforderungen an die Hardware. Das Abrufen und Erzeugen dynamischer Inhalte, insbesondere das Erzeugen von Grafiken, fordert mehr CPU-Zeit als auch eine erhöhte Speicherauslastung. Eine solch hohe Last kann dazu führen, dass sich die Auslieferung der Seite verzögert oder dass es in Spitzenzeiten zu einem Timeout kommt.

3.1.1 Fazit

Mit Hilfe eines CMS können nicht nur Inhalte dynamisch und vom Design getrennt erzeugt, sondern auch Nutzer erstellt und verwaltet werden. Diese erhalten Zugriff für die Bereiche, die für sie freigegeben sind. Mögliche Erweiterungen des CMS geben SWACI neue Möglichkeiten, Inhalte und Funktionen zur Verfügung zu stellen.

Das Erstellen der Inhalte durch Redakteure hat den Vorteil der Entlastung des Administrators. Zudem können einige Redakteure besser und detaillierter über die verschiedenen Themen informieren als der Administrator, der sich erst mit dem Thema befassen müsste oder nicht immer zur Verfügung steht. Die Funktionen, die durch das gewählte CMS nicht umsetzbar sind, müssen durch eigene Skripte oder Programme realisiert werden.

3 <http://www.php.net/>

4 http://www.libgd.org/Main_Page

5 <http://www.imagemagick.org/>

3.2 Vergleich verschiedener Web-Content-Management-Systeme

Es gibt sehr viele verschiedene Web-Content-Management-Systeme. Für die engere Auswahl kamen nur Systeme in Frage, die kostenlos zur Verfügung standen, da für kostenpflichtige Lösungen keine Mittel zur Verfügung stehen. Zur Zeit gelten als besonders Leistungsstark Drupal⁶, Joomla!⁷ und TYPO3⁸.

Vorweg müssen noch einige gebräuchliche Begriffe für Web-Content-Management-Systeme geklärt werden. Es gibt einen Frontend-Bereich. Dies ist die eigentliche Internetseite, die für alle oder angemeldete Nutzer zur Verfügung steht. Die registrierten Nutzer werden Frontend-Nutzer bzw. Frontend-Nutzergruppen genannt [siehe Kapitel 4.2.1]. Wenn von dem Backend-Bereich gesprochen wird, dann ist das der administrative Teil eines Web-CMS. Es ist meist in Form einer Internetseite mit einem separaten Login gestaltet. Für den Backend-Bereich gibt es ebenfalls Nutzer [siehe Kapitel 4.2.2] und Nutzergruppen.

3.2.1 Joomla!

Joomla! ist ein Web-Content-Management-System, welches für kleine Projekte geeignet ist. Es zeichnet sich durch seine kurze Einarbeitungszeit und eine einfache Installation aus. Daher ist es besonders für Einsteiger geeignet. Als Datenbank wird ausschließlich MySQL⁹ genutzt. Es ermöglicht eine Druckfunktion für die jeweilige Internetseite. Dadurch kann die Seite für einen Ausdruck optimiert werden. Ein weiterer Vorteil ist die Möglichkeit, die Seite auch als PDF-Datei zu exportieren, ohne dazu eine Erweiterung installieren zu müssen.

Der geringe Administrationsaufwand, für den Joomla! auch bekannt ist, kommt durch die relativ geringe Administrationsfähigkeit zu Stande. Für Frontend-Nutzer gibt es die Möglichkeit, Nutzer zwischen den Rollen, also vorgefertigten Nutzergruppen, *Registered*, *Author*, *Editor* und *Publisher* zu unterscheiden. *Registered* ist eine Gruppe, die Zugang zu nicht öffentlichen Daten hat. Es können aber keine unterschiedlichen Gruppen dieses Typs angelegt werden. Deshalb ist es nicht möglich, einer weiteren Gruppe diesen Typs begrenzte Informationen zur Verfügung zu stellen und diese vor dem Zugriff einer anderen Gruppe zu schützen. Die von den Autoren erstellten Inhalte werden vom Publisher veröffentlicht. Der Editor kann von allen Nutzern Inhalte bearbeiten und neue erstellen. Ein weiterer Nachteil ist, dass Redakteure nicht auf bestimmte Abschnitte der Seite beschränkt werden können. Jeder kann frei wählbar Artikel verfassen. Ob diese veröffentlicht werden, hängt vom Publisher ab. Im Backend-Bereich befindet sich der Administrator und ein Manager. Der Manager kann die Artikel und Menüs bearbeiten. Eine Untergliederung ist hier nicht

6 <http://www.drupal.org/>

7 <http://www.joomla.org/>

8 <http://www.typo3.org/>

9 <http://www.mysql.de/>

möglich. Der Administrator und ein ihm übergeordneter Superadministrator übernehmen alle restlichen administrativen Aufgaben.

Zur Zeit ist es mit Joomla! nicht möglich, W3C-konformen Inhalt zu generieren. Ein weiteres Problem ist die fehlende Versionierung. Das macht es besonders schwierig Änderungen wieder zurück zu nehmen. Der nicht intuitive Backend-Bereich erfordert eine längere Einarbeitungszeit.

Joomla! ist ein relativ junges CMS. Es ist zwar 2005 aus dem Mambo-CMS hervorgegangen, besteht aber seit Version 1.5 aus einem eigenem Code. Daher ist damit zu rechnen, dass einige Mängel in nächster Zeit behoben werden. Der Grundsatz der Einfachheit lässt aber darauf schließen, dass der Funktionsumfang nicht wesentlich erweitert wird.

3.2.2 Drupal

Drupal ist ein in PHP geschriebenes Web-CMS. Es greift wahlweise auf eine MySQL- oder eine PostgreSQL-Datenbank¹⁰ zu. Die Installation ist sehr einfach und kann wahlweise in der eigenen Sprache durchgeführt werden.

Die Funktionalität lässt sich durch zahlreiche Erweiterungen verbessern. Da es PHP und keine eigene Skriptsprache verwendet, ist die Einarbeitungszeit stark verkürzt. In den Templates lässt sich ein PHP-Code direkt ausführen. Ungewollte Änderungen an Inhalten können über ein Versionierungssystem rückgängig gemacht werden.

Zwar gibt es wie bei Joomla! für Backend-Nutzer vordefinierte Rollen aber nicht die Möglichkeit, sie auf bestimmte Seiten zu beschränken. Auch fehlt hier die Möglichkeit Frontend-Nutzergruppen zu unterscheiden. Der Frontend ist mehrsprachig gestaltbar. Dabei bietet Drupal die Möglichkeit einer linksläufigen Schreibrichtung an, um beispielsweise Inhalte in Arabisch umzusetzen. Damit effektiv mit Drupal gearbeitet werden kann, erfordert es entsprechende Kenntnisse in HTML, CSS und PHP. Für die meiste Redakteursarbeit sind diese Kenntnisse jedoch nicht notwendig. Auch hier ist zu bemängeln, dass das Backend nicht intuitiv gestaltet ist.

3.2.3 TYPO3

TYPO3 ist ein sehr funktionsreiches OpenSource-CMS. Daher bietet es viele Vorteile. Einer davon ist die Möglichkeit der schnellen Umsetzung einer Idee. Dies unterstreichen weitere Vorteile. Es ist sehr intuitiv und es werden zum Arbeiten von TYPO3 weder HTML-, CSS- noch PHP-Kenntnisse benötigt. Weitere Pluspunkte sind die Multidomainfähigkeit und die Möglichkeit der Versionierung. Dabei werden allerdings nur Änderungen berücksichtigt, die in kürzester Zeit entstanden. Für eine komplexere Versionierung gibt es entsprechende Erweiterungen. Auch bei TYPO3 ist es möglich, Seiten in verschiedenen Sprachen anzubieten.

¹⁰ <http://www.postgresql.org/>

Die größten Vorteile gegenüber den anderen Kandidaten liegen im Rechtemanagement. Es können eigene Frontend-Nutzergruppen angelegt werden. Inhalte sind einer oder mehreren Gruppen zugänglich. Um weitere Rechte zu erhalten, können Gruppen von untergeordneten Gruppen Rechte erben. Das kann den Administrationsaufwand später verkürzen, da es möglich ist, eine Gruppe nur zur Vererbung von Eigenschaften und Rechten zu erstellen. Für den Backend-Bereich gilt ebenfalls die Möglichkeit der Vererbung. Im Gegensatz zu Joomla! und Drupal können den Redakteuren bestimmte Seitenbäume, also Abschnitte der Internetseite, gezielt zugeordnet werden. Diese Abschnitte können durch das gezielte Setzen von Rechten und durch Freischalten von Modulen für diese Nutzer durch Redakteure bearbeitet werden. So ist es möglich, verschiedene Profile für Autoren anzulegen. Das so genannte Taskcenter bietet die Möglichkeit Redakteuren Aufgaben zuzuweisen. Es dient dazu Projekte anzulegen und zu managen. Mitglieder können dann offene Seiten editieren.

Der hohe Umfang an Funktionen erhöht die Einarbeitungszeit für den Administrator und ist daher in gewisser Weise ein Mangel. Dies wird durch die eigene Skriptsprache TypoScript noch verstärkt.

3.2.4 Zusammenfassung

Die verschiedenen CMS-Systeme zeigen sowohl Gemeinsamkeiten als auch Unterschiede. Besonders groß sind diese beim Rechtemanagement. Hier soll noch einmal auf die wichtigsten Kriterien und Eigenschaften eingegangen werden, um die richtige Wahl treffen zu können.

Das richtige CMS muss es ermöglichen, Inhalte oder Bereiche nur für bestimmte Frontend-Nutzergruppen zur Verfügung zu stellen. Ebenso ist es notwendig Funktionen, die das CMS nicht zur Verfügung stellt, durch eigene Skripte lösen zu können. Den Redakteuren soll die Einarbeitungsphase relativ leicht fallen, um schnell dem Administrator Arbeit abzunehmen. Jedoch sollten diese Redakteure Zugang mit eingeschränkten Rechten erhalten. Da die bisherige Lösung für SWACI zweisprachig war, soll auch diese die Möglichkeit besitzen, Inhalte zweisprachig anzubieten.

Die nachfolgende Tabelle 3.2.1 soll noch einmal die wesentlichen Eigenschaften der betrachteten CMSs für eine Entscheidung gegenüber stellen. Dabei werden auch bisher nicht genannte Eigenschaften verglichen.

Kriterium	Joomla!	Drupal	TYPO3
allgemein			
Lizenz	GPL	GPL	MIT
Kosten	kostenlos	kostenlos	kostenlos
Datenbank	MySQL	MySQL, Oracle, PostgreSQL	MySQL, Oracle, PostgreSQL
erweiterbar	Ja	Ja	Ja
Eigene Skriptsprache	Nein	Nein	TypoScript
Metadaten			
Manuelle Eingabe von Schlagwörtern	Ja	Ja	Ja
Zeitliche Steuerung	Ja	Nein	Ja
Integration eigener Skripte	Ja	Nein	Ja
Redaktion			
Backend mehrsprachig	Ja	Ja	Ja
Versionierung	Nein	Ja	Ja
WYSIWYG	Ja	Ja	Ja
Vorschaufunktion	Ja	Ja	Ja
Caching-Mechanismus	Ja	Ja	Ja
Administration			
Rechtmanagement	rollenbasiert	rollenbasiert	Eigene Rollen und Nutzer
Administration über Browser	Ja	Ja	Ja
Sandbox	Nein	Nein	Ja
Inhalte			
Barrierefreiheit	Nein	möglich	möglich
Suchmaschinen- und benutzerfreundliche URLs	Ja	Ja	Ja
Interne Suchmaschine	Ja	Ja	Ja
Suche innerhalb von Dateien	Nein	Nein	PDF, DOC, ODT, XLS
Mehrsprachigkeit	Nur durch Erweiterungen	Ja	Ja

Tabelle 3.2.1: Vergleich verschiedener CMS

[CMS04] [CMS05] [CMS08] [CMS09]

3.2.5 Fazit

Joomla! ist für die Umsetzung der Aufgabe ungeeignet, da neben dem mangelndem Rechtemanagement auch kein W3C-valider Code möglich ist. Drupal hat ebenfalls den Mangel des Rechtemanagements. Es bietet aber durch zahlreiche Erweiterungen die Möglichkeit die Funktionalität zu verbessern.

TYPO3 ist ein Web-CMS mit sehr vielen Möglichkeiten. Es können damit nicht nur die bestehenden Anforderungen umgesetzt werden, sondern bietet auch genügend Raum für Weiterentwicklungen. Durch die große Anzahl an Erweiterungen, die schon heute verfügbar sind, kann SWACI in seinem Funktionsumfang leicht erweitert werden. Neue Nutzer für den Frontend-Bereich können problemlos eingefügt und deren Zugriffsrechte geregelt werden. Eine hierarchische Strukturierung der Nutzerrechte durch Untergruppen macht es möglich, Bereiche zu erstellen, die für mehrere Nutzergruppen zugänglich sind. Dies gibt zusätzliche Möglichkeiten zur Integration neuer Projekte. Durch eine Integration neuer Projekte bietet es sich an, neue Backend-Nutzer als Redakteure für diese Inhalte zu erstellen.

Die Backend-Nutzer, somit die Redakteure, können schnell ihre Arbeit aufnehmen, da sie keine Kenntnisse in PHP oder HTML benötigen. Die Umsetzung von SWACI-II wird mit dem freien Content-Management-System TYPO3 umgesetzt, da es allen Anforderungen entspricht.

4 Konzeption SWACI-II

4.1 Konzept SWACI-II

Dieses Kapitel soll Aufschluss geben, wie die konzeptionellen Lösungen auf die Anforderungen der neuen SWACI-Internetpräsenz gestaltet sind. Außerdem gibt es Auskunft, in welchem Zusammenhang SWACI-II mit beteiligten Diensten stehen soll, da es keine autarke Lösung ist.

Einer dieser Dienste mit den SWACI-II im Zusammenhang steht, ist das EOWEB. Damit seine Funktionalität zur Verfügung gestellt werden kann, ist eine One-Stop-Shopping-Funktion vorgesehen.

Die Abbildung 4.1.1 zeigt, dass DIMS die Produkte von SWACI-II aktualisiert. Das IKN liefert neben DIMS ebenfalls Daten für die Produktgruppen. Das EOWEB dient als Schnittstelle zu DIMS.

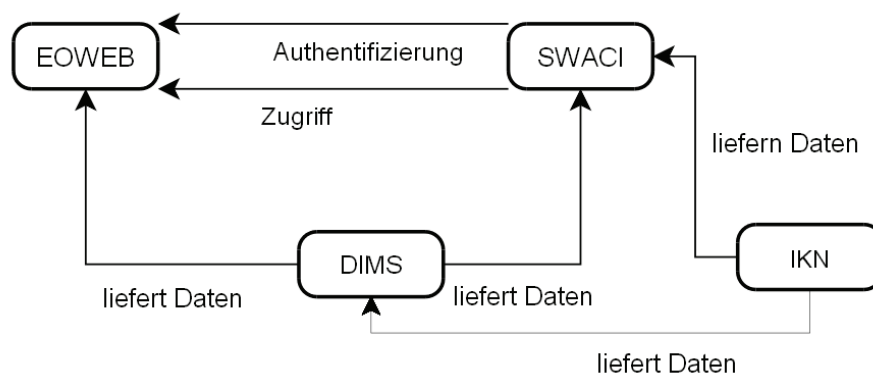


Abbildung 4.1.1: SWACI und andere Kommunikationspartner

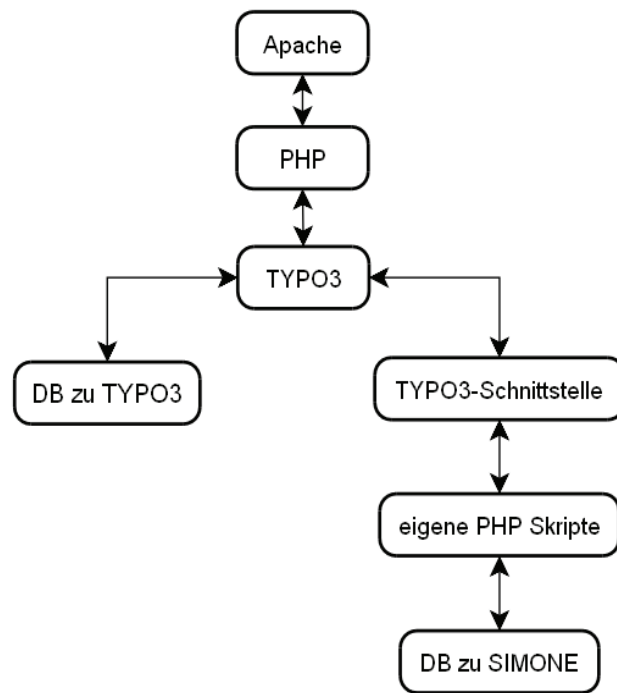


Abbildung 4.1.2: Komponenten von SWACi

In Abbildung 4.1.2 zeigt sich, dass die SWACi-Internetpräsenz durch einen Apache-Webserver mit PHP-Unterstützung betrieben wird. PHP ist ein notwendiger Interpreter für TYPO3. Daher wird dieser vorhandene serverseitige Interpreter für eigene Skripte genutzt, die die Funktionalität von TYPO3 ergänzen sollen. Um dies zu realisieren, muss eine Schnittstelle zwischen TYPO3 und den PHP-Skripten geschaffen werden. Für die Integration von SIMONE sowie für die Produkte sind PHP-Skripte notwendig. Für die Produkte sind diese wichtig, um beispielsweise die Metadaten auszulesen, die in den Dateien stehen.

Als Standardsprache ist bei SWACi bisher Englisch gewählt worden. Im Hinblick darauf, dass durch SIMONE besonders viele deutsche Nutzer die Seite besuchen, sollen sich die Inhalte daher anpassen. Ein weiterer Grund ist die Mehrsprachigkeit in TYPO3. Wenn als Standardsprache Deutsch gewählt ist, kann die Übersetzungsunterstützung innerhalb TYPO3s genutzt werden. Die Seitenstruktur soll sich vornehmlich an der bestehenden Seite orientieren. Die Inhalte werden übernommen.

4.2 Nutzergruppenstruktur

Nun wird auf die verschiedenen Nutzergruppen eingegangen. Dies umfasst die Gruppen für den Frontend und den Backend-Bereich. Für den Frontendbereich ist zusätzlich die Art der Datenhaltung und Registrierung ein wesentlicher Bestandteil dieses Kapitels.

4.2.1 Frontend-Nutzer

Die eigentlichen Nutzer von SWACI-II sind die Frontend-Nutzer. Diese werden unter TYPO3 in die Tabelle *fe_users* abgelegt. Mit dem Wissen des Aufbaus dieser Tabelle können Nutzer auch durch eigene Skripte oder Programme zur Registrierung in die Datenbank von TYPO3 aufgenommen werden.

Tabellenstrukturen

In der Tabelle *fe_users* findet sich neben Nutzernamen, Passwort, UID (ID des Nutzers) und persönlichen Angaben über den Nutzer eine *pid*. Die *pid* gibt die eindeutige Identifikation der Seite an, die im TYPO3-Projekt die Nutzer verwaltet. Diese Angaben sind wichtig um einen Nutzer erfolgreich in die Datenbank aufnehmen zu können. Diese Seite ist nur im Backend erreichbar. Weitere Angaben, die für diese Thematik unbedeutend sind, sind nicht Gegenstand dieser Betrachtung. In der Tabelle *fe_groups* werden die Nutzergruppen verwaltet. Abbildung 4.2.1 zeigt die Beziehung zwischen *fe_users* und *fe_groups*.

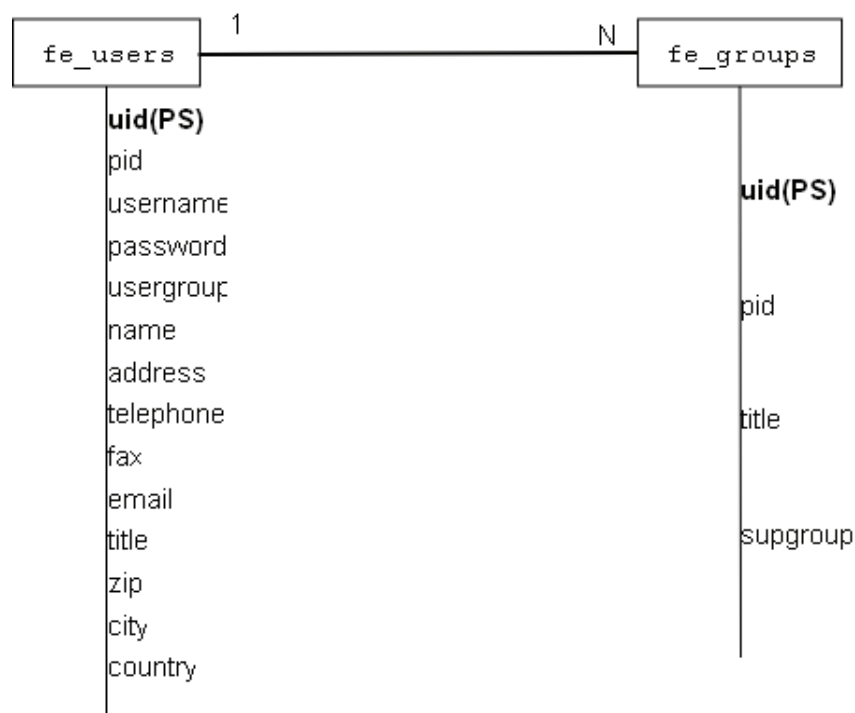


Abbildung 4.2.1: Auszug der Nutzerverwaltung in TYPO3

4.2.1.1 Nutzergruppen

Gruppenname	Beschreibung	Zugriff
research	Hochschulen, Universitäten usw.	• Produktebene: Research • Intern
commercial	Nutzer, die für die Daten zahlen.	• Produktebene: Research, Commercial • Intern
consortium	Ähnlich <i>commercial</i> , jedoch mit höheren Privilegien	• Produktebene: Research, Commercial, Consortium • Intern
simone	Nutzer des SIMONE-Projekts	• SIMONE: Profil, Upload • Intern

Tabelle 4.2.1: Nutzergruppen

Die Tabelle 4.2.1 zeigt die zu erstellenden Nutzergruppen. Sie gibt auch Hinweise, warum Nutzer einer bestimmten Gruppe zugeordnet sind. In der Spalte Zugriff sind die geschützten Seiten aufgeführt zu denen die Mitglieder dieser Gruppe Zugriff haben. Mit *Intern* ist eine Seite gemeint, die allgemein für registrierte Nutzer zur Verfügung steht, um beispielsweise nicht öffentliche Nachrichten zentral zu verbreiten. Damit die Funktionalität des EOWEB genutzt werden kann, müssen die Nutzer auch dort bekannt sein.

4.2.1.2 Registrierungsverfahren

Für die Registrierung ist ein auf der SWACI-Seite zu findendes Registrierungsformular auszufüllen. Dieses Formular beinhaltet die Personalien (Name, Adresse, Telefon, E-Mail usw.) als auch die gewünschte Nutzergruppe. Die Informationen werden an eine dafür zuständige Person gesendet. Sie soll der Einfachheit halber *Nutzeragent* genannt werden. Dieser entscheidet, ob dem Registrierungsgesuch stattgegeben und der gewählten Nutzergruppe entsprochen wird.

Möglichkeiten zur Haltung der Benutzerdaten

Die Registrierung erfolgt nach Zustimmung des Nutzeragenten. Da es notwendig ist, dass Nutzer auch dem EOWEB bekannt sein müssen, gibt es verschiedene Ansätze für eine Lösung. Nachfolgend soll beleuchtet werden, in wie fern es Vor- oder Nachteile hat, die Nutzerhaltung nur in einem der beiden Webservices (SWACI-II und EOWEB) oder in beiden vorzunehmen.

1. Datenhaltung durch einen Webservice:

1. Daten im EOWEB

Bei dieser Methode würden die Nutzer ausschließlich im EOWEB gehalten werden. Mit TYPO3 können Seiten oder Seiteninhalte in jeglicher Form nur bestimmten Nutzern, Nutzergruppen oder generell eingeloggten Nutzern zur Verfügung gestellt oder entzogen werden. Diese wesentliche Option würde dann nicht mehr zur Verfügung stehen. Außerdem ist es nicht möglich, die Datenbank des EOWEB direkt anzusprechen und Informationen weiter zu leiten. Die Verwaltung der SIMONE-Daten müsste dieser Lösung angepasst werden. Die Nutzer wären allerdings in der Lage auf die Kollektionen im EOWEB zuzugreifen.

Registrierung:

Es wird dazu ein entsprechendes Formular zur Erstellung von Nutzern innerhalb des EOWEBs vom Nutzeragenten ausgefüllt.

2. Daten in SWACI-II

In Falle, dass die Nutzerhaltung nur in SWACI-II erfolgt, ist kein Zugriff auf das EOWEB und seine Kollektionen möglich. Damit die Nutzer trotzdem das EOWEB nutzen können, müssten sie sich ebenfalls separat beim EOWEB registrieren und authentifizieren.

Registrierung:

Die Erstellung der Nutzer erfolgt im Backend-Bereich von TYPO3.

2. Datenhaltung durch zwei Webservices

Das bedeutet, dass die Nutzer im EOWEB und in SWACI-II angelegt werden. Bei dieser Lösung muss die Konsistenz der beiden Datenbanken gewahrt bleiben. Diese Lösung hat den Vorteil, dass es den Zugriffsschutz durch TYPO3 ermöglicht. Außerdem ist es durch eine One-Stop-Shopping-Funktion möglich, auf die Kollektionen im EOWEB mit entsprechenden Rechten zuzugreifen.

Die Datenhaltung durch zwei Webservices hat einen weiteren Vorteil. So können Nutzergruppen angelegt werden, die es im EOWEB nicht gibt und die damit keinen Zugriff auf deren Funktionalitäten haben.

Registrierung:

Bei einer positiven Entscheidung des Nutzeragenten trägt dieser den Nutzer in die Datenbank des EOWEB und in die von TYPO3 ein. Sollte er Mitglied der Gruppe SIMONE sein, wird die *UID* in *benutzer* der Tabelle *schule* in die SIMONE-Datenbank hinzugefügt. Dieser Ablauf wird noch einmal in der Abbildung 4.2.2 grafisch verdeutlicht.

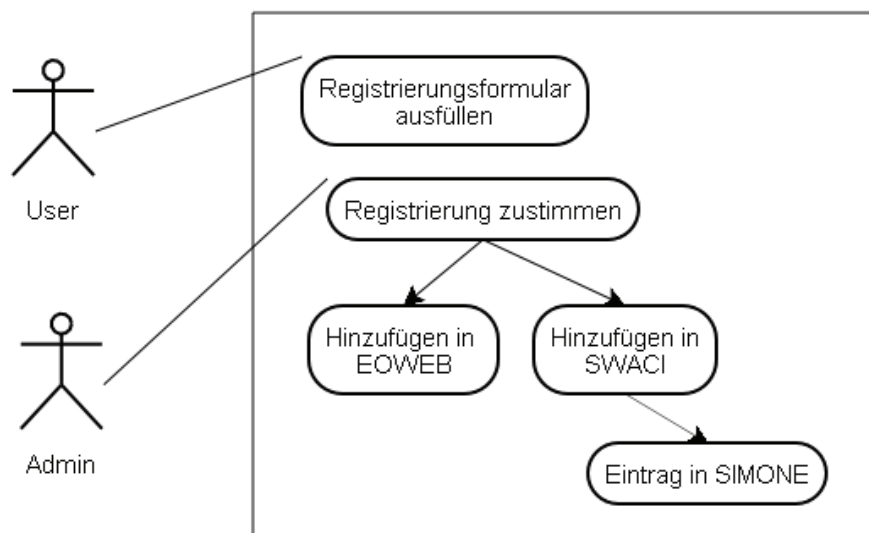


Abbildung 4.2.2: Registrierung

Fazit

Es ist für SWACI-II leider nicht möglich, einen direkten Zugriff auf die Datenbank des EOWEB zu erhalten. Das ist vor allem der Sicherheit geschuldet. Daher ist es notwendig, dass die Nutzer auf beiden Seiten angelegt werden. Die Registrierung der Nutzer erfolgt in beiden Webservices durch den Nutzeragenten.

4.2.1.3 Authentifizierung

Damit die Konsistenz zwischen den beiden Webservices gewährleistet wird, müssen sich die Nutzer gegenüber dem EOWEB und SWACI-II authentifizieren. Dabei ist zu prüfen, ob der Nutzer in beiden Systemen bekannt ist. Sollte es zu Inkonsistenzen kommen, muss darauf reagiert werden. Die Reaktion kann eine automatische Registrierung des Nutzers in SWACI-II bedeuten, falls dieser dem EOWEB bekannt ist.

Vor- und Nachteile einer automatisierten Nutzerregistrierung

Ein automatisiertes Registrierungsverfahren bedeutet weniger Aufwand gegenüber einem per Hand durchgeführten Verfahren. Allerdings ist diese bei wenigen Neuregistrierungen, wie es bisher bei SWACI-II der Fall ist, zu vernachlässigen. Ein viel wichtigerer Vorteil ist die Konsistenz. Es kann nicht passieren, dass Nutzer mit unterschiedlichen Angaben von Benutzernamen und Passwort in den Datenbanken existieren. Abhilfe kann aber auch hier geschaffen werden, in dem ein Formular für den Nutzeragenten zur Verfügung gestellt wird, welches zur selben Zeit die Nutzer parallel im EOWEB und in SWACI-II anlegt.

Da SWACI-II keinen direkten Zugriff auf die Datenbank des EOWEB hat, würden bei einer automatischen Registrierung einige Metainformationen zu den Nutzern fehlen. Bei einem manuellen Verfahren, kann entschieden werden, welche Informationen ergänzend einzutragen sind. Die folgenden Abbildungen geben Auskunft über den schematischen Ablauf der Authentifizierungsmethoden.

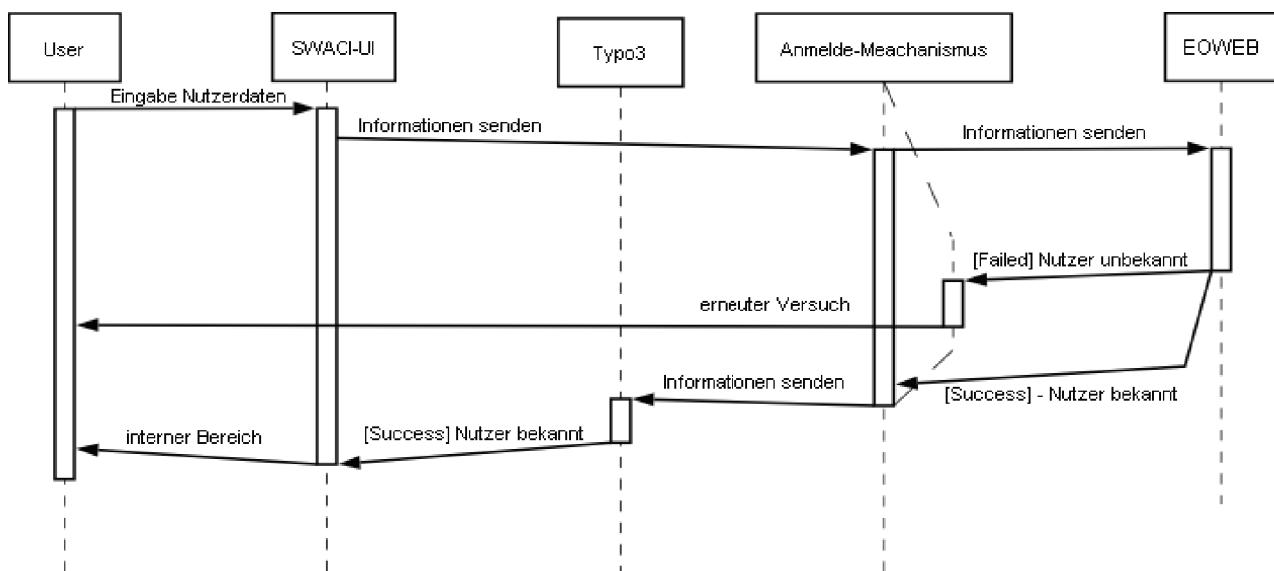


Abbildung 4.2.3: Authentifizierung ohne automatischer Registrierung

Abbildung 4.2.3 zeigt den schematischen Ablauf einer Authentifizierung ohne eine automatisierte Registrierung.

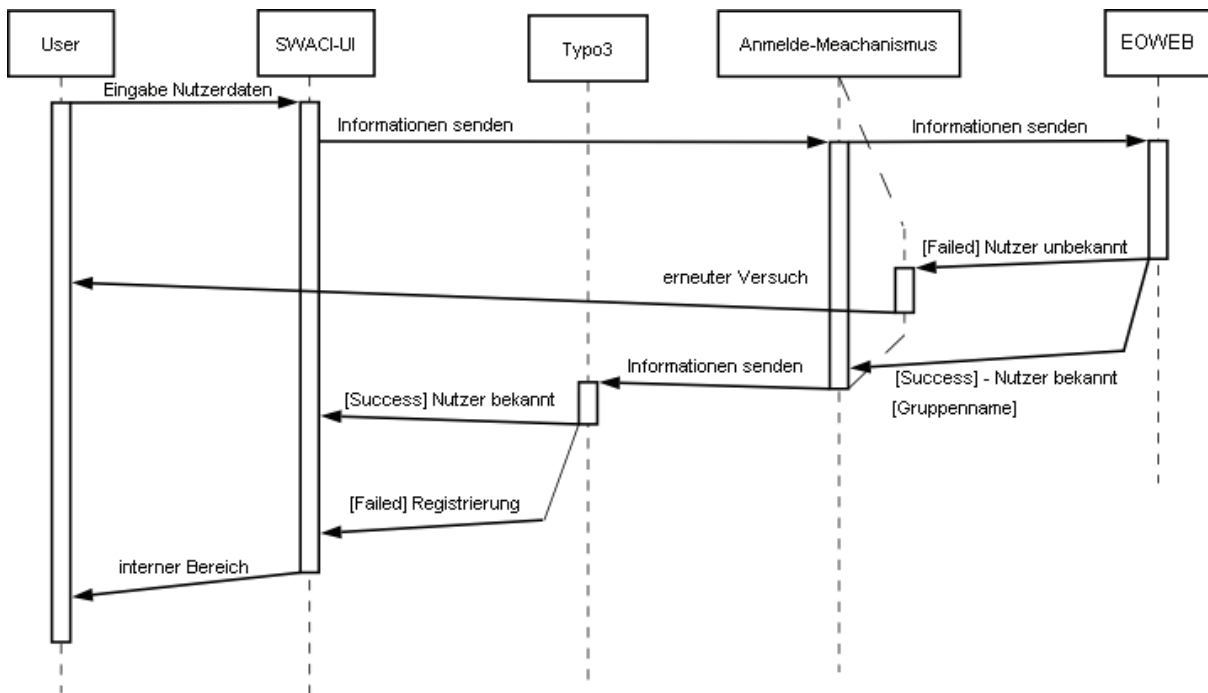


Abbildung 4.2.4: Authentifizierung mit automatischer Registrierung

Abbildung 4.2.4 zeigt hingegen den Ablauf einer Authentifizierung mit einer automatisierten Registrierung.

Fazit

Zur Zeit ist es leider nicht möglich, dass das EOWEB die Gruppenzugehörigkeit der bestehenden Nutzer, bei der Authentifizierungsschnittstelle mitsenden kann. Daher muss für die praktische Umsetzung, auf eine automatische Registrierung zunächst verzichtet werden.

4.2.2 Backend-Nutzer

Wie zu Beginn erwähnt, können den Backend-Nutzern und den Backend-Nutzergruppen verschiedene Rechte zugewiesen werden. Dazu gehören die Verzeichnisfreigaben. Dies sind Verzeichnisse, auf die Gruppen zugreifen, um Bilder für ihre Inhalte hinzuzufügen oder um auf Dateien verlinken zu können. Es ist ebenfalls für diese Gruppen möglich, neue Daten hochzuladen, insofern die entsprechenden Rechte gesetzt sind. Weiterhin können den Gruppen oder direkt den Nutzern Datenbankfreigaben zugeordnet werden, welche Seitenbäume der Internetseite darstellen. Zu jeder Seite können wie in UNIX Rechte vergeben werden, also gegenüber einem Besitzer, der Gruppe und allen anderen. Über Zugriffslisten ist es möglich den Nutzern und Gruppen die verfügbaren Module zuzuordnen. Die Backend-Nutzer können dann mit den ihnen zugeordneten Modulen arbeiten. Es ist ebenfalls für einzelne Module möglich einige Schaltflächen und Optionsfelder Nutzern oder Gruppen vorzuenthalten.

Innerhalb von TYPO3 gibt es keine Backend-Nutzergruppe für Administratoren. Einzelne Nutzer können Administrator werden. Der Administrator hat dann allerdings einen uneingeschränkten Zugriff.

Gruppen

Für den Backend-Bereich ist es sinnvoll, für die neue Struktur von SWACI drei Nutzergruppen einzurichten. Zum einen gibt es die Gruppe *simone*. Diese Gruppe soll auf den entsprechenden Abschnitt der Seite beschränkt bleiben. Die Gruppe *redaktion* ist eine übergeordnete Gruppe von Redakteuren, die auf den kompletten Seitenbaum Zugriff hat. Sie kann auch überall Änderungen vornehmen. Mit der Gruppe *intern*, sollen Administrationsaufgaben vereinfacht werden. Sie dient dazu allgemeine Rechte an andere Gruppen zu vererben. Ihr werden keine Nutzer direkt zugeordnet. Die folgenden Tabellen 4.2.2 – 4.2.4 geben übersichtlich Aufschluss über die zu erstellenden Gruppen und ihre Eigenschaften.

Name	intern
Untergruppen	<i>Keine</i>
Verzeichnisfreigaben	pics

Tabelle 4.2.2: Gruppe intern

Name		simone
Zugriffsliste		
Erlaubte Module		
	Web	Seite, Anzeigen, Liste, Info, Funktionen
	Datei	Dateiliste
	Benutzerwerkzeuge	Aufgaben, Benutzereinstellungen, Arbeitsumgebung
Untergruppen		intern
Verzeichnisfreigaben		simone

Tabelle 4.2.3: Gruppe simone

Name		redaktion
Zugriffsliste		
Erlaubte Module		
	Web	Seite, Anzeigen, Liste, Info, Funktionen
	Datei	Dateiliste
	Benutzerwerkzeuge	Aufgaben, Benutzereinstellungen, Arbeitsumgebung
Untergruppen		intern, simone
Verzeichnisfreigaben		redaktion

Tabelle 4.2.4: Gruppe redaktion

allgemeine Verzeichnisfreigaben

Die Freigaben befinden sich im Verzeichnis *fileadmin* innerhalb der TYPO3-Installation. Folgende Verzeichnisfreigaben aus Tabelle 4.2.5 stehen entweder durch Vererbung allen Nutzern oder nur den Administratoren zur Verfügung.

Freigabename	Zweck	Nutzer/Gruppen	Zugriff
pics	Bilder	intern	lesen
scripts	Skripte, auch SIMONE	Administratoren	schreiben
template	Templates	Administratoren	schreiben

Tabelle 4.2.5: Verzeichnisfreigaben

4.3 One-Stop-Shopping

Für die One-Stop-Shopping Funktion wird auf das Cookie zugegriffen, dass TYPO3 bei einem Login anlegt. Cookies sind kleine Textdateien, die der Browser auf Veranlassung des Servers anlegt. Diese beinhalten Informationen, die für den Besuch der Internetseite nötig sind. Dieses Cookie enthält eine Session-ID. Anhand dieser Session kann die UID des Nutzers, sein Name sowie das Passwort aus der TYPO3-Datenbank ermittelt werden.

Es folgt eine HTTP-Anfrage mit dem Benutzernamen und Passwort, der dem EOWEB zugesandt wird. Diese Anfrage entspricht der normalen Authentifizierung. Bei Erfolg wird im EOWEB für den Nutzer eine Session gestartet. Nun folgt ein Zugriff auf den Katalog des EOWEB. Bei der Antwort wird der Frameset der EOWEB-Seite ausgetauscht. Es wird ein SWACI-Logo angezeigt. Dadurch bleibt der Eindruck von SWACI erhalten. Außerdem wird die Session durch die Katalog-Abfrage, durch die Session der Authentifizierung ersetzt. So ist der Nutzer im EOWEB angemeldet und kann dessen Funktionen nutzen. Abbildung 4.3.1 verdeutlicht den Ablauf der One-Stop-Shopping-Funktion grafisch.

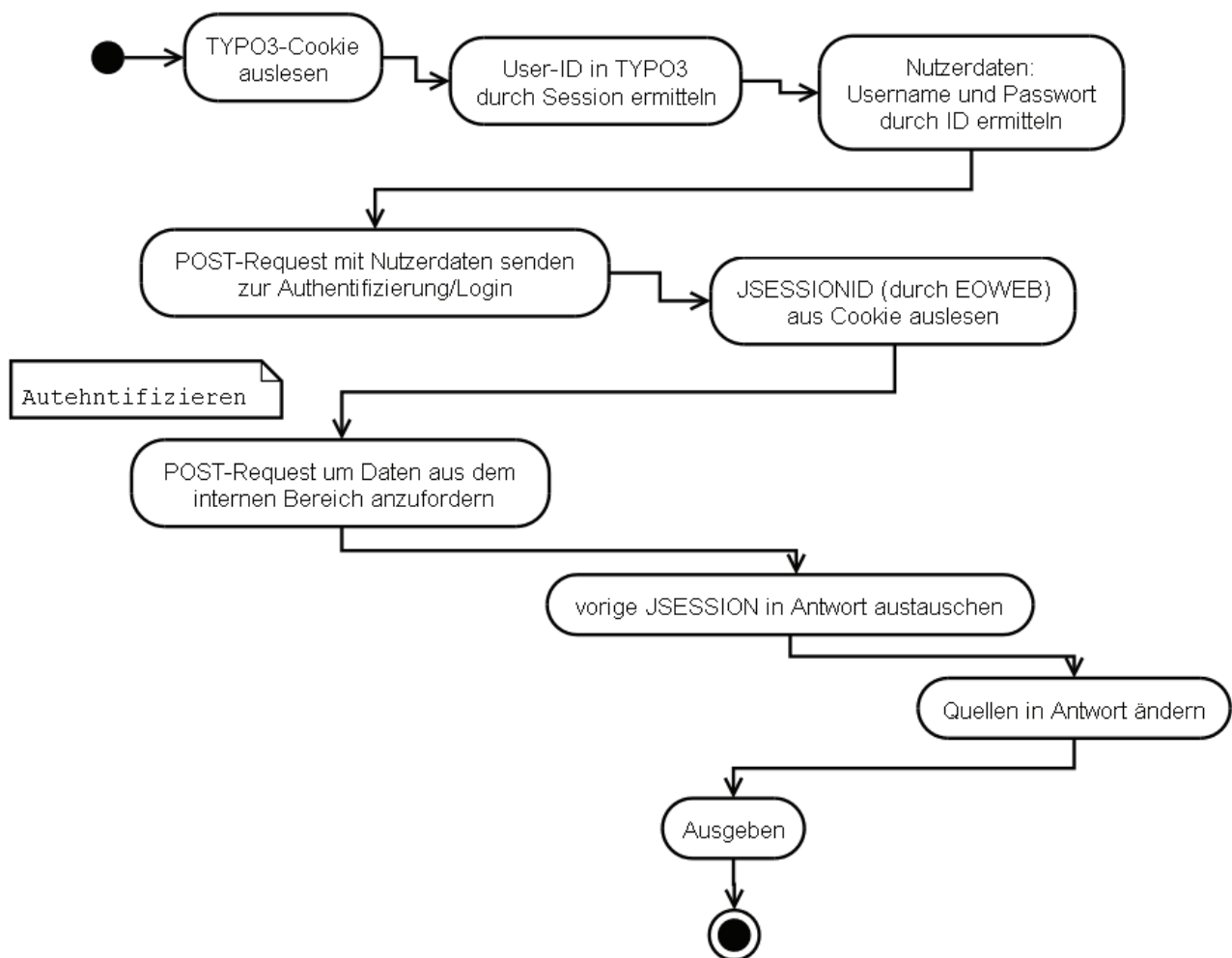


Abbildung 4.3.1: One-Stop-Shopping

4.4 Produkte

Es gibt für die Produkte vier Zugangsarten oder vielmehr Produktebenen. Über *Public* ist es für jeden möglich, auf dafür freigegebene Datensätze zuzugreifen. Die weiteren drei Zugänge entsprechen den Gruppennamen der entsprechenden Frontend-Gruppen. Dadurch, dass der Zugang zu den Produkten prinzipiell der Gleiche geblieben ist, sind wesentliche Schnittstellen erhalten geblieben. Für die Datenhaltung der Produkte wird zu jeder Zugangsart ein Verzeichnis angelegt. Darin befindet sich jeweils zu jeder Produktgruppe Verzeichnisse, wie bei der vorigen SWACI-Lösung. Diese beinhalten jeweils die einzelnen Produkte. Das Schema zur Produktwahl auf der Abbildung 4.4.1 auf der folgenden Seite stellt grafisch den Zugriff auf die Produkte dar.

GroundBased

Für die Produktgruppe *GroundBased* werden die Plots, die immer unter dem gleichen Namen gespeichert werden, in die Seite eingebunden. Die Ansicht wird regelmäßig aktualisiert um die neuen Daten anzeigen zu können. Dazu gibt es Links zu den Textdateien mit den Einzelwerten. Im Bereich *Public* werden keine Links zu den Textdateien angeboten.

TopsideReconstruction und RadioOccultation

Für die beiden Produktgruppen *TopsideReconstruction* und *RadioOccultation* wird ein einheitlicher Weg bestritten. Ein zentrales Skript ermöglicht den Zugriff auf die Produkte. Dazu ist zunächst die gewählte Produktgruppe auszuwerten. Dies geschieht über einen Parameter.

Anschließend listet dieses Skript die verfügbaren Verzeichnisse zu den Tagesdaten tabellarisch auf und erstellt Links zu diesen. Die Namen entsprechen dem Tag des Jahres. Das kann beispielsweise der Wert 126 sein.

Da jeweils nur Daten der letzten 31 Tagen dem Nutzer zur Verfügung stehen sollen, müssen ältere entfernt werden. Ein Löschskript sorgt dafür, dass nicht mehr Datensätze als zu 31 Tagen für den Nutzer zur Verfügung stehen. Es zählt die Verzeichnisse und sortiert sie chronologisch, alle älteren Verzeichnisse werden gelöscht. Dies geschieht im Hintergrund mittels eines Cronjobs. Ein Cronjob ist ein zeitgesteuerter Prozess, der in regelmäßigen Abständen ausgeführt.

Da das Löschskript nicht unbedingt synchron mit dem Upload der neuen Daten durch DIMS abläuft, ist eine Begrenzung im Skript nötig, welches den Zugriff auf die Daten ermöglicht. Die Anzahl an Links ist auf 31 begrenzt und berücksichtigt nur die neuesten Verzeichnisse.

Nach der Wahl des jeweiligen Tages werden die verfügbaren Plots und Textdateien ebenfalls tabellarisch aufgelistet. Dies geschieht durch Skripte für die jeweiligen unterschiedlichen Produkte. Neben den Links zu den Plots und den Textdateien werden Metadaten ausgegeben.

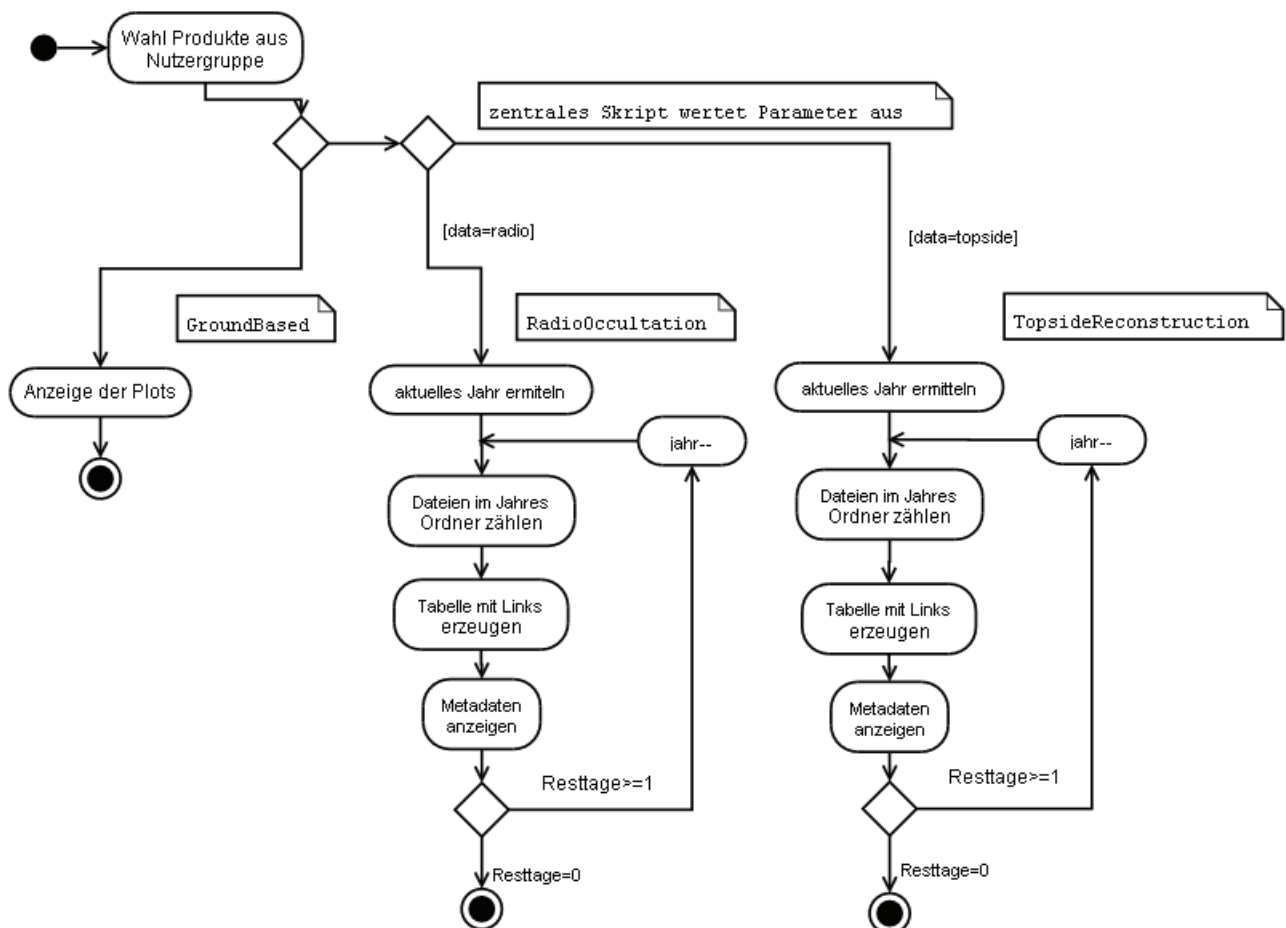


Abbildung 4.4.1: Schema zur Produktgruppenwahl und Anzeige

Zugriffsschutz

Um den unbefugten Zugriff zu den Produkten zu verhindern, wird wie zuvor htaccess eingesetzt. Hierbei hat es nicht die Aufgabe der Authentifizierung, sondern den des Zugriffsschutzes. Der Schutz ist so gestaltet, dass jegliche Zugriffe clientseitig unterbunden sind. Da PHP serverseitig arbeitet, sind die gewünschten Dateien erreichbar. Der direkte Zugriff auf die Skripte ist ebenfalls durch htaccess unterbunden. Die einzelnen Internetseiten und Inhaltselemente sind durch TYPO3 geschützt. TYPO3 übernimmt also den primären Zugriffsschutz.

4.5 Integration SIMONE

Komponenten	Zugriff
Statistik	frei zugänglich
Datenplot	
Profil	Nach Login für Mitglieder der Gruppe SIMONE sicht- und zugreifbar.
Datenupload	
Archivprozedur	Kein Zugriff

Tabelle 4.5.1: Komponenten SIMONE

Die Tabelle 4.5.1 zeigt die fünf Komponenten von SIMONE und den Zugriffsmöglichkeiten. Zu diesen Komponenten und der Datenbank, die Voraussetzung für diese Komponenten ist, folgt eine konzeptionelle Betrachtung zur Umsetzung.

Für SIMONE wird eine separate Nutzergruppe *simone* erstellt. Diese hat alleinigen Zugriff zum Profil und dem Datenupload.

4.5.1 Datenbank

In diesem Abschnitt soll auf die Datenbank, die für SIMONE erstellt wurde, eingegangen werden. Das zentrale Element der Datenbank ist die Tabelle *misst*. Dort werden zu jedem Datensatz das Datum der Messung, die Schule und ein Fremdschlüssel zu möglichen Ereignissen auf der Sonne angelegt. Auf die Tabelle wird auch in weiteren Tabellen verwiesen, wie beispielsweise in *daten*. Dort wird der Name der Datei hinterlegt, in der sich die Messdaten befinden. So ist es möglich, dass zu einem Tag durch Messunterbrechungen verschiedene Dateien mit Messdaten entstehen. Wetterereignisse sollen ebenso berücksichtigt werden. Dies geschieht durch die Kombination der Tabellen *wetter* und *misst* in *beeinflusst*.

Für die Zuordnung der Schulen existiert die Tabelle *schule*. Dort finden sich zusätzliche Metainformationen sowie die Verbindung zur TYPO3-Datenbank. So ist in dem Datensatz *benutzer* die UID der Tabelle *fe_users* abgelegt, um einen Nutzer der Schule zuordnen zu können. Die oben genannten Strukturen und Inhalte der Datenbank werden in Abbildung 4.5.1 auf Seite 37 mit der Darstellung des ER-Modells verdeutlicht.

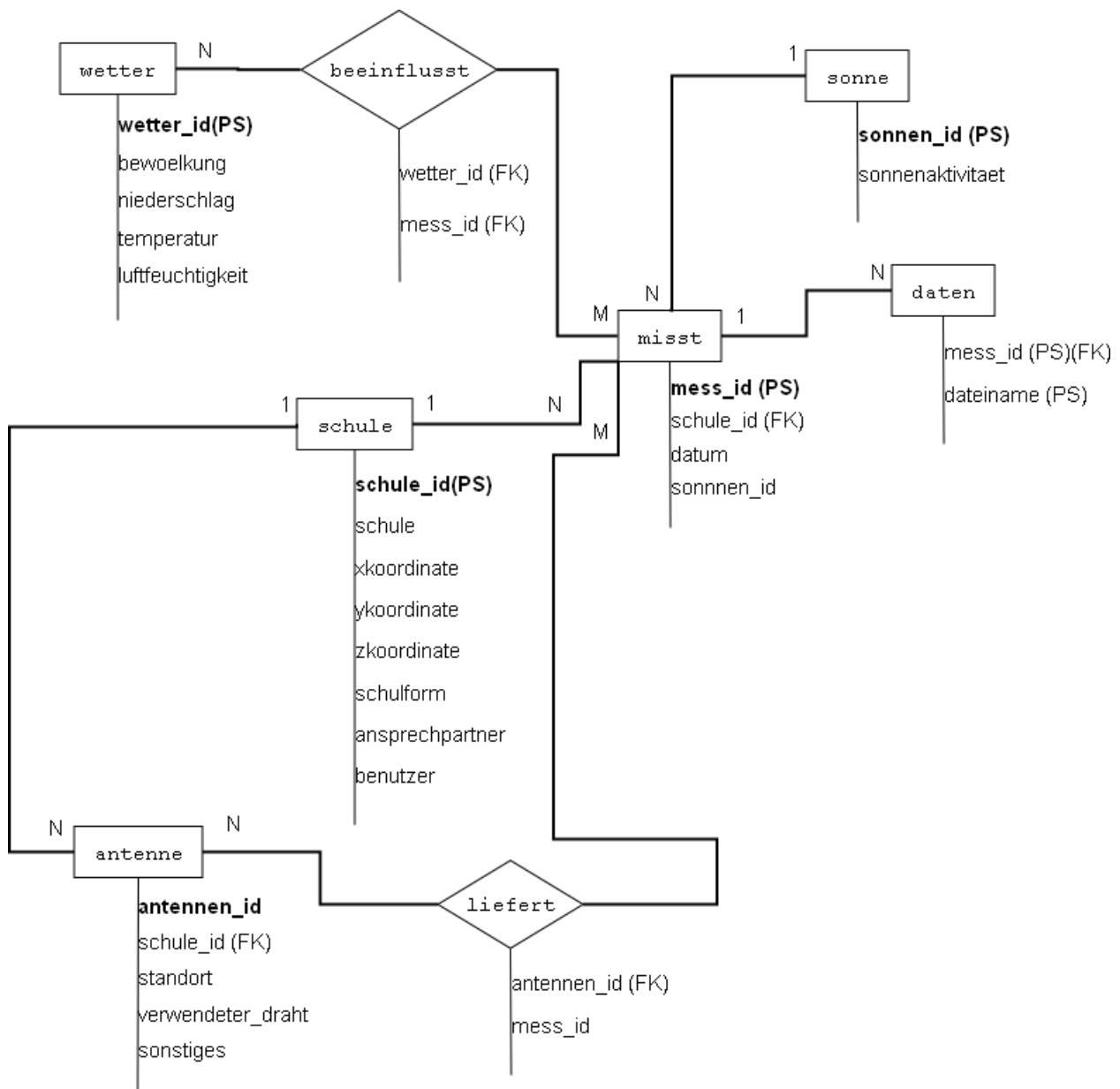


Abbildung 4.5.1: ER-Modell SIMONE

Die Tabellen *antenne*, *liefert*, sowie *wetter*, *beeinflusst* und *sonne* sind Gegenstand zukünftiger Erweiterungen der Funktionalität von SIMONE.

4.5.2 Statistik

Eine Statistik für SIMONE gibt an, wie viele Schulen im Projekt beteiligt sind. Des Weiteren gibt sie die Anzahl der bisher gesendeten Tagesdaten dieser Schulen an. Darüber hinaus soll auch zu jeder einzelnen Schule die Angabe der bisher übermittelten Tagesdaten angezeigt werden.

4.5.3 Datenupload

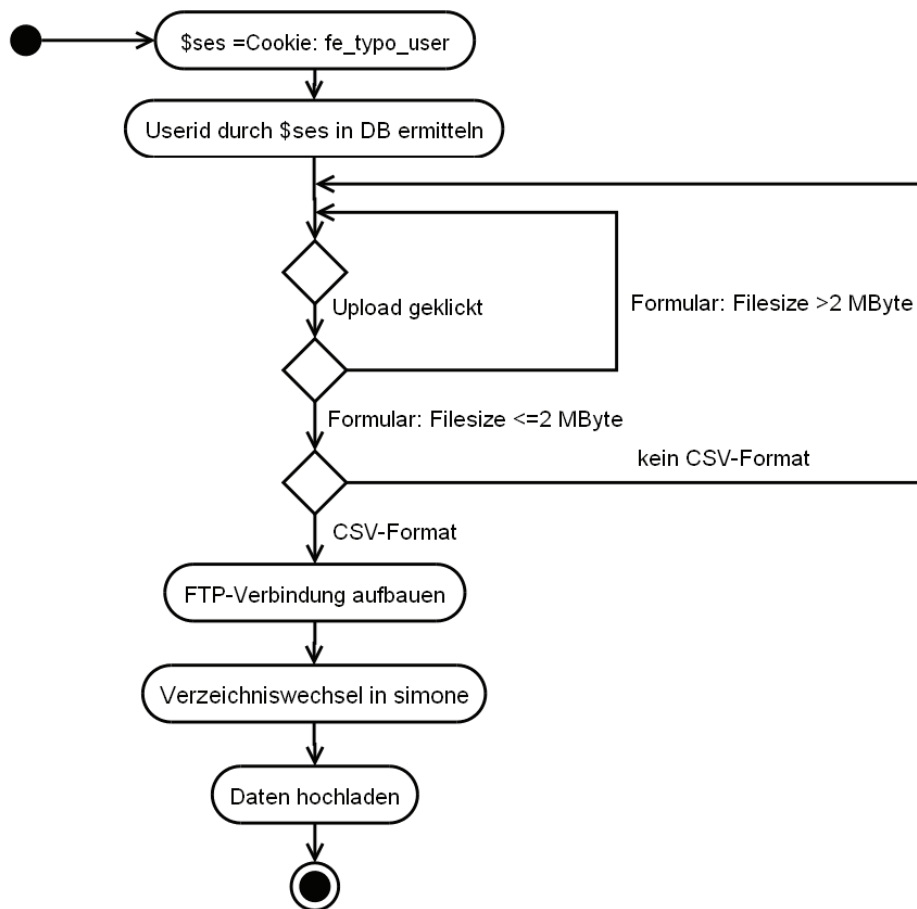


Abbildung 4.5.2: Schema Dateiupload

Mit einem Datenupload können die Nutzer ihre Dateien im CSV-Format hochladen. Dies ist nur im angemeldeten Zustand möglich. Für den so genannten Upload wird eine FTP-Verbindung aufgebaut.

Die Dateien sollen nach dem Upload, durch die UID der Nutzer aus der TYPO3-Datenbank unterscheidbar sein. Dazu wird diese UID im automatisch gesetzten Cookie gesucht. Anschließend wird geprüft, ob die Datei die Größe von 2 MByte nicht überschreitet. Die Größe leitet sich aus der Größe der CSV-Dateien ab, die 2 MByte nicht überschreiten. Ebenfalls wird geprüft, ob die Datei dem Format CSV entspricht, um das Hochladen von anderen Dateien zu verhindern. Abgelegt werden die Daten in ein dafür angelegtes Verzeichnis mit dem Namen *simone*. Das Verzeichnis befindet sich im Home-Verzeichnis des Unix-Nutzers, unter dessen Rechten die FTP-Verbindung aufgebaut wird. Dem Unix-Nutzer wird aus Sicherheitsgründen jegliches Arbeiten auf der Konsole und somit am System verwehrt. Die Abbildung 4.5.2 zeigt das Schema des Uploads zur Verdeutlichung grafisch.

4.5.4 Archivprozedur

Eine vom Webservice unabhängige Archivprozedur dient dazu, die Daten aus dem Uploadverzeichnis in das Archiv zu verschieben und alle notwendigen Informationen in die Datenbank zu übertragen. Für diese Prozedur werden nur Dateien im CSV-Format berücksichtigt. Sie wird mittels Cronjob, durch den Root durchgeführt. Der Cronjob startet alle zehn Minuten, um die verfügbaren Plots stets aktuell zu halten. Um die notwendigen Informationen zur Übertragung der Daten zu bekommen, werden die Dateinamen untersucht.

Auflösung der Namensstruktur der Messdateien nach dem Upload

UID-des Nutzers innerhalb TYPO3



Abbildung 4.5.3: Namensstruktur SIMONE-Dateien

Abbildung 4.5.3 zeigt die Auflösung der Namensstruktur der Messdateien nach dem Upload. Wichtig sind für die Übertragung der Informationen die UID, das Messdatum und der gesamte Dateiname. Die UID ist, wie schon erwähnt, in der Tabelle *schule* und *benutzer* zu finden. In der Tabelle *misst*, wird die *schule_id* der dazugehörigen UID mit dem Messdatum hinzugefügt. Jeder Eintrag in dieser Tabelle beschreibt eine Messung. Die Dateinamen liegen in der Tabelle *daten* und verweisen auf die Messung. Anschließend werden die Rechte der Dateien im Archivverzeichnis so angepasst, dass sie für den Plot genutzt werden können.

4.5.5 Plot

Um einen Plot darzustellen, müssen einige Informationen über ein Formular und über Datenbankabfragen gesammelt werden. In dem Formular, welches vom Nutzer auszufüllen ist, wird der Standort und das Datum der Messung des angeforderten Plots ermittelt. Aus der Tabelle *daten* werden zur ermittelten *mess_id* aus die verfügbaren CSV-Dateien zusammen getragen. Diese Informationen werden über eine Schnittstelle an GNUPlot übergeben. Das Ergebnis ist eine Bilddatei, die dann auf der entsprechenden Seite von SIMONE eingeblendet ist.

```
# Site = EMAG
# Longitude = 13.43
# Latitude = 54.42
#
# UTC_Offset = +02:00
# TimeZone = Westeuropäische Sommerzeit
#
# UTC_StartTime = 2007-06-29 00:00:00
# StationID = NAA
# Frequency = 24.0 KHZ
# MonitorID = S-0087-FB-0087
# SampleRate = 5
2007-06-29 00:00:00, +0.645
2007-06-29 00:00:05, +0.664
2007-06-29 00:00:10, +0.684
2007-06-29 00:00:15, +0.605
```

Metadaten

Messung, in 5-Sekundenschritten

Abbildung 4.5.4: Beschreibung der Messdatei

Die Abbildung 4.5.4 zeigt den Aufbau der Messdateien. Die Metadaten geben die geographische Länge und Breite, sowie die Startzeit der Messung und weitere Daten an. Es folgen Informationen, die für den Plot wichtig sind. Es folgt der Abschnitt mit den Messdaten. Er enthält in jeder Zeile das Datum, sowie die Uhrzeit und die gemessene elektrische Spannung. Die Messung erfolgt für einen ganzen Tag und die Datendichte wird durch den Ausdruck *SampleRate* aus den Metadaten bestimmt. Dieser gibt an, dass alle fünf Sekunden der aktuelle Wert zu speichern ist. Für den Plot wird aus dem Abschnitt der Messdaten die Uhrzeit und der dazugehörige Messwert benötigt. Das Skript für den Plot wertet zuerst die Dateien mit dem früheren Start der Messung aus, falls es mehrere Dateien für diese gibt.

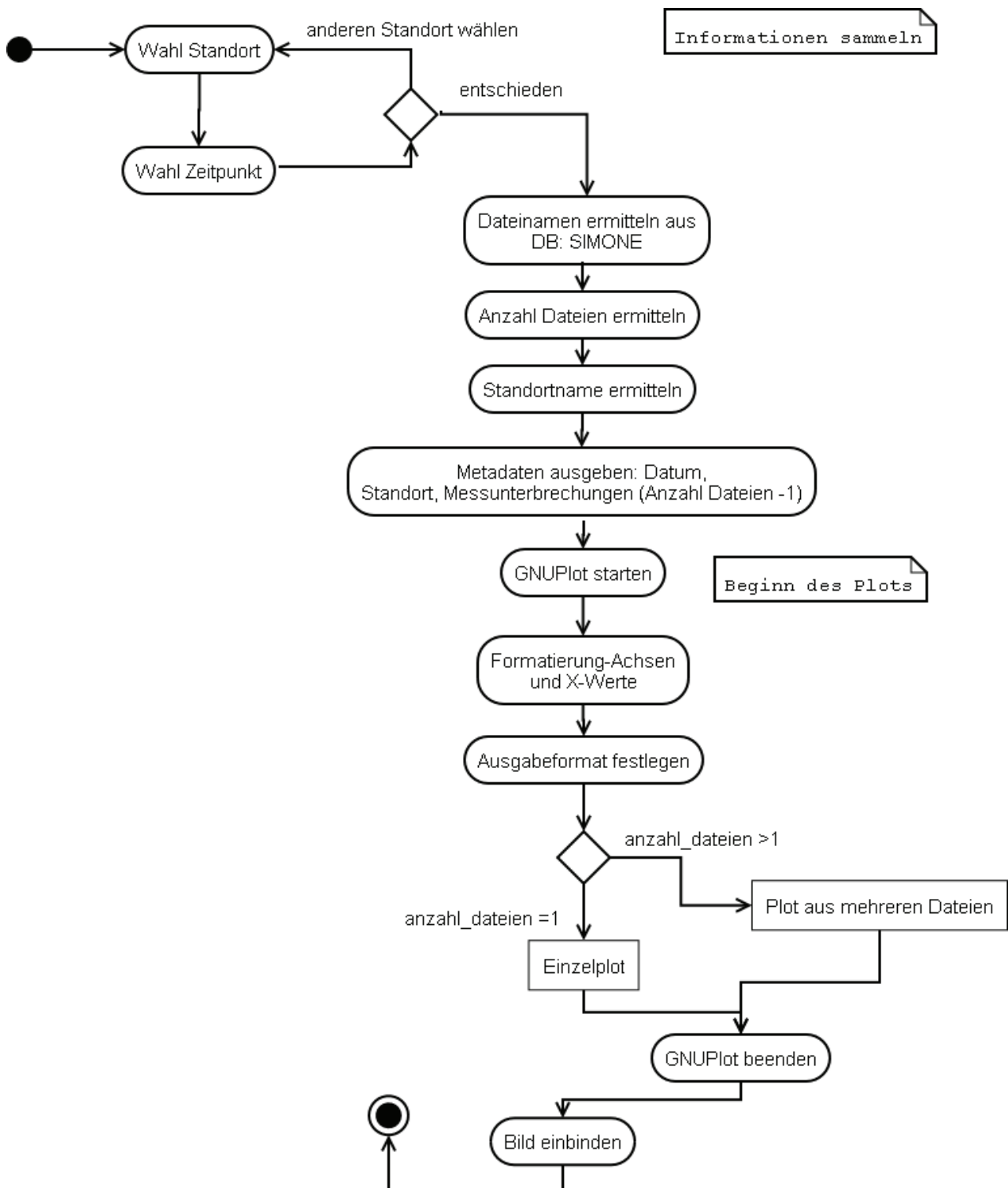


Abbildung 4.5.5: Schema Plot ausgeben

Abbildung 4.5.5 macht deutlich, dass der Vorgang zur Erstellung des Plots im Wesentlichen aus zwei Teilen besteht. Zunächst werden die erforderlichen Daten ermittelt und anschließend ein Plot, beruhend auf den ermittelten Ergebnissen durchgeführt.

GNUPlot
+ph = NULL +exe = NULL
+gnuplot() +exe(befehl:String) +format(tag:String,standortname:String) +ausgabe(dateien:String) +plot(dateien:Array) +plotn(dateien:Array,anzahl_daten=Integer) +close()

Abbildung 4.5.6: Klasse GNUPlot

Um die Schnittstelle zu GNUPlot zu realisieren, ist folgende Klasse konzipiert worden. Diese ist in der Abbildung 4.5.6 dargestellt. Sie zeigt die verfügbaren Funktionen und Parameter der Klasse GNUPlot. Diese dienen dem Starten von GNUPlot und dazu Befehle zu übergeben. Des Weiteren dienen sie zur Formatierung der Ausgabe und der Erstellung des Plots.

4.6 Logfunktion

Um eine übersichtliche Darstellung der Inhalte der Apache-Logdatei zu erhalten, wird eine Logfunktion für SWACI eingebaut. Für die Logfunktion wird eine Erweiterung¹¹ für TYPO3 genutzt, die AWStats integriert. AWStats ist ein so genannter Logfile-Analyser. Die Erweiterung ermöglicht den Zugriff und die Auswertung der Apache-Logdatei im Backend-Bereich. Der Zugriff ist nur für den Administrator möglich. Dadurch wird nicht nur ein genereller Zugang zu der Logdatei ermöglicht, sondern auch ein bequemer Weg bestritten.

AWStats gibt die Häufigkeit der Zugriffe pro Tag, Monat oder stündlich; Server-Fehlermeldungen und Browsertypen an. Zu einigen weiteren Informationen gehören, wie die Besucher die Seite gefunden haben. Es wird dabei zwischen Suchmaschinen, externe Links, Newsgruppen, dem direkten Zugriff aus den persönlichen Favoriten oder der unbekannten Herkunft unterschieden. Für die Auswertung sind die Ergebnisse tabellarisch dargestellt und durch Balkendiagramme veranschaulicht. Der Zugriffsschutz zur Logdatei wird mittels htaccess ermöglicht.

¹¹ http://typo3.org/extensions/repository/view/cc_awstats/0.10.0/

5 Umsetzung

Dieses Kapitel befasst sich mit der Umsetzung des Konzepts. Dabei geht es auf die Besonderheiten der Umsetzung ein. Es erläutert kurz die Installation des Servers, die Schnittstelle zu PHP innerhalb von TYPO3, sowie die Nutzergruppen, den Zugriff auf die SWACI-Produkte und die Integration des SIMONE-Projektes.

Einführend soll auf das geforderte Forum hingewiesen werden, welches Teil der Anforderungen ist. Hierfür kommt die Erweiterung CHC-Forum zum Einsatz. Diese kann auf die Frontend-Nutzer von TYPO3 zurückgreifen. Da die Konfiguration nicht von dem üblichen Vorgehen zu dieser Erweiterung abweicht, soll dies nicht Gegenstand der Ausführungen sein.

5.1 Installation des Servers

Die Installation wird wie vom DLR gefordert, auf einer virtuellen Maschine mit SuSE 9.3 Professional durchgeführt. Die virtuelle Maschine wird durch VMWare umgesetzt. Zu den zu installierenden Paketen gehören der Apache-Webserver, PHP 5.2, MySQL, ImageMagick, ein FTP-Server sowie Module zu PHP und dem Webserver Apache. Des Weiteren gehören zu den zu installierenden Paketen einige für deren Entwicklung nötige Programme.

Für den Betrieb des FTP-Server wird ein entsprechender Nutzer angelegt.

5.2 Schnittstelle PHP

Um eine Schnittstelle zu eigenen PHP-Skripten zu realisieren, muss das Template der SWACI-Seite angepasst bzw. ein Template angelegt werden. Da im Template der Pfad zum Skript angegeben sein muss, ist ein Template zu jeder Seite, die Skripte enthalten sollen, anzulegen. Die Konfiguration kann auf mehreren Wegen erfolgen. Damit das PHP-Skript und damit der Inhalt der Seite besser gestaltet werden kann, ist das PHP-Skript über das Inhaltselement *Skript* einzufügen.

```
tt_content.script = CASE
tt_content.script {
    key.field = select_key
    MARKER = PHP_SCRIPT_EXT
    MARKER.file = fileadmin/scripts/public_radio.php
}
```

Abbildung 5.2.1: Template als PHP-Schnittstelle

Abbildung 5.2.1 zeigt das Template, welches die Ausführung eines PHP-Skriptes ermöglicht. Im Template ist ein Schlüssel namens MARKER definiert. Dieser wird, wie in Abbildung 5.2.2 auf der nächsten Seite zu sehen, im Feld *CODE* eingetragen, um dem Inhaltselement das Skript im Template zuordnen zu können.

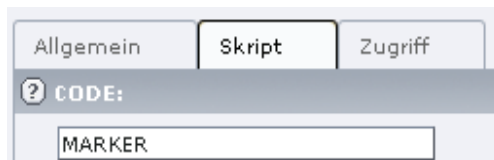


Abbildung 5.2.2: Inhaltselement Skript

5.3 Nutzer

Zur Umsetzung der Nutzergruppenstruktur für den Frontend-Bereich wurden im EOWEB äquivalente Gruppen zu SWACI geschaffen, um dort die Nutzer einzuordnen. Diese sind in der Tabelle 5.3.1 auf der nächsten Seite dargestellt.

Nutzergruppen und ihre Zuordnung im EOWEB

SWACI	EOWEB
research	SWACI-RES
commercial	SWACI-COMM
consortium	SWACI-CONS
simone	ANNONYMOUS

Tabelle 5.3.1: Nutzergruppen im EOWEB

Das Anlegen der Backend-Nutzer erfolgt innerhalb von TYPO3 mit den entsprechenden Werkzeugen.

5.3.1 Authentifizierung

Die Authentifizierung der Frontend- und Backend-Nutzer erfolgt über SSL-geschützte Seiten. Dies dient der sicheren Übertragung der Passwörter und Benutzernamen. Um dies für den Frontend-Bereich zu gewährleisten, kommt die Erweiterung *HTTPS Enforcer* zum Einsatz. Damit kann festgelegt werden, dass eine Seite durch TYPO3 nur mittels HTTPS übertragen wird.

Damit die Übertragung der Authentifizierungsdaten auch für den Backend-Bereich verschlüsselt geschieht, ist eine Änderung mittels dem *Install Tool* durchzuführen. Dies ist ein Werkzeug, das zu TYPO3 gehört um grundlegende Einstellungen vornehmen zu können. Im Install Tool wird im Abschnitt [lockSSL] die Option [BE][lockSSL] von 0 auf 2 gesetzt. Das hat zur Folge, dass der HTTP-Aufruf der Seite für den Backend-Bereich sofort auf HTTPS umgeleitet wird.

Es kann bei der Authentifizierung für den Backend-Bereich dazu kommen, dass einige Nutzer vor der Adresse ein *www* setzen und andere nicht. TYPO3 löst bei einem Aufruf, der von der Installationsadresse abweicht, eine Fehlermeldung aus. Um dies zu unterbinden, wird ebenfalls mittels dem Install Tool die Option [doNotCheckReferer] von 0 auf 1 gesetzt.

Bei dem Vorgang der Authentifizierung der Frontend-Nutzer überträgt ein PHP-Skript die eingegebenen Authentifizierungsdaten per POST-Anfrage an eine speziell dafür eingerichtete Schnittstelle. Sollten die Daten korrekt sein, wird die Meldung *SUCCESS* zurückgegeben. Bei einem Fehler ist es *FAILED*. Die Antwort wird in einem String gespeichert und nach dem Ausdruck *SUCCESS* durchsucht. Ist die Suche erfolgreich, erfolgt die Übertragung der Daten an TYPO3. Dafür greift das PHP-Skript auf die TYPO3-Variable `$content`. Dieser Variable übergibt das Skript ein Formular in Form des TYPO3-Anmeldeformular. Die Übertragung erfolgt per POST. Die Abbildung 5.3.1.1 soll den Sachverhalt verdeutlichen.

Das Anmeldeformular in SWACI-II ist auf der Abbildung 5.3.1.2 dargestellt. Dieses Formular ist über den entsprechenden Menüpunkt zu erreichen. Es erscheint auch, wenn ein Nutzer nicht die Berechtigung hat, einen Produktbereich einer jeweiligen Gruppe zu betreten.

Neben der eigentlichen Anmeldung können Besucher gleich über einen bereitgestellten Link zum Registrierungsformular gelangen.

```
$content .= '  
    <form method="post" id="fe_cfu_user_login_form"  
    action="http://swaciweb.nz.dlr.de/index.php?id=10">  
    <input type="hidden" id="user" name="user" value=".'.$userid.'" />  
    <input type="hidden" id="pass" name="pass" value=".'.$passwd.'" />  
    <input type="hidden" name="logintype" value="login" />  
    <input type="hidden" name="pid" value="7" />  
    </form>  
    <script type="text/javascript">  
    document.getElementById("fe_cfu_user_login_form").submit();  
    </script>';  
  
return $content;
```

Abbildung 5.3.1.1: Login für TYPO3

Abbildung 5.3.1.2: Anmeldeformular

5.4 One-Stop-Shopping

Für eingeloggte Nutzer steht eine Schaltfläche für das EOWEB bereit. Bei der Wahl dieser Schaltfläche öffnet sich ein neues Fenster und löst die One-Stop-Shopping-Funktion aus.

Zunächst ermittelt das ausgeführte PHP-Skript den eingeloggten Nutzer. Dazu liest es das Cookie namens *fe_typo_user* aus. Dann folgt die POST-Anfrage an den Server des EOWEB. Dies ist der gleiche Prozess, wie bei der Authentifizierung. Die Antwort wird nach dem Schlüsselwort *JSESSIONID=* durchsucht und dessen Wert in einer Variable hinterlegt. Diese beinhaltet die noch gültige Session des authentifizierten Nutzers.

Nachdem dies erfolgreich verlaufen ist, löst das Skript eine weitere Anfrage an den EOWEB-Server aus. Dieser richtet sich an eine spezielle Schnittstelle des Kataloges des EOWEB und nicht an die Authentifizierungsschnittstelle. Diese Schnittstelle ist für SWACI-II speziell mit dessen Logo ausgerüstet worden. Da bei dieser Anfrage keine Nutzerdaten mitgesendet werden, würde eine Ausgabe der Antwort dazu führen, dass der Nutzer nur als *Anonymous* Zugriff hätte. Somit hätte er nicht im vollem Umfang Zugriff auf das EOWEB. Der voll Zugriff wird dadurch gewährleistet, dass nach dem Begriff *JSESSIONID=* gesucht und durch den vorher in einer Variable abgelegten Wert ersetzt wird. Nun sind noch einige Pfade der Seite des EOWEBs anzupassen, die aus drei Frames besteht.

Da das Skript für den Katalog vom EOWEB aus dessen Sicht lokal ausgeführt wird, müssen die Pfade angepasst werden. Die Antwort, die in der Variable *\$nonstop* gespeichert ist, wird nach den Pfaden der Frames durchsucht und deren Pfade um *https://server.de:8443/* erweitert vielmehr der ganze Ausdruck durch einen vollständig neuen ersetzt. Die Abbildung 5.4.1 zeigt die Funktion zur Ersetzung.

```
$suchwort="/eoweb-ng/template/swaci/welcome/subsysLogo.vm;";  
$mixed=str_replace($suchwort , "https://server.de:8443/eoweb-  
ng/template/swaci/welcome/subsysLogo.vm;" , $nonstop);
```

Abbildung 5.4.1: Ersetzung

Es wird zudem ein neues Cookie gesetzt, mit dem Inhalt der Session-ID aus der Authentifizierung. Da diese Session noch für den kurzen Moment der Abarbeitung des Skriptes gültig ist, stehen dem eingeloggten Nutzer alle Funktionen des EOWEB zur Verfügung. Abbildung 5.4.2 zeigt das EOWEB im Design von SWACI-II.

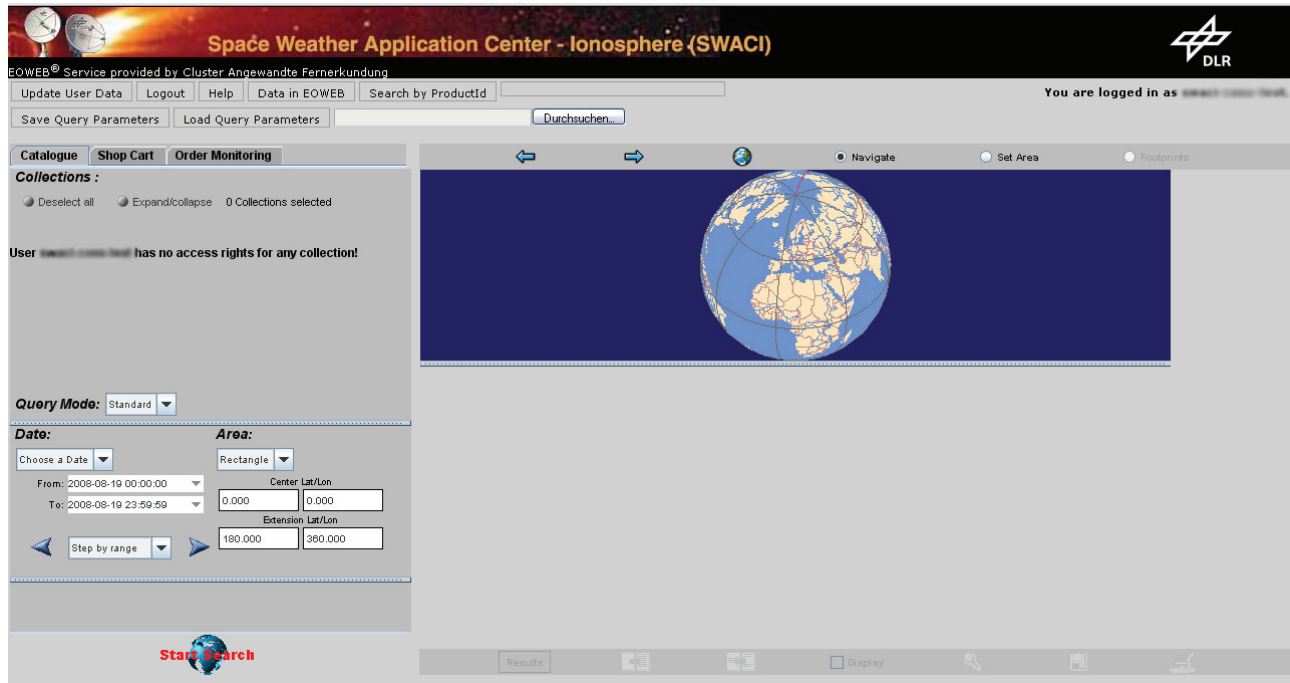


Abbildung 5.4.2: EOWEB im SWACI-II-Stil

5.5 SWACI-II-Produkte

Die htaccess-Datei beinhaltet die Option *Order Allow, Deny* mit der Einstellung *Deny from all*. Damit wird jeder Zugriff von Außen auf diese Dateien unterbunden. Da PHP serverseitig ist, können TYPO3- und eigene PHP-Skripte auf die Dateien zugreifen. Der Zugriff zu den Seiten ist durch TYPO3 beschränkt, welches es ermöglicht Seiten für Nutzer sichtbar und zugreifbar zu machen. Jede Nutzergruppe hat eigene Skripte für die jeweiligen Produktgruppen. Das hat den Vorteil, dass der Zugang zu den Daten und deren Präsentation differenzierbar ist und die Sicherheit weiter erhöht wird.

```
page.10.marks.FORECAST= IMAGE
page.10.marks.FORECAST{
    file = fileadmin/data/PUBLIC/GroundBased/DLR-RST-PRO-MAP- FORECAST.png
    altText = FORECAST
    titleText = FORECAST
}
```

Abbildung 5.5.1: TypoScript-Code für Einbindung eines Bildes

Für die Dateien der Produktgruppe *GroundBased*, wird der Zugriff durch TypoScript realisiert. In der Abbildung 5.5.1 ist zu sehen, dass ein Objekt angelegt wird. Dieses Objekt repräsentiert das Bild. Diesem Objekt ist ein Pfad zu der Datei und ein alternativer Text, sowie ein Titel zugewiesen. Da TYPO3 das Bild über einen HTML-Code einfügt, würde htaccess den Zugriff verweigern. Um dies zu umgehen, wird die Erweiterung *naw_securedl* genutzt. Diese Erweiterung ermöglicht den Zugriff mittels der TYPO3-Zugriffsrechte. Im Bereich Public, der frei zugänglich ist, sind für diese Produktgruppe nur die Plots zu sehen. In den zugriffsgeschützten Bereichen ist es möglich, auch auf die Dateien mit den einzelnen Werten zugreifen zu können. In allen Zugriffsebenen werden die Produkte regelmäßig neu geladen, um neuere Produkte zeigen zu können, die die alten regelmäßig ersetzen.

GroundBased

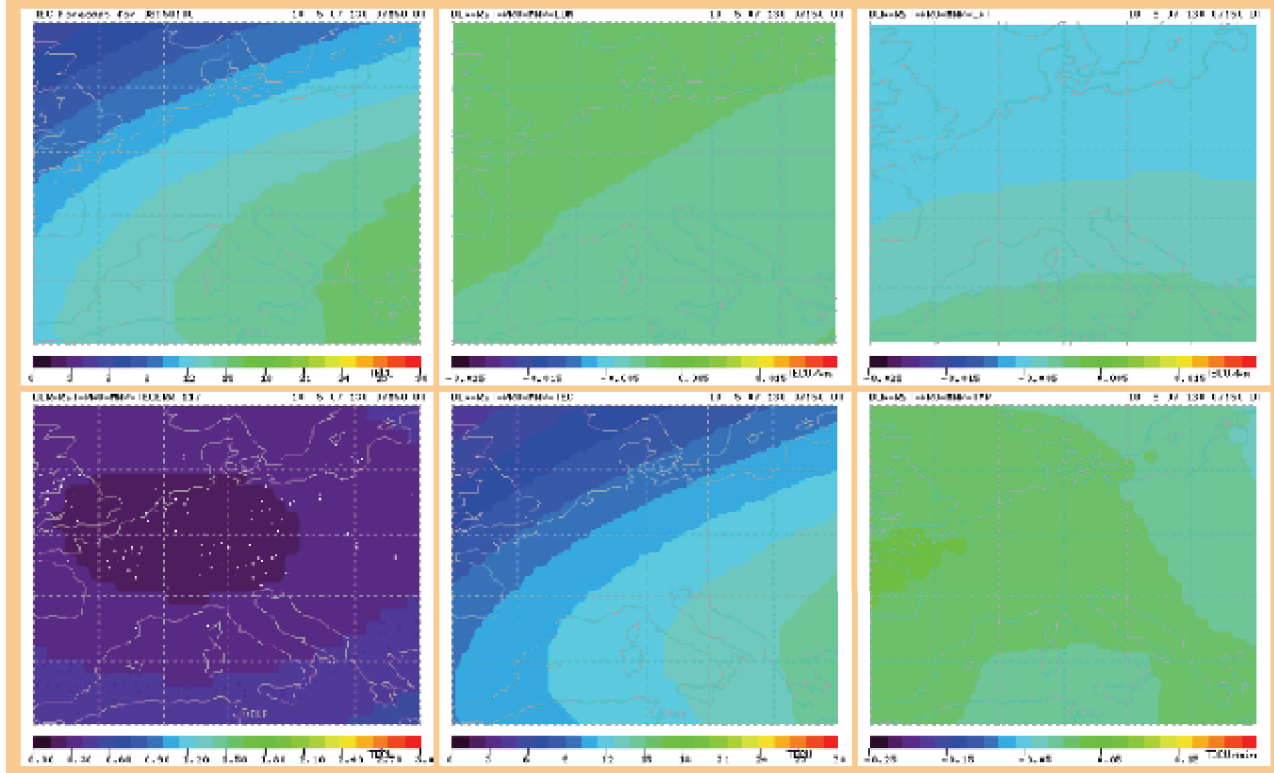


Abbildung 5.5.2: Produktgruppe GroundBased

Abbildung 5.5.2 zeigt, wie ein Zugriff auf die Daten der Produktgruppe *GroundBased* aussieht.

Da sich die Dateinamen und Verzeichnisse in den Produktgruppen *RadioOccultation* und *TopsideReconstruction* regelmäßig ändern, musste ein anderer Weg gefunden werden den geeigneten Zugriff zu realisieren.

Über das PHP-Skript *dir_list.php* werden die verfügbaren Verzeichnisse mit den Datensätzen aufgelistet. Diese sind auf 31 Tage begrenzt, da SWACI-II nur Daten über einen kurzen Zeitraum anbietet. Sollten die nötigen 31 Verzeichnisse nicht gefunden werden, wird auf das Verzeichnis des Vorjahres gewechselt. Diese ausstehenden Einträge sind in einer weiteren Tabelle zu dem Vorjahr zu finden. Dieses Skript wird für beide Produktgruppen und alle Zugriffsebenen genutzt. Die Unterscheidung erfolgt zum Teil über Parameter. Für jede Nutzergruppe existiert ein Skript, welches die jeweiligen Pfade enthält und dann *dir_list.php* importiert. Die Produktgruppe wird, über einen Parameter durch die Links entschieden. Dieser Parameter heißt *data* und enthält entweder *radio* oder *topside*.

2008

Day of Year
218
217

Abbildung 5.5.3: Ergebnisse *dir_list.php*

RadioOccultation
[< zurück](#)

Year-DOY UTC	Metadata							Data
	lat/°	lon/°	prn	nocc	hbot/km	htop/km	Quality	
2007-127 UTC: 0: 0:43	-2.2	209.6	4	32	80.1	341.6	1	File##Plot
2007-127 UTC: 0: 0:55	-77.5	40.8	8	27	80.9	374.0	0	File##Plot
2007-127 UTC: 0: 1:19	-26.7	210.1	13	26	81.0	352.5	2	File##Plot

Abbildung 5.5.4: Auflistung der Tagesdaten

Die Abbildung 5.5.3 zeigt die Auflistung der Tabellen zu den Tagesdaten. Nach Auswahl eines Tages, erfolgt die Auflistung der einzelnen Daten in einer Tabelle durch ein weiteres Skript. Es gibt zu jeder Zugriffsebene jeweils ein Skript zur jeweiligen Produktgruppe. Dort sind die Links zu den Plots, Dateien mit den Einzelwerten und einigen Metadaten. In der Abbildung 5.5.4 sind die Metainformationen zu den Produkten und die Links dargestellt.

Für den Produktbereich *RadioOccultation* sind Links zu dem Plot und zur Datei mit den Messwerten in einer Textdatei vorhanden. Die Links öffnen ein neues Fenster mit dem jeweiligen Inhalt. Damit die Bild- und die Textdatei durch PHP dargestellt werden kann, wird an dem Browser eine Information gesendet, dass sich der Datentyp geändert hat. Dies geschieht über den Header, der den Mime-type, also das Format der Informationen, enthält. Gewöhnlich sendet TYPO3 die Informationen als *text/html* aus. Um dies zu ändern, muss im Template zunächst der normale Header deaktiviert und durch einen neuen ersetzt werden. Dies ist in der Abbildung 5.5.5 auf Seite 51 dargestellt.

```

config.disableCharsetHeader = 1
page.config {

additionalHeaders = Content-Type:image/png
}

```

Abbildung 5.5.5: Senden eines neuen Content-Types

Nach diesen Festlegungen erfolgt die Einbettung des PHP.-Skripts, welches die Datei öffnet und somit den Inhalt dem Browser übergibt. In TYPO3 ist es notwendig, das Zwischenspeichern der Seite im Cache zu deaktivieren, da der Browser die Ziele der Links sonst nicht darstellt. Für die Produktgruppe *TopsideReconstruction* sind nur Links zu den Plots notwendig.

```
$shellBefehl = "rm -Rv $source";
```

Abbildung 5.5.6: Shellbefehl

Das Skript zur Löschung der veralteten Produkte funktioniert ähnlich *dir_list.php*. Der Start dieses Skripts erfolgt regelmäßig über den Cronjob. Dieser geht zunächst nach den Produktebenen und dann nach den Produktarten vor. Nach dem die neuesten 31 Verzeichnisse zu den Produkten gefunden wurden, löscht es jedes weitere, das es findet. Da die Verzeichnisse nicht leer sind, nutzt es die Bashshell um sie zu löschen und den Programmiervorgang zu erleichtern. Der Befehl aus der Bash ist in Abbildung 5.5.6 dargestellt.

5.6 SIMONE

In diesem Unterkapitel werden die Schwerpunkten während der Umsetzung der SIMONE-Integration erläutert. Hier kommen fast ausschließlich eigene PHP-Skripte zum Einsatz.

5.6.1 Statistik

Global	
Anzahl der gesamten Messungen	13
Anzahl der beteiligten Schulen	3

Schulname	Anzahl der hochgeladenen Messungen
EMAG	4
Bergen	
Testschule	9

Nutzer ohne Messergebnisse werden nicht aufgelistet.

Abbildung 5.6.1: Statistik

Die Abbildung 5.6.1. zeigt die Statistikfunktion für SIMONE. Die Statistik ist für alle Besucher und Nutzer einsehbar. Diese Funktion ist mit zwei entsprechenden SQL-Anfragen realisiert. Für die Ausgabe der Anzahl der einzelnen Ergebnisse der Teilnehmer, sind nur diese namentlich aufgeführt, die Ergebnisse vorweisen können.

5.6.2 Profil

Im Bereich Profil von SIMONE werden alle Angaben des Nutzers angezeigt. Durch die Schaltfläche *Änderungen speichern* werden die Daten an den Server übertragen. Die neuen Einträge sind sichtbar, wenn der Link *aktualisieren* gewählt wird. Diese Link lädt die Seite neu. Name und Passwort können nicht geändert werden, da diese Änderungen nicht in das EOWEB übertragbar sind. Die Abbildung 5.6.2 zeigt ein Beispiel für den Zugriff zu dem Profil.

Persönliche Angaben
Hier können Sie ihre persönlichen Angaben einsehen und ändern.

Nutzer: **simone**

Schule:	EMAG Bergen	
Postleitzahl:	17033	
Ort:	Bergen	
Straße und Hausnummer:	Hauptstraße 4	
x-Koordinate:	-0	
y-Koordinate:	-0	
z-Koordinate:	-0	
Schulform:	Gymnasium	
Ansprechpartner:	Max Mustermann	
Telefon:	039933	
Email:	test@test.de	

[aktualisieren](#)

Abbildung 5.6.2: Profil

5.6.3 Datenaupload

Für den Datenaupload wird ein FTP-Server eingesetzt. Die Konfiguration ist so gestaltet, dass der Server die System-Nutzer für die Anmeldung zulässt. Es wird ein Nutzer erstellt, der speziell für den Datenaupload von SIMONE genutzt wird. Das hat zur Folge, dass diesem Nutzer aus Sicherheitsgründen die Loginshell, also die Konsole für Befehlseingaben, entzogen werden muss. Um dies umzusetzen, ist unter SuSE Linux die Login-Shell auf */bin/false* zu setzen.

Die Daten werden in das Verzeichnis *simone* abgelegt. Dieses befindet sich innerhalb des Home-Verzeichnisses des Nutzers, der für diese FTP-Verbindung vorgesehen wurde. Die Archivprozedur verschiebt regelmäßig alle hochgeladenen Daten in das Archivverzeichnis und nimmt Eintragungen in die Datenbank vor.

5.6.4 Archivprozedur

Die Archivprozedur wird mittels Cronjob durchgeführt. Der Cronjob führt ein Bash-Skript aus, dass wiederum ein PHP-Skript startet. Dieses Skript durchsucht das Uploadverzeichnis nach Dateien vom Typ CSV. Da es mehrere Arten gibt, wie CSV-Dateien definiert werden, sind diese durch das Skript abgedeckt. Die Abbildung 5.6.3 stellt die Abfrage auf das Datenformat dar.

```
if(($_FILES['Datei']['type']=="text/comma-separated-values")  
($_FILES['Datei']['type']=="application/vnd.ms-excel")){  
    $UploadErgebnis = ftp_put($status, $user."_".$_FILES['Datei']['name'], $_FILES['Datei']  
        ['tmp_name'], FTP_BINARY);  
}
```

Abbildung 5.6.3: Datentypprüfung

Die Archivprozedur arbeitet jede Datei einzeln ab. Anhand der Namensstruktur nimmt diese Prozedur alle notwendigen Eintragungen in der SIMONE-Datenbank vor. Nach den Einträgen in die Datenbank verschiebt es die jeweilige Datei in das Archivverzeichnis. Hier kann dann später das Skript für den Plot auf die Dateien zugreifen. Nach der vollständigen Übertragung aller Dateien setzt das Skript die Rechte der Dateien so, dass der Webserver diese nutzen kann.

5.6.5 Plot

Damit GNUPlot durch PHP ausgeführt werden kann, wird die Funktion *fopen* aus PHP benötigt. Diese Funktion startet das Programm. Mit der PHP-Funktion *fwrite* werden die Befehle an das Programm übergeben.

Für den Plot sammelt ein Skript die nötigen Informationen aus der Datenbank und importiert das Skript mit der Klasse GNUPlot. Durch die Klasse GNUPlot werden verschiedene Funktionen zur Verfügung gestellt, die in der Abbildung 3.5.5 dargestellt sind. Bevor eine Funktion für den Plot startet, ermittelt das aktuelle Skript die Anzahl an Dateien für diese Messung. Mittels der Funktion *plot* wird der Plot für eine Grafik aus einer Datei realisiert. Um eine Grafik aus mehreren Dateien zu erstellen, wird die Funktion *plotn* aufgerufen. Diese Funktion ist in Abbildung 5.6.4 abgebildet.

```
function plotn($dateien,$anzahl_daten){
    $i=2;$j=1;
    $plot="\"./fileadmin/data/SIMONE/\".$dateien[0]."\\" using 2:3 with line not";
    while($i<=$anzahl_daten){
        $plotn="\", \"./fileadmin/data/SIMONE/\".$dateien[$j]."\\" using 2:3 with line not";
        $i++;
        $j++;
        $plot=$plot.$plotn;
    }
    $this->exe( "plot ".$plot."\n");
}
```

Abbildung 5.6.4: Plotfunktion

Zur Formatierung des Plots ist die Funktion *format* zuständig. Diese setzt die Beschriftung, sowie den Titel. Für die Beschriftung der X-Achse sind diese Werte in das Format *time* zu bringen. Dieses Format wird durch den GNUPlot-Befehl *timefmt* genau definiert. Zur Darstellung wird festgelegt, dass nur volle Stunden ausgegeben werden. Dann folgt die Häufigkeit der Ausgabe, der X-Werte. Die Y-Achse gibt den Wert der elektrischen Spannung zur Uhrzeit in der X-Achse an. Der Bereich der X-Achse wird von 00:00:00 – 24:00:00 und für die Y-Achse auf -5 bis 5 gesetzt. Die Grenze der Y-Achse entspricht den maximalen Spannungsspitzen.

Das Ergebnis wird als Datei im PNG-Format in einem temporären Verzeichnis abgelegt und im Browser aufgerufen.

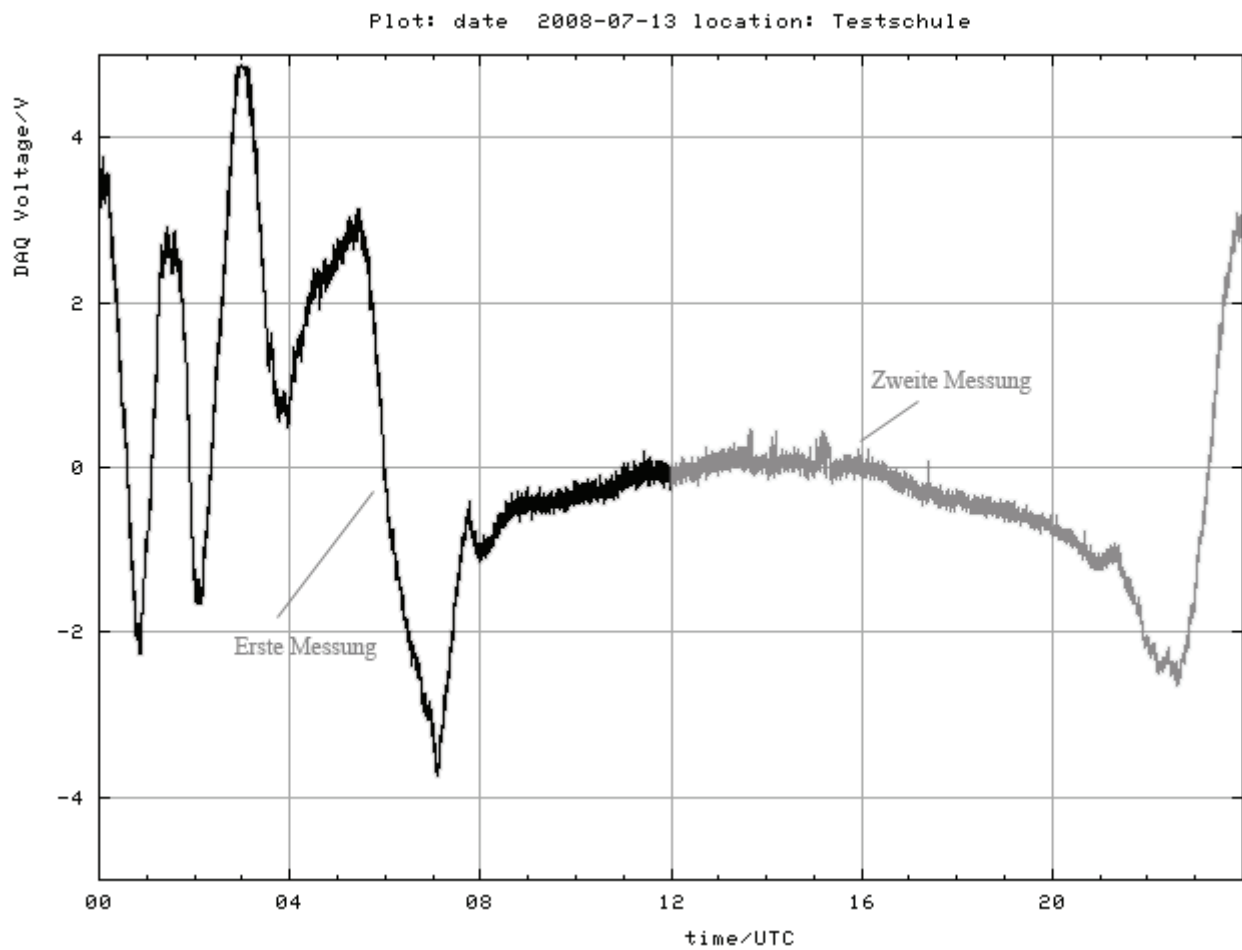


Abbildung 5.6.5: GNUPlot mit 1 Messunterbrechung

Abbildung 5.6.5 zeigt ein Ergebnis der Plotfunktion. Die Messunterbrechungen, also jede einzelne Datei mit Messwerten, wird im Plot durch unterschiedliche Farben gekennzeichnet.

5.7 Ergebnisse

Die Systemanforderungen, die sich aus den Nutzeranforderungen ableiten ließen, sind durch verschiedene Verfahren umgesetzt worden. Dabei kommen neben Erweiterungen zu TYPO3 auch eigene Skripte zum Einsatz.

Die Zugänge zu den Produktgruppen orientieren sich an den bisherigen Schnittstellen. Die Produkte selber stehen nur den entsprechenden Nutzergruppen zur Verfügung. Die Mitglieder dieser Nutzergruppen erhalten Zugriff auf die Daten, die auf 31 Tage begrenzt sind. Dies geschieht durch Skripte, die diese Daten zur Verfügung stellen und einen Prozess der im Hintergrund wirkt.

Die Anmeldung der Nutzer erfolgt für den Frontend- und den Backend-Bereich über SSL-geschützte Seiten. Die gegenüber dem EOWEB authentifizierten Frontend-Nutzer haben Zugriff auf die Funktionalitäten des EOWEB mittels einer One-Stop-Shopping-Funktion.

Das integrierte SIMONE-Projekt bietet den Upload von Daten an, die in kurzer Zeit später auch für den Plot zur Verfügung stehen. Dieser ist durch den freien Plotter GNUPlot umgesetzt. Der Upload steht neben der Möglichkeit der Aktualisierung des Profils nur Mitgliedern der Gruppe SIMONE zur Verfügung.

Über ein bereitgestelltes Forum können sich Nutzer aus verschiedenen Gruppen austauschen. Das Forum wird über das Backend konfiguriert. In dem Backend können auch verschiedene Nutzergruppen von Redakteuren ihre Inhalte bereitstellen. Über das Backend kann der Administrator auf die Logdaten des Apaches zugreifen und diese auswerten.

Während der Umsetzung traten Probleme bei der Erstellung des Zugriffs auf die SWACI-Produkte auf. Durch Cache-Effekte entstand der Eindruck, dass die Skripte fehlerhaft arbeiten. Daher ist das Cachen, also Zwischenspeichern der gesamten Seite, für die Produktzugriffe jeweils deaktiviert.

6 Tests und Abgleich

Abschließend nach der Entwicklung des Systems wurden einige Tests und ein Abgleich mit den Systemanforderungen durchgeführt. Da die Inhalte von SWACI-II durch TYPO3 gesteuert und entsprechend eines Templates ausgegeben werden, sind die Systemanforderungen SI01 – SI02 erfüllt. Das Erstellen von Inhalten auch in verschiedenen Sprachen über verschiedene Nutzer ist erfolgreich überprüft worden, womit SI03 und SI04 ebenso erfüllt sind. Die Seitenbäume und andere Objekte sind verschiedenen Backend-Nutzergruppen zugeordnet und dessen Funktion ist in Versuchen bestätigt. Dabei erfolgte der Login in den Backend über verschiedene Nutzer. Damit ist SN21 und das damit verbundene Rechtemanagement nachgewiesen.

Für den Administrator steht der Zugriff auf die Logdaten über das Modul AWStats zur Verfügung. Dieses Modul ist den anderen Backend-Nutzern verwehrt und erfüllt SF09. Über Schnittstellen, die den bisherigen entsprechen, ist SF05 erfüllt. Durch verschiedene Einträge in das Forum ist die Funktionalität der Anforderung SF06 und SF07 erfolgreich überprüft worden.

Die Funktionalität der Frontend-Nutzergruppen wurde durch verschiedene Szenarien geprüft. In der Entwicklungsumgebung des EOWEB sind vergleichbare Nutzer angelegt worden. Dadurch ist die Funktionalität der Anforderungen SN23 – SN24 erfolgreich getestet. Die Mitglieder der eingerichteten Nutzergruppen konnten nur auf ihre freigegebenen Inhalte zugreifen. Dies geschah über eine systematische Überprüfung mit den Testnutzern über alle Nutzergruppen. Die Darstellung der Produkte und deren Zugriff sind erfolgreich über verschiedene Nutzer getestet worden. Dazu gehört auch die begrenzte Anzahl von verfügbaren Produkten durch *dir_list.php* und dem im Hintergrund selbständig arbeitendem Löschskript. Somit konnte erfolgreich die Funktionalität von SF10, SN18 – SN20 nachgewiesen werden.

Die Funktionalität von SIMONE wurde ebenfalls geprüft. Die geschützten Zugriffe sind durch vorige Tests bestätigt worden. Die Änderung der Angaben im Profil konnten erfolgreich übertragen und wieder ausgelesen werden. Die eingeloggteten Nutzer der Gruppe SIMONE konnten erfolgreich Daten hochladen. Dabei ist auch erfolgreich Geprüft worden, ob der Upload von Daten abweichend vom Typ CSV verhindert wurde. Das Archivskript konnte die Daten in die Datenbank übertragen. Dies klappte auch mit Testdaten die Messunterbrechungen beinhalteten. Auch der Plot konnte erfolgreich auf Grundlage von einzelnen und mehreren Dateien durchgeführt werden. Somit sind die Systemanforderungen SO11 – SO17 erfolgreich umgesetzt.

Da der Server wie gefordert unter SuSE 9.3 Professional auf VMWare installiert wurde und mit den angelegten Nutzern und Diensten erfolgreich läuft, ist auch SU26-SU28 erfolgreich nachgewiesen. Die Funktionalität von SN25 wurde auch erfolgreich getestet. Die Seiten für den Login des Front- und Backends, Logout, sowie das Profil für SIMONE und die One-Stop-Shoppingfunktion sind nun nur über eine HTTPS-Verbindung aufrufbar.

7 Zusammenfassung und Ausblick

Die Verwendung des Content-Management-Systems TYPO3 hat sich bewährt, um verschiedene Projekte innerhalb der Internetpräsenz SWACI-II zu integrieren, wie es SIMONE beweist. Diese Arbeit hat dargestellt, wie es durch TYPO3 möglich ist, Redakteure für das System bereit zu stellen, die allein für ihren Bereich inhaltlich verantwortlich sind. Diese können mit ihrem Fachwissen die Inhalte schneller und mit höherer Qualität bereichern, als ein Einzelner es könnte. Die verschiedenen Nutzer der Projekte oder Teilprojekte können in Gruppen zusammengefasst werden. Es ist möglich, diesen Gruppen bestimmte Zugriffsrechte auf einzelne Seiten oder Seiteninhalte zu geben. Das wiederum macht es möglich, bestimmte Bereiche gegenüber anderen Nutzergruppen abzugrenzen.

TYPO3 ist in seinem Funktionsumfang durch Erweiterungen für den Frontend- und den Backend-Bereich stark erweiterbar. Diese umfassen neben offensichtlichen, wie dem umgesetzten Forum, auch nicht so offensichtliche Erweiterungen, wie beispielsweise den zwingenden Aufruf bestimmter Seiten über HTTPS. Fehlende Funktionen können durch das Einbinden von PHP-Skripten nachgeliefert werden. Durch die starke Verbreitung von PHP ist eine Einarbeitung für beteiligte Personen relativ schnell möglich. Selbst Funktionalitäten, die mit PHP nicht oder sehr kompliziert lösbar sind, sind wiederum durch andere Programme umsetzbar. Damit ist nicht zuletzt GNUPlot gemeint, mit dem relativ einfach qualitativ hochwertige Plots zu erzeugen sind. Das schließt auch Hintergrundprozesse ein, wie das Skript zur Löschung veralteter Produkte. Es nutzt die Bash um Vorgänge zu beschleunigen und Lösungen zu erleichtern.

Hier sei darauf hingewiesen, dass eine Lösung für eine solch komplexe Internetpräsenz nicht nur aus Komponenten besteht, die erst auf eine Anfrage reagieren, also passiv sind. Diese Komponenten sind Prozesse, die im Hintergrund laufen und wichtige Aufgaben ausführen. Der Ablauf im Hintergrund hat den Vorteil, dass diese sicher vor Dritte und unabhängig vom der Erreichbarkeit des Webserverns sind.

Diese Arbeit soll aber nicht das Ende aller Bemühungen an SWACI-II sein. Zukünftig sollte ein Weg gefunden werden, dass das EOWEB die Zugehörigkeit der Nutzergruppe mitsendet, so dass der fehlende Nutzer in SWACI-II automatisch mit aufgenommen werden kann.

Die bisher noch ungenutzten Tabellen innerhalb der SIMONE-Datenbank könnten genutzt werden. So könnte neben dem Plot, auch die Ausgabe des Wetters zu dem Datum an dem jeweiligen Ort mit ausgegeben werden. Ebenfalls kann die bekannte Sonnenaktivität zu der Zeit eingeblendet werden. Vielleicht wäre hier die automatisiert Kommunikation mit anderen Diensten sinnvoll.

Ein GNUPlot könnte innerhalb der Statistik die Entwicklung des Projektes gut darstellen und somit den Informationswert erhöhen.

Verzeichnis der Tabellen und Abbildungen

Tabellenverzeichnis

Tabelle 2.1.1: bisherige Nutzergruppen und Zugriffsrechte.....	8
Tabelle 2.3.1: notwendige Gruppen und Zugriffsrechte.....	11
Tabelle 2.3.2: Nutzeranforderungen.....	13
Tabelle 2.3.3: Systemanforderungen.....	14
Tabelle 3.2.1: Vergleich verschiedener CMS.....	21
Tabelle 4.2.1: Nutzergruppen.....	26
Tabelle 4.2.2: Gruppe intern.....	31
Tabelle 4.2.3: Gruppe simone.....	32
Tabelle 4.2.4: Gruppe redaktion.....	32
Tabelle 4.2.5: Verzeichnisfreigaben.....	32
Tabelle 4.5.1: Komponenten SIMONE.....	36
Tabelle 5.3.1: Nutzergruppen im EOWEB.....	44

Abbildungsverzeichnis

Abbildung 2.1.1: Use-Case-Diagramm SWACI-I.....	8
Abbildung 2.3.1: Use-Case-Diagramm SWACI-II.....	12
Abbildung 4.1.1: SWACI und andere Kommunikationspartner.....	23
Abbildung 4.1.2: Komponenten von SWACI.....	24
Abbildung 4.2.1: Auszug der Nutzerverwaltung in TYPO3.....	25
Abbildung 4.2.2: Registrierung.....	28
Abbildung 4.2.3: Authentifizierung ohne automatischer Registrierung.....	29
Abbildung 4.2.4: Authentifizierung mit automatischer Registrierung.....	30
Abbildung 4.3.1: One-Stop-Shopping.....	33
Abbildung 4.4.1: Schema zur Produktgruppenwahl und Anzeige.....	35
Abbildung 4.5.1: ER-Modell SIMONE.....	37
Abbildung 4.5.2: Schema Dateiupload.....	38
Abbildung 4.5.3: Namensstruktur SIMONE-Dateien.....	39
Abbildung 4.5.4: Beschreibung der Messdatei.....	40
Abbildung 4.5.5: Schema Plot ausgeben.....	41
Abbildung 4.5.6: Klasse GNUPlot.....	42
Abbildung 5.2.1: Template als PHP-Schnittstelle.....	43
Abbildung 5.2.2: Inhaltselement Skript.....	44
Abbildung 5.3.1.1: Login für TYPO3.....	45
Abbildung 5.3.1.2: Anmeldeformular.....	45
Abbildung 5.4.1: Ersetzung.....	46
Abbildung 5.4.2: EOWEB im SWACI-II-Stil.....	47
Abbildung 5.5.1: TypoScript-Code für Einbindung eines Bildes.....	48
Abbildung 5.5.2: Produktgruppe GroundBased.....	49
Abbildung 5.5.3: Ergebnisse dir_list.php.....	50
Abbildung 5.5.4: Auflistung der Tagesdaten.....	50
Abbildung 5.5.5: Senden eines neuen Content-Types.....	51
Abbildung 5.5.6: Shellbefehl.....	51
Abbildung 5.6.1: Statistik.....	51
Abbildung 5.6.2: Profil.....	52
Abbildung 5.6.3: Datentypprüfung.....	53
Abbildung 5.6.4: Plotfunktion.....	54
Abbildung 5.6.5: GNUPlot mit 1 Messunterbrechung.....	55

Verzeichnis der Abkürzungen

EOWEB	Earth Observation on the WEB	3
SIMONE	Sonnen & Ionosphären MONitoring NETzwerk	3
DLR	Deutsche Zentrum für Luft- und Raumfahrt	5
DFD	Deutsches Fernerkundungsdatenzentrum	5
IKN	Institut für Kommunikation und Navigation	5
SID	Sudden Ionospheric Disturbances	6
DIMS	Data Information Management System	7
PNG	Portable Network Graphics	7
HTML	Hypertext Markup Language	7
CSS	Cascading Style Sheets	7
FTP	File Transfer Protocol	7
SSL	Secure Sockets Layer	8
EOS	Earth Observing System	9
PL	Product Library	9
CMS	Content-Management-System	15
PHP	Hypertext Preprocessor	15
W3C	World Wide Web Consortium	16
GDlib	Gif Draw Library	17
CPU	central processing unit	17
PDF	Portable Document Format	18
URL	Uniform Resource Locator	21
CSV	Comma-Separated Values	38
HTTPS	Hypertext Transfer Protocol Secure	44

Quellen

Internet

[DLR01] DLR, Das DLR im Überblick

URL: http://www.dlr.de/desktopdefault.aspx/tabid-636//1065_read-1465/
(21.07.2008)

[DLR02] DLR, Standort Neustrelitz

URL: <http://www.nz.dlr.de/>
(21.07.2008)

[DLR03] DLR, Datenzugriff

URL: http://www.dlr.de/caf/desktopdefault.aspx/tabid-2668//4024_read-6012/
(21.07.2008)

[DLR04] DLR, Data and Information Management

URL: http://www.dlr.de/caf/desktopdefault.aspx/tabid-2698/4085_read-6116/
(22.07.2008)

[MSN01] MSN, Deutsches Zentrum für Luft- und Raumfahrt

URL: http://de.encarta.msn.com/encyclopedia_721550370/Deutsches_Zentrum_f%C3%BCr_Luft-_und_Raumfahrt.html
(21.07.2008)

[CYR01] CyRes, Nachteile von Content Management Systemen

URL: <http://www.cyres.de/cms-grundlagen/cms-gruende/cms-nachteile.htm>
(22.07.2008)

[W3C01] W3C, HTML 5

URL: <http://www.w3.org/html/wg/html5/>
(23.07.2008)

[W3C02] W3C, CSS2 Specification

URL: <http://www.w3.org/TR/2008/REC-CSS2-20080411/>
(23.07.2008)

[AWS01] AWStats, AWStats official web site

URL: <http://awstats.sourceforge.net/>
(24.07.2008)

[SIM01] Institut für Experimentelle und Angewandte Physik, Weltraumwetterprojekt SIMONE

URL: http://www.ieap.uni-kiel.de/et/ag-heber/ihy2007/aktivitaeten/simone/simone_v104.pdf
(23.07.2008)

[CMS01] Projekt 777, Die Vorteile von TYPO3

URL: http://www.projekt-77.de/typo3_leistungsmerkmale.html
(23.07.2008)

[CMS02] VDI Technologiezentrum GmbH, TYPO3 (CMS)

URL: <http://www.vditz-internetagentur.de/leistungen/content-management-systeme-cms/typo3-cms/>
(24.07.2008)

[CMS03] VDI Technologiezentrum GmbH, Drupal(CMS)

URL: <http://www.vditz-internetagentur.de/leistungen/content-management-systeme-cms/drupal-cms/>

(24.07.2008)

[CMS04] CMS-Vergleich.de, TYPO3

URL: <http://www.cms-vergleich.de/cms/?s=cmsliste&cid=4&>

(25.07.2008)

[CMS05] CMS-Vergleich.de, Drupal

URL: <http://www.cms-vergleich.de/cms/?s=cmsliste&cid=118&>

(25.07.2008)

[CMS06] Narres, Drupal

URL: <http://www.narres.com/drupal>

(25.07.2008)

[CMS07] Narres, TYPO3

URL: <http://www.narres.com/typo3>

(25.07.2008)

[CMS08] 24iX Systems, Typo3 und Joomla im Vergleich

URL: <http://www.24ix.de/fileadmin/documents/Typo3/Typo3-vs-Joomla.pdf>

(25.07.2008)

[CMS09] VDI Technologiezentrum GmbH, Joomla (CMS)

URL: <http://www.vditz-internetagentur.de/leistungen/content-management-systeme-cms/joomla-cms/>

(25.07.2008)

[CMS10] JUST-ONE-CMS, Das just-one-cms als Mambo / Joomla Alternative?

URL: http://joomla-oder-typo3-alternative.just-one-cms.de/cms/k109.Alternative-zu-Joomla_Das-just-one-cms-als-Mambo-Joomla-Alternative.htm

(25.07.2008)

[CMS11] akademie.de, Joomla! als CMS? Überblick über Features, Vor- und Nachteile

URL: <http://www.akademie.de/programmierung-administration/website-administration/tipps/content-management-systeme/mambo-vor-und-nachteile.html>

(25.07.2008)

[CMS12] Wikipedia.de, Content-Management-System

URL: <http://de.wikipedia.org/wiki/Content-Management-System>

(23.07.2008)

[CMS13] TAO Beratungs- und Management GmbH, Drupal vs. Joomla! vs. Typo3

http://www.tao.at/ressourcen/drupal_vs_joomla_vs_typo3

(26.07.2008)

[CMS14] contentmanager.de, Systemvorstellung: Joomla! im Überblick

URL: http://www.contentmanager.de/magazin/artikel_1421_joomla_cms_open_source.html

(26.07.2008)

[CMS15] erdfisch, Drupal

URL: <http://erdfisch.de/book/export/html/7>

(26.07.2008)

[TYPO01] Typo3-Forum der HostingAgency, Backend SSL Verschlüsselung
URL: <http://typo3forum.hosting-agency.de/backend-ssl-verschluesselung-t55.html>
(01.08.2008)

[TYPO02] MiTTWALD, Deutsche Typo3-Dokumentation
URL: <http://www.typo3.net/fileadmin/downloads/dokumentation.pdf>
(01.08.2008)

[TYPO03] Wiki der typo3.org, 45 Minuten TypoScript
<http://wiki.typo3.org/index.php/De:ts45min>
(02.08.2008)

[TYPO04] admin-wissen.de, Typo3 Workshop
<http://www.admin-wissen.de/tutorials/eigene-tutorials/webentwicklung/typo3-workshop/>
(02.08.2008)

[TYPO05] offizielle TYPO3-Homepage, 1.5. Passwort-geschützte Seiten
http://typo3.org/documentation/document-library/tutorials/doc_tut_quickstart_de/0.0.6/view/1/5/
(02.08.2008)

[TYPO06] Typo3-Forum der HostingAgency, PHP Scripte in TYPO3 einbinden
<http://typo3forum.hosting-agency.de/php-scripte-in-typo3-einbinden-t46.html>
(04.08.2008)

[TYPO07] Mailing-Liste von netfielders.de, [TYPO3] Automatic FE user creation and auto-login
<http://lists.netfielders.de/pipermail/typo3-english/2007-October/043510.html>
(07.08.2008)

Literatur

Daniel J. Barrett: Linux – Kurz und Gut, O'Reilly, 3-89721-501-2

Anhang

A1

Nutzeranforderungen		Zuordnung der Systemanforderungen																											
		SI01	SI02	SI03	SI04	SF05	SF06	SF07	SF08	SF09	SF10	SO11	SO12	SO13	SO14	SO15	SO16	SO17	SI18	SI19	SI20	SI21	SI22	SI23	SI24	SI25	SU26	SU27	SU28
	NI01	X	X																										
	NI02	X		X																									
	NI03				X																	X							
	NI04																						X						
	NF05					X																X							
	NF06						X														X								
	NF07							X																					
	NF08								X																				
	NF09									X																			
	NF10										X																		
	NO11											X		X															
	NO12												X																
	NO13											X			X														
	NO14											X				X	X	X											
	NN15																		X										
	NN16																			X									
	NN17																		X										
	NN18																			X				X	X				
	NU19																												