

Alexander Waldhör

Evaluierung von LBS-Architekturen am Beispiel von Campus 2011

MASTERARBEIT

zur Erlangung des akademischen Grades
Diplom-Ingenieur/in

Masterstudium Geomatics Science



Technische Universität Graz

Betreuer:

Univ.-Prof. Dipl.-Ing. Dr.phil. Norbert Bartelme
Institut für Geoinformation

Graz, Jänner 2011

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am 24. Jänner 2011

Alexander WALDHÖR, Bakk.techn.

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, am 24. Jänner 2011

Alexander WALDHÖR, Bakk.techn.

Abstract

In today's highly connected world, there is a wealth of information coming from various sources which are available to a user. To prevent losing track of things, the information must be filtered and processed for the end user. Location Based Services (LBS) process information with respect to the current location of the user.

In the spatial and semantic domain of a University as well as in its history there is a myriad of information that results from the different parties involved. The Campus 2011 system provides links for this information and presents them to a user.

This master thesis deals first with the basic concepts such as LBS and the requirements of the Campus 2011 system. To enable an open and expandable system, special attention is given to the use of standards and norms. The resulting requirements and structures are then being applied to the architecture of a Campus 2011 system.

The last section of this thesis is concerned with methods and technologies that can be used for an actual implementation of such a system.

Kurzfassung

In der heutigen stark vernetzten Welt gibt es eine Fülle an Informationen aus verschiedensten Quellen, welche einem Benutzer zur Verfügung stehen. Damit man sich darin zurechtfindet, müssen diese Informationen für den Endkunden gefiltert und aufbereitet werden. Eine Ausprägung dieses Prozesses stellen Location Based Services (LBS) dar, welche die Information auf Grund des aktuellen Aufenthaltsorts des Benutzers verarbeiten.

Auch im Bereich einer Universität, für deren örtliche, semantische und geschichtliche Gegebenheiten, gibt es unzählige Informationen, welche durch die unterschiedlichen Beteiligten entstehen. Das Campus 2011 System verknüpft diese Informationen und präsentiert sie einem Benutzer.

Diese Arbeit beschäftigt sich zuerst mit den grundlegenden Begriffen wie LBS und den Anforderungen des Campus 2011 Systems. Um ein möglichst offenes und zukunftssicheres System zu ermöglichen, wird insbesondere auf die Verwendung von Standards und Normen Wert gelegt. Die daraus entstehenden Anforderungen und Strukturen werden auf die Architektur eines Campus 2011 Systems angewendet.

Im letzten Abschnitt beschäftigt sich diese Arbeit mit Methoden und Technologien, welche zur tatsächlichen Implementierung eines Campus 2011 Systems verwendet werden können.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Forschungsprojekt Campus 2011	2
1.2	Motivation	3
1.3	Aufbau und Struktur	5
2	Grundlagen.....	7
2.1	Location Based Services.....	8
2.1.1	Definition.....	8
2.1.2	LBS-Schnittstellen Definition: OpenLS.....	12
2.2	Kommunikation	18
2.2.1	Allgemein.....	18
2.2.2	OpenLS Kommunikationsschnittstelle	18
2.3	Information.....	22
2.3.1	Definition.....	22
2.3.2	Informationskontext.....	23
2.3.3	Informationshandhabung in OpenLS	25
2.4	Position	26
2.4.1	Positionshandhabung in OpenLS.....	26
3	Systemkomponenten	29
3.1	Akteure	30
3.1.1	Benutzer	30
3.1.2	Geräte.....	31
3.2	Daten	35
3.2.1	Datenarten	35
3.2.2	Datenstruktur	35
4	Systemarchitektur.....	43

4.1	Systemschichten	44
4.2	Systemmodule	47
4.2.1	Client Control Module	48
4.2.2	Application Control Module	49
4.2.3	Resource Control Module	50
5	Technologien	51
5.1	Server	52
5.1.1	Hardware	52
5.1.2	Software	53
5.2	Client	53
5.2.1	Hardware	53
5.2.2	Software	56
6	Zusammenfassung und Ausblick	58
6.1	Zusammenfassung	58
6.2	Ausblick	59
A.	Literaturverzeichnis	I
B.	Abbildungsverzeichnis	III
C.	Codeverzeichnis	IV
D.	Tabellenverzeichnis	IV

Abkürzungsverzeichnis

ACM	Application Control Module	PDE	Position Determining Entity
ADT	Abstract Data Type	PLZ	Postleitzahl
A-GPS	Assisted Global Positioning System	POI	Point of Interest
AOA	Angle of Arrival	QR-Codes	Quick Response Codes
API	Application Programming Interface	RCM	Resource Control Module
AW	Application Worker	RDA	Resource Data Access
AWR	Application Worker Registration	RFID	radio-frequency identification
CCM	Client Control Module	RR	Resource Registration
CGI	Common Gateway Interface	RSEC	Resource Security
CH	Client Handle	RTK GPS	Real Time Kinematic GPS
CM	Connection Module	SOAP	Simple Object Access Protocol
COO	Cell of Origin	SQL	Structured Query Language
CS	Client Scheduler	TA	Timing Advance
DBMS	Datenbankmanagementsystem	TOA	Time of Arrival
D-GPS	Differential Global Positioning System	TU	Technische Universität
E-OTD	Enhanced Observed Time Difference	TUG	Technische Universität Graz
GIS	Geographisches Informationssystem	TUGC 2011	Technische Universität Graz Campus 2011
GML	Geography Markup Language	UL-TDOA	Uplink Time Difference of Arrival
GMLC	Gateway Mobile Location Center	UMTS	Universal Mobile Telecommunications System
GMS	GeoMobility Server	WC3	World Wide Web Consortium
GNSS	Global Navigation Satellite System	WGS84	World Geodetic System 1984
GPS	Global Positioning System	WLAN	Wireless Local Area Network
GSM	Global System for Mobile Communications	WMS	Web Map Service
HTTP	Hypertext Transfer Protocol	WWW	World Wide Web
LAN	Local Area Network	XLS	XML for Location Services
LBS	Location-Based Service	XML	Extensible Markup Language
LDS	Location Dependent Service		
LIF	Location Interoperability Forum		
LTE	Long Term Evolution		
MGI	Militärgeographisches Institut		
MLP	Mobile Location Protocol		
MPC	Mobile Positioning Center		
OGC	Open Geospatial Consortium		

1 Einleitung

In diesem Kapitel wird versucht die Motivation, welche hinter dem Campus 2011 Projekt steckt, zu verdeutlichen. Dies umfasst die grundlegende Motivation, welche zu diesem Forschungsprojekt geführt hat, als auch die daraus resultierende grundlegende Zielsetzung.

Da es sich bei diesem Campus System um ein sehr komplexes System handelt, mit vielen unterschiedlichen Beteiligten und Aufgabenbereichen, wird im Zuge dieser Arbeit nur auf einen Teilbereich dieses Systems eingegangen.

Dieser Teilbereich beschäftigt sich mit der Möglichkeit der Bereitstellung der unterschiedlichen Dienstleistungen des Systems, nicht nur von einem festen Standort, wie zum Beispiel einem TU-Terminal, sondern auch von jedem beliebigen Standort aus. Des Weiteren befasst sich diese Arbeit mit den Möglichkeiten, wie sich der Standort auf die vom System bereitgestellten Dienste auswirken kann.

Aus den hier getroffenen Zielsetzungen lassen sich in weiterer Folge die genaueren Anforderungen an die Systemarchitektur ableiten, welche in den darauffolgenden Kapiteln zur Anwendung kommen.

Im letzten Abschnitt der Einleitung wird der generelle Aufbau dieser Arbeit aufgeführt, welcher dem Leser einerseits einen generellen Überblick über die Arbeit geben soll und ihm andererseits als Leitfaden durch die Arbeit dient.

1.1 Forschungsprojekt Campus 2011

Im Jahr 2011 feiert die Technische Universität Graz ihr zweihundertjähriges Bestehen, da die Gründung auf eine Stiftung des Erzherzog Johann von Österreich aus dem Jahr 1811 zurückzuführen ist [1]. Aufgrund dieses besonderen Jubiläums will die TU Graz seine Geschichte, als auch aktuelle Vorhaben den Angestellten und Besuchern in angemessener Form präsentieren.

Mit diesem besonderen Ereignis im Hintergrund wurde das Forschungsprojekt Campus 2011 von den TU Instituten für Stadt- und Baugeschichte, Geoinformation, Navigation und Satellitengeodäsie, Fernerkundung und Photogrammetrie, Computer Graphik und Wissensvisualisierung, Maschinelles Sehen und Darstellen, Informationssysteme und Computer Medien, Architektur und Medien initiiert.

Ausgangsgedanke bzw. Ziel dieses Projektes ist es, den Blickwinkel auf die TU zu erweitern und so noch unbekannt Seiten aufzuzeigen. Es soll eine individuell erkundbare, interaktive Ausstellung geschaffen werden, welche den Besucher in Form von lebendigen Führungen durch die Geschichte und das aktuelle Geschehen führt. Des Weiteren soll das dabei entstehende System für zukünftige Anforderungen erweiterbar und anpassbar sein.

Ausgehend von einem beliebigen Punkt auf oder um den Campus der TU Graz soll sich dem Benutzer eine große, neue Welt an Informationen rund um die TU Graz eröffnen, in der er sich in Raum und Zeit bewegen kann.

Während sich der Benutzer nun in der Informationswelt der Technischen Universität bewegt, entdeckt er immer neue Verbindungen zwischen den verschiedenen Informationsentitäten. Diese Verbindungen können mit Pfaden verglichen werden, welche den Benutzer durch Raum und Zeit leiten. Eine solche Reise könnte wie folgt ablaufen:

Ein Student befindet sich im Gebäude des Tesla Labors und ruft dort an einem TU Terminal das Campus 2011 System auf. Jetzt erhält er Informationen über seinen aktuellen Aufenthaltsort, das Tesla Labor. Des Weiteren entdeckt er, dass dieses Labor nach dem Erfinder Nikola Tesla benannt wurde. Nun könnte er der Frage nachgehen wollen, warum dieser Name für das Gebäude verwendet wurde. Dazu öffnet er nun den Informationsknoten Nikola Tesla und entdeckt, dass dieser Mann Erfinder war und einige Geräte entwickelt hat, darunter auch die Hochfrequenzmaschine. Diese Information führt zu einer weiteren Verbindung, welche zur Hochspannungshalle der TU Graz führt, die den Namen „Tesla Labor“ trägt. Dies erklärt dem Studenten zwar, das Tesla etwas mit Strom zu tun hat und es dadurch zu diesem Gebäudenamen kommt, jedoch gibt es auch bei anderen Universitäten solche Hochspannungshallen welche jedoch nicht nach dieser Person benannt wurden.

In weiterer Folge betrachtet der Student den Informationsknoten über Nikola Tesla etwas genauer und entdeckt eine Verbindung zum Lebenslauf von Tesla. Dort erfährt er den Geburtsort von Nikola Tesla (in Kroatien) und kann diesen in einer Karte betrachten. Über weitere Verbindungen erfährt er mehr über Biographie von Tesla. Nun bewegt sich der Benutzer

weiter entlang des Lebenslaufs und öffnet den Informationsknoten über Teslas Studienzeit. Hier entdeckt der Benutzer, dass Tesla in Graz studiert hat. Diese Informationen führt ihn schließlich wieder zurück zur Technischen Universität Graz und in weiterer Folge zum Tesla Labor.

Nun ist die Frage des Studenten zwar beantwortet, jedoch bietet sich ihm noch immer eine Welt, welche aufgrund ihrer Verknüpfungen zu weiteren Entdeckungen einlädt.

Bei der Reise durch Raum und Zeit ergeben sich für den Benutzer mehrere Möglichkeiten diese durchzuführen, wobei einige Einschränkungen beachtet werden müssen.

Eine Bewegung in der Zeit kann natürlich nur virtuell mit Hilfe eines technischen Hilfsmittels geschehen. Dagegen kann eine Fortbewegung im Raum sowohl virtuell als auch real durchgeführt werden.

Aufgrund der Möglichkeit, dass sich der Benutzer von einem Ort, an dem er Informationen erhalten hat, zu einem anderen Ort bewegen kann und dort wieder etwas Neues erfährt, hat der aktuelle Aufenthaltsort des Benutzers einen maßgeblichen Einfluss auf die Informationen, welche der Benutzer erhält. Damit besteht die Möglichkeit, dass der Student aus dem vorangegangenen Beispiel sich vom System an den tatsächlichen Ort, an dem Tesla studiert hat, navigieren lässt und dort zum Beispiel dessen Zeugnisse betrachten kann.

Durch die Verknüpfung der einzelnen Informationsentitäten untereinander entsteht ein sehr großes Netzwerk, welches die verschiedenen Informationen über Raum und Zeit verknüpft. Damit ist es möglich, den Zusammenhang von aktuellen mit historischen Ereignissen herzustellen. Durch diese Verbindungen entstehen immer neue Informationen, somit bietet das System den Benutzern einen Zugang zu einem Meer an Informationsquellen.

1.2 Motivation

Der Zugang zu Information ist ein uraltes und grundlegendes Bedürfnis eines jeden Menschen. Information beeinflusst das Handeln entscheidend, da Informationen als Grundlage für eine Entscheidung benutzt werden und so die Aktionen eines Menschen beeinflussen. Es ist somit wichtig, dass die dafür verwendete Information aktuell und richtig ist. Da immer wieder neue Informationen in das Campus 2011 System einfließen und durch die Verknüpfungen dieser untereinander wiederum neue Informationen entstehen, sieht sich der Benutzer einer Informationsflut gegenüber gestellt, welche ihm durch eine Reihe von Informationsquellen (z.B.: TU-Terminal, Smartphone) jederzeit und an jedem Ort zur Verfügung stehen.

Da der Informationsbedarf individuell je nach Benutzer unterschiedlich sein kann, muss die Information dementsprechend eingeschränkt werden. Diese Aufgabe kann bei digitalen Daten von einem Informationssystem erledigt werden, welches darüber hinaus auch noch die

Verwaltung und Speicherung der der Information zu Grunde liegenden Daten erledigt. Des Weiteren ermöglicht es die einfache Verknüpfung der unterschiedlichsten Information und die Aufbereitung zu neuen Informationen (z.B.: Zusammenfassen).

Jede Information hat oder baut auf Informationen auf, welche einen Raumbezug haben und kann so auch für das Filtern der Information herangezogen werden, indem ein Gültigkeitsbereich festgelegt wird. Die Dienste welche den Zugriff auf diese Information bereitstellen, können unter dem Begriff der raum- bzw. ortsabhängigen Dienste (Location Based Services) zusammengefasst werden.

Die Verfügbarkeit von ortsabhängigen Diensten macht in den meisten Fällen erst Sinn, wenn das dahinter liegende Informationssystem nicht nur einen einzelnen Punkt, sondern ein bestimmtes Gebiet mit mehreren Informationsquellen versorgt. Diese Informationsquellen müssen dabei nicht zwingend einen fixen Punkt in der Realität haben, sondern können auch mobil sein (z.B. Handy). Auch kann eine jede Quelle zu anderen führen. Dieses Prinzip kann so ein Individuum an die Hand nehmen und es durch den Bereich des Informationssystems leiten und so auch den Forscher und Entdecker in einer Person wecken.

Die Unterstützung des Benutzers bei der Auffindung bestimmter Orte wird umso wichtiger, je größer und unbekannter das Gebiet, in dem er sich bewegt, ist. Dies trifft unter anderem auf den weitläufigen und heterogenen Campus der TU-Graz (siehe Abbildung 1) zu. Mit Hilfe von ortsabhängigen Diensten wird es so Besuchern, Studenten und TU-Mitarbeitern ermöglicht, alle Winkel des Campus kennenzulernen. Im Zuge des 200-jährigen Bestehens der TU-Graz [1], wurde wie bereits erwähnt das Forschungsprojekt Campus 2011 ins Leben gerufen. Ein Teil dieses Projekts umfasst das Themengebiet der ortsabhängigen Dienste.



Abbildung 1: Campus Bereich der TU-Graz [2]

1.3 Aufbau und Struktur

Die Arbeit gliedert sich in sieben Kapitel, wobei diese teilweise aufeinander aufbauen. Auch wird an bestimmten Stellen auf folgende oder zurückliegende Kapitel verwiesen.

In **Kapitel 1** wird die Motivation, welche hinter dieser Arbeit steckt, angeführt, als auch die Zielsetzung des Forschungsprojekts Campus 2011 betrachtet. Des Weiteren gibt es Auskunft über die Struktur und den Aufbau dieser Arbeit.

Kapitel 2 befasst sich mit der grundlegenden Thematik und Begriffsdefinitionen. Des Weiteren wird auf die Definition und den grundlegenden Aufbau eines LBS-Systems eingegangen und deren Aufgabenbereich definiert.

Kapitel 3 befasst sich mit den Hauptkomponenten eines LBS-Systems und zeigt so einerseits welche Komponenten Einfluss auf das System haben, und andererseits auch wie dessen Strukturen aussehen.

Kapitel 4 betrachtet nun die eigentliche Systemarchitektur. Wobei zuerst auf die logische Struktur eingegangen wird und dann in einem weiteren Schritt auf die Abbildung dieser Architektur in Systemmodulen und eine Ausführung deren Aufgabenbereiche und Funktionsweise.

Kapitel 5 befasst sich mit der Thematik welche Hardware und Software benötigt wird und gibt einige Beispiele welche existierende Software bzw. Technologien herangezogen werden können und wie diese eingesetzt werden.

Kapitel 7 bietet eine Zusammenfassung der Arbeit und gibt einen Ausblick auf die Einsatzmöglichkeiten eines solchen LBS-Systems.

Der **Anhang** schließlich bietet ein Verzeichnis über die im Text vorkommenden Abbildungen, Tabellen, Code-Beispiele.

2 Grundlagen

Wie im vorangegangenen Kapitel angeführt sind ortsbezogene Dienstleistungen ein kleiner, aber wichtiger Bestandteil des Campus 2011 Systems, da beinahe jede Information in irgendeiner Art und Weise einen Ortsbezug aufweist.

Aus diesem Grund wird im folgenden Kapitel der Begriff eines Location Based Services Systems definiert. Dies umfasst einerseits eine Definition des Begriffs hinsichtlich seiner Verwendung als auch andererseits die Analyse welche Hauptkomponenten bzw. Themengebiete mit einem solchen System verbunden sind.

Des Weiteren wird im Zuge der LBS-Komponentenanalyse der Einfluss des TU-Campus 2011 Projekts angeführt und welche Anforderungen im Speziellen dadurch an ein solches LBS-System gestellt werden und wie diese grundsätzlich erfüllt werden können.

Die Anwendung von Normen oder Standards führt zu einer guten Interoperabilität mit anderen Systemen, da allgemein anerkannte und definierte Schnittstellen und Funktionalität unterstützt werden. Dazu wird hier auf den vom Open Geospatial Consortium (OGC, [3]) entwickelten OpenLS Standard eingegangen. Es wird seine Verwendung im Zuge des LBS-Systems definiert und dargestellt wie die Besonderheiten, welche sich durch den Campus 2011 ergeben, sich auf diesen Standard auswirken.

Dies soll ein grundlegendes Verständnis für die Thematik und Definition von LBS im Rahmen des Campus 2011 hervorrufen und so die Basis für die Entwicklung eines Systemkonzepts für ein LBS-System bilden.

2.1 Location Based Services

Durch die zunehmende Verbreitung von sogenannten Smartphones haben immer mehr Menschen die Möglichkeit, unterwegs bequem auf das Internet zuzugreifen und eine Vielzahl unterschiedlicher Informationsquellen zu benutzen. Im Zuge dieser Verbreitung wird der Begriff der ortsabhängigen Dienstleistungen (Location Based Services) vermehrt verwendet.

2.1.1 Definition

Für den Begriff von Location Based Services gibt es eine Reihe unterschiedlicher Definitionen. So findet man im Glossar des UMTS Forums folgende Definition:

Mobile services and applications that make use of the subscribers geographical location. These could include mapping or localised shopping and entertainment services. [4]

Eine zusammenfassende Definition, welche auf der Basis mehrere Quellen getroffen wurde, beschreibt LBS wie folgt:

Standortbezogene Dienste (Location Based Services, LBS) sind mobile Mehrwertdienste, die dem Anwender mit den Mitteln der mobilen IT selektive Informationen oder Dienste auf Grundlage einer Lokalisierung in Echtzeit zur Verfügung stellen. [5], S. 12

An den oben angeführten Zitaten sieht man, dass je nach Quelle und Zugang zur Thematik leicht unterschiedliche Auffassungen der Definition von LBS bestehen, je nachdem welches höhere Ziel (z.B. Bereitstellung von Einkaufsfunktionen oder Mehrwertdienst) sie verfolgen. Jedoch haben sie gemeinsam, dass die Position des Anwenders ausschlaggebend für das Vorhandensein bzw. den Umfang der bereitgestellten Funktionen ist. Die Abhängigkeit der Funktionalität von der aktuellen Position lässt sich dabei unterschiedlich definieren.

Betrachtet man die Definitionen der internationalen Normungsorganisation in der Norm ISO 19132:2007, so kann man die Ortsabhängigkeit der Dienstleistung differenzierter betrachten:

Location Based Service (LBS)

*service whose return or other property is dependent on the **location** of the client requesting the service or of some other thing, object or person [6]*

Im Falle von LBS ist der Dienst an jedem Ort, an dem eine Verbindung mit diesem aufgebaut werden kann, verfügbar. Der Ort beeinflusst nur die vom Dienst entgegengenommenen Parameter und das dem Benutzer zurückgelieferte Resultat des Location Based Services.

Location Dependent Service (LDS)

*service whose availability is dependent upon the **location** of the client [6]*

Im Falle von LDS schränkt der Ort die Verfügbarkeit des Dienstes auf einen bestimmten Ort bzw. ein bestimmtes Gebiet ein. Die Funktionalität des Dienstes ist aber nicht unbedingt vom Ort abhängig.

In weiterer Folge wird nicht zwischen LBS und LDS unterschieden sondern der Begriff LBS so verwendet, dass die Ortsabhängigkeit sowohl auf das Ergebnis als auch auf die eigentliche Verfügbarkeit des Dienstes Einfluss hat.

So wird die allgemeiner gefasste Definition von OGC zu Grunde gelegt und die Definition auf beliebige Anwendungsfälle (Dienste) erweitert:

A wireless-IP service that uses geographic information to serve a mobile user. Any application service that exploits the position of a mobile terminal. [7]

Aus den vorangegangenen Definitionen von LBS lassen sich die Hauptkomponenten eines jeden LBS-Systems festlegen und so die grundlegenden Anforderungen definieren (siehe Abbildung 2: Komponenten eines LBS-Systems):

Kommunikation:

Eine grundlegende Voraussetzung für das Funktionieren eines LBS ist die Möglichkeit eines Endbenutzers mit dem System zu kommunizieren und so die Dienstleistungen in Anspruch nehmen zu können. Dies geschieht im digitalen Zeitalter über geeignete Eingabegeräte (z.B.: PC, Smartphone, Handy, TU-Terminal) und digitale Kommunikationsmedien (z.B.: Internet). Die Kommunikationskomponente ist eine der wichtigsten Teile eines LBS-Systems, da ohne sie keine Interaktionen zwischen dem eigentlichen Kernsystem und den Konsumenten (Benutzern) möglich ist.

Information:

Das Beanspruchen einer digitalen Dienstleistung beruht in jedem Fall auf dem Austausch von Daten. Da diese digitalen Daten im weiteren Sinne Information darstellen, sind sie ein weiterer Eckpfeiler eines jede LBS-Systems. Dieser Punkt beschäftigt sich einerseits mit der Verarbeitung und somit mit der Ermittlung der jeweiligen Daten, welche eine angeforderte Information bilden, und andererseits die Verpackung von dem System übergebenen Informationen in geeignete Datenstrukturen bzw. Datensätze. Die Information ist nicht direkt frei zugänglich, sondern wird erst durch das Benutzen von Dienstleistungen (Services) dem Benutzer bereitgestellt.

Position:

Die beiden vorher angeführten Eckpunkte beschreiben jedoch nur ein allgemeines digitales Informationssystem und reichen aus um prinzipiell dem Benutzer Information zur Verfügung zu stellen.

Im Falle von LBS kommt als ein weiterer Eckpunkt die aktuelle Position bzw. eine vom Benutzer angegebene Position zum Tragen. Dies ist notwendig, da die Verfügbarkeit einzelner Dienstleistungen vom aktuellen Aufenthaltsort des Benutzers abhängig ist. So macht eine Dienstleistung, welche die Routenberechnung bereitstellt, nur am Campus der TU Graz Sinn, da im Falle des Campussystems nur hier die benötigten Graphen vorhanden sind und so keine Routen nach beispielsweise Wien berechnet werden können.

Darüber hinaus beeinflusst der Ort des Benutzers auch die Ergebnisse der einzelnen Dienstleistungen. So liefert die Umkreissuche nur jene gesuchten Objekte, welche sich auch tatsächlich im vom Benutzer definierten Umkreis um seinen Standort befinden. Um diese wichtige Komponente auszufüllen müssen geeignete Methoden und Techniken vorhanden sein, welche die Positionsbestimmung des Benutzers, um genauer zu sein, des vom Benutzer verwendeten Gerätes, zu ermöglichen.

Für die Bestimmung der Position des Konsumenten wird einerseits auf die Kommunikationskomponente zurückgegriffen, um die Position vom Benutzergerät zu empfangen. Andererseits wird auch die Informationskomponente benutzt, um zum Beispiel eine Adresse in vom System verarbeitbare Koordinaten umwandeln zu können.

Aus diesem Grund bilden die drei Basiskomponenten Kommunikation, Information und Position das Grundgerüst eines LBS-Systems.

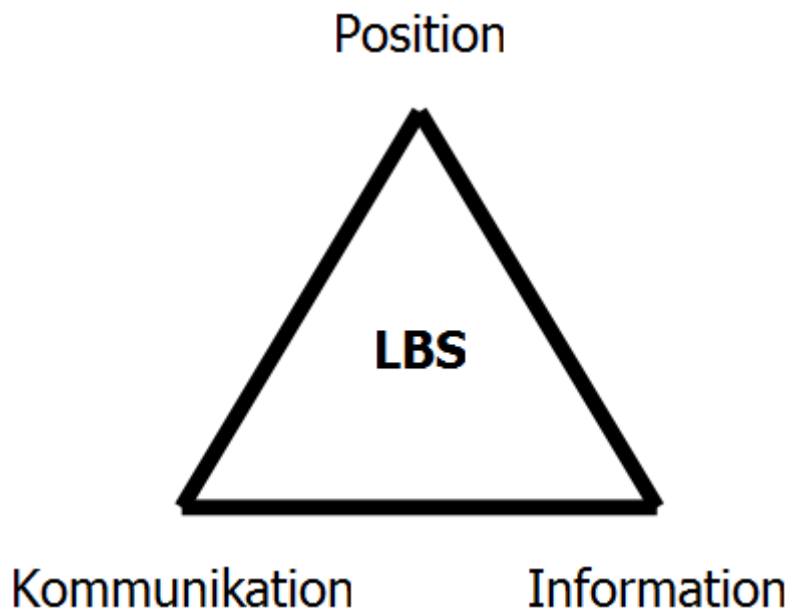


Abbildung 2: Komponenten eines LBS-Systems

Aus diesen recht allgemein formulierten Komponenten eines jeden LBS-System müssen in einem nächsten Schritt die konkreten, grundlegenden Dienstleistungen und deren Schnittstellen nach außen hin zum Benutzer definiert werden.

Aufgrund der heterogenen und sich ständig ändernden Struktur der Geräte, welche ein Benutzer für die Interaktion mit dem System verwenden kann, ist es sinnvoll auf gute Interoperabilität zu achten. Dies erreicht man, indem man auf Standards bzw. Normen zurückgreift, da man in diesem Fall meist davon ausgehen kann, dass diese allgemein akzeptiert und somit jedem bekannt sind. Dies trifft auf das Campus 2011 System zu, da an diesem Forschungsprojekt eine Reihe unterschiedlicher Institute beteiligt sind und durch eine Standardisierung eine leichte Erweiterbarkeit möglich ist. So greift das Institut für Maschinelles Sehen und Darstellen mit seiner Augmented Reality Anwendung über die gleichen Schnittstellen zu wie das Institut für Stadt- und Baugeschichte mit ihrer Ausstellungsanwendung.

Für die grundlegende Ausstattung eines LBS-Systems hat das OGC in der OpenLS Spezifikation einen offenen Standard für die Schnittstellen und Kerndienstleistung eines LBS-Systems geschaffen. Da bei der Definition einige der führende Organisationen [7] im Bereich Geoinformation beteiligt waren und dies in einer offenen Weise geschehen ist, bietet dieser Standard eine gute Zukunftssicherheit.

Aus diesem Grund wurde diese Spezifikation für die Systemarchitektur herangezogen um später möglichst offen gegenüber etwaigen Entwicklungen zu sein.

2.1.2 LBS-Schnittstellen Definition: OpenLS

Der OpenLS Standard baut teilweise auf den von OGC definierten Standards hinsichtlich der Handhabung von geographischen Daten (Geographic Markup Language [8], GML) auf und erweitert diese um die LBS-Komponenten. Dies umfasst unter anderem die Definition der grundlegenden Kommunikationsschnittstellen und Protokolle, als auch den grundsätzlichen Aufbau eines LBS-Systems. Darüber hinaus werden noch die grundlegende Anwendungen und deren Aufgabenbereiche definiert.

Die OpenLS Spezifikation definiert einen GeoMobility Server (GMS) welcher den Kern des LBS-Systems darstellt. OGC definiert hierfür den GMS wie folgt:

The open service platform comprising the Core Services developed under the OGC OpenLS initiative [7].

Der OpenLS GeoMobility Server beschreibt nur den grundlegenden Rahmen eines LBS-Systems. Die dahinter liegende Strukturen sind hierbei nicht vorgegeben, sondern die Definition des GeoMobility Servers umfasst folgende grundlegende Punkte, welche den Aufbau des Servers beschreiben:

- OpenLS Core Services und die dazugehörigen Schnittstellen.
- OpenLS Informationsmodell und die Abstrakten Datentypen (ADTs).

Optional fallen noch folgende Punkte in den Zuständigkeitsbereich eines GeoMobility Servers:

- Anwendungen, welche auf den OpenLS-Core Services aufbauen und auf diese zurückgreifen.
- Geodaten (z.B.: Kartendaten) welche von den Core Services benutzt werden.
- Weitere unterstützende Funktionalität (z.B.: Protokollierung, Abrechnungen).

Betrachtet man die Rolle, welche der GeoMobility Server in einem LBS-System wie auch im Falle des Campus 2011 System übernimmt, ergibt sich folgendes Bild (siehe Abbildung 3), welches in Anlehnung zur Rolle in der OGC Spezifikation [7] erstellt wurde:

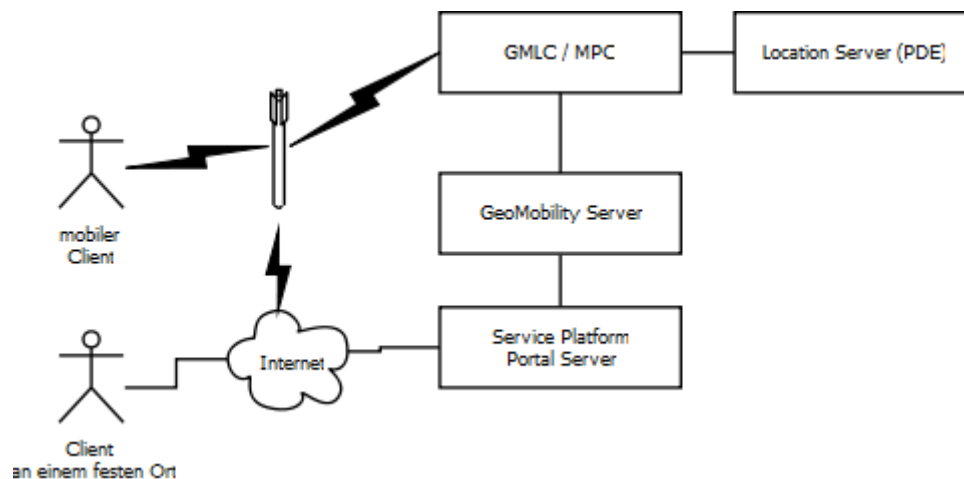


Abbildung 3: Rolle des GeoMobility Servers im Rahmen eines LBS-Systems
Quelle: in Anlehnung an [7]

Wie in obiger Darstellung abgebildet nimmt ein Client (das Gerät das ein Benutzer verwendet), welches sich an einem festen Standort befindet (z.B.: TU-Terminal), die Verbindung über das Internet zum Portal Server bzw. zur Serviceplattform auf. Im Falle des Campus 2011 Systems stellt dies in erster Linie einen Webserver dar, welcher in weiterer Folge gegebenenfalls auch die Authentifizierung des Clients übernimmt. Es besteht jedoch auch die Möglichkeit, die Service Plattform zu umgehen und direkt mit dem GMS zu kommunizieren.

Im Falle eines mobilen Clients erfolgt der erste Teil der Kommunikationsstrecke über ein Funknetzwerk (z.B.: WLAN, GPRS, UMTS, LTE). Der Betreiber des Funknetzes speist die Kommunikation dann für den Rest der Kommunikationsstrecke in das Internet ein. Ab diesem Punkt läuft die Kommunikation mit dem Client wie bei einem Client mit festem Standort ab und der Netzbetreiber kümmert sich um die Einspeisung ins Funknetz.

Der Client kann dem GMS seine Position selbst mitteilen oder der GMS greift auf das GMLC (Gateway Mobile Location Center) des Funknetzbetreibers zu, welches ihm die Position des mobilen Clients mitteilt.

Der Betreiber kann die Position des mobilen Clients zum Beispiel auf Grund der Kommunikationsparameter, (z.B.: Empfangsstärke und verwendete Antenne lassen auf die Entfernung und Winkel zu einer Funkbasisstation schließen), ermitteln. Hierfür benutzt das GMLC den Location Server, welcher die eigentliche Positionsbestimmung vornimmt.

Betrachtet man den Aufbau des GeoMobility Server etwas genauer, so ergibt sich folgender Aufbau (siehe Abbildung 4 analog zu [7]):

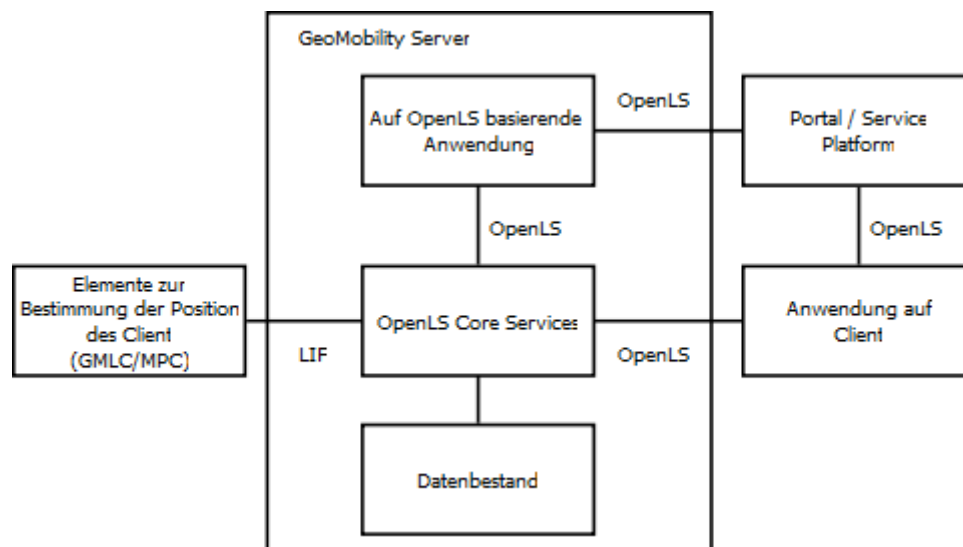


Abbildung 4: Aufbau des GeoMobility Servers (GMS)
Quelle: in Anlehnung an [7]

Eine mobile Anwendung auf dem Gerät des Endbenutzers (z.B.: Web-Browser, App) kann entweder direkt oder über eine Applikation auf dem GeoMobility Server mit den Core Services kommunizieren oder den Umweg über einen Service Provider bzw. OpenLS nehmen, welcher zum Beispiel die Authentifizierung übernimmt. Anschließend kommuniziert der Serviceprovider mit einer OpenLS Anwendung, welche erst dann mit den eigentlichen Core Services kommuniziert.

Die OpenLS Core Services greifen des Weiteren auf die für die Verarbeitung benötigten Daten (z.B.: Straßennetz, Adressenverzeichnis), als auch ggf. auf Dienstleistungen externer Server zurück (z.B.: für die Positionsbestimmung, LIF [9]).

Der GMS sollte in minimaler Ausführung folgende Grunddienste zur Verfügung stellen, die auch des Weiteren den grundlegenden LBS-Dienstleistungen entsprechen, welche im Zuge des Campus 2011 zum Tragen kommen:

OpenLS Core Services:

Zur Grundausstattung eines jeden GeoMobility Servers sollten folgende fünf Core Services zählen, um die grundsätzlichen Anforderung an ein LBS-System zu erfüllen:

- Directory Service
- Gateway Service
- Location Utility Service (Geocode / Reverse Geocode)
- Presentation Service
- Route Services

Die Kommunikation mit den Services erfolgt hierbei nach dem Request- und Response- Prinzip und beruht auf dem Austausch von in XLS codierten Nachrichten (siehe 2.2.2 OpenLS Kommunikationsschnittstelle).

Der folgende Abschnitt befasst sich mit den vom GeoMobility Server bereitgestellte Dienstleistungen (Services) und deren Aufgabenbereichen im Zuge des Campus 2011 Systems:

Der **Directory Service** ermöglicht einem Benutzer die Abfrage eines bestimmten Verzeichnisses zum Auffinden von Produkten oder Dienstleistungen. Dies könnte mit einer Suche in den Gelben Seiten eines Telefonbuchs verglichen werden. Diese Anfrage kann sich in zweierlei Weise auf eine Ortsangabe beziehen:

- Pinpoint Directory Service
- Proximity Directory Service

Der Pinpoint Directory Service ist vom aktuellen Aufenthaltsort des Benutzers unabhängig. Dies ermöglicht dem Benutzer ortsunabhängige Suchen (z.B.: Suche nach dem Gebäude der Alten Technik) oder auch die Suche an anderen Orten, welche nicht der aktuellen Position entsprechen (z.B.: Suche nach Hörsälen in der Alten Technik; Annahme: beispielsweise aktuelle Position im Gebäude der Neuen Technik).

Im Zuge des Campus 2011 Systems kann der Pinpoint Directory Service einerseits vom Benutzer selbst direkt benutzt an einem von ihm definierten Ort benutzt werden, um dann so zum Beispiel alle Hörsäle des Gebäudes Steyrergasse 30 zu erhalten.

Andererseits wird diese Ausprägung des Directory Service auch dazu benutzt, zu einem Informationsknoten die umliegenden Informationsknoten zu erhalten. So könnte man zum Beispiel die Informationsknoten zum Objekt Tesla Labor erhalten.

Beim Proximity Directory Service handelt es sich um eine Suche, welche sich auf die aktuelle Position des Benutzers bezieht. Ansonsten bietet er dieselben Möglichkeiten (z.B.: Angabe eines Radius bei der Umkreissuche) wie der Pinpoint Directory Service.

Ein Beispiel im Zuge des Campus 2011 Systems ist die Suche eines Studenten nach den nächst liegenden Sanitäreinrichtungen.

Da beide Services sich nur durch die Art der aktuellen Position unterscheiden, erfolgt die Verarbeitung für beide auf dieselbe Weise. So können vom Ort unabhängige Suchparameter (z.B. Sortierung, Anzahl der Ergebnisse) angegeben werden. Dies wird genutzt um die Anforderungen, welche durch das Campus 2011 System entstehen, zu erfüllen (siehe 2.3.2 Informationskontext).

Das Ergebnis der Suche mit Hilfe des Directory Services ist eine Liste der gesuchten Objekte mit den Eigenschaften, welche in der Anfrage definiert wurden. Diese Liste kann nun von der Clientanwendung weiterverarbeitet bzw. dem Benutzer präsentiert werden.

Der **Gateway Service** dient dazu, die Position einer mobilen Einheit aufgrund von übermittelten Daten zu bestimmen und dies auch der mobilen Einheit mitzuteilen. Der Zugriff auf die Positionsdaten eines mobilen Geräts erfolgt hierbei über den Betreiber des Netzwerks. Somit stellt dieser Dienst die Schnittstelle zwischen dem GMS und einem Location Server (vom Netzbetreiber) dar. Der Zugriff darauf erfolgt mit Hilfe der Mobile Location Protocol Specification (MLP). Somit ist der Zugriff auf die Positionsdaten der mobilen Geräte unabhängig vom tatsächlich genutzten Netzwerk (z.B.: GSM, UMTS).

Im Falle von Campus 2011 wird der Gateway Service für das interne TU-Netzwerk (LAN, WLAN) verwendet um dann so einerseits die Position der ortstreuen (z.B.: TU-Terminal) als auch der mobilen Clients zu ermitteln.

Der **Location Utility Service** stellt die Verbindung zwischen geographischen Koordinaten und der Angabe von Adressen her. Hierbei können zwei Unterscheidungen getroffen werden, welche die Art der Abfrage beschreiben:

- Geocode
- Reverse Geocode

Wenn eine Geocode-Abfrage durchgeführt wird, so werden zur einer gegebenen Adresse oder Positionsbeschreibung (z.B.: Parkplatz vor Neuer Technik) die dazugehörigen Koordinaten ermittelt. Die Angabe der Adresse erfolgt hierbei in Form der Codierung der Adresse in einen Abstract Data Type (ADT) (siehe 2.3.3 Informationshandhabung in OpenLS) und die zurückgelieferten Koordinaten werden in Form von Geography Markup Language (GML) codiert. Das Ergebnis besteht aus einer Liste der Adresse(n) mit den dazugehörigen Koordinaten. Somit ist es auch möglich, bei unvollständiger Angabe einer Adresse die möglichen Koordinaten zu ermitteln.

Sucht ein Benutzer des Campus 2011 Systems eine bestimmte Adresse, zum Beispiel die Steyrergasse 30, so erhält er die Koordinaten aller Eingänge, welche diese Adresse tragen. Die Geocode-Abfrage wird auch dann angewendet, wenn bei der Routenberechnung eine Adresse als Ausgangspunkt bzw. Zielpunkt vorliegt. Diese muss nun in Koordinaten umgewandelt werden, um den Ausgangspunkt im Routengraphen festzustellen.

Die Reverse Geocode-Abfrage erfüllt genau den entgegengesetzten Aufgabenbereich der Geocode-Abfrage und liefert zu übergebenen Koordinaten die zugehörige Adresse. Mit Hilfe weiterer Parameter kann man die Form des zurückgegebenen Ergebnisses beeinflussen (z.B.: Adresse mit oder ohne PLZ).

Der Fall, dass zu einer Koordinate die dazugehörige Adresse benötigt wird tritt dann auf, wenn der Benutzer zu einem bestimmten Punkt auf einer Karte die Adresse benötigt um vielleicht später einen Brief an diese Adresse zu senden.

Beim **Presentation Service** wird eine graphische Repräsentation von einer geographischen Informationen zur Verfügung gestellt. Bei der Anforderung werden Parameter übergeben,

welche einerseits die Eigenschaften des geographischen Inhalts festlegen (Lage, Ausdehnung, Layer), die das Endprodukt betreffen, und andererseits solche, die die Darstellung des Inhalts (Farbe, Symbole) und die Eigenschaften des Endprodukts (Bildformat, Größe) festlegen. Als Ergebnis erhält man einen Verweis (Link) auf das erzeugte kartographische Produkt. Die tatsächliche Generierung wird zum Beispiel von einem WebMapService (WMS) [10] übernommen werden.

Der Presentation Service wird im Zuge des Campus 2011 Systems sehr häufig benötigt um dann so zum Beispiel eine berechnete Route nicht nur in Form einer Manöverliste zu präsentieren, sondern auch um dem Benutzer eine graphische Repräsentation der Route anzubieten. Auch können so die aktuelle Position und die umgebenden Objekte in der Karte dargestellt werden um die Topologie darzustellen.

Der **Route Service** ermöglicht es dem Benutzer von der aktuellen Position oder von einer von ihm definierten Position aus eine Route zu einem Zielpunkt zu berechnen. Dabei können die Eigenschaften der Route (z.B.: kürzeste, schnellste) oder weitere Wegpunkte definiert werden. Das Ergebnis umfasst die gewünschte Route in diversen Formen (Instruktionsfolge, Wegkarte) und kann für weitere Abfragen auf dem Server abgespeichert werden. Das zurückgelieferte Ergebnis muss in minimalster Form eine Zusammenfassung der Route beinhalten.

Dieser Service ist auch ein wichtiger Bestandteil des Campus 2011 Systems. Einerseits wird er für die normale Routenberechnung zwischen zwei vom Benutzer definierten Punkten verwendet, stellt aber auch auf Abfrage bereits im Voraus berechnete Routen zur Verfügung. Eine solche Route wäre zum Beispiel die Route der Entstehung der TU Graz wo die Reihenfolge der zu besuchenden Zwischenpunkte vorgeben ist und somit nicht mehr unbedingt den Kriterien schnellste oder kürzeste Route entspricht.

Neben diesen fünf Core Services kommen noch folgende zwei Services zum Einsatz, welche einerseits die Zügführung eines mobilen Client ermöglichen, als auch andererseits die Überwachung dieser:

- Navigation Service
- Tracking Service

Der **Navigation Service** entspricht in den Grundzügen und im Funktionsumfang grundsätzlich jenen des Route Service, mit der einzigen Einschränkung, dass der Ausgangspunkt immer der aktuellen Position des Clients entspricht. Ändert sich diese, wird dem Client eine neue Route übermittelt. Dafür muss das Clientgerät seine Position in ausreichender Häufigkeit und Genauigkeit dem Service mitteilen können. Dies kann mit Hilfe der im Gerät verbauten Sensoren geschehen oder auch mit Hilfe des Gateway Service.

Ein Campus 2011 Benutzer wird mit Hilfe dieses Services von einem Punkt des Campus zum anderen geführt, wobei der Benutzer nicht allen Anweisungen folgen muss sondern auch andere Wege benutzen kann. Das System passt sich automatisch an die Position des Benutzers an und berechnet von diesem aus die gewählte Route zum Zielpunkt.

Der **Tracking Service** steht nicht direkt den Konsumenten des Campus 2011 Systems zur Verfügung, sondern übernimmt die Überwachung der Clientgeräte. Dies ist notwendig, damit die Position möglicher ausgegebener Clientgeräte überwacht werden kann und man so immer über deren Verbleib informiert ist. Eine weitere Anwendung besteht darin, dass sie für das Auslösen bestimmter Ereignisse (z.B.: Starten einer Präsentation) benutzt werden oder für die Benutzung interaktiver Touren, wobei der Benutzer bei Erreichen eines bestimmten Ortes eine Nachricht erhält oder sich die gerade dargestellte Information ändert.

Für die Funktion des Systems besteht ein wichtiger Punkt in der Kommunikation eines Clients mit dem System bzw. den vom System bereitgestellten Dienstleistungen. Das folgende Kapitel beschäftigt sich damit.

2.2 Kommunikation

2.2.1 Allgemein

Damit ein Benutzer mit dem System überhaupt interagieren kann, muss ein Kommunikationskanal von ihm zum System aufgebaut werden, über den die Daten ausgetauscht werden. Im digitalen Fall stellt dies eine Netzwerkverbindung dar, welche über unterschiedliche Netze ablaufen kann. So kann die Kommunikation zuerst über ein UMTS-Netz laufen um sich anschließend dann in Internet fortzusetzen. Dabei müssen aber die Eigenschaften (z.B.: Bandbreite) berücksichtigt werden. Dies ist notwendig um das benutzte Netzwerk hinsichtlich Bandbreite und Kosten optimal ausnutzen zu können.

Beim Austausch der Datensätze muss dabei ein Protokoll eingehalten werden (beide Gegenstellen müssen dieselbe Sprache sprechen, sonst verstehen sie sich nicht). Für diesen Fall bietet OpenLS (siehe Abbildung 5) ein Request/Response Schema an, welches die Core Services in Form von Webservices ausführt (siehe 5 Technologien).

2.2.2 OpenLS Kommunikationsschnittstelle

Für den Austausch der Daten zwischen dem Benutzer und dem System definiert OpenLS einerseits einen Client und andererseits einen Server. Diese beiden kommunizieren über ein Netzwerk, über das sie ihre Daten in einem normierten Protokoll austauschen. Die OpenLS Core Services basieren auf folgender grundsätzlicher Vorgehensweise, welche auf den Austausch von XML-Codes aufsetzt (siehe Abbildung 5):

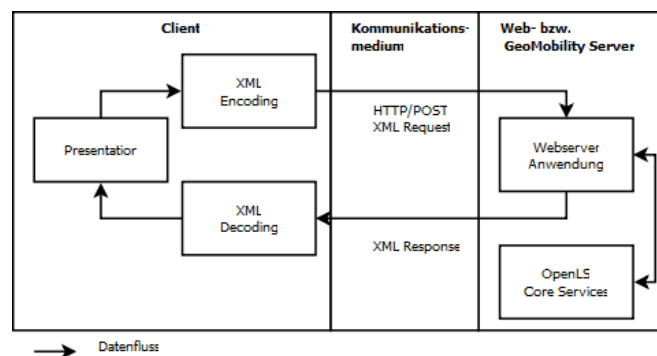


Abbildung 5: Ablauf eines OpenLS Request/Response Paares
 Quelle: in Anlehnung an [7]

Der Benutzer spezifiziert auf einer graphischen Oberfläche (Präsentationsschicht) seine Anfrage. Anschließend codiert die Clientanwendung seine Anfrage (Request) in die für die aktuelle Funktion benötigte XLS-Struktur und sendet diese mittels HTTP/POST an den GMS. Dieser liest die benötigten Daten aus dem XLS-Request-Code heraus und verarbeitet die angeforderten Daten (z.B.: Routensuche), um anschließend das Daten-Ergebnis in einen XLS-Response-Code zu verpacken und dann an den Client zurückzusenden. Dieser decodiert das Resultat und leitet dieses an die Präsentationsschicht, welche sich um die Darstellung kümmert, weiter.

Die Übermittlung der Daten im XLS-Format kann entweder direkt mittels HTTP (siehe [11]) übertragen, oder auch noch in ein höheres Übertragungsprotokoll (z.B.: SOAP) eingepackt werden.

Die Codierung der Daten bei der Kommunikation in beide Richtungen erfolgt an Hand des OGC XLS-Formats, welches wiederum eine XML-Codierung darstellt und so einerseits für Menschen und andererseits für Maschinen lesbar ist.

XML for Location Services (XLS):

The method for encoding request/response messages and associated Abstract Data Types for the GeoMobility Server. [7]

Der Aufbau eines solchen XLS- Request/Response Paares sieht wie folgt aus:

Request:

Sucht der Client zum Beispiel einen bestimmten Hörsaal welcher ein „A“ im Namen hat (z.B. Hörsaal A), so sieht die Anfrage (Request) wie folgt aus (siehe Code 1: XLS-Request):

Code 1: XLS-Request

1	<XLS>
2	<REQUEST HEADER />
3	<REQUEST>
4	<DIRECTORY REQUEST>

```

5      <POIProperties directoryType="Hoersaal">
6      <POIProperty name="Bezeichnung" value="%A%"/>
7      </POIProperties>
8      </DIRECTORY REQUEST>
9      </REQUEST>
10     </XLS>

```

Im optionalen `<REQUEST HEADER />` werden Informationen zum Client und für die Kommunikation notwendige Informationen übergeben (z.B.: Client Name, Client Passwort, Client ID).

Das `<REQUEST>` Tag beinhaltet den, je nach angesprochenem Core Service unterschiedlichen Request (z.B. Directory Request, Route Request). Dies ist die grundsätzliche Struktur eines jeden Requests an den GMS.

Innerhalb eines jeden Core-Service-Request-spezifischen Bereichs werden die Parameter, welche der Dienstleistung übergeben werden, abgespeichert. In dem Fall des Beispiels ist dies die Angabe des Verzeichnis (z.B. Tabelle in der Datenbank) in welchem gesucht werden soll (`directoryType="Hoersaal"`). Innerhalb dieses Bereichs werden nun die Eigenschaften des zu suchenden Elements definiert. Das Beispiel sucht nach allen Hörsälen, bei welchen in der Spalte „Bezeichnung“ an irgendeiner Stelle der Bezeichnung ein „A“ vorkommt (z.B.: **AE01** bzw. Hörsaal **A**).

Response:

Eine mögliche Antwort auf diese Anfrage könnte wie in folgendem Beispiel (siehe Code 2: XLS-Response) gezeigt aussehen, welches eine Liste der nächsten Hörsäle zurückliefert.

Code 2: XLS-Response

```

1      <XLS>
2      <RESPONSE HEADER />
3      <RESPONSE>
4      <DIRECTORY RESPONSE>
5      <POIContext>
6      <POI ID="2">
7      <POIAttributeList>
8      <POIInfoList>
9      <POIInfo name="Bezeichnung" value="A01"/>
10     </POIInfoList>
11     </POIAttributeList>
12     <gml:POINT>
13     <gml:POS>15.24 46.7</gml:POS>
14     </gml:POINT>
15     </POI>
16     </POIContext>
17     <!-- ... Weiter POIContext Ergebnisse ... -->
18     </DIRECTORY RESPONSE>
19     </RESPONSE>
20     </XLS>

```

Im `<RESPONSE HEADER/>` werden analog zum `<REQUEST HEADER/>` dem Client allgemeine Informationen zum Resultat des Core Services mitgeteilt (z.B.: Fehlermeldungen). Danach kommt der XLS-Response Bereich. Der Response-Header und Responsebereich definieren wieder die grundlegenden Elemente eines XLS-Responses. Der Inhalt des Responsebereichs ist davon abhängig, welcher Core Service ihn erzeugt.

Im Falle des Beispiels ist dies ein Directory-Response welcher für jedes gefundene Element oder jeden Point of Interest (POI) die in der Datenbank gespeicherte ID und die dazugehörige Position (in GML codierte Koordinaten) zurückliefert.

Neben der Position oder Adresse eines gefundenen Elements (`<POIContext>`) können auch noch weitere Informationen in Form einer Attribute Liste (`<POIAttributeList>`) angegeben werden. Bei diesem Beispiel ist das die Bezeichnung des Hörsaals.

Im der Zusammenfassung stellt sich ein typischer Request/Response Ablauf wie folgt dar (siehe Abbildung 6):

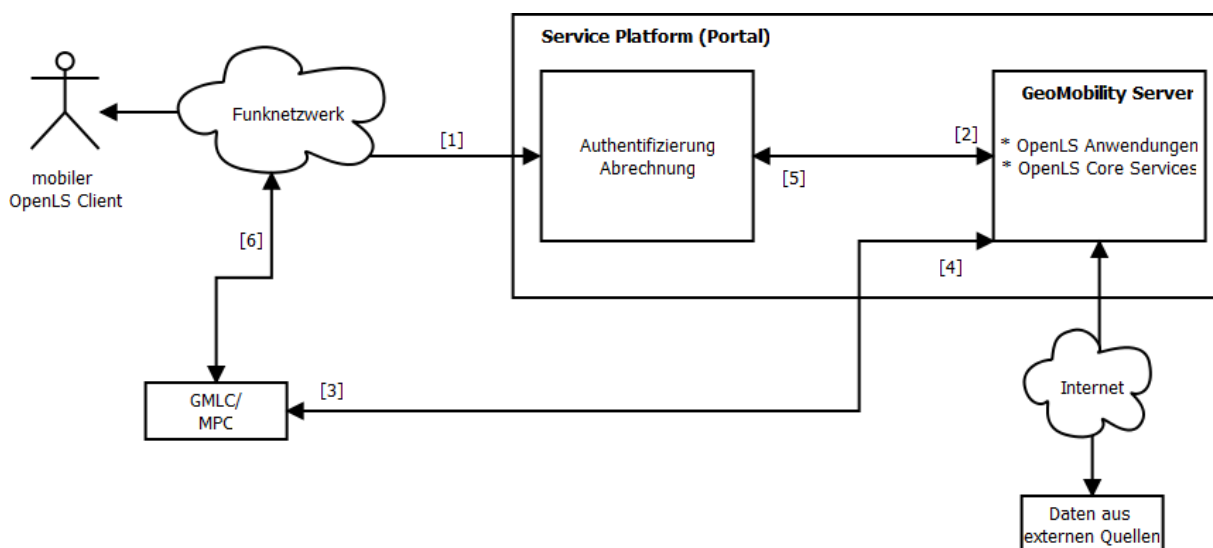


Abbildung 6: Ablauf (1-6 Reihenfolge der Abarbeitung) einer typischen Request / Response Kommunikation
Quelle: in Anlehnung an [7]

Der Client verbindet sich (Abbildung 6: #1) mit der Service Plattform (Portal) um eine Dienstleistung (Service) in Anspruch zu nehmen (Abbildung 6: #2). Die OpenLS Anwendung oder der Service ermittelt gegebenenfalls die Position des Client (Abbildung 6: #3) und bekommt diese zurückgeliefert (Abbildung 6: #4). Des Weiteren greift der jeweilige Core Service auf die für die Verarbeitung benötigten Daten zu welche sich einerseits im System des GeoMobility Servers befinden, aber auch außerhalb, von Dritten, bereitgestellt werden können. Ein Beispiel für den Zugriff auf Daten Dritter ist zum Beispiel der Zugriff auf externes Kartenmaterial wie von OpenStreetMap oder der Zugriff auf die Datenbank des alumniTUGraz 1887. Wurden die Daten verarbeitet, wird ein OpenLS Response formuliert und dem Portal übermittelt (Abbildung 6: #5). Das Portal übernimmt die Aufgabe, diese Antwort dem Client zu übermitteln und somit die Verbindung auf- und wieder abzubauen. Der Betreiber des Funk-

netzwerks kann die Position des Clients feststellen (Abbildung 6: #6), da er zu mindestens die Basisstation, über die die Kommunikation läuft, kennt.

2.3 Information

2.3.1 Definition

Der Sinn und Zweck eines Informationssystems, zu dem ein LBS-System auch gehört, besteht aus dem Austausch von Informationen. Diese Informationen basieren immer auf Daten, welche in Form eines Datensatzes abgespeichert werden.

Ein Datensatz basiert einerseits auf grundsätzlichen atomaren Datentypen (z.B.: Ziffern, Buchstaben) oder andererseits auch auf bereits bestehende abstrakten Datentypen (z.B.: Adressen). Diese abstrakten Datentypen basieren ihrerseits wieder auf atomaren Datentypen oder anderen abstrakten Datentypen.

Ein einfacher Adressdatentyp besteht so zum Beispiel aus einem Datentyp für Straße und Postleitzahl. Diese bauen wiederum auf den Datentypen Zeichenkette (z.B.: String), für Straße und Ganzzahl (z.B.: Integer), für PLZ auf.

Die Daten, auf welche die Datensätze aufbauen, werden dabei von Sensoren erstellt, welche in Form von tatsächlichen technischen Sensoren (z.B.: GPS-Sensor, WLAN-Sensor) vorkommen oder auch ein realer Mensch sein können, welcher seine Beobachtungen aufschreibt.

Betrachtet man isoliert einen Datensatz, so stellt dieser zwar immer Information dar, jedoch ist der Datensatz meist nicht eindeutig genug, damit ein Benutzer die richtige Information (z.B.: Bedeutung) daraus generieren kann.

So ist zum Beispiel der Datensatz „1:2“ nicht eindeutig genug und kann unter anderem für ein Fußballspielergebnis oder ein Mischverhältnis stehen. So sind noch weitere Daten (Metadaten) über diesen Datensatz notwendig, wie zum Beispiel Mischverhältnis von Zucker und Mehl. Durch diesen Zusatz wird erst die Bedeutung des Datensatzes ersichtlich, nämlich dass dieser Datensatz beispielsweise ein Rezept ist.

Da ein Mensch seine Entscheidungen auf Grund von Informationen trifft, welche ihm bekannt sind (=Weltwissen, z.B.: dass Objekte nach unten fallen) oder welche sie durch Interpretieren der Daten generieren, ist es wichtig, dass die Deutung der Daten korrekt erfolgen kann.

Die eindeutige Zuordnung von Daten und ihrer Bedeutung durch Metadaten führt erst zur Entstehung von verwertbarer Information. Die Metadaten können mit dem eigentlichen Datensatz mitgegeben werden oder ergeben sich aus dem Kontext, indem der Datensatz verwendet wird. Steht der Datensatz „1:2“ in einem Kochbuch, so ist es meist ersichtlich, dass es sich um ein Mischverhältnis handelt.

Diese Verbindung zwischen Datensatz und Kontext kann auch bei der Suche nach bestimmten Informationen verwendet werden. Bei der Vorgabe des Kontexts „Kochbuch“, erhält man dann keine Fußballergebnisse sondern das Mischverhältnis „1:2“, wobei das resultierende Datenergebnis dann noch durch weitere Metadaten, zum Beispiel „Rezept für Marmorkuchen“, verfeinert werden kann.

Der Zusammenhang und das Entstehen von Information wird in folgender Abbildung dargestellt (siehe Abbildung 7):

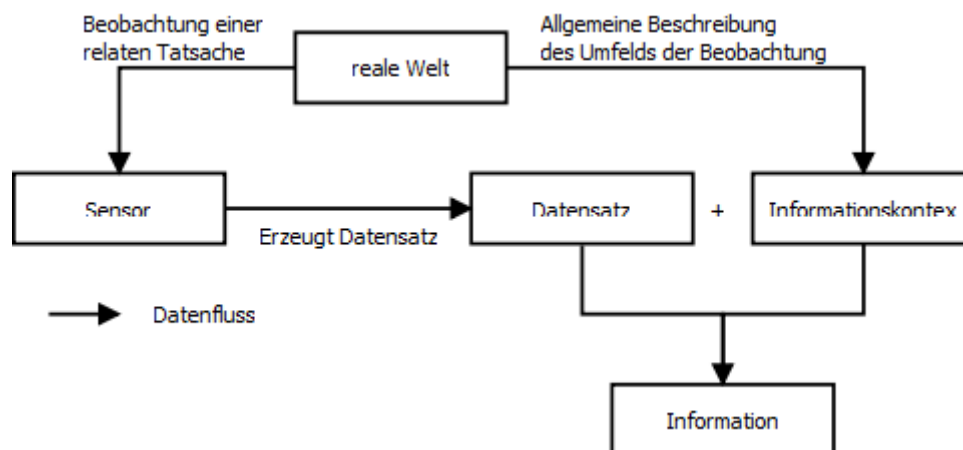


Abbildung 7: Zusammenhang zwischen Realer Welt, Datensatz und Information

Der Sensor beobachtet die reale Welt und empfängt die Daten, welche die Welt aussendet. Aus diesen Beobachtungen formuliert er nun einen Datensatz, welcher in einem im Vorhinein festgelegten Datenformat (z.B.: Zeile in einer Datenbanktabelle) abgespeichert wird. Damit dieser Datensatz zu Information wird, muss dieser mit dem Informationskontext kombiniert werden. Der Aufbau des Informationskontexts muss im Zuge des Erstellens des Systems festgelegt werden, da die Datenverwaltung auf dieser Information aufbaut. Dabei reicht es, den Informationskontext einmal für einen Sensor festzulegen, da dieser das allgemeine Umfeld der Beobachtung beschreibt. Die Kombination eines individuellen Datensatzes mit dem allgemeinen Informationskontext erzeugt so für den Benutzer eindeutige und verständliche Information.

2.3.2 Informationskontext

Der Informationskontext beschreibt die Parameter, welche für die Suche als auch für die Beschreibung eines Datensatzes verwendet werden können. Beim Betrachten eines einzelnen Datensatzes entsteht dadurch die benötigte Information.

Ein wesentlicher Bestandteil eines LBS-Systems ist die Abhängigkeit der verfügbaren Funktionalität bzw. der verwendeten Daten vom aktuellen Ort an dem der Zugriff erfolgt. Somit wird die Ortsinformation zu einer Hauptkomponente, die den Kontext eines Datensatzes bildet. Aus diesem Grund nehmen die Geoinformation und die dazugehörigen Funktionen (z.B.: Routenplanung) einen hohen Stellenwert in Rahmen eines LBS-Systems ein. Neben der

Beziehung der Daten zu einem einzelnen geographischen Punkt (z.B.: Koordinaten) können sich die Daten auch auf Objekte (z.B.: Fläche für ein Gebäude), welche aus mehreren Koordinaten bestehen, beziehen.

Da im Zuge des Campus 2011 Projekts auch zwischen Daten unterschieden werden soll, welche zu unterschiedlichen Zeitpunkten bzw. Zeitspannen Gültigkeit haben, wird die Zeitinformation zu einer weiteren Komponente. Hierbei ist zu beachten, dass die Zeit als ein Zeitpunkt (einmaliges Ereignis) oder als Zeitraum (andauernder Vorgang) angesehen werden kann [12].

Jeder Datensatz kann auch einem oder mehreren Themengebieten (bzw. Tags) zugeordnet werden (z.B.: Thema Hörsaal). Dies hat den Sinn, eine weitere Filterstufe zu erhalten und die Daten besser spezifizieren zu können.

Das System wird des Weiteren von unterschiedlichen Interessensgruppen bedient, wobei nicht jeder Benutzer Interesse an allen verfügbaren Datensätzen hat bzw. der Zugriff auf diese, aus Datenschutzgründen, nur eingeschränkt möglich sein soll. Dadurch wird der Benutzer zu einem weiteren Element des Informationskontexts.

So ergibt sich für den Informationskontext eines jeden Datensatzes folgendes Spannungsfeld (siehe Abbildung 8), welches bei der Benutzung von Systemfunktionen benutzt wird um einerseits die Funktionalität und andererseits das Ergebnis anpassen zu können.

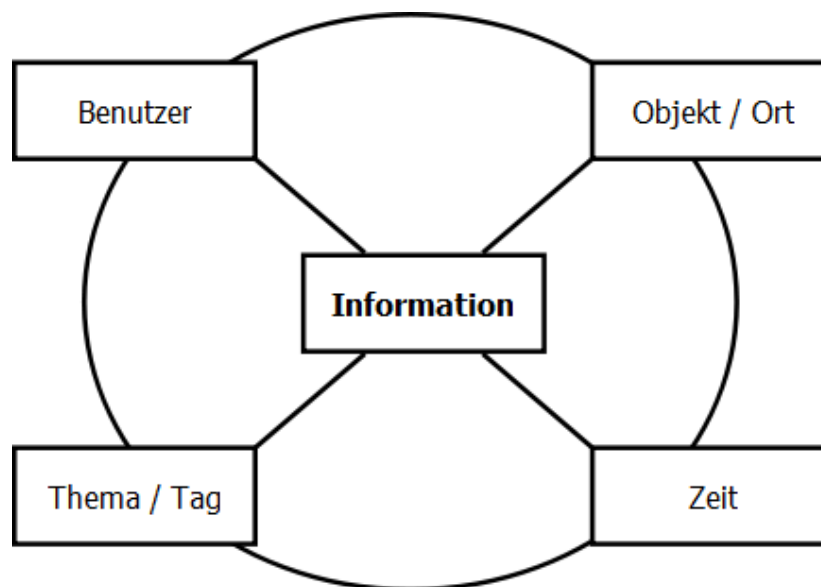


Abbildung 8: Informationskontext im Überblick

2.3.3 Informationshandhabung in OpenLS

Da OpenLS in seinem Informationsmodell (siehe 2.4.1 Positionshandhabung in OpenLS) eigentlich nur Datenstrukturen für die Beschreibung von Ortsinformationen zur Verfügung stellt, müssen die weiteren Kontextparameter auf anderem Weg ins System gelangen. Das Problem lässt sich durch die Übergabe weiterer Parameter lösen. So kann man im Falle des Directory Services bei der Abfrage in Form eines POIProperty weitere Parameter in der Form eines Key-Values Prinzips übergehen:

Code 3: XLS-Request mit Erweiterung für den Campus 2011 Informationskontext am Beispiel eines Directory Services Requests

```

1 <XLS>
2 <REQUEST HEADER />
3 <REQUEST>
4   <DIRECTORY REQUEST>
5     <POILocation>
6       <Address countryCode="AT">
7         <StreetAddress>
8           <Street officialName="Petersgasse"/>
9           <Building number="12"/>
10        </StreetAddress>
11       </Address>
12     </POILocation>
13     <POIProperties directoryType="Hoersaal">
14       <POIProperty name="Bezeichnung" value="%A%"/>
15       <POIProperty name="tag" value="Neue Technik"/>
16       <POIProperty name="event" value="1.1.2011"/>
17       <POIProperty name="user" value="xyz"/>
18     </POIProperties>
19   </DIRECTORY REQUEST>
20 </REQUEST>
21 </XLS>

```

`<POILocation>` übernimmt die Aufgabe des Ortskontexts und liegt noch innerhalb des OGC Standards. `<POIProperty name="tag">` bestimmt den Kontext, der sich auf das aktuelle Themengebiet bezieht. Analog dazu funktionieren auch die anderen `<POIProperty>` Angaben für Zeit (event) und Benutzer (user).

Beim Presentation Service könnte der Kontext durch Angabe eines Layers für jede Kontextausprägung angegeben werden. Die Behandlung der zusätzlichen Kontextparameter als eigenständige Parameter, ohne den OpenLS Standard zu zerstören, ermöglicht einerseits, dass der Kontext nicht immer vollständig angegeben werden muss und andererseits auch dass die Zusammenarbeit mit anderen Anwendungen, welche auf OpenLS basieren, weiterhin möglich ist.

2.4 Position

Im Zuge eines LBS-Systems muss auch mit einer spezielleren Art von Information umgegangen werden. Bei dieser Art der Information handelt sich um Information über geographische Daten (Geoinformation). Dies umfasst unter anderem Koordinaten, aber auch abstraktere Ortsangaben wie zum Beispiel Adressen.

Bei der Handhabung von geographischen Daten ist es wichtig, dass sich diese auf ein Bezugssystem beziehen:

Der Raumbezug stellt eines der wesentlichsten Charakteristika von Geodaten dar. Fragen, in denen das Wo?, das Wie groß?, das Wie weit bis? eine entscheidende Rolle spielt, können nur dann fundiert beantwortet werden, wenn es ein für Fragende und Antwortende eindeutiges Bezugssystem gibt. Dies erlaubt es uns, Geodaten an einer bestimmten Stelle im Raum bzw. auf der Erdoberfläche zu fixieren. [13], S. 212-213

Die Definition eines Bezugssystems umfasst einerseits die Festlegung eines Punktes als Ursprungs-, als Ausgangspunkt für alle Bezugssystemachsen (Koordinatenachsen) und die Ausrichtung (z.B.: Nordrichtung) und Einteilung (z.B.: Meter, Hausnummern) dieser. Innerhalb des Koordinatensystems beschreibt die Position (z.B.: Koordinaten) der einzelnen Objekte die Verhältnisse der Objekte zueinander.

Da sich die Definition eines Bezugssystems auf bekannte, reale Verhältnisse (z.B.: Erdmittelpunkt, Nordrichtung) bezieht ist es möglich, zwischen den einzelnen Positionen in einem Bezugssystem in ein anders überzuführen, um dann so die realen Verhältnisse unterschiedlicher Geoinformationen mit verschiedenen Bezugssystemen in einem gemeinsamen Bezugssystem darstellen zu können (z.B.: Transformation WGS84 → MGI).

Das LBS-System muss geeignete Strukturen zur Speicherung solcher geographischen Informationen zur Verfügung stellen, aber auch Methoden für die Verarbeitung dieser bereitstellen (siehe 3.2.2 Datenstruktur).

2.4.1 Positionshandhabung in OpenLS

Bei der Kommunikation mit dem LBS-System (siehe 2.2 Kommunikation) werden beim Austausch der Datensätze, welche geographischen Informationen repräsentieren, auf den Abstract Data Type (ADT) zurückgegriffen. Unter dem Abstract Data Type ist laut OGC folgendes zu verstehen:

Abstract Data Type (ADT):

The basic information construct used by the GeoMobility Server and associated Core Services. Consists of well-known data types and structures for location information. Defined as application schemas that are encoded in XML for Location Services (XLS) [7].

Die ADTs bauen dabei einerseits auf den Grunddatentypen aber auch auf den abstrakten Daten der OGC GML Spezifikation (z.B.: `gml:pos` für Positionsangaben) auf. Im Überblick werden die Daten im Zuge des OpenLS Information Model wie folgt verwendet (siehe Abbildung 9):

Tabelle 1: Übersicht über die von OpenLS verwendeten abstrakten Datentypen

Position ADT	Punkt Koordinaten in einem bekannten Koordinatensystem.
Address ADT	Angabe eines Objekts in Form einer Adresse.
Point of Interest (POI) ADT	Ein Ort an dem sich etwas von Interesse oder eine Dienstleistung bzw. Produkt befindet.
Area of Interest (AOI) ADT	Ein Bereich der für die Suche als Vorlage dient. Dies kann mit Hilfe eines Polygons, eines Rechtecks oder eines Kreises geschehen.
Location ADT	Beschreibung eines Ortes (Position, Adresse oder POI).
Map ADT	Beschreibt die Karte und die darüber liegenden Objekte (Routen & POI).
Route Summary ADT	Allgemeine Beschreibung (Metadaten) der Route.
Route Geometry ADT	Die Geometrie der Route.
Route Manourvers ADT	Angabe der Manöver welche zur Bewältigung der Route durchgeführt werden müssen.
Route Directions ADT	Schrittweise Instruktionen für die Zielführung einer Route.

Ein Address ADT beschreibt zum Beispiel alle jene Arten von Informationen, von denen eine Adressangabe dem System übergeben werden kann. Dies umfasst die Angabe einer Adresse in einer beliebigen Form (`freeFormAddress`) oder durch Angabe eines Straßennamens, Orts und Postleitzahl.

Das Campus 2011 System baut beim Austausch von Information auf diesen abstrakten Datentypen auf. Um den durch das Campus 2011 System entstehenden Anforderung an den Informationskontext der direkt auf das Verhalten einer Dienstleistung Einfluss hat werde die Parameter des Informationskontext (Zeit, Tag, Benutzer), in anderen Parametern des jeweiligen Dienstes, wie zum Beispiel im Falle der Zeit in dem Parameter `<POIProperty name="event" value="1.1.2011"/>` (siehe 2.3.3 Informationshandhabung in OpenLS) dem Dienst übergeben. Dies ist möglich, da der OpenLS Standard nur den Aufbau der Schnittstellen und die grundsätzliche vorhandenen Services vorgibt. Wie die übergebenen Parameter tatsächlich verwendet bzw. verarbeitet werden liegt in der Hand der Implementierung.

Betrachtet man die OpenLS ADTs in Kombination mit den fünf Core Services so erhält man folgendes Bild in dem der Datenfluss und das Vorkommen der ADTs veranschaulicht wird (siehe Abbildung 9):

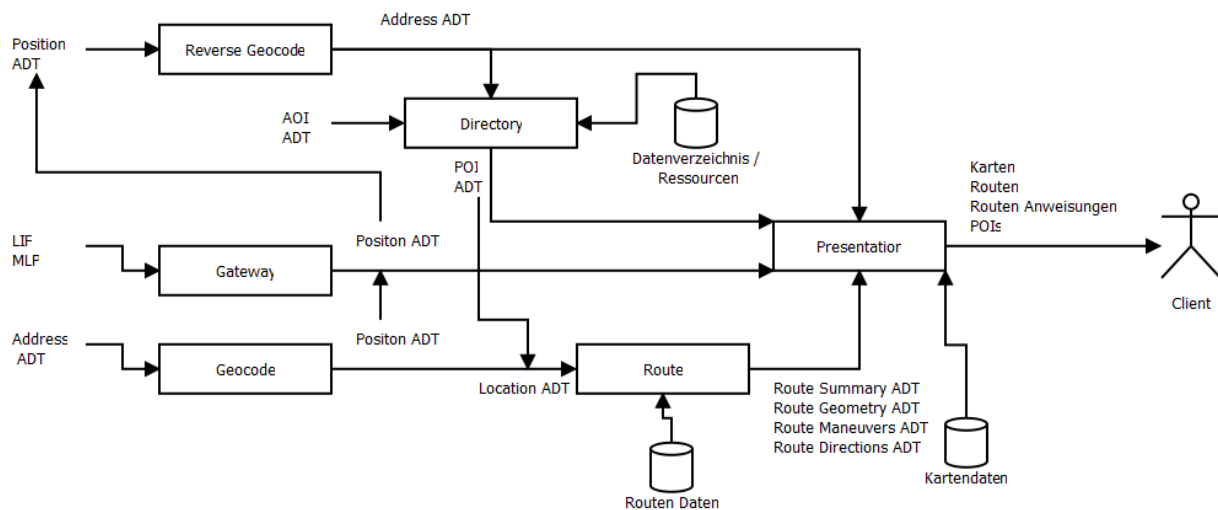


Abbildung 9: OpenLS Information Model
Quelle: in Anlehnung an [7]

Wie in Abbildung 9 veranschaulicht erwarten die verschiedenen Core Services die ADTs als Parameter und liefern unter anderem andere ADTs zurück, welche in weiterer Folge von anderen Core Services oder von der Clientanwendung verarbeitet werden. Neben den an die Dienste übergebenen ADT-Parameter greifen die unterschiedlichen Services auf die für die Verarbeitung des Requests benötigten Daten zu.

Dieser Datenverlauf betrifft grundsätzlich alle jene Daten, welche in dieses OpenLS bzw. ADT Schema passen. Werden nicht ortsbezogene Daten übergeben, so wird auf die weiteren von der jeweiligen Dienstleistung bereitgestellten Parameter zurückgegriffen und so zum Beispiel eine Zeitinformation, parallel zu einem ADT, in einem eigenen Parameter übergeben, die somit nicht Teil des OGC ADT Schemas ist.

3 Systemkomponenten

Wurden im Kapitel vorher die Eigenschaften und Anforderungen, welche sich aus der Definition eines LBS-System ergeben betrachtet, so wird in diesem Kapitel das System aus einer abstrakten Ansicht behandelt. Dabei wird auf den Teilbereich des LBS-Systems nicht eingegangen, sondern das Campussystem wird hinsichtlich seines grundlegenden Aufbaus analysiert, um somit die weiteren, allgemeineren Anforderungen an das System herauszufinden. Dabei wird in erster Linie ein Augenmerk auf das Umfeld des Systems gelegt, um somit die Komponenten zu identifizieren, welche mit dem System kommunizieren und es somit beeinflussen.

So kann man die Akteure als jene Komponente identifizieren, welche jene Elemente (Benutzer + Geräte) umfasst, die mit dem System in Interaktion treten können. So definieren sie auch den Funktionsumfang und die Schnittstellen, welche sie benutzen können, da sie die Endbenutzer bzw. Konsumenten des Systems darstellen.

In einem weiteren Schritt wird der Blick auf den grundlegenden internen Aufbau des Systems gerichtet. So können zwei grundsätzliche Bereiche identifiziert werden, welche in jedem System vorkommen sollten.

Der erste Bereich umfasst die Struktur der bereitzustellenden Daten, welche für die Verarbeitung der Anfragen der Akteure benötigt werden. Es ist notwendig zu wissen, welche grundlegenden Datenarten im System vorkommen und in welcher Struktur diese gespeichert werden.

Der zweite betrifft das Bindeglied zwischen dem interne Datenbereich und dem externen Bereich der Akteure. Dieser Bereich des Systems bereitet die gespeicherten Daten für die Akteure auf und verarbeitet bzw. speichert die von einem Akteur gesendeten Daten im Datenbereich ab. Dieser Bereich (Prozesse) umfasst hauptsächlich die OpenLS Core Services und wurde bereits in Kapitel 2.1.2 LBS-Schnittstellen Definition: OpenLS behandelt und in weiterer Folge vom Application Control Modul. Die Prozesse müssen verwaltet (siehe 4.2.2 Application Control Modul) werden und können teilweise gleichzeitig ablaufen (Voraussetzung: Multitasking) um gleichzeitig mehrere Benutzer mit unterschiedlichen Anfragen zu behandeln.

Um die Verfügbarkeit und Zuverlässigkeit der Prozesse zu gewährleisten sollten die Prozesse auch auf unterschiedlichen physikalischen Maschinen laufen (siehe 4.1 Systemschichten und 3.1.2 Geräte).

3.1 Akteure

Die Akteure sind jene äußeren Systemkomponenten welche mit dem System in Interaktion treten können. Da die realen Personen (Benutzer) nicht direkt mit dem System kommunizieren können benötigen sie eine Schnittstelle (Benutzer-Maschine-Schnittstelle = Gerät), welche ihre Eingaben in digitale Befehle umwandelt.

3.1.1 Benutzer

Betrachtet man die Benutzer unter dem Blickwinkel der Möglichkeiten zur Interaktion mit dem GMS, so lassen sich drei Benutzergruppen bilden, welche jeweils unterschiedliche Rechte haben und aufeinander aufbauen:

- Gruppe der Konsumenten
- Gruppe der Inhaltserzeuger
- Gruppe der Administratoren

Die wichtigste Gruppe stellen die Konsumenten dar, da sie der Grund sind, weshalb das System entwickelt wird. Sie können nur bereits auf vorhandene Informationen und Funktionalität zugreifen.

Um den Konsumenten überhaupt Information anbieten zu können müssen die Daten, auf denen die Informationen basieren, von jemand erzeugt und im System gespeichert werden. Dafür ist die Gruppe der Inhaltserzeuger vorgesehen. Sie sind für die Verwaltung (Erstellen, Ändern, Löschen von Daten) verantwortlich, wobei jede Veränderung an den Daten mitprotokolliert wird.

Die letzte Gruppe stellt die Gruppe der Administratoren dar, welche fundamentale Verwaltungsaufgaben am System übernehmen und sich dafür verantwortlich zeigen. So können sie z.B. Änderungen an der Datenstruktur oder Funktionalität vornehmen. Dadurch sind sie auch für die Wartung und den Betrieb des Systems verantwortlich.

Aufgrund der unterschiedlichen Anforderungen der Benutzer an das System ist es somit notwendig, eine Benutzerverwaltung einzuführen.

Betrachtet man die Benutzer aus einem weiteren Blickwinkel des TUGC 2011, so kann man hinsichtlich des Zugangs noch folgende zwei Unterscheidungen treffen:

- externe Benutzer
- interne Benutzer

Die externen Benutzer sind jene Personen, welche zum ersten Mal mit dem System in Kontakt kommen und auch sonst nicht näher mit der TU-Graz zu tun haben. Sie benötigen die Möglichkeit eines Gastzugangs (Konsumentenzugang ohne Registrierung) oder die Möglichkeit sich als Konsument zu registrieren. In beiden Fällen benötigen sie bei der erstmaligen Benutzung eine Hilfestellung.

Bei den internen Benutzern handelt es sich um jene Personengruppe, welche direkt mit der TU-Graz in Verbindung zu bringen ist (Studenten, Angestellte), und die ein Benutzerkonto (TUG-online) besitzen. Dadurch kann die Authentifizierung mit Hilfe dieses Kontos auch am GMS erfolgen.

Kombiniert man die beiden Betrachtungsweisen, so erhält man in folgender Tabelle (siehe Tabelle 2) ein Bild von der Einteilung eines Benutzers in die jeweilige Gruppe.

Tabelle 2: Benutzereinteilung

Aufgabe im System	<ul style="list-style-type: none"> • Konsument • Inhaltserzeuger • Administrator
Zugang zum System	<ul style="list-style-type: none"> • Interner Benutzer • Externer Benutzer

3.1.2 Geräte

Ein Benutzer kann nicht direkt auf das System zugreifen, sondern benötigt ein Gerät, welches die Schnittstelle zwischen einer realen menschlichen Person und der digitalen Welt darstellt. Da es eine beinahe unüberschaubare Anzahl von Gerätemodellen gibt, können diese aus Sicht eines LBS-Systems in eine zweidimensionale Geräteklassenmatrix eingeteilt werden, wobei die erste Dimension die Art des Standorts eines jeden Geräts definiert, und die andere die Funktionalität bzw. den Zugang zum System beschreibt.

Wird die Dimension des Standorts betrachtet, so kann man zwei Unterscheidungen treffen:

- Standorttreue Geräte
- Mobile Geräte

Standorttreue Geräte sind jene Geräte, welche sich an einem fixen Ort befinden, womit die Positionen dem System bereits bekannt sind oder diese auf Grund von Parametern des Kommunikationsmediums (z.B.: IP-Adresse oder eindeutige Identifikationsnummer) genau bestimmt werden können. Voraussetzung ist es, dass die Beziehung Standort \leftrightarrow Geräteidentifikator dem System bekannt und eindeutig ist.

Mobile Geräte haben keinen fixen Standort und können diesen auch während der Interaktion mit dem System ändern. Je nach Sensorausstattung und aktuellem Aufenthaltsort kann die aktuelle Position mit unterschiedlicher Genauigkeit ermittelt werden.

Aufgrund der Kommunikationsdefinition durch OpenLS (siehe 3.2 OpenLS) existieren noch die Begriffe des Servers und des Clients. Ein Gerät, welches von einem Benutzer verwendet wird befindet sich hierbei immer auf Seite des Clients.

Mit der Berücksichtigung der Client-Server Struktur lässt sich die Geräteklasse um eine weitere Dimension hinsichtlich der Einteilung erweitern.

Die zweite Dimension der Geräteklassenmatrix befasst sich mit der Aufgabenverteilung, d.h. welche Aufgaben vom Client erledigt werden können, und welche vom Server ausgeführt werden. Die Einteilung geschieht analog zu [14], wobei sie um die Definition des Ultra Thin Client erweitert wurde:

- Fat Clients
- Thin Clients
- Ultra Thin Clients

Bei der Einteilung der Geräte in die drei Klassen unterscheidet man zwischen der Benutzer-eingabe (Eingabe), bzw. der Ausgabe des Geräts in einer vom Benutzer erfassbaren Form (Ausgabe; z. B. Rendern einer Webseite am Bildschirm). Des Weiteren gibt es noch die Komponente der Datenverwaltung, welche die im System vorhandenen Daten vorrätig hält und mit einer Lagerverwaltung verglichen werden kann. Zuletzt gibt es noch den Bereich der Datenverwaltung welcher die Daten anfordert und damit Berechnungen durchführt (siehe Abbildung 10, Abbildung 11, Abbildung 12).

Ein **Fat Client** kümmert sich neben der Ein- und Ausgabe der verarbeiteten Daten auch um die eigentliche Datenverarbeitung. Dafür kommuniziert er direkt mit dem GMS. Dies trifft im Falle von eigenständigen Applikation (oder auch Apps) zu (siehe Abbildung 10). In diese Geräteklasse fallen alle jene Geräte welche eine Installation einer eigenständigen Anwendung ermöglichen und der Anwendung auch einen Zugriff auf Bereiche bzw. Sensoren des Gerätes ermöglichen. Diese Geräteklasse erstreckt sich von einem aktuellen Smartphone bis hin zu einem an einem fixen Standort befindlichen Computer.

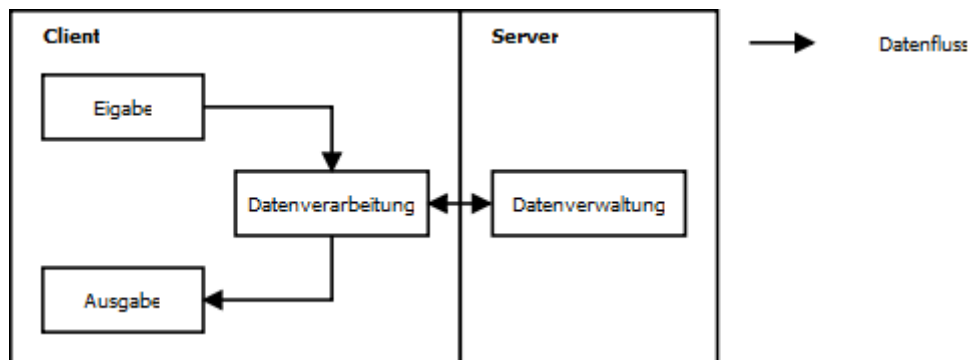


Abbildung 10: Struktur eines Fat Clients

Der **Thin Client** bietet auch eine Ein- und Ausgabe, lagert jedoch die Datenverarbeitung auf den Server aus. So ist ein Webbrowser ein Thin Client. Er kümmert sich um die eingegebenen Daten und sendet diese an den Server. Dort stellt zum Beispiel eine PHP-Seite eine Verbindung zur Datenbank her und erstellt aus den resultierenden Daten eine HTML Code, welcher dann vom Browser angezeigt wird (siehe Abbildung 11). Da der Client grundsätzlich seine Position nicht selbstständig feststellen kann, muss bei der Ermittlung der Position auf Eigenschaften des Kommunikationswegs zurückgegriffen werden (z.B.: IP-Adresse, verwendete Basisstation).

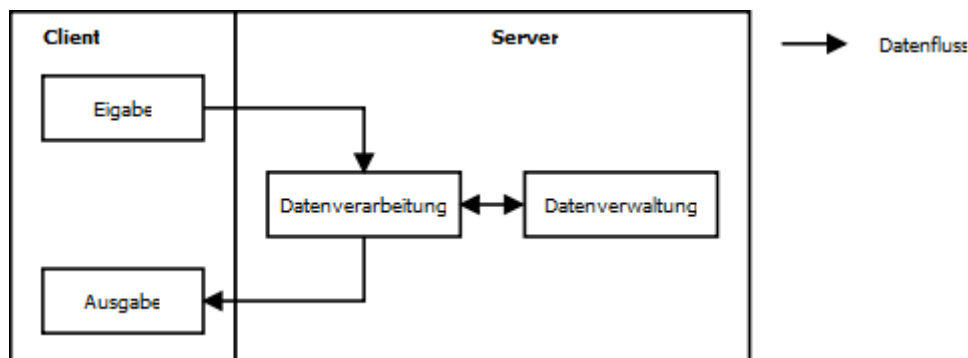


Abbildung 11: Struktur eines Thin Clients

Ein **Ultra Thin Client** bietet nur eine Ausgabemöglichkeit (z.B.: Info Screen). Er empfängt die darzustellenden Daten vom Server. Dies muss auf Grund der fehlende Eingabemöglichkeit automatisiert entweder vom Client aus oder vom Server angestoßen werden (siehe Abbildung 12). Da diese Geräte keine direkte Interaktion mit dem Benutzer ermöglichen, werden meist nur allgemeinere Informationen bereitgestellt. Es ist jedoch möglich durch das Benutzen einer anderen Güteklasse dem Server mitzuteilen, die Information nicht direkt am Benutzergerät darzustellen sondern auf einen Ultra Thin Client umzuleiten, und so das eine Gerät als reines Eingabegerät zu benutzen und den Ultra Thin Client als reines Ausgabegerät.

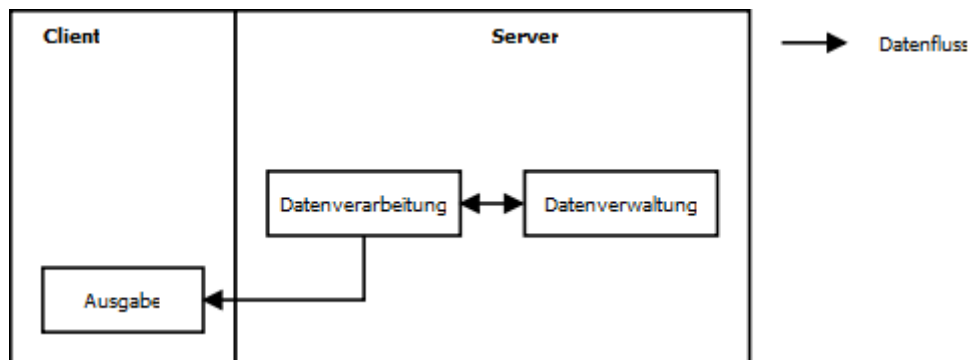


Abbildung 12: Struktur eines Ultra Thin Client

In der nächsten Tabelle werden einige Beispiele der Geräte, welche das Campus 2011 System benutzen könnten, vorgestellt. Des Weiteren wird ihre Position in der Gerätematrix gezeigt:

Tabelle 3: Beispiele für die Einteilung unterschiedlicher Geräte

Standorttreu	Desktop-Anwendung  Quelle: www.office.com	TU-Terminal Browser  Quelle: eigene	Hörsaalankunft  Quelle: eigene
Mobil	Smartphone-App  Quelle: eigene	Handy mit Browser  Quelle: eigene	-
	Fat Client	Thin Client	Ultra Thin Client

In Tabelle 3 dargestellt können die unterschiedlichen Geräteklassen einen unterschiedlichen Bezug zu ihrer aktuellen Position haben, sodass die Feststellung der aktuellen Position auf sehr unterschiedliche Weise und mit unterschiedlicher Genauigkeit erfolgen kann. So muss bei der Implementierung des Campus 2011 Systems auf diese unterschiedlichen Anforderungen geachtet und gegebenenfalls eine geeignete Infrastruktur geschaffen werden, um die unterschiedlichen Geräte zu unterstützen.

3.2 Daten

3.2.1 Datenarten

Grundsätzlich basieren alle verwendeten Daten auf einigen wenigen Datentypen (Zahlen, Buchstaben). Darauf aufbauend entstehen komplexe Datenstrukturen, welche in einem geeigneten DBMS gespeichert und verwaltet werden.

Bei den Geodaten kann man zwei Datenarten unterscheiden. Einerseits Vektordaten, welche auf der Angabe von einer oder mehrerer Koordinaten beruhen und andererseits Rasterdaten, welche das Abbild geographischer Information (z.B. Luftbild) darstellen.

Für die Handhabung dieser unterschiedlichen Geodaten reicht ein rationales DBMS nicht mehr aus. In diesem Fall muss man auf einen Aufsatz, auf ein DBMS zurückgreifen, welches die Verwaltung dieser Datenstrukturen ermöglicht, indem es Datentypen (Datenarten) zur Speicherung bereithält, aber auch Methoden bietet, mit diesen umzugehen.

3.2.2 Datenstruktur

Dem Grundgedanken aus Kapitel 2.2 Kommunikation folgend, werden zu jedem Datensatz ein Verweis und der Typ des Verweises abgespeichert um es so zu ermöglichen, dass der Datensatz nicht unbedingt im eigentlichen System vorhanden sein muss. Die Angabe beschreibt die Art der Quelle auf die die Verweise zeigen, so z.B. DB-Tabelle oder Weblink. In weiterer Folge wird diese Verlinkung als Ressource bezeichnet.

Um möglichst unabhängig von der eigentlichen Struktur eines Datensatzes zu sein, werden im Kernbereich der Datenstruktur keine Informationsdaten gespeichert. Es gibt nur eine Tabelle, welche einerseits einen Verweis auf den tatsächlichen Datensatz und andererseits einen Verweis auf die Quelle des Datensatzes beinhaltet. Aufgrund der Quelleninformation (z.B.: Textdatei, SQL-Server) wird die Art des Zugriffs geregelt (z.B. direkter Zugriff bei Textdatei oder SQL-Abfrage bei DBMS-Server). Somit weiß das System wie der Zugriff zu erfolgen hat, und mit Hilfe des Verweises findet es auch den aktuellen Speicherort des gesuchten Datensatzes (siehe Abbildung 13).

Aufgrund dieser Abstraktionsebene erfolgt der Zugriff auf die Informationsdatensätze immer nach dem gleichen Schema und es wird keine Unterscheidung zwischen systeminternen Quellen oder externen Datenquellen gemacht. Die informationstragenden Datensätze werden systemintern mittels Ressourcentabelle repräsentiert.

Diese Trennung in Verzeichnis und Ressource ermöglicht es im Campus 2011 System, dass zum Beispiel jedes Institut jede Informationen im Campus 2011 System veröffentlicht, jedoch noch weiterhin volle Gewalt über seine Daten hat, da bei einer Abfrage die Daten immer online abgerufen und nicht im System gespeichert werden.

Der Nachteil hierbei ist, dass bei einem Ausfall des Verzeichnisservers keine Verknüpfung möglich ist. Dies kann durch Redundanz oder Speichern im direkten Bereich des Systems gelöst werden. Eine Speicherung im Bereich des Systems würde jedoch eine teilweise Entziehung des direkten Zugriffs des Inhaltserstellers auf die im Systembereich gespeicherten Daten zur Folge haben.

Auch müssten die Daten zwischen dem Campus 2011 System und dem Inhaltsersteller bei Änderungen wieder aktualisiert werden. Dies ist bei dem gewählten Konzept mit dem direkten Zugriff nicht nötig, da immer auf die aktuellsten Daten zugegriffen wird. Ein weiterer Vorteil ist, dass bei Änderungen an der Datenstruktur auf Seiten des Inhaltsbereitstellers nur der Zugriff auf diese geändert werden muss und es bei der Darstellung und Datenverarbeitung zu keinen Änderungen kommt.

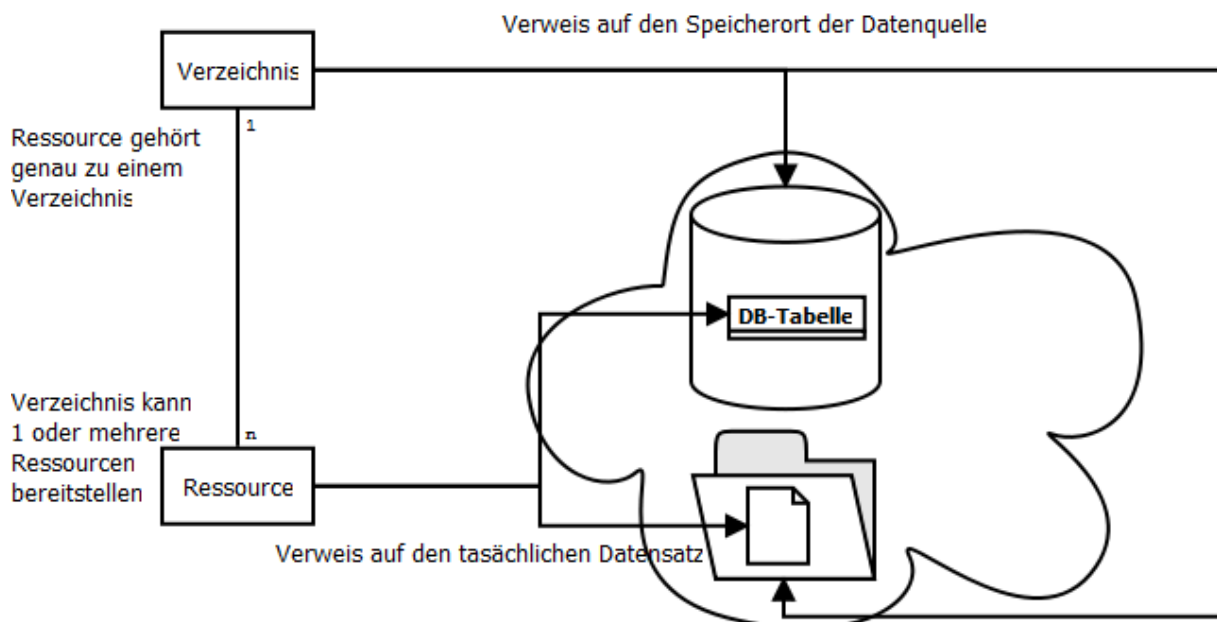


Abbildung 13: Beziehung zwischen Ressource und dem tatsächlichen Datensatz

Damit die Datenverarbeitung die Ressource korrekt handhabt muss für jede einzelne Ressource noch der dazugehörige Informationskontext modelliert werden, da die Ressource sich im Zuge eines LBS-Systems meist auf ein Objekt (z.B.: Gebäude, Raum, Informationspunkt) bezieht. Diese Objektabelle verwaltet alle im System existierenden geographischen Objekte.

Eine weitere Komponente des Informationskontexts stellt ein Ereignis dar, welches die Zeit abbildet. Es wird jedoch keine Unterscheidung zwischen einem Zeitpunkt und einer Zeitspanne getroffen (siehe 2.3.2 Informationskontext). Dabei hat jedes Ereignis einen Anfang und ein Ende, und so kann man jedem Ereignis noch einen Ereignistyp (z.B.: einmalig, periodisch) zuweisen. Im Falle eines Zeitpunkts bestehen Anfang und Ende aus denselben Daten.

Jede Ressource befasst sich auch mit einem speziellen Themengebiet (z.B.: Hörsaal, Gebäude, Geschichte) welches in einer Tag- (Thema-)Tabelle abgebildet wird (siehe Abbildung 15).

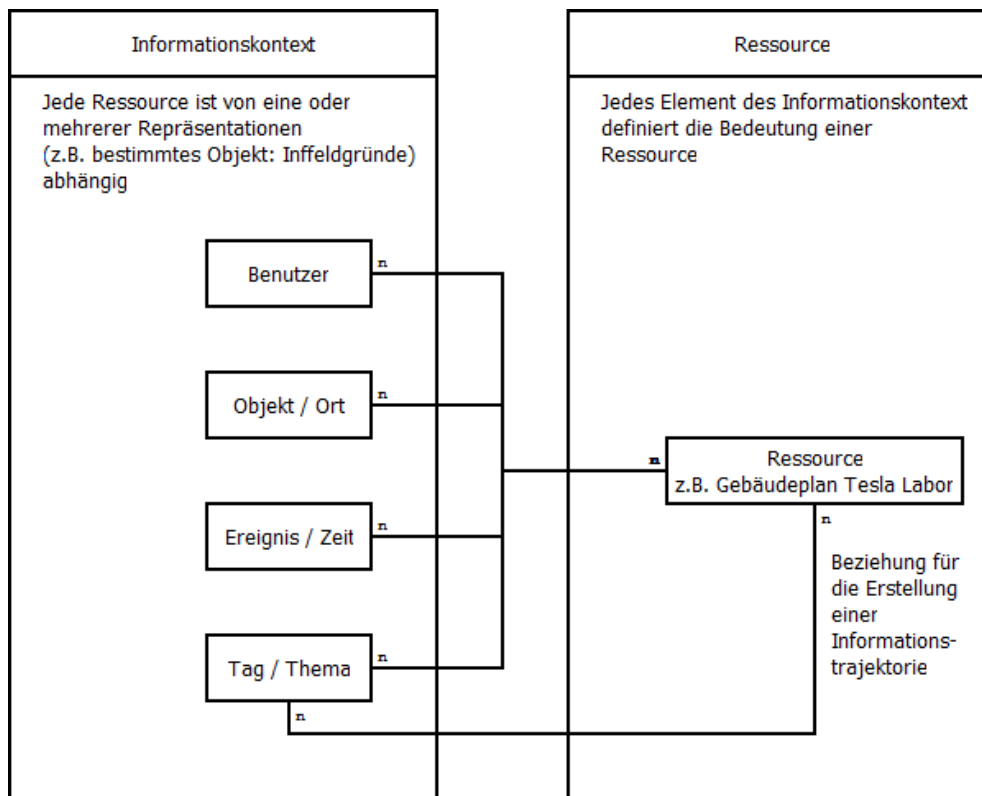


Abbildung 15: Beziehungen des Informationskontexts zu den Ressourcen

Im Rahmen der Campus 2011 LBS-Komponente ist der Bereich des Objekts bzw. Ortes die häufigste gebrauchte Komponente des Informationskontexts. Jede einzelne Komponente des Informationskontexts besitzt eine direkte Verbindung zur Ressource. Durch diesen Ansatz wird für das Auffinden von einer oder mehreren Ressourcen nur ein Bereich des Informationskontexts benötigt. Werden mehrere Informationen zum verwendeten Kontext angegeben, können die Ressourcen genauer eingeschränkt werden.

Da auch die Funktionalität vom Kontext (Benutzer, Objekt, Zeit und Thema) abhängig ist (z.B.: wird die Routenberechnung nur am Campus selbst angeboten), muss sich die Tabelle für die Verwaltung der Funktionalität in ähnlicher Weise wie die Ressourcentabelle mit den einzelnen Kontextelementen verbinden (siehe Abbildung 16).

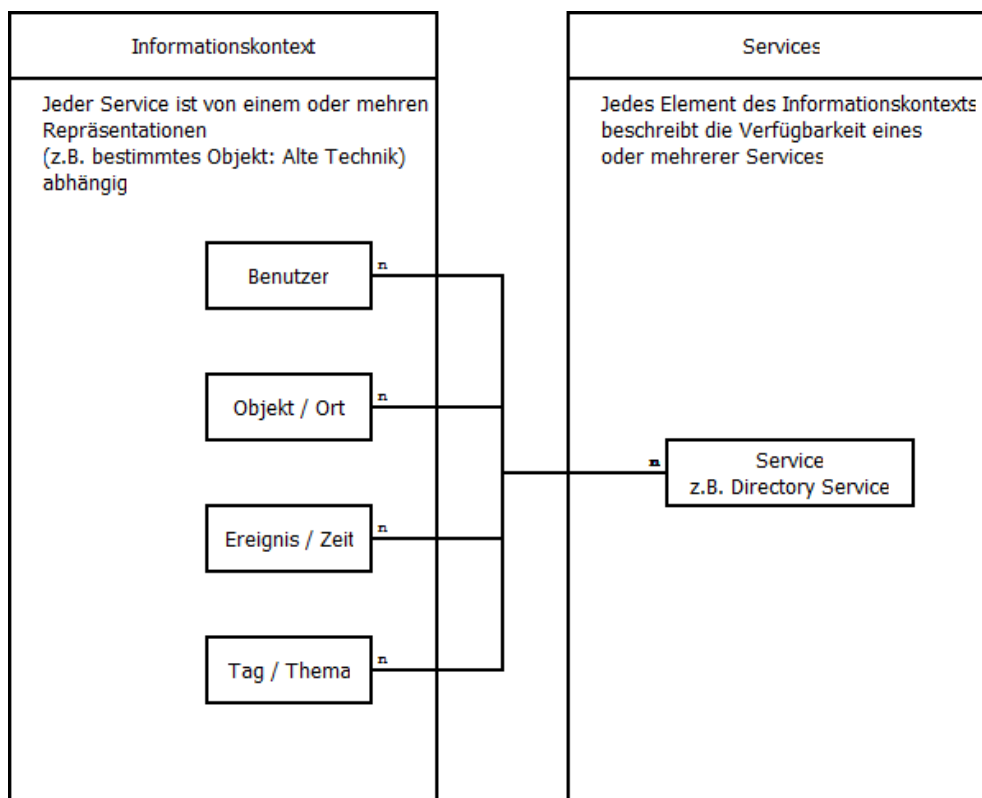


Abbildung 16: Beziehung des Informationskontexts zu den Services

Im Gegensatz zur Beziehung des Informationskontexts zu einer Ressource gibt es bei den Diensten keine welche aufeinander aufbauen, sondern jeder Dienst wird einzeln vom Client aufgerufen und greift immer auf die gleichen internen Funktionen zu. Dadurch werden die Dienste voneinander unabhängig und immer vom Client aufgerufen und die Resultate immer diesem übermittelt. Somit ist keine direkte Kommunikation zwischen zwei Diensten vorgesehen.

Der Informationskontext hilft einerseits dem System festzustellen, ob ein angeforderter Dienst dem Clientgerät zur Verfügung gestellt werden darf (z.B.: Ein bestimmter Benutzer darf nur einen bestimmten Service benutzen). Andererseits dienen diese Verknüpfungen auch dazu, direkt die Funktion zu steuern und so vom Informationskontext abhängige Informationen zu erstellen. So werden zum Beispiel abhängig vom aktuellen Benutzer die Routenanweisungen in Deutsch oder Englisch erstellt.

Eine weitere Hauptaufgabe eines LBS-Systems ist die Verwaltung und Handhabung von Geoinformation. Hierfür ist die abstrakte Objekttable (siehe Tabelle 1) die wichtigste Komponente weil sie einerseits als Verbindung zu den Informationsdaten dient und andererseits auch die Verhältnisse der einzelnen geometrischen Objekte untereinander regelt. So besteht ein Gebäude einerseits aus einem Gebäudekorpus und auch andererseits aus Stockwerken, Räumen und Türen.

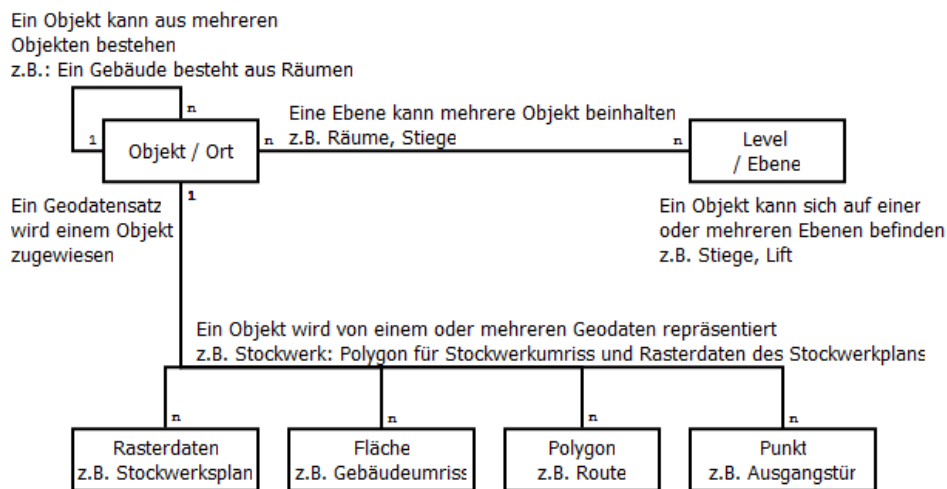


Abbildung 17: Beziehung von Geographischen Daten zur Objekt-Tabelle

Diese Zuordnung ist unabhängig vom eigentlichen Geoinformationsdatentyp und ermöglicht ggf. auch die Verbindung zwischen Raster und Vektordaten. Bei der Speicherung von geographischen Objekten, welche sich direkt auf der Erdoberfläche befinden, reicht es meist, diese in nur zwei Dimensionen abzuspeichern, da sie von keinen anderen Objekten überdeckt werden. Bei der Speicherung von Innenräumen von Gebäuden geht dies nicht mehr, da sich die einzelnen Stockwerke zwangsläufig überdecken.

So kann man einerseits die Koordinaten der Objektpunkte mit Hilfe dreidimensionaler Koordinaten (x, y, z) abspeichern, andererseits kann man die Objektkoordinaten zweidimensional belassen und in der Datenbank dem Geometriedatensatz eine Höheninformation hinzufügen (Tabelle: Level). Das hat den Vorteil, dass bei einer Abfrage nicht alle Koordinaten jeder Geometrie abgefragt werden müssen, sondern dies nur noch einmal je Objekt geschieht und sich auch beschleunigen lässt, indem man einen Index auf die Höheninformationstabelle bildet.

Das Zuweisen einer Höheninformation (z.B. Level) ermöglicht es auch bei tatsächlichen dreidimensionalen Objekten (z.B.: Schrank), dass sich die Höheninformation nicht mehr auf das globale Referenzsystem bezieht sondern auf ein bestimmtes Stockwerk. So kann man das Objekt ohne Probleme auf ein anderes Stockwerk versetzen ohne die Objektgeometrie zu bearbeiten.

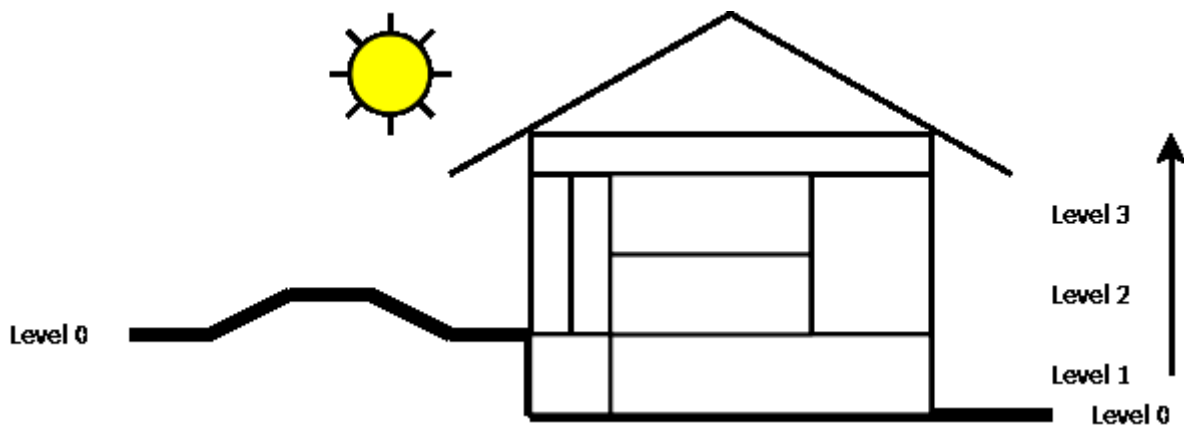


Abbildung 18: Level Behandlung der Höheninformation

Wie Abbildung 18 zeigt wird die Erdoberfläche als Level 0 behandelt. Alle Levels in künstlichen Bauten (Gebäude) werden aufsteigend behandelt, wobei das unterste Stockwerk jeweils dem Level 1 entspricht, egal ob es ebenerdig oder unterhalb der Erdoberfläche liegt. Die Verbindungsobjekte (Türen, Lift, Stiegen) werden mit der Levelinformation für diejenigen Levels, die sie verbinden, gespeichert.

Eine weitere Aufgabe eines LBS-Systems ist die Berechnung von Routen und die Verwaltung dieser in Touren (i.e. eine bereits berechnete Route). Je nach Routenberechnungsalgorithmus können unterschiedliche Strukturen zum Tragen kommen (siehe [15]). Um von dem tatsächlich eingesetzten Algorithmus unabhängig zu sein wurde für die Verwaltung des notwendigen Netzes folgende Struktur gewählt, welche sich in die Objekt-Geometrie Struktur eingliedert, da ein Graph für die Routenberechnung aus Knoten (vertex) und Kanten (edge) besteht (siehe Abbildung 19):

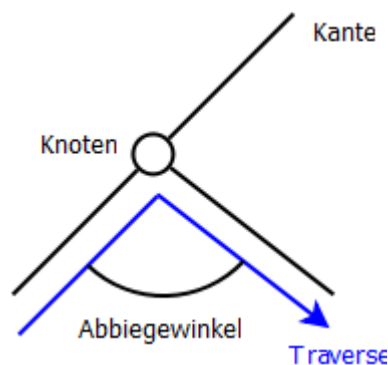


Abbildung 19: Grundsätzliche Elemente eines Graphen

Um noch zusätzliche Richtungen in den Graphen zu modellieren werden vorhandene Traversen (z.B.: sind Feuertüren nur in eine Richtung begehbar), als zusätzliche Informationstabelle in die Graphenstruktur gespeichert (siehe Abbildung 20):

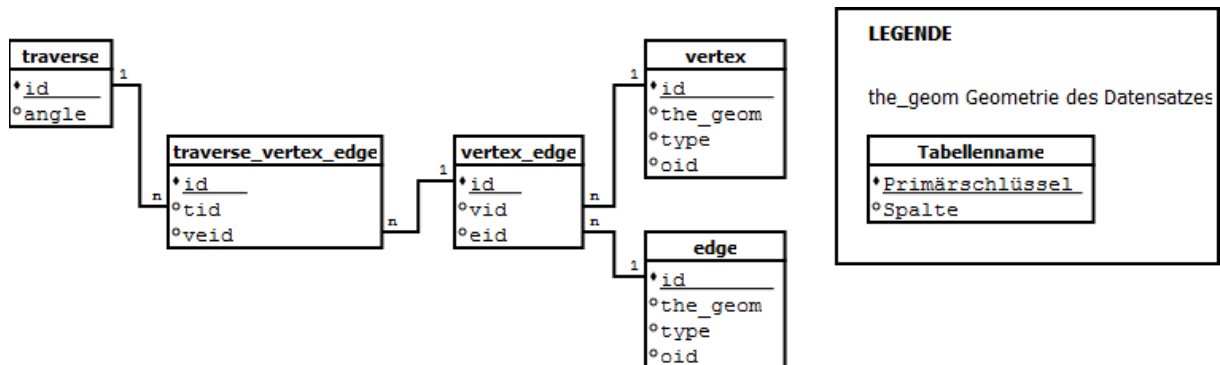


Abbildung 20: Struktur der Speicherung der Graphendaten

Die Winkelangabe (`traverse::angle`) beschreibt den Winkel zwischen den beiden Vektoren aus denen eine Traverse besteht (siehe Abbildung 19). Sie wird bei der Routenberechnung für das Erkennen von Abbiegungen benötigt (d.h. ab welchem Winkel spricht man z.B. von Rechtsabbiegen).

Vorausberechnete Routen, welche in einem bestimmten Informationskontext zur Verfügung stehen (z.B. Entstehung der TU-Graz) werden als Touren in folgender Struktur abgespeichert (siehe Abbildung 21):

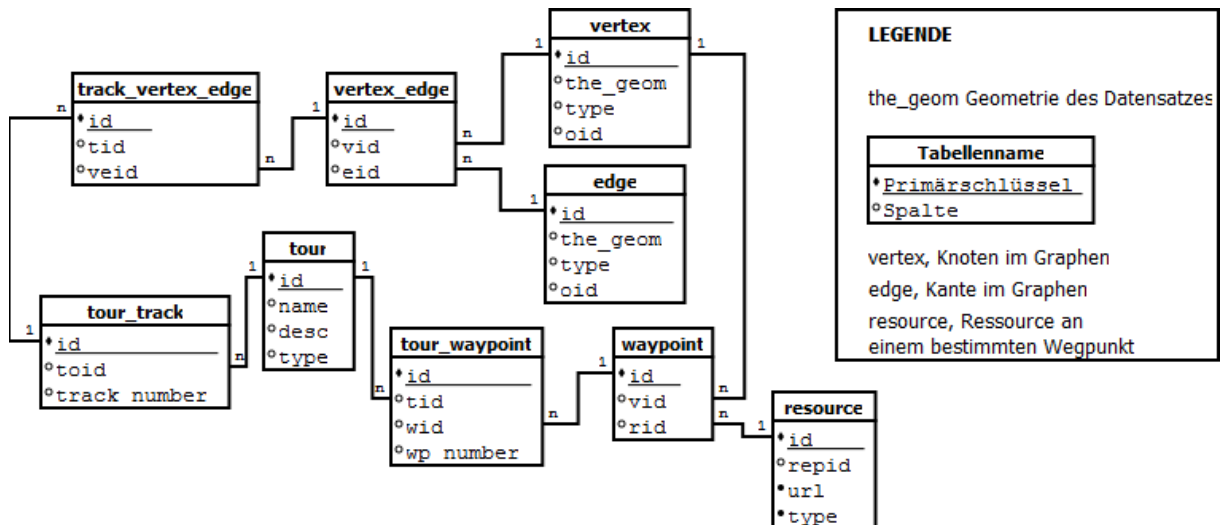


Abbildung 21: Datenstruktur zur Speicherung von Routeninformation

Jede Tour besteht aus einer Reihe von Wegpunkten (`waypoints`), welche mit einer Resource in Beziehung stehen können (z.B.: Wohnhaus von Tesla). Zwischen den einzelnen Wegpunkten muss ein Benutzer dem Pfad (`track`) folgen. Jeder dieser Pfade besteht aus einer Reihe von Knoten-Kanten (`vertex_edge`) Abfolgen. Dadurch erhält man die resultierende Tourgeometrie. Aufgrund der Knoten-Kanten Beziehungen erhält man auch die Anweisungen für den Benutzer (z.B.: „Gehen Sie gerade aus“ oder „Biegen Sie links ab“).

4 Systemarchitektur

Die im vorangegangenen Kapitel definierten Anforderungen an das System, als auch der grundlegende Aufbau des Systems werden nun in den Aufbau einer möglichen Systemarchitektur übergeführt.

In einem ersten Schritt wird die grundlegende Systemstruktur noch weiter verfeinert und die Anforderungen an das System angepasst. Aufgrund der unterschiedlichen heterogenen Akteursstruktur gibt es eine Vielzahl unterschiedlicher grundlegender Systemarchitekturen welche alle zur Anwendung kommen können.

Aus diesen grundlegenden Architekturen wird eine für das System anwendbare Systemarchitektur abgeleitet. Diese wird in einem weiteren Schritt noch weiter verfeinert um dann so noch differenziertere Strukturen zu erhalten, welche bei einer tatsächlichen Einführung des Systems in Form von Systemmodulen implementiert werden können.

Dies umfasst die Definition der Struktur des Systemmoduls als auch dessen Submodulen und die Beschreibung der Aufgabenbereiche, welche diese übernehmen. Somit sollte es möglich sein diese Strukturen für die tatsächliche Erstellung des Campus 2011 Systems zu benutzen.

Die Gliederung des Systems in die unterschiedlichen Module soll ermöglichen, das System um weitere Komponenten erweitern zu können um dann so auch für zukünftige Aufgabenbereiche gerüstet zu sein.

Bei der Kommunikation der einzelnen Module untereinander wird, wo es möglich ist, auf bereits definierte Standards zurückgegriffen bzw. darauf aufgebaut. Ein weiterer positiver Aspekt des modularen Aufbaus ist die Skalierbarkeit des Systems. Die verschiedenen Aufgaben werden auf mehrere, gleichartige Module aufgeteilt, welche auch auf unterschiedlichen physikalischen Rechnern laufen können.

4.1 Systemschichten

Betrachtet man die Eigenschaften eines LBS-Systems, so weist es viele Eigenschaften eines verteilten Systems auf:

Ein **verteiltes System** (distributed system) besteht aus unabhängigen, über ein Rechnernetz kommunizierenden Rechnern, wobei keine zentrale Systemsteuerung existiert und der Verteilungsaspekt für die Benutzer des Systems möglich transparent (→ 15.1.3) ist. [14]

Auch entspricht das Kommunikationsprinzip (vgl. 2.2.2 OpenLS) eines verteilten Systems dem eines LBS-Systems (siehe [14], [16]).

Die Komponenten eines verteilten Systems kommunizieren in der Regel nach dem Client-Server Prinzip:

- Ein **Client** (Kunde, Auftragnehmer) ist eine Komponente, die von einem Server eine bestimmte Dienstleistung anfordert (Request) und auf eine Antwort wartet.
- Ein **Server** (Dienstbringer) ist eine Komponente, die Aufträge von Clients entgegennimmt, diese Aufträge bearbeitet und eine Antwort (Response) an den Client zurücksendet. [14]

Aufgrund der Client-Server Kommunikation kommt es zu einer mehrschichtigen Architektur.

Im einfachsten Fall besteht das System nur aus zwei Komponenten (siehe Abbildung 22), der Clientanwendung und dem Server, wobei der Server sich neben der Datenverarbeitung, auch um die Verwaltung der Daten kümmert (z.B.: Datenbankanwendung fragt direkt einen DB-Server ab. Ein typisches Beispiel hierfür ist ein Webbrowser der ein PHP Skript aufruft.).

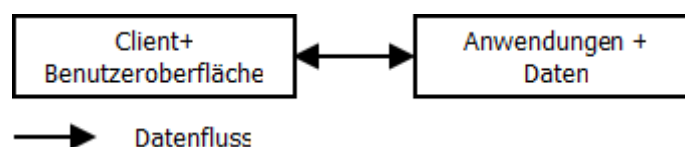


Abbildung 22: Aufbau einer zweischichten Systemarchitektur

Trennt man nun die eigentliche Datenverarbeitung (Anwendungen) und Datenverwaltung voneinander so erhält man eine dreischichtige Systemarchitektur (siehe Abbildung 23). Je nachdem ob die Anwendungsschicht auf dem Client oder einem eigenen Server läuft, spricht man von einem Fat- oder Thin-Client (siehe 3.1.2 Geräte)

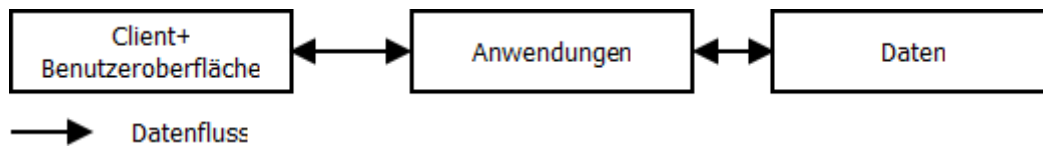


Abbildung 23: Dreischichtige Systemarchitektur

Da auch der Zugriff durch einen Browser möglich sein soll (siehe 3.1.2 Geräte), ist es notwendig, die eingeschränkten Möglichkeiten eines Browsers (z.B.: ohne JavaScript nur GET/POST) zu unterstützen. Um die Kommunikation mit dem GMS zu ermöglichen ist die Einführung einer weiteren Schicht (siehe Abbildung 24) notwendig, welche sich um die Umsetzung der vom Browser gesendete Daten in die Kommunikationsprotokolle des GMS (z.B.: OpenLS Request) kümmert. Hierfür eignen sich zum Beispiel Skriptsprachen welche auf der Common Gateway Interface-Schnittstelle (CGI-Schnittstelle) aufbauen (z.B.: PHP).

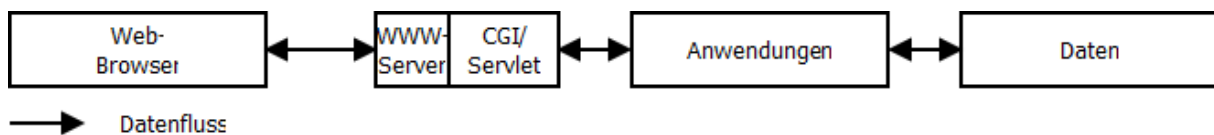


Abbildung 24: Vierschichtige Systemarchitektur

Zusammenfassend entsteht durch diese Anforderungen an das System folgende vierschichtige Architektur (siehe Abbildung 25):

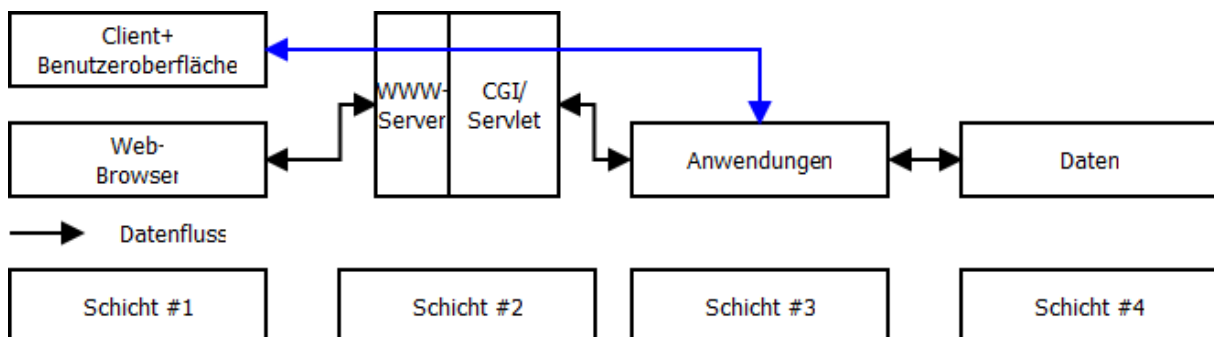


Abbildung 25: Resultierende Systemschichten (Kommunikation mit Clients welche das OpenLS-Protokoll verwenden werden direkt an den GMS (Anwendungen) weitergereicht (blauer Pfeil))

Durch die Trennung des Systems in diese vier Schichten wird es notwendig, diese komplexe Struktur vor dem Client zu verbergen. So kontaktiert jeder Client, gleich welcher Art, den WWW-Server. Im Falle eines Clients mit Benutzeroberfläche (z.B.: Smartphone App, Anwendung für einen Desktop) wird hierbei die Anfrage an die Anwendung durchgereicht und für den Client ist es so, als würde er direkt mit der Anwendungsschicht kommunizieren (Abbildung 26, blauer Pfeil).

Ein einfacherer Browser Client kontaktiert auch den WWW-Server, welcher aber in weiterer Folge die Anfragen mit Hilfe eines Skripts an die Anwendungsschicht stellt. Für die einzelnen Anwendungen (Core Services) ist es so egal, welcher Client die Anfrage stellt und sie muss

sich nicht um die Aufbereitung der berechneten Daten kümmern. So reicht die Entwicklung einer Anwendungsfunktion für alle möglichen Clients.

Man spricht in diesem Fall von Verteilungstransparenz:

Da die Nutzung verteilter Systeme in der Regel komplizierter ist als die Nutzung eines einzelnen Rechners, wurden auf unterschiedlichen Ebenen (Programmiererebene, Benutzerebene usw.) Mechanismen eingeführt, die den Aspekten der Verteilung verbergen; ein verteiltes System erscheint wie ein zentrales System. Man spricht von Verteilungstransparenz. [14]

Die Verteilungstransparenz lässt sich noch in folgende Arten untergliedern, welche Sicherheits-, Zuverlässigkeits-, Verfügbarkeits- und Fehlertoleranzaspekte bedienen [14]:

- **Zugriffstransparenz:** Der Zugriff auf lokale und ferne Ressourcen erfolgt in derselben Weise.
- **Ortstransparenz:** Der Zugriff auf Ressourcen ist ohne Kenntnis des Ortes, an dem sich die Ressourcen befinden, durchführbar.
- **Namenstransparenz:** Der Name einer Ressource ist für alle Rechner des verteilten Systems gleich.
- **Skalierungstransparenz:** Die Erweiterung des Systems um zusätzliche Rechner ist möglich, ohne dass die Gesamtstruktur oder die Anwendungsprogramme geändert werden müssen.
- **Replikationstransparenz:** Falls das verteilte System zur Erhöhung der Verfügbarkeit und Fehlertoleranz mehrere Kopien von einer Ressource hält, so bleibt dies den Prozessen bzw. Benutzern verborgen.
- **Nebenläufigkeitstransparenz:** Mehrere Prozesse bzw. Benutzer können nebenläufig auf dieselben Ressourcen zugreifen, ohne dass es dabei zu Problemen kommt.
- **Ausführungstransparenz:** Es ist für die Prozesse bzw. Benutzer nicht erkennbar, auf welchem Rechner ein angeforderter Dienst ausgeführt wird. Es ist sogar möglich, dass ein Prozess während der Ausführung eines Dienstes auf einen anderen Rechner bewegt wird. (Migration).

Um die Anforderungen an die Verteilungstransparenz zu erfüllen und unter Berücksichtigung der Schichtenarchitektur führt dies zu folgenden Systemmodulen:

4.2 Systemmodule

Aus den vorangegangenen Definitionen und Aspekten lassen sich folgende drei Systemmodule aus den Schichten der Serverseite ableiten:

- Client Control Module (CCM)
- Application Control Module (ACM)
- Resource Control Module (RCM)

Jedes dieser Module baut sich aus einer Reihe weiterer Submodule auf und ist so eine konsequente Weiterführung der Schichtenarchitektur mit Übertragung der Verteilungstransparenz auf die Systemmodule. So kann das System ohne weiteres um zusätzliche Anwendungen erweitert werden (Skalierbarkeit).

Im Überblick sieht die Systemstruktur wie folgt aus (siehe Abbildung 26):

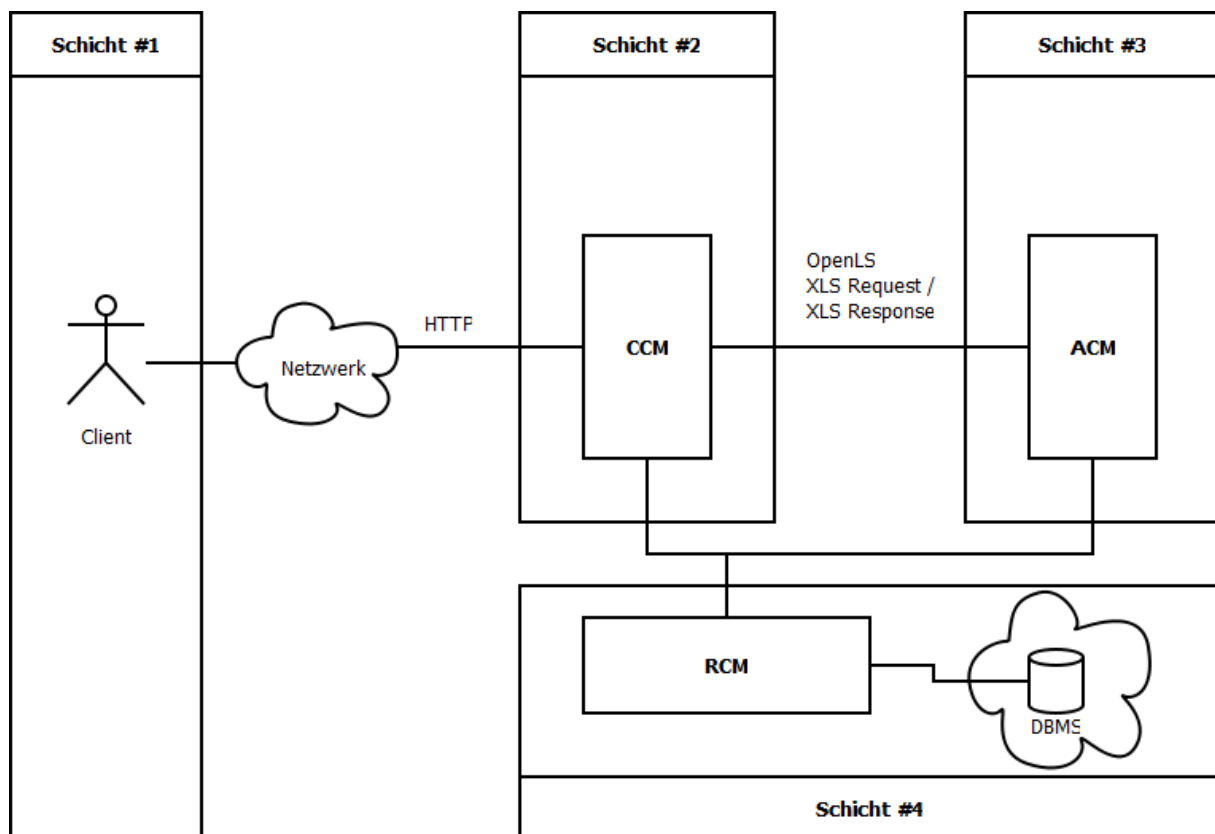


Abbildung 26: Systemmodule und ihre Zuordnung zur jeweiligen Schicht

Die Systemmodule können hierbei auf einem einzelnen physikalischen Server laufen, als auch auf mehrere Server verteilt werden. Die Kommunikation der einzelnen Systemmodule sollte hierbei unabhängig vom Clientnetzwerk und wenn nicht anders möglich verschlüsselt erfolgen.

4.2.1 Client Control Module

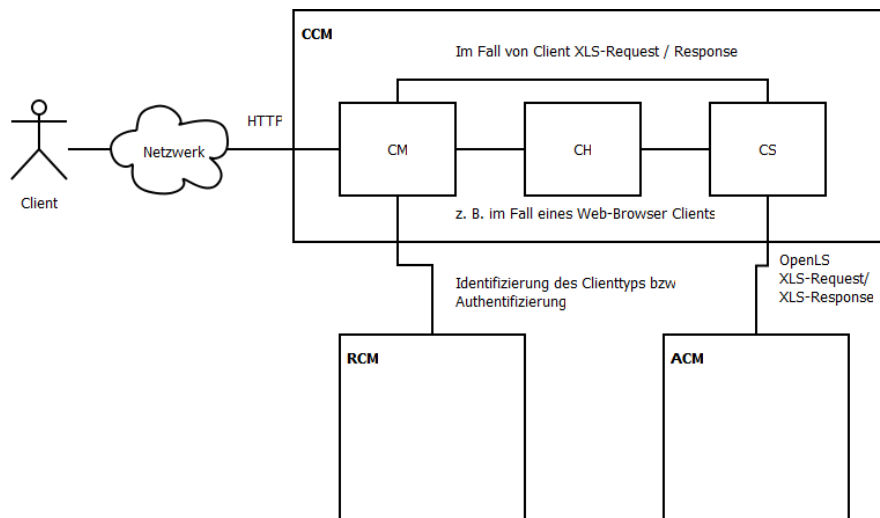


Abbildung 27: Aufbau des CCM

Das Client Control Module (CCM) und seine Submodule (siehe Abbildung 27) repräsentieren die Schnittstelle des Kernsystems zum Client und bilden somit die aktive Schnittstelle nach außen, da sie aktive Befehle zur Systemsteuerung (Requests) annehmen und weiterreichen.

Um mit dem System in Interaktion treten zu können, wird über eine HTTP-Verbindung (siehe 2.2.2 OpenLS) eine Verbindung zum GMS aufgebaut. Das Connection Module (CM) nimmt diese Verbindung entgegen. Es kümmert sich um die grundsätzliche, direkte Kommunikation mit dem Client. Des Weiteren greift das CM auf die im RCM angesiedelten Identifizierungs- und Authentifizierungsfunktionen zu um den verbundenen Client zu erkennen.

So regelt der CM das Empfangen und Senden der Request- und Response-Nachrichten, als auch das Übersetzen der verschiedenen Clientnachrichtenprotokolle (z.B.: im Falle eines Web-Browsers) in OpenLS-Protokolle für das Application Control Module, welches über die OpenLS Kommunikationsschnittstelle angesprochen wird. Für einen Browser wird so das benötigte Formular in Form einer HTML-Seite generiert und an den Client gesendet, auch ist es möglich, die Daten für den Client zu optimieren (z.B.: Komprimieren der zu sendenden Bilddaten). Diesen Aufgabenbereich der Anpassung und Umcodieren übernimmt das Client Handle (CH).

Anschließend kümmert sich der Client Scheduler (CS) um die Verteilung und Verwaltung der einzelnen Anforderung an das Applikation Control Module (ACM). Alle für eine spätere Verarbeitung notwendigen Kommunikationsparameter (z.B.: IP-Adresse) werden an das jeweilige Application Control Module übergeben.

4.2.2 Application Control Module

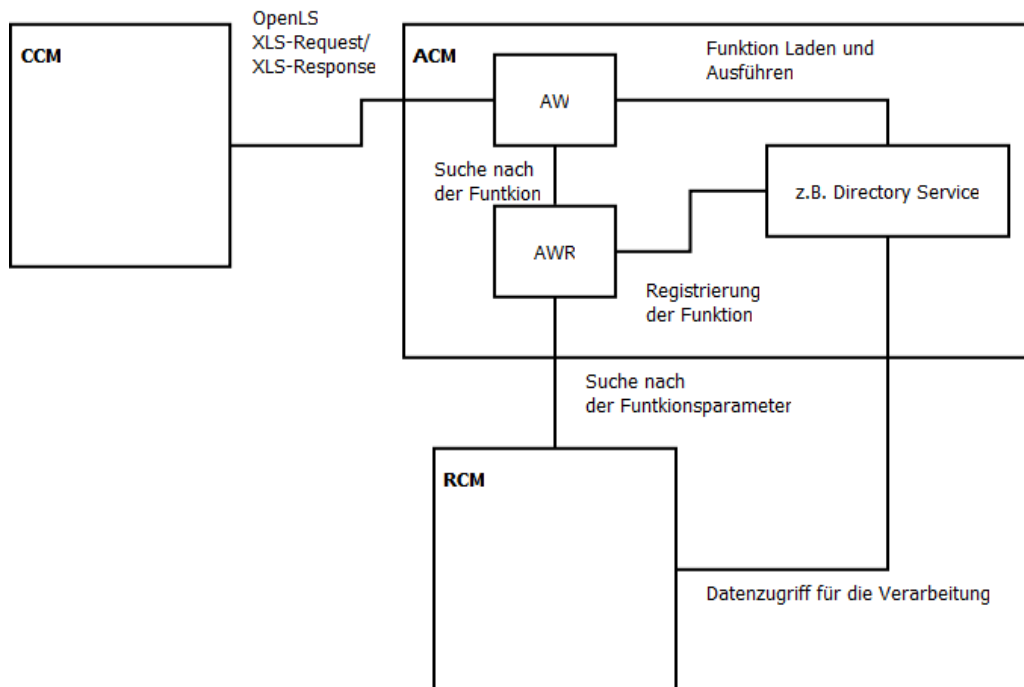


Abbildung 28: Aufbau des ACM

Das Application Control Module (siehe Abbildung 18) kümmert sich um die tatsächliche Abarbeitung der Anforderungen, als auch um die Verwaltung der Funktionalität des Systems, welche vom Client in Anspruch genommen werden können.

Der vom Control Scheduler übergebene Request wird vom Application Worker (AW) übernommen und abgearbeitet und das Ergebnis dem CS wieder übermittelt. Ein AW lädt die jeweilig in XLS-Request angeforderte Funktion, welche die Datenverarbeitung übernimmt (z.B.: Route Service). Die Funktion greift für die Erfüllung ihrer Aufgabe auf die dafür benötigten Daten (z.B.: Routengraph) und auf das Resource Control Module (RCM) zu. Um den aktuellen Ort der Funktion zu erhalten wird die Funktion in der Application Worker Registration (AWR) gesucht und man erhält den Ort (bzw. Datei) an dem sich der auszuführende Code (z.B. PHP-Skript) der Funktion befindet.

Neben der eigentlichen Behandlung der Abarbeitung der Anfragen bietet das ACM noch eine Funktionalität zur Verwaltung der Abarbeitung der Anfragen an. Die Application Worker Registration (AWR) teilt so zum Beispiel dem CS mit, welche AWs zur Verfügung stehen und gibt so Auskunft über die dem Client bereitzustellende Funktionalität. Die statische Verwaltung der Funktionalität erfolgt innerhalb der Systemdatenbank. Somit greift die AWR über das RCM auf die benötigten Daten der Funktion (z.B.: Speicherort, Typ, Rechte) zu.

4.2.3 Resource Control Module

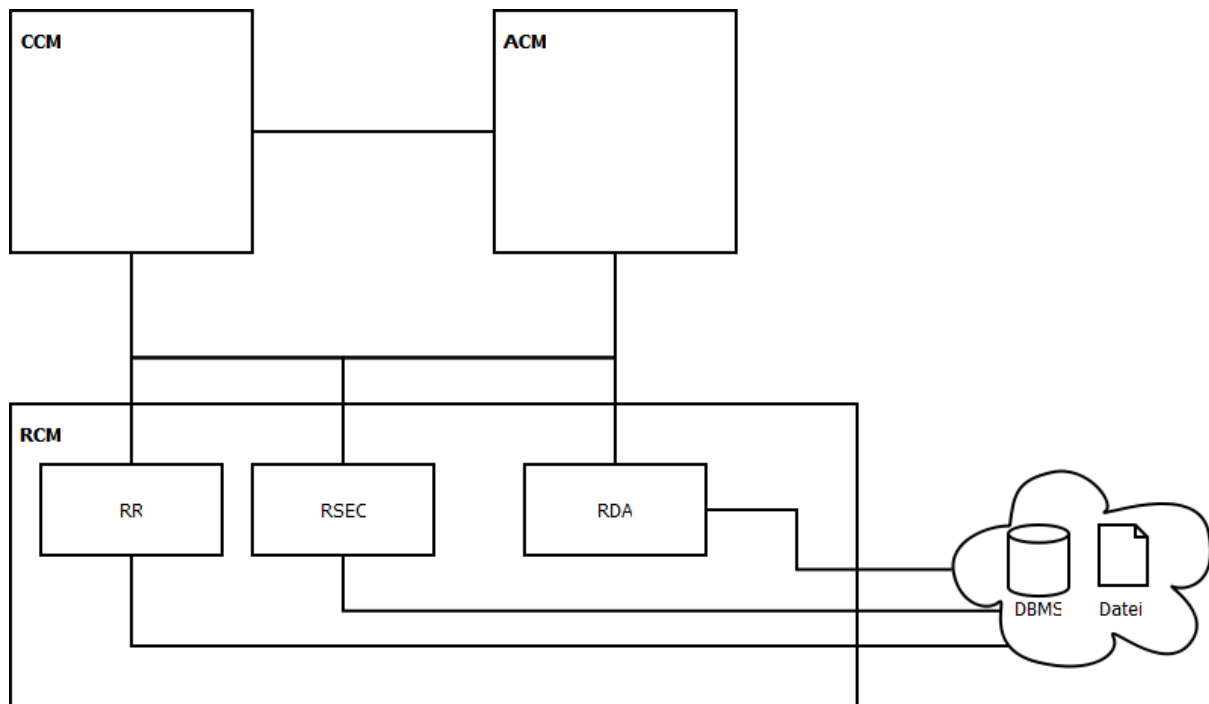


Abbildung 29: Aufbau des RCM

Das Resource Control Module (siehe Abbildung 29) kümmert sich um den Zugriff auf die eigentlichen Datenkomponenten (z.B.: Datenbankmanagementsystem (DBMS), externe DBMS, andere Datenquelle). Hierfür stellt sie mit der Resource Registration (RR) ein Informationsmodul bereit, welches Auskunft über die Verfügbarkeit der einzelnen Ressourcen und deren Typ gibt. Des Weiteren gibt er der Resource Registration Auskunft über die Art des Zugriffs (z.B.: SQL- Abfrage, oder direkter Dateizugriff).

Die Resource Security (RSEC) stellt Funktionalität zur Verifizierung von Daten (z.B.: Benutzername und Passwort oder Überprüfung von Daten auf Manipulation durch Hashwerte) bereit. Daneben gibt es auch Auskunft über die zur Verfügbarkeit stehenden Funktionen (z.B. OpenLS Core Services) des Systems.

Die eigentliche Datenanfrage erfolgt über den Resource Data Access (RDA). Dieser greift auf die Daten direkt zu und bereitet sie für die abfragenden Module auf (z.B.: ersetzt `\n` durch `
`). Da das RDA auch auf externe Datenquellen zugreifen kann, können diese durch das RSEC Modul noch verifiziert werden. Dies soll sicherstellen, dass keine falschen oder manipulierten Daten weitergereicht werden.

Die einzelnen Module des RCM können sowohl vom CCM als auch von ACM für die diversen Aufgaben (z.B.: Verifikation, Datenzugriff) benutzt werden.

5 Technologien

Um das System auch tatsächlich in der Realität umsetzen zu können muss auf bereits vorhandenen Technologien bzw. Techniken zurückgegriffen werden.

Dies umfasst einerseits Technologien welche die Hardware des Systems betreffen, als auch die Hardware des Endbenutzers definieren und so die Kommunikationswege beeinflussen.

Andererseits betrifft dies Softwaretechnologien auf welche bei der Implementierung der Komponenten des Systems zurückgegriffen werden kann.

Da es besonders im Softwarebereich eine Menge von möglichen Realisierungen gibt, wird hier nur auf jeweils eine Möglichkeit eingegangen bzw. die Anforderungen an diese bei der Realisierung angeführt. Man kann die verschiedenen Teile des Systems mit den unterschiedlichsten am Markt befindlichen Produkten erstellen.

Durch diese große Vielfalt wird im folgenden Abschnitt wie gesagt kein allumfassender Überblick über die benötigte Hardware und Software gegeben, sondern nur welche Anforderung die Systemkomponenten erfüllen müssen bzw. welche Möglichkeiten es für die Implementierung gibt.

5.1 Server

5.1.1 Hardware

Die Grundanforderung auf der Serverseite an die Hardware besteht darin, dass in erster Linie eine Verbindung zu einem Client hergestellt werden kann. Bei mobilen Clients geschieht dies über eine Netzwerkverbindung über ein auf Funk basierendes Netzwerk (z.B. WLAN, GSM, UMTS). Bei der TU-Graz ist eine Internetverbindung notwendig, falls eine Kommunikation außerhalb der Campusbereiche (siehe Abbildung 1: Campus Bereich der TU-Graz) notwendig ist (z.B.: online Zielführung).

Die weiteren Anforderungen, welche die Performance und Zuverlässigkeit betreffen, ergeben sich schlussendlich aus der Anzahl und Art der Nutzung der Clients. Aufgrund der Systemarchitektur (siehe 4 Systemarchitektur) ist es kein Problem, das System auf einen bis mehreren Rechnern zu skalieren und die einzelnen Aufgabenbereiche (siehe Systemschichten 3 Systemkomponenten) auch aufzuteilen.

Eine mögliche Serverstruktur könnte wie folgt aussehen (siehe Abbildung 30):

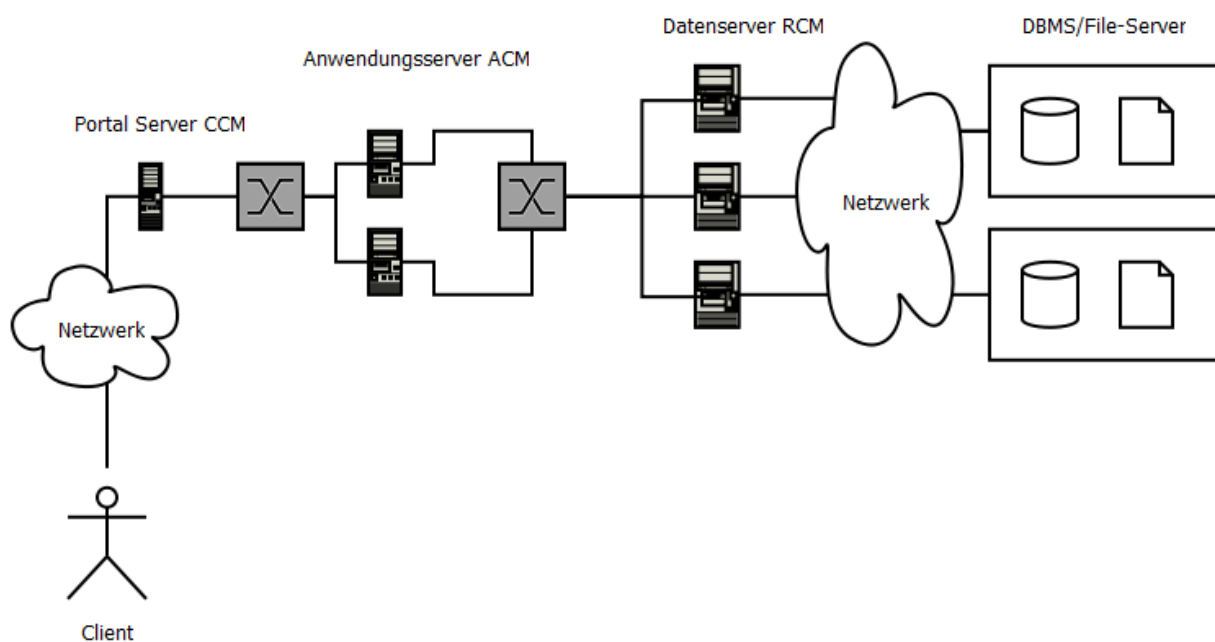


Abbildung 30: Mögliche Severinfrastruktur

Die einzelnen Server (CCM, ACM, RCM) sind zum Beispiel über Switches (siehe Abbildung 31) verbunden oder befinden sich auch in unterschiedlichen Netzen (siehe RCM Datenserver und DBMS/File-Server in Abbildung 31).

5.1.2 Software

Auf Seite der Software wird an vorderster Front auf dem Portalserver ein Webserver benötigt, welcher mit dem Client kommuniziert. Dies ist notwendig um die Kommunikation mit Web-Browsern zu ermöglichen.

Je nach Wahl der Entwicklungstechnologien der GMS bzw. Applikationsschicht müssen diese die verschiedensten Technologien unterstützen. Wird der GMS als C-Applikation entwickelt muss z.B. CGI unterstützt werden, bei einer Realisierung in PHP muss auch diese Skriptsprache unterstützt werden. Auch müssen die höheren, auf dem HTTP-Protokoll aufbauenden Protokolle unterstützt werden, das reicht von standardisierten Protokollen wie zum Beispiel SOAP bis hin zu eigenen Entwicklungen für spezielle Clients.

Der Webserver übernimmt auch die Aufgabe der ersten Schicht auf Seiten des Servers und verteilt die empfangenen Anfragen auf die Applikationsserver (bzw. Anwendungsserver) welche nicht zwingend auf dem gleichen Server laufen müssen. Der Applikationsserver entspricht in seiner Ausführung auch dem OpenLS GMS. Er kann seine Funktionalität in Form von Webservices (siehe [17]) den Clients zur Verfügung stellen. Für die Durchführung steht wiederum eine Vielzahl an Möglichkeiten bereit, sei es eine Entwicklung in PHP oder als eigene Entwicklung (Beispiele für Applikationsserver siehe [18]).

Die Datenverwaltungsschicht wird zum einen Teil (RCM) auch als Applikationsserver und zum anderen Teil in Form eines Datenbankservers ausgeführt, welcher mit den anderen Servern kommunizieren muss. Dieser muss einerseits die normale Sachdaten verwalten können. Dies kann zum Beispiel in Form eines DBMS wie PostgreSQL (siehe [19]) ausgeführt werden. Andererseits muss er darüber hinaus auch noch geographische Daten verwalten können (siehe 2.4 Position). Für diesen Zweck kann man zum Beispiel einen Aufsatz auf das relationale DBMS wie PostGIS (siehe [20]) für PostgreSQL verwenden.

5.2 Client

5.2.1 Hardware

Bei der Hardware auf Clientseite gibt es eine große Vielzahl an verschiedensten Konfigurationen, welche sich ständig ändert. Jedoch lassen sich gewisse Grundvoraussetzungen definieren.

So muss der Client die Möglichkeit besitzen eine Kommunikationsverbindung mit dem Server aufbauen zu können. Dies kann von einem kabelgebunden Kommunikationsmedium (z.B.: TUG-Terminal) bis hin zu den diversen Funktechnologien (z.B.: WLAN, GSM, UTMS, LTE) gehen. Der Server muss dabei jedoch erkennen, um welches Kommunikationsmedium es sich handelt. (So weist beispielsweise eine bestimmte IP-Adresse auf ein kabelgebundenes Medium hin, oder der Client teilt dem Server mit, welches Medium er verwendet, z.B. indem ein

Benutzer den Link für die mobile Webseite des Systems auswählt). Dies ist notwendig um das Kommunikationsmedium optimal ausnutzen zu können, um dann so z.B. die Kosten gering halten zu können oder für eine optimale Darstellung zu sorgen.

In einem LBS-System liegt ein großes Augenmerk auf der Bestimmung der aktuellen Position, da diese für diverse Aufgaben (z.B.: Routenberechnung, Zielführung) benötigt wird. Hierbei entscheidet die unterschiedliche Hardwareausstattung der Clients über die Quantität (Wie oft kann die Position bestimmt werden) und die Qualität (wie genau kann die Position bestimmt werden; siehe Abbildung 31) der Positionsbestimmung. In jedem Fall wird bei der Nutzung der verschiedenen Sensoren oder Sensorkombinationen spezielle Software benötigt, welche die Handhabung dieser erledigt.

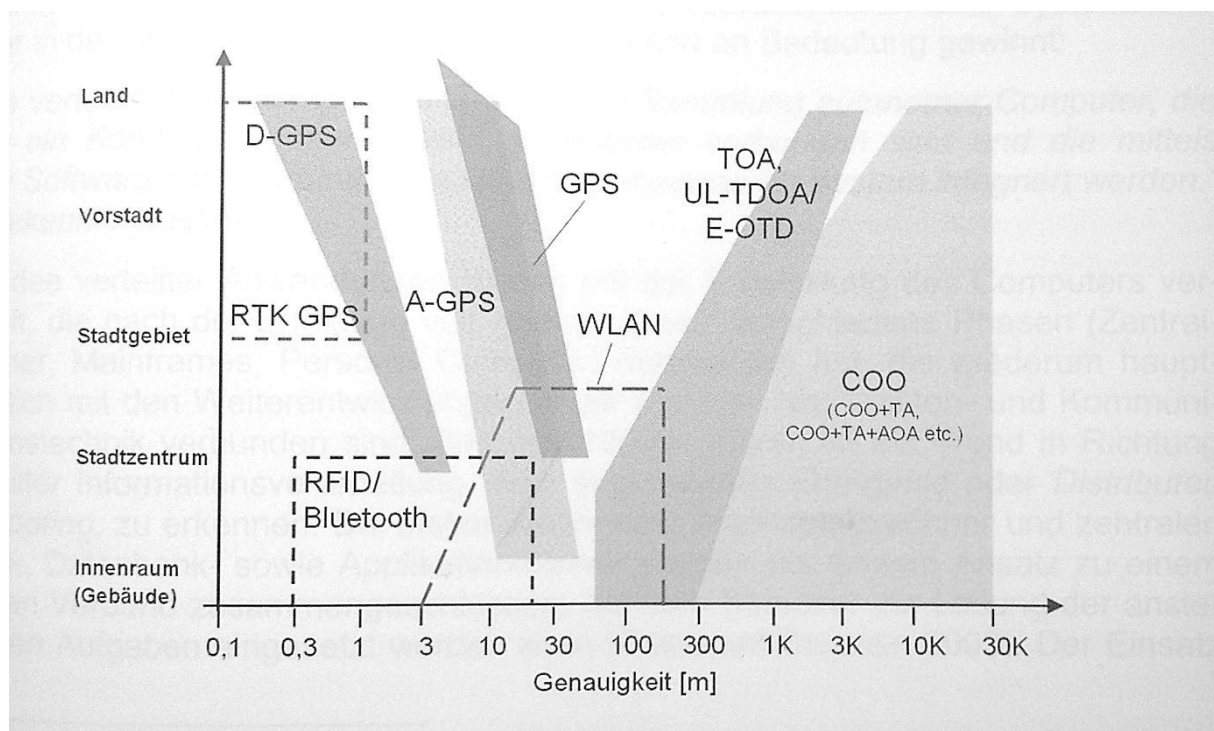


Abbildung 31: Überblick über die verschiedenen Technologien zur Positionsbestimmung
Quelle: [5] S. 119

Bei standorttreuen Geräten sind meist keine weiteren Sensoren notwendig, da die Zuordnung des Clients aufgrund der Kommunikation geschieht und die Position bereits bekannt ist und sich nicht ändert.

Bei mobilen Geräten werden die weiteren Sensoren wichtiger, da die Position des Clients im Vorfeld noch nicht bekannt ist, und sich auch ändern kann. Die Genauigkeit der Position sollte eine Zuordnung zu einzelnen Räumen bzw. Stockwerken zulassen. Bei einer angenommenen Raumhöhe von ca. drei Meter landet man in dem Bereich indem man die Position z.B. noch mittels WLAN bestimmen könnte.

Im Folgenden soll ein kurzer Überblick über mögliche Sensoren und deren Funktionsweise gegeben werden, wobei diese einzeln oder in Kombination verwendet und auch durch weitere Sensoren (z.B.: Beschleunigungssensoren) unterstützt werden können.

Eine einfache Möglichkeit die Position eines mobilen Benutzers zu ermitteln, ist die Ausnutzung eines optischen Sensors (Handykamera) in Verbindung mit einem zu erkennenden Muster. Hierfür könnte man zum Beispiel QR-Codes einsetzen [21]. Da die Aufnahme nur in näherem Umfeld und nur in einem einschränkbareren Blickwinkel geschehen kann (siehe Abbildung 33), ist die Position und ungefähre Blickrichtung des Clients für den Moment der Bildaufnahme bestimmbar.



Abbildung 32: Beispiel für QR-Code [21]

Durch Codierung eines Weblinks mit einer eindeutigen ID für den Standort in einem QR-Code (siehe Abbildung 32) wird beim Aufruf dieses Links die ID übermittelt und das System kann auf Grund der bekannten Beziehung $ID \leftrightarrow \text{Standort}$ den Standort des Benutzers bestimmen und auch die ungefähre Blickrichtung der Handykamera zum Zeitpunkt der Aufnahme.

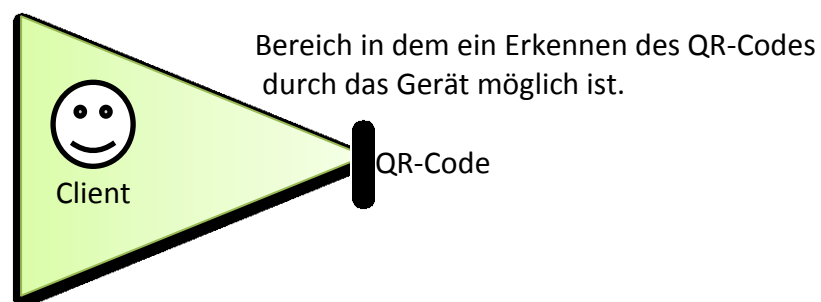


Abbildung 33: Draufsicht einer QR-Aufnahmesituation

Jedoch basieren alle weiteren ortsabhängigen Anfragen des Clients auf diesem Ort und berücksichtigen nicht eine etwaige Positionsänderung. Eine weitere Anwendung des Kamerasensors ist das Vergleichen des aktuellen Kamerabilds mit Referenzbildern. Diese Möglichkeiten fallen in den Bereich der Mustererkennung und erfordern eine ausreichende Rechenleistung (siehe [15]).

Will man die Position des Clients während er sich bewegt feststellen ohne dabei bekannte Referenzbilder oder feste Standorte zu benutzen, muss man auf Funk basierende Technologien zurückgreifen. Grundsätzlich basieren sie auf dem Messen von Laufzeiten bzw. Laufzeitunterschieden von bekannten Sendern. Da die ausgesendeten elektromagnetischen Wellen diversen Effekten (Dämpfung, Mehrwegeeffekte) unterliegen, kommt es zu anderen gemessenen Laufzeiten, als der Empfänger eigentlich erwarten würde (siehe [15]).

So eignet sich ein Sensorsystem nicht für jedes Anwendungsgebiet und muss von anderen Sensoren unterstützt werden.

GNSS (z.B. GPS) bietet zum Beispiel in offenen Geländeformen (z.B.: Wiese) eine gute Qualität und vollständige Überdeckung, jedoch kommt es zu Problemen in verbauten Gebieten und in Gebäuden ist je nach Bauweise meist keine Verwendung möglich.

Im Falle, dass eine Positionsbestimmung durch GPS nicht oder nur schlecht möglich ist, könnte man auf die Verwendung von WLAN ausweichen (siehe Abbildung 31, Beispielsprojekt siehe [22]). Da im Falle von WLAN die Positionsbestimmung aufgrund der Empfangsstärke erfolgt (siehe [23]), kann man auf die mittlerweile weit verbreiteten WLAN-Stationen zurückgreifen.

Hierfür wird im Vorfeld eine Karte der Empfangsstärken ermittelt. Der Client übermittelt die von ihm empfangen Signalstärken dem Server, welcher aus diesen die Position in der Karte bestimmen kann. Das vorangehende Messen der Empfangsstärken ist notwendig, da nicht alle Standorte aller WLAN-Sender bekannt sind, bzw. deren Empfangsstärke aufgrund der Gebäudetopologie keine konzentrischen Empfangsstärkeisolinien ergeben. Bei dem Erstellen solcher WLAN-Karten sollte es dabei vermieden werden, dass Nutzdaten mitgeschnitten werden (siehe [24])

Des Weiteren gibt es noch eine Vielzahl an verschiedenen anderweitigen Sensoren, die zur Positionsbestimmung herangezogen werden können. In der Zukunft dürften dies beispielsweise Beschleunigungssensoren leisten können, welche einerseits die Lage des Gerätes im Raum ermitteln, als auch die Beschleunigungen und durch Integration dieser Werte den vom Gerät zurückgelegten Weg feststellen können. Mittels RFID wird es auch möglich sein beinahe unsichtbare, günstige Tags zu erstellen, welche eine sehr genaue Positionsbestimmung ermöglichen. Diese beiden Technologien haben eine gute Zukunftssicherheit, da sie in den kommenden Betriebssystemen namhafter Hersteller von Smartphone Betriebssystemen unterstützt werden [25].

5.2.2 Software

Die Software übernimmt auf den Clientgeräten in erster Linie die Darstellung der Benutzeroberfläche und die Interaktion mit dem menschlichen Benutzer. In zweiter Linie kümmert sie sich um die Interaktion mit der Clienthardware. Dies beginnt bei der Benutzung der diversen

Sensoren (z.B.: WLAN, GPS) bis hin zur Ausnutzung der Grafikkarte zur Darstellung von 3D-Inhalten.

Im einfachsten Fall sollte auf dem Client ein Webbrowser laufen, welcher eine statische HTML-Seite darstellen kann. Ein weiterer Evolutionsschritt ist die Unterstützung von JavaScript um dynamischere Seiten zu generieren bzw. direkter mit den GMS-Core Services zu kommunizieren (mittels AJAX).

Die höchste Ausbaustufe besteht in der Entwicklung einer eigenen Applikation (auf Desktop-Systemen) bzw. Apps (auf einem Smartphone). In diesem Fall sind dem Funktionsumfang nur Grenzen durch die Clienthardware und der vom System zur Verfügung gestellten Daten gegeben.

Dies reicht zum Beispiel von einer einfachen Benutzeroberfläche, ähnlich einer Webseite, bis hin zu komplexeren Applikationen wie zum Beispiel Google Earth mit einer 3D-Darstellung.

Die Entwicklung der Anwendungen für das Campus 2011 System ist sehr stark von dem auf den Geräten laufenden Betriebssystem abhängig, welches den Zugriff auf die in den Geräten verbaute Hardware ermöglicht. Dadurch wird auch der mögliche Funktionsumfang einer Clientanwendung festgelegt.

Zusammenfassen kann gesagt werden, dass dem Funktionsumfang einer Campus 2011 Clientanwendung hauptsächlich durch das verwendete Clientbetriebssystem und der darunterliegende Hardware definiert wird. Durch die modulare Schichtenbauweise des Serverbereichs kann auch flexibel auf Anforderungen durch die Clientsoftware reagiert werden.

6 Zusammenfassung und Ausblick

In diesem, letzten, Kapitel wird einerseits auf die bei der Entwicklung der Systemarchitektur des Campus 2011 Systems aufgetretenen Schwierigkeiten und Probleme eingegangen.

Andererseits befasst sich ein weiterer wichtiger Punkt mit der zukünftigen Entwicklung und Einsatzmöglichkeit des Campus 2011 Systems. Dies betrifft einerseits den Einsatz des Systems an der Technischen Universität Graz als auch andererseits den Einsatz eines solchen Systems in anderen Anwendungsscenarios.

6.1 Zusammenfassung

Die Problematik bei der Erstellung dieser Arbeit bestand einerseits in der Komplexität der Umgebung in der das System einmal seinen Dienst versehen soll, als auch andererseits auf der sich erst im Laufe der Zeit verfeinernden Zielsetzung. So wurde das Campus 2011 System zunächst als reines Ausstellungssystem angesehen, wobei die Notwendigkeit einer LBS-Komponente noch nicht genau zu sehen war und sich erst während der Erstellung zeigte, da beinahe jegliche Information sich mit Ortsinformation verknüpfen lässt und sich dies somit zur Filterung als auch Spezifikation der verarbeiteten Daten nutzen lässt.

Um auf das heterogene Umfeld reagieren zu können und gleichzeitig auch eine möglichst gute Zukunftssicherheit des Campus Systems zu gewährleisten, wurde auf die umfangreiche Spezifikation des OpenLS Standards zurückgegriffen. Dieser definiert zwar die Schnittstellen und den grundsätzlichen Aufbau eines LBS-Systems, jedoch steht die tatsächliche Implementierung den Systementwicklern offen. So kann auch auf die Eigenheiten des Campus 2011 Systems besser eingegangen werden.

Aufgrund des großen Umfangs des Systems wurde auf eine vollständige Implementierung verzichtet. Es wurde jedoch die zu Grunde liegende Datenbank erstellt und auszugsweise wurden Teilbereiche des Systems implementiert und getestet. Dies umfasst einerseits die Kommunikation einer Anwendung direkt mit dem OpenLS Services, als auch andererseits die Kommunikation eines Clients (Webbrowser) über das Systemportal. Auch wurde die Integra-

tion von Inhalten Dritter getestet indem externes Kartenmaterial (OpenStreetMap) und auch eine Informationsspur über die Person Tesla (externer Datenserver) verarbeitet wurden.

Somit sind die grundlegenden Strukturen für die Entwicklung des Campus 2011 Systems geschaffen. Systemintern benötigt es noch Anstrengungen, alle OpenLS Services vollständig zu implementieren, und auf Clientseite die geeigneten Anwendungen bzw. Webinterfaces zu schaffen, welche die von den Services gelieferten Informationen den Benutzern ansprechend und leicht verständlich darstellen.

6.2 Ausblick

Die offene und auf Verteilungstransparenz ausgelegte Architektur ermöglicht eine problemlose Erweiterung auf zukünftige, noch nicht bekannte Einsatzmöglichkeiten.

So könnte das Campus 2011 System stärker mit dem bereits existierenden Informationssystem integriert werden und mit Hilfe der dort verwendeten Benutzerauthentifizierung und den verfügbaren Informationen verknüpft werden. Das hieße zum Beispiel, dass ein Student immer die aktuellen Änderungen an seinem Stundenplan auf sein mobiles Gerät mitgeteilt bekommt. Des Weiteren könnten insbesondere die erstsemestrigen Studenten das System für die Zügelführung zwischen den verschiedenen Lehrveranstaltungen nutzen und dadurch das Campusgelände noch besser kennen lernen.

Darüber hinaus könnte das System noch um eine Web 2.0 Komponente erweitert werden, wo nicht nur bestimmte berechnete Inhaltsersteller Informationen in der virtuellen und realen Welt erstellen, sondern dies auch die Studenten durch die Institute tun können. Dies kann zu einer noch besseren Zusammenarbeit zwischen den verschiedenen Instituten führen, da die netzartige Struktur des Campus 2011 alle Informationen in Raum, Zeit, Begriff und Benutzer miteinander vernetzen kann und dies nicht nur auf die virtuelle Welt beschränkt, sondern mit der realen Welt verknüpfbar ist. So kann zum Beispiel eine virtuelle Pinnwand erstellt werden, welche kaum bis keinen Platz in der realen Welt benötigt, jedoch in der virtuellen Campus-2011-Welt ein riesiges Angebot an Informationen bereitstellt, welche wiederum zu anderen Informationen führen.

Prinzipiell eignet sich das LBS-System bei denjenigen Einsatzbereichen, bei denen es darum geht, für viele mobile Einheiten Informationen bereitzustellen bzw. diese damit zu versorgen. Dies trifft z.B. auf die Touristen einer fremden Stadt zu, welche sich am ersten Tag im Hotel ihren Reiseführer herunterladen und somit zu den Sehenswürdigkeiten der Stadt geleitet werden.

Ein weiteres Einsatzgebiet ergibt sich in den Bereichen, in denen es um die Überwachung von Objekten geht, welche wiederum mobil sind und so keinen fixen Standort haben. Dies ist zum Beispiel bei Verkehrsbetrieben möglich, um Auskunft darüber zu erhalten, welche Haltestelle die richtige ist, um zu einem bestimmten Ort zu kommen, wobei der aktuelle Stand-

ort der Busse berücksichtigt wird. Es kann so möglich sein, dass ein Ziel schneller erreichbar wird, obwohl man zu einer weiter entfernten Haltestelle geht.

Die weiteren Anwendungsmöglichkeiten dieses Systems sind sehr vielfältig. Aufgrund der zu Grunde liegenden Technologien und Architektur ist eine modulare Kombination mehrerer, unterschiedlicher Anwendungs- und Informationskomponenten möglich.

A. Literaturverzeichnis

1. **WOHINZ, Josef W.** *Die Technik in Graz*. Graz ; Wien [u.a.] : Verl. Styria, 2002.
2. **TUGraz.** TUGraz. [Online] 30. Oktober 2010. <http://www.tugraz.at>.
3. **OGC.** Open Geospatial Consortium. [Online] 22. November 2010. <http://www.opengeospatial.org>.
4. **umts-forum.org.** UMTS forum - LBS - Location Based Services. *www.umts-forum.org*. [Online] 22. November 2010. <http://www.umts-forum.org/content/view/1595/197/>.
5. **BLANKENBACH, Jörg.** *Handbuch der mobilen Geoinformation*. Heidelberg : Wichmann, 2007.
6. **ÖNORM.** Geoinformation - Standortbezogene Dienste - Referenzmodell (ISO 19132:2007).
7. **OGC.** OpenGIS Location Services (OpenLS): Core Services. [Online] 22. November 2010. http://portal.opengeospatial.org/files/?artifact_id=22122.
8. —. OpenGIS® Geography Markup Language (GML) Encoding Standard. [Online] 30. Oktober 2010. http://portal.opengeospatial.org/files/?artifact_id=20509.
9. **LIF.** Location Interoperability Forum. [Online] 22. November 2010. <http://locationforum.org>.
10. **OGC.** Web Map Service Implementation Specification Version 1.1.1. [Online] 22. November 2010. <http://www.opengeospatial.org/specs>.
11. **HTTP.** Hypertext Transfer Protocol. [Online] 21. November 2010. <http://www.ietf.org/rfc/rfc2616.txt>.
12. **BILL, Ralf.** *Grundlagen der Geo-Informationssysteme (Band)*. Berlin [u.a.] : s.n., 1991.
13. **BARTELME, Norbert.** *Geoinformatik*. Berlin, Heidelberg : Springer-Verlag, 2005.
14. **SCHNEIDER, Uwe und WERNER, Dieter.** *Taschenbuch der Informatik*. München, Wien : Carl Hanser Verlag, 2000.
15. **HOFFMANN-WELLENHOF, Bernhard, LEGAT, Klaus and WIESER, Manfred.** *Navigation*. Wien : Springer, 2003.
16. **SCHILL, Alexander und SPRINGER, Thomas.** *Verteilte Systeme*. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2007.
17. **WC3.** Working Group - Web Services Glossary. [Online] 22. November 2010. <http://www.w3.org/TR/ws-gloss>.
18. **wikipedia.org.** Liste von Anwendungsservern. [Online] 30. Oktober 2010. http://de.wikipedia.org/wiki/Liste_von_Anwendungsservern.

19. **PostgreSQL**. The world's most advanced open source database. [Online] 22. November 2010. <http://www.postgresql.org>.
20. **PostGIS**. PostGIS: Home. [Online] 22. November 2010. <http://www.postgis.org>.
21. **tugraz.at**. QR-Codes | TU Graz. [Online] 1. November 2010. <http://qr.tugraz.at>.
22. **IBACH, Peter, HÜBNER, Tobias und SCHWEIGERT, Martin**. Magic Map - Kooperative Positionsbestimmung über WLAN. [Online] 30. Oktober 2010. <http://events.ccc.de/congress/2004/fahrplan/files/48-ad-hoc-collaboration-paper.pdf>.
23. **FULLER, Richard [Hrsg.]**. *Mobile entity localization and tracking in GPS-less enviroment*. Berlin Heidelberg : Springer-Verlag, 2009.
24. **derstandard.at**. Datenschutz gibt Google Street View in Österreich freie Fahrt. [Online] 10. November 2010. <http://derstandard.at/1289609151771/Datenschutzrat-gibt-Google-Street-View-in-Oesterreich-freie-Fahrt>.
25. **Heise Zeitschriften Verlag**. *heise online*. [Online] 20. Novmeber 2010. <http://www.heise.de/mac-and-i/artikel/Was-wir-2011-von-Apple-erwarten-1163378.html?artikelseite=4>.

B. Abbildungsverzeichnis

ABBILDUNG 1: CAMPUS BEREICH DER TU-GRAZ [2]	5
ABBILDUNG 2: KOMPONENTEN EINES LBS-SYSTEMS.....	11
ABBILDUNG 3: ROLLE DES GEOMOBILITY SERVERS IM RAHMEN EINES LBS-SYSTEMS QUELLE: IN ANLEHNUNG AN [7]	13
ABBILDUNG 4: AUFBAU DES GEOMOBILITY SERVERS (GMS) QUELLE: IN ANLEHNUNG AN [7]	14
ABBILDUNG 5: ABLAUF EINES OPENLS REQUEST/RESPONSE PAARS QUELLE: IN ANLEHNUNG AN [7]	19
ABBILDUNG 6: ABLAUF (1-6 REIHENFOLGE DER ABARBEITUNG) EINER TYPISCHEN REQUEST / RESPONSE KOMMUNIKATION QUELLE: IN ANLEHNUNG AN [7].....	21
ABBILDUNG 7: ZUSAMMENHANG ZWISCHEN REALER WELT, DATENSATZ UND INFORMATION.....	23
ABBILDUNG 8: INFORMATIONSKONTEXT IM ÜBERBLICK	24
ABBILDUNG 9: OPENLS INFORMATION MODEL QUELLE: IN ANLEHNUNG AN [7].....	28
ABBILDUNG 10: STRUKTUR EINES FAT CLIENTS	33
ABBILDUNG 11: STRUKTUR EINES THIN CLIENTS.....	33
ABBILDUNG 12: STRUKTUR EINES ULTRA THIN CLIENT	34
ABBILDUNG 13: BEZIEHUNG ZWISCHEN RESSOURCE UND DEM TATSÄCHLICHEN DATENSATZ	36
ABBILDUNG 14: LAUFFÄHIGES BEISPIEL FÜR EINE INFORMATIONSSPUR.....	37
ABBILDUNG 15: BEZIEHUNGEN DES INFORMATIONSKONTEXTS ZU DEN RESSOURCEN	38
ABBILDUNG 16: BEZIEHUNG DES INFORMATIONSKONTEXTS ZU DEN SERVICES	39
ABBILDUNG 17: BEZIEHUNG VON GEOGRAPHISCHEN DATEN ZUR OBJEKT-TABELLE.....	40
ABBILDUNG 18: LEVEL BEHANDLUNG DER HÖHENINFORMATION	41
ABBILDUNG 19: GRUNDSÄTZLICHE ELEMENTE EINES GRAPHEN	41
ABBILDUNG 20: STRUKTUR DER SPEICHERUNG DER GRAPHENDATENWINKELANGABE (TRAVERSE : : ANGLE) BESCHREIBT DEN WINKEL ZWISCHEN DEN BEIDEN VEKTOREN AUS DENEN EINE TRAVERSE BESTEHT (SIEHE ABBILDUNG 19). SIE WIRD BEI DER ROUTENBERECHNUNG FÜR DAS ERKENNEN VON ABBIEGUNGEN BENÖTIGT (D.H. AB WELCHEM WINKEL SPRICHT MAN Z.B. VON RECHTSABBIEGEN).....	42
ABBILDUNG 21: DATENSTRUKTUR ZUR SPEICHERUNG VON ROUTENINFORMATION	42
ABBILDUNG 22: AUFBAU EINER ZWEISCHICHTEN SYSTEMARCHITEKTUR.....	44
ABBILDUNG 23: DREISCHICHTIGE SYSTEMARCHITEKTUR.....	45
ABBILDUNG 24: VIERSCHTIGE SYSTEMARCHITEKTUR	45
ABBILDUNG 25: RESULTIERENDE SYSTEMSCHICHTEN (KOMMUNIKATION MIT CLIENTS WELCHE DAS OPENLS-PROTOKOLL VERWENDEN WERDEN DIREKT AN DEN GMS (ANWENDUNGEN) WEITERGEREICHT (BLAUER PFEIL))	45

Codeverzeichnis

ABBILDUNG 26: SYSTEMMODULE UND IHRE ZUORDNUNG ZUR JEWEILIGEN SCHICHT	47
ABBILDUNG 27: AUFBAU DES CCM.....	48
ABBILDUNG 28: AUFBAU DES ACM	49
ABBILDUNG 29: AUFBAU DES RCM.....	50
ABBILDUNG 30: MÖGLICHE SEVERINFRASTRUKTUR	52
ABBILDUNG 31: ÜBERBLICK ÜBER DIE VERSCHIEDENEN TECHNOLOGIEN ZUR POSITIONSBESTIMMUNG [5][sic!].....	54
ABBILDUNG 32: BEISPIEL FÜR QR-CODE [21]	55
ABBILDUNG 33: DRAUFSICHT EINER QR-AUFNAHMESITUATION.....	55

C. Codeverzeichnis

CODE 1: XLS-REQUEST	19
CODE 2: XLS-RESPONSE	20
CODE 3: XLS-REQUEST MIT ERWEITERUNG FÜR DEN CAMPUS 2011 INFORMATIONSKONTEXT AM BEISPIEL EINES DIRECTORY SERVICES REQUESTS.....	25

D. Tabellenverzeichnis

TABELLE 1: ÜBERSICHT ÜBER DIE VON OPENLS VERWENDETE ABSTRAKTE DATENTYPEN	27
TABELLE 2: BENUTZEREINTEILUNG.....	31
TABELLE 3: BEISPIELE FÜR DIE EINTEILUNG UNTERSCHIEDLICHER GERÄTE	34