

# Universität Stuttgart

## Fakultät Informatik

Prüfer: Prof. Dr. Kurt Rothermel  
Betreuer: Dr. C. Burger

Beginn am: 01.03.2001  
Beendet am: 31.08.2001

CR-Klassifikation H.5.2, H.5.3, H.4.3, J.7

Studienarbeit Nr. 1817  
Prototypische Erstellung einer  
Sitzungsverwaltung für den  
Einsatz in Lehrveranstaltungen  
Marcus Sommer

Institut für Parallele und Verteilte  
Höchstleistungsrechner  
Breitwiesenstraße 20-22  
D-70565 Stuttgart

## Inhaltsverzeichnis

Inhaltsverzeichnis.....	i
Abbildungsverzeichnis.....	iii
1. Einleitung.....	1
1.1 Motivation.....	1
1.2 Ziele.....	3
1.3 Übersicht.....	4
1.4 Quellenangaben.....	6
2. Projekt SASCIA.....	7
2.1 Projektbeschreibung.....	7
2.2 Quellenangaben.....	13
3. Anforderungsanalyse.....	15
3.1 Eingabe der Sitzungsdaten.....	15
3.2 Sitzungsverzeichnis.....	16
3.3 Benutzerverwaltung.....	17
3.4 Sitzungsverwaltung.....	18
4. Architektur.....	23
4.1 Server Architektur.....	23
4.1.1 Lokaler Administrationsclient.....	23
4.1.2 Benutzerdatenbank.....	25
4.1.3 Sitzungsverzeichnis.....	28
4.1.4 Sitzungsverwaltung Serverseite.....	31
4.1.5 Sitzungsverwaltung Administrationsserver.....	33
4.1.6 Serverkonfiguration.....	33
4.1.7 Ablauf beim Start einer Sitzung auf Serverseite.....	35
4.2 Client Architektur.....	36
4.1.8 Sitzungsverwaltung Clientseite.....	36
4.1.9 Sitzungsverwaltung Administrationsclient.....	39
4.2 Zusammenspiel der Sitzungsverwaltungskomponenten.....	40
4.3 Quellenangaben.....	44
5. Benutzungsoberfläche.....	45
5.1 Serveroberflächen.....	45

5.1.1	Allgemeine Bemerkungen.....	45
5.1.2	Erfassung der Sitzungsdaten .....	46
5.1.3	Benutzerverwaltung .....	49
5.2	Clientoberflächen .....	51
5.2.1	Sitzungsverwaltung Clientseite.....	51
5.2.2	Sitzungsverwaltung Administrationsclient .....	53
5.3	Quellenangaben.....	57
6.	Test und Bewertung .....	58
6.1	Test und Bewertung .....	58
6.2	Quellenangaben.....	66
7.	Zusammenfassung und Ausblick .....	67
8.	Anhang .....	69
8.1	Projektplan .....	69
8.1.1	Einleitung .....	69
8.1.2	Beschreibung der Arbeitspakete.....	70
8.1.3	Zeitplan.....	73
8.1.4	Dokumente und Meilensteine.....	74

## Abbildungsverzeichnis

Abbildung 1: Funk-LAN-Anbindung an den Sitzungsserver .....	2
Abbildung 2: Systemarchitektur SASCIA .....	8
Abbildung 3: Standardwerte für Sitzungsparameter .....	24
Abbildung 4: Ablauf einer Authentifizierungsanfrage.....	26
Abbildung 5: Ablauf einer Veröffentlichung im Sitzungsverzeichnis.....	29
Abbildung 6: Komponenten des Sitzungsverzeichnisses.....	30
Abbildung 7: Ablauf beim Start einer Sitzung auf Serverseite .....	35
Abbildung 8: Ablauf beim Start einer Sitzung auf Clientseite.....	38
Abbildung 9: Zusammenspiel der Sitzungsverwaltungskomponenten .....	40
Abbildung 10: Abfrage der Sitzungsdaten .....	47
Abbildung 11: Benutzerverwaltung .....	49
Abbildung 12: Änderung/Hinzufügen eines Benutzers .....	50
Abbildung 13: Session Management Client Oberfläche .....	51
Abbildung 14: Remote Admin Client .....	53
Abbildung 15: Aufbau des Testsystems.....	62
Abbildung 16: Testergebnisse.....	65
Abbildung 17: Abhängigkeiten der Arbeitspakete.....	71

# 1. Einleitung

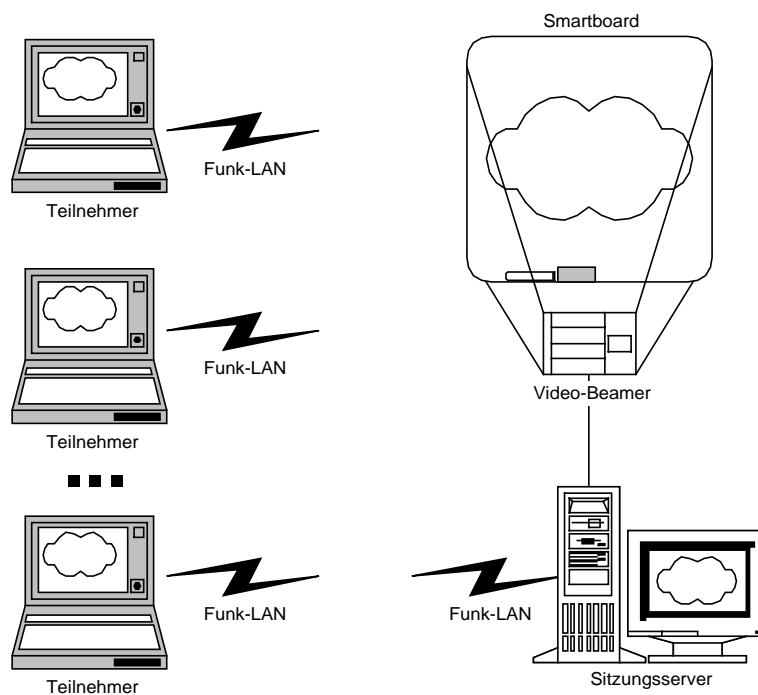
## 1.1 Motivation

Die Lehre an Universitäten leidet seit einiger Zeit an der fehlenden Interaktion zwischen Dozenten und Studierenden in den Vorlesungen. Dies wird, sowohl von Seiten der Dozenten als auch von Seiten der Studierenden, immer wieder beanstandet. Hervorgerufen wird dieses Symptom nicht nur durch die Studierenden selbst. Vielmehr fehlen in den Vorlesungen oft die Möglichkeiten die es den Studierenden erlauben sich aktiv daran zu beteiligen. Sei es zum einen, indem der Dozent selbst den Studierenden keine Möglichkeit bietet, oder es keine geeigneten Instrumente gibt, die es ermöglichen vorhandene Visualisierungsprogramme interaktiv ablaufen zu lassen. Als weiterer Grund für die fehlende Interaktion lassen sich auch die oft großen Vortragsräume nennen. Diese erfordern es, dass Studierende oft große Wege vom Sitzplatz zum Dozenten zurücklegen müssen, um selbst Anmerkungen zum Vortrag machen zu können. Auch fühlen sich die Studierenden oft durch die Größe eingeschüchtert, dass manche Fragen und Anmerkungen erst gar nicht gestellt werden.

In zunehmendem Maße werden die Vorlesungen deshalb an Universitäten und anderen Weiterbildungseinrichtungen mittels elektronischer Hilfsmittel wie Video Beamer und Computereinsatz durchgeführt. In neuerer Zeit kommen hierbei die sogenannten Smart Boards zum Einsatz. Bei diesen Boards handelt es sich um eine „elektronische Tafel“. Entwickelt wurden die Smart Boards (wie der Name schon aussagt) von der Firma [SMART] Technologies Inc. Diese Boards dienen als Projektionsfläche für die Präsentationsfolien, die mittels einem Video Beamer erzeugt werden. Neben dieser Funktion der Projektionsfläche bringen die Smart Boards noch zwei weitere Eigenschaften mit, die den Einsatz lohnend machen. Zum einen wird dem Vortragenden ermöglicht per Hand Anmerkungen auf der dargestellten Präsentationsfolie zu machen bzw. diese wieder zu löschen ohne die Präsentationsfolie zu zerstören, zum anderen gibt das Smart Board die Möglichkeit den Präsentationscomputer so zu steuern, als ob man direkt am Computer sitzt. Somit ist es möglich, die verschiedensten Programme und Medien mit in den Vortrag einfließen zu lassen.

Durch die Möglichkeit des Einsatzes von verschiedensten Programmen, Medien und der Smart Boards innerhalb eines Vortrages, lassen sich die Vorträge attraktiver und abwechslungsreicher gestalten. Allerdings ist das Problem der Interaktivität zwischen Dozent und Zuhörer damit immer noch nicht gelöst. Zuhörer sind bei dieser Art des Vortrages noch gezwungen, den Weg zwischen Sitzplatz und Dozent zurückzulegen, um eigene Anmerkungen machen zu können. Zusätzlich ist die Möglichkeit des Application-Sharings, der Möglichkeit zur gleichzeitigen Bedienung und Benutzung einer Applikation durch mehrere Teilnehmer, durch dieses Vorgehen noch nicht verwirklicht. Auch sind die Zuhörer dazu gezwungen, immer beim Vortrag lokal anwesend zu sein, um etwas davon mitzubekommen. Dies ist aber aufgrund von räumlicher Distanz zwischen den einzelnen Vortragsorten oft zeitlich nicht möglich.

Ein weiterer Schritt in Richtung Einsatz modernster Mittel innerhalb der Lehre, bietet ein neues Projekt. Im Rahmen dieses Projektes, das von der Universität und der Fachschaft in Zusammenarbeit mit verschiedenen Notebookanbietern angeboten werden soll, werden den Studierenden jeweils Notebooks im Leasingverfahren während der Studienzzeit zur Verfügung gestellt. Aber auch hier fehlt noch die Möglichkeit zur besseren Interaktion und zum Application-Sharing, da die Vernetzung zwischen den einzelnen Notebooks noch nicht ermöglicht ist.



**Abbildung 1: Funk-LAN-Anbindung an den Sitzungsserver**

In der Vernetzung von Computern hat sich durch die Einführung von Funk-LANs einiges geändert. Funk-LANs gelten in letzter Zeit immer mehr als Alternative für altbekannte Computernetzwerke, die noch auf drahtgebundener Technik beruhen. Mittels einem Funk-LAN ist es möglich, einzelne Computer drahtlos über Funk zu einem Netzwerk zusammen zu schließen. Somit können diese ohne aufwendige Verkabelungen in das Netzwerk eingebunden werden. Dies erleichtert den unkomplizierten Anschluss neuer Rechner in ein bestehendes Netzwerk enorm. Ein Funk-LAN besteht in der Regel aus einem sogenannten Access Point der die Verbindung zwischen den Computern im Funk-LAN und den Computern im fest verkabelten Netz herstellt. Die einzelnen Computer hierbei sind jeweils mit einer Funk-LAN-Karte ausgerüstet welche die Verbindung zum Access Point herstellt.

Wenn man nun die den Studenten zur Verfügung gestellten Notebooks, das Funk-LAN und das Smart Board zusammennimmt, hat man zumindest aus Hardware-Sicht ein System, mittels dem eine Interaktion zwischen Dozent und Studierenden möglich wird. Das Smart Board ersetzt hierbei die Tafel, auf der die Lehrinhalte angezeigt werden. Die Notebooks bei den Studierenden dienen als Client für die Kontaktaufnahme zum Dozenten und das Funk-LAN dient zum Transport der Anfragen und Anmerkungen zwischen Studierenden und Dozent. Allerdings fehlt hier immer noch ein System welches den ordnungsgemäßen Ablauf einer Vorlesung gewährleistet, die Vorlesung elektronisch aufbereitet und die Möglichkeit des Application-Sharings bietet.

### **1.2 Ziele**

Bei den im vorherigen Abschnitt aufgeworfenen fehlenden Bestandteilen, setzt nun das Projekt SASCIA (System Architecture Supporting Cooperative and Interactive Applications) an. Das Ziel des Projektes ist es, die Interaktivität zwischen Zuhörern und Vortragenden zu verbessern. Dies wird erreicht, indem eine kontrollierte Fernsteuerung von Präsentationen mittels Einsatz mobiler Endgeräte ermöglicht wird und die Möglichkeit des Application-Sharings implementiert wird. Hierzu wird ein System entwickelt, welches folgende Eigenschaften aufweist:

- Fernsteuerung einer Präsentation
- Anschluss mehrerer Endgeräte (bis zu 20 Teilnehmer)

- Annotationen innerhalb des Präsentationsinhalts von allen Endgeräten aus
- Protokollierung der Präsentationen inklusive Annotationen
- Koordination des Zugriffs
- Application-Sharing

Bei dieser Studienarbeit handelt es sich um eine Teilarbeit im Gesamtprojekt SASCIA. Der Schwerpunkt dieser Studienarbeit liegt in der prototypischen Erstellung einer Sitzungsverwaltung. Diese Sitzungsverwaltung dient als Hauptkomponente im Projekt. Ihre Aufgaben sind das Starten der Applikationen, die Benutzerverwaltung, die Bereitstellung der Kommunikationsplattform und die Publikation bereits laufender Sitzungen. Mittels den Applikationen wird im SASCIA-Projekt die Möglichkeit des Application-Sharings ermöglicht, indem eine Applikation bei jedem Teilnehmer der Sitzung lokal durch die Sitzungsverwaltung gestartet wird. Das Augenmerk bei der Entwicklung der Sitzungsverwaltung soll darauf liegen, dass diese für den Einsatz in Lehrveranstaltungen geeignet sein soll.

Neben dem Entwurf der Funktionalitäten für die Sitzungsverwaltung wird bei dieser Studienarbeit auch die Entwicklung einer Benutzungsoberfläche für die Sitzungsverwaltung angestrebt. Die genaue Aufgabenstellung der Studienarbeit lautet:

Zusammen mit anderen Studienarbeiten soll ein Konferenzsystem mit der entsprechenden Funktionalität erstellt werden. Im Rahmen dieser Studienarbeit sind dazu die Komponenten für einen fakultätsweiten Verzeichnisdienst und Sitzungsverwaltungen für die einzelnen Lehrveranstaltungen zu entwerfen, und prototypisch zu realisieren. Der Verzeichnisdienst verwaltet die aktuell laufenden Sitzungen. Die Verwaltung einer einzelnen Sitzung ist für die bei ihr angemeldeten Teilnehmer einschließlich ihres Sitzplatzes und die benötigten Dienste zuständig. Die Kommunikation zwischen allen beteiligten Komponenten soll drahtlos erfolgen.

### **1.3 Übersicht**

Um dem Leser einen besseren Überblick über dieses Dokument zu geben werden hier kurz die einzelnen Abschnitte dieser Ausarbeitung näher erläutert. Bis auf Kapitel zwei



spiegeln die einzelnen Kapitel auch die Vorgehensweise während der Studienarbeit wieder.

Im zweiten Kapitel wird zuerst das Gesamtprojekt SASCIA vorgestellt. Hiermit soll dem Leser die Möglichkeit gegeben werden, einen Überblick über das Projekt zu bekommen. Dies soll die Einordnung der Studienarbeit in das Gesamtprojekt erleichtern.

In der Anforderungsanalyse in Kapitel drei wurden, in Zusammenarbeit mit den anderen Teilnehmern des Projekts SASCIA, zuerst die Anforderungen an die Sitzungsverwaltung ausgearbeitet. Diese dienen als Grundlage für die Erstellung der Architektur.

Die Architektur in Kapitel vier spiegelt die in der Anforderungsanalyse herausgefundenen Anforderungen wieder. Hier werden diese Anforderungen genauer untersucht und in logische Programmeinheiten aufgeteilt. Des Weiteren wird hier anhand einer Grafik das Zusammenspiel der einzelnen Komponenten innerhalb und außerhalb der Sitzungsverwaltung erläutert. Hierdurch werden die einzelnen Schnittstellen deutlich.

In Kapitel fünf werden im ersten Abschnitt zuerst einige allgemeine Erläuterungen zur Gestaltung der Benutzungsoberflächen innerhalb der Sitzungsverwaltung gemacht. Das Hauptaugenmerk liegt hier aber auf der Beschreibung der entworfenen Benutzungsoberflächen.

Nach der Implementierung, die durch die Architektur bestimmt wurde, werden in Kapitel sechs nun die erstellten Funktionen und Oberflächen der Sitzungsverwaltung auf ihre Lauffähigkeit und ihre Korrektheit mittels Testfällen überprüft.

Zum Abschluss wird in Kapitel sieben neben einer Zusammenfassung der einzelnen Ergebnisse, die bei der Bearbeitung dieser Studienarbeit ermittelt werden konnten, auch ein Ausblick auf mögliche weitere Vervollständigungen im Bereich der Sitzungsverwaltung gegeben.

## **1.4 Quellenangaben**

SMART                    SMART Technologies Inc.  
Webseite der Firma Smart Technologies Inc  
<http://www.smarttech.com>

## **2. Projekt SASCIA**

### **2.1 Projektbeschreibung**

Das Projekt [SASCIA] ist ein Unterprojekt des [FESTIVAL] Projektes in der Abteilung [VERTEILTE SYSTEME] am [IPVR] der Universität Stuttgart. Im Projekt FESTIVAL wird einerseits untersucht, inwiefern neuere Entwicklungen im Bereich verteilter Systeme wie z. B. die Technologie mobiler Agenten sowie Vermittlungsverfahren gewinnbringend zur Interaktionsunterstützung eingesetzt werden können. Andererseits soll ein generisches Baukastensystem entwickelt werden, durch das beliebige interaktionsunterstützende Software komfortabel erstellt werden kann.

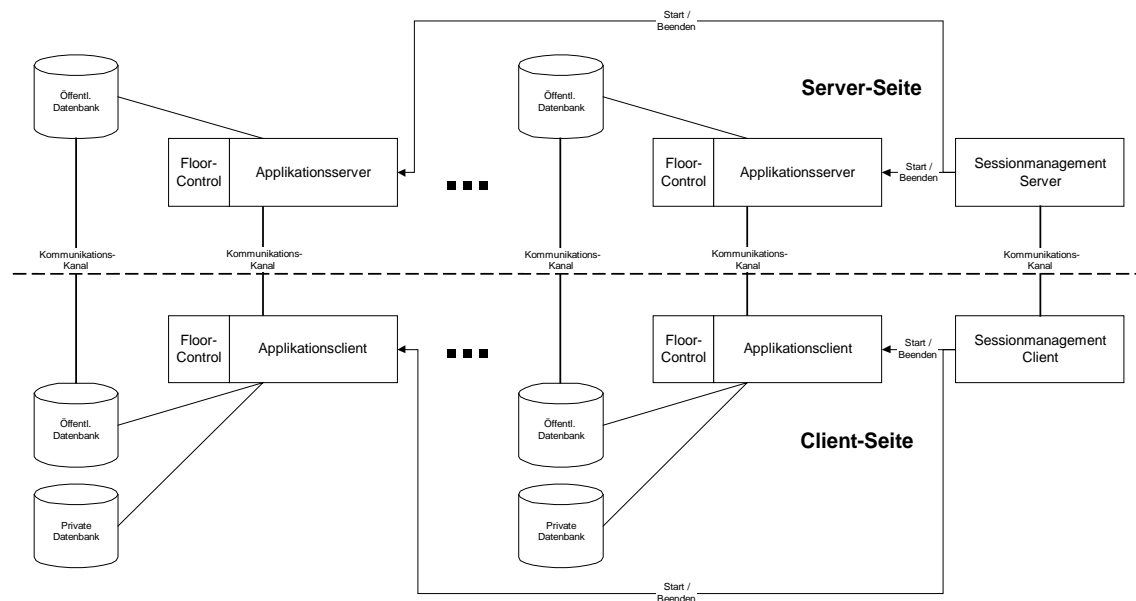
Der Name des Projekts SASCIA steht für **S**ystem **A**rchitecture **S**upporting **C**ooperative and **I**nteractive **A**pplications. Im Projekt soll ein Framework-System entwickelt werden, welches bereits bestehende Applikationen hinsichtlich der Interaktivität unterstützt und weiteren Applikationen diese Möglichkeit anbietet. Das System ist als sogenanntes Konferenzsystem realisiert, welches die Möglichkeit für die gemeinsame Anwendungsnutzung (application-sharing) bietet. Somit ist man in der Lage, mittels dem SASCIA System nicht nur wie bisher eine Präsentation vorzuführen, sondern auch verschiedene Applikationen zu zeigen, auf die gemeinsam (Zuhörer und Dozent) zugegriffen werden kann. Der Zugriff erfolgt hier nicht wahllos sondern kontrolliert.

Um die Möglichkeit der gemeinsamen Anwendungsnutzung zu realisieren baut das Projekt SASCIA auf Notebooks auf, die zumindest während der Probezeit jedem Zuhörer zur Verfügung gestellt werden. Später gibt es die Möglichkeit, dass jeder Studierende ein eigenes Notebook über das bereits angesprochene Projekt erhält. Die Verbindung der Notebooks untereinander erfolgt über ein Funk-LAN. Um die Kosten des Projektes zu senken, und um erste Erfahrungen zu sammeln, ist die Obergrenze der Teilnehmer im Moment auf zwanzig beschränkt. Trotzdem wird bei der Implementierung des Systems bereits jetzt darauf geachtet, dass dieses System auch für mehr Teilnehmer ausgelegt werden kann.

Das System besteht aus mehreren Komponenten. Die wichtigsten Komponenten sind:

- Kommunikationssystem
- Floor Control
- Applikationsserver und Applikationsclients
- Datenbank
- Session Management Server und Session Management Client

Folgendes Schaubild soll das Zusammenspiel der einzelnen Komponenten im Projekt SASCIA verdeutlichen.



**Abbildung 2: Systemarchitektur SASCIA**

Das Kommunikationssystem von SASCIA bietet die Schnittstelle, um eine Kommunikation zwischen den einzelnen Komponenten des Systems aufbauen zu können. Hierzu wird auf Serverseite ein Kanal geöffnet, dem die Komponenten auf Clientseite beitreten können. Der Nachrichtenversand zwischen den Komponenten erfolgt dann entweder an alle Teilnehmer außer an sich selbst oder an alle inklusive sich selbst. Für den Empfang der Nachrichten muss von jeder Komponente ein Konsument implementiert werden. Ohne diese ist ein Nachrichtenempfang nicht möglich, da der alleinige Beitritt zum Kanal dafür nicht genügt. Der Konsument ist dann in der Lage die empfangene Nachricht zu verarbeiten, und dann an die betreffende Komponente

weiterzugeben. Das in SASCIA implementierte Kommunikationssystem kapselt das eigentlich verwendete Kommunikationssystem. Dies ermöglicht es ohne weitere Änderungen in den anderen Systemkomponenten von Scia das eigentliche Kommunikationssystem, welches im Moment durch [JSDT] (**J**ava **S**hared **D**ata **T**oolkit) realisiert wird, zu ersetzen, falls es erforderlich werden sollte.

Die zweite wichtige Komponente ist die Floor Control, welche von Peter [Oberparleiter] in der Studienarbeit „Vergabe von Zugangsberechtigungen (floor control) für die gemeinsame Nutzung von Smartboards in der Vorlesung“ (2001) entworfen und implementiert wurde. Die Floor Control dient als Mechanismus zur Realisierung einer allgemeinen Rederechtvergabe, welche in vielen bisher auf dem Markt vorhandenen Programmen nicht oder nur unzureichend verfügbar ist. Diese allgemein gehaltene Floor Control steht jeder Applikation in SASCIA zur Verfügung, um bestimmte Funktionen oder Dienste freizugeben oder zu sperren. Mittels dieser Komponente kann ein geregelter Ablauf einer Lehrveranstaltung erreicht werden. Bei der Vergabe der Rederechte gibt es drei verschiedene Verfahren:

- zentrale Vergabe
- dezentrale Vergabe
- delegierte Vergabe

Bei der zentralen Vergabe gibt es einen Vorsitzenden für die jeweilige Sitzung. Dieser Vorsitzende hat eine höhere Priorität als alle anderen Teilnehmer einer Sitzung. Er entscheidet, wer das Rederecht bekommt und er hat zusätzlich die Möglichkeit anderen das Rederecht auch wieder zu nehmen. Diese Situation entspricht zum Beispiel der eines Professors während einer Vorlesung. Die dezentrale Vergabe gibt den Fall wieder, wenn es keinen Teilnehmer mit einer Priorität gibt die höher als die aller anderen Teilnehmer ist. In solch einem Fall bekommt automatisch der erste Teilnehmer das Rederecht und behält dies solange, bis er es wieder freigibt. Erst dann kann ein nächster Teilnehmer das Rederecht bekommen. In diesem Fall spiegelt sich die Gruppenübung mit Studierenden wieder, bei der jeder gleichberechtigt ist, was das Rederecht angeht. Die delegierte Vergabe gibt den Fall wieder, dass ein Vorsitzender gewählt werden soll. Hierfür findet eine Abstimmung unter den Teilnehmern statt. Bei dieser Abstimmung hängt das Gewicht einer Stimme von der Priorität des jeweiligen

Teilnehmers ab. Nachdem ein Vorsitzender nun gewählt wurde, vergibt dieser, wie bei der zentralen Vergabe, die Rederechte. Dies spiegelt eine Gruppenübung mit einem studentischen Übungsleiter wieder.

Die Applikationsserver und die Applikationsclients bilden die Komponenten, mittels derer der Teilnehmer auf den Vortrag und die Anmerkungen des Vortragenden Einfluss nehmen kann. Bis jetzt existieren folgende Applikationen:

- Whiteboard Applikation
- Chat Applikation
- HiSaP Applikation

Mittels der Whiteboard Applikation ist es möglich, Präsentationsfolien auf allen Geräten der Teilnehmer darzustellen. Sofern eine Berechtigung für Anmerkungen besteht, hat auch jeder Teilnehmer die Möglichkeit von seinem Platz aus Anmerkungen anzufügen.

In der ersten Phase des Projekts war die Chat Applikation als Test des Kommunikationssystems gedacht. In der Zwischenzeit bietet diese aber eine einfache Möglichkeit eine schriftliche Diskussion zwischen den Teilnehmern aufbauen zu können, sofern ein Teilnehmer der Sitzung nicht lokal anwesend ist.

Die dritte Applikation ist die [HiSaP] Applikation. HiSaP ist die Abkürzung für **H**ighly **i**nteractive **S**imulation of **A**lgorithms and **P**rotocols. Das Ziel von HiSaP ist, das Verständnis komplexer Protokolle bzw. Algorithmen in der Lehre sowie beim Design durch geeignete Visualisierung und die Möglichkeit von Interaktionen zu erleichtern. Die HiSaP Applikation bietet den mit Hilfe des [HISAP]-Baukastens erstellten Java Applets und Applikationen die Möglichkeit, die Kontrollmechanismen und die Möglichkeit des Application Sharings von SASCIA mitzubeneutzen. Ein erster Prototyp dieser HiSaP Applikation wird derzeit von Xue [Bai] in der Diplomarbeit „Gemeinsames Lernen von Kommunikationsprotokollen durch elektronisch unterstütztes Rollenspiel“ (2001) entworfen und auch implementiert. Hierbei wird das in der Floor Control von SASCIA zur Verfügung gestellte Gesamtederecht noch in einzelnen Rollenrederechte aufgeteilt. Nötig wurde dies da innerhalb der Java Applets und Applikationen bei der Visualisierung von Protokollen verschiedene Rollen angeboten werden, die von verschiedenen Teilnehmern wahrgenommen werden können.

Implementiert in der ursprünglichen Floor Control war pro Applikation aber nur ein Gesamtederecht.

Die vierte Komponente ist die Datenbank, welche das bei SASCIA verfolgte Ziel der Wiedergabe von Lehrinhalten nach der Lehrveranstaltung unterstützt. Diese Datenbank wird zeitgleich mit dieser Studienarbeit von Andreas [Schmid] in der Studienarbeit „Prototypische Erstellung einer Protokollierung für den Einsatz in Lehrveranstaltungen“ (2001) entworfen und implementiert. Aufgeteilt ist die Datenbank jeweils in einen privaten und einen öffentlichen Bereich. Im öffentlichen Bereich werden alle Daten gespeichert die über das Kommunikationssystem beim einzelnen Teilnehmer ankommen. Welche Daten hierfür in Frage kommen zeigt folgende Auflistung:

- Daten der elektronischen Präsentationsfolien
- Annotationen des Dozenten
- Annotationen anderer Teilnehmer, sofern diese ein Rederecht haben
- Java Applets oder Java Applikationen

Der private Bereich hingegen enthält nur Daten die vom Teilnehmer selbst erzeugt wurden und nicht für die Allgemeinheit bestimmt sind. Durch eine permanente Abspeicherung der Daten inklusive Reihenfolgeindex in einer Datenbank beim einzelnen Teilnehmer wird die Wiedergabe auch nach der Lehrveranstaltung möglich. Selbst eine Wiedergabe in Echtzeit wird durch die Datenbank unterstützt, wenngleich aber noch kein Frontend für den Teilnehmer existiert, um diese Wiedergabe auch wirklich nutzen zu können. Zusätzlich zur Abspeicherung der zur Zeit übermittelten Daten ermöglicht die Datenbank auch die Abfrage bereits übermittelter Daten, um diese auch noch abspeichern zu können. Dies ist vor allem dann von Nutzen wenn ein Teilnehmer einen Vortrag nicht von Beginn an mitverfolgen konnte oder nicht alle Daten von der Datenbank korrekt empfangen werden konnten.

Die fünfte Hauptkomponente ist das Session Management, welches Hauptbestandteil dieser Studienarbeit ist. Durch das Session Management werden die einzelnen Sitzungen die mit SASCIA möglich sind gestartet und auch verwaltet. Hinzu kommt das die Sitzungsverwaltung die Möglichkeit bietet, Sitzungen zu veröffentlichen, so dass Teilnehmer sich über die vorhandenen Sitzungen informieren und sich ohne weitere

Probleme mit einer der Sitzungen verbinden können. Eine weitere wichtige Funktion des Session Managements ist die Authentifizierung der Teilnehmer. Dies ermöglicht die Beschränkung einer Sitzung auf bestimmte Teilnehmer und gewährleistet auch, dass die Teilnehmer die sind für die sie sich ausgeben.

Zur Zeit ist SASCIA nach dem Client-Server-Prinzip strukturiert. Für diesen Aufbau sprechen zwei Gründe. Zum einen ist es durch diesen Aufbau möglich, auf der Serverseite alle Daten des Vortrages zu speichern, und diese dann später zu archivieren, oder an zu spät kommende Studenten zu übermitteln. Zum anderen kann durch diesen Aufbau sichergestellt werden, dass die Floor Control von Anwendungen nicht umgangen werden kann, indem auf Serverseite noch eine weitere Floor Control vorhanden ist, welche das Rederecht eines anfragenden Teilnehmers überprüft. Die Serverkomponenten laufen auf einem Serverrechner. Dieser ist für die Verteilung der einzelnen Daten an die Teilnehmer, und die Kommunikation der Teilnehmer untereinander zuständig. Auf diesem läuft neben dem Session Management Server und dem Kommunikationssystem auch jeweils für jede gestartete Applikation ein sogenannter Applikationsserver. Parallel dazu gibt es bei jedem Teilnehmer den Session Management Client. Um sich bei einer Sitzung anmelden zu können verbindet sich dieser mit dem Session Management Server. Für den eigentlichen Austausch von Daten werden bei den einzelnen Teilnehmern noch die jeweils passenden Applikationsclients benötigt. Diese nehmen dann direkten Kontakt mit den zuständigen Applikationsservern auf und erhalten so die von anderen Teilnehmern gesendeten öffentlichen Daten.



## 2.2 Quellenangaben

SASCIA	Projekt SASCIA (System Architecture Supporting Cooperative and Interactive Applications) Webseite des Projekts SASCIA <a href="http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/Festival/SASCIA.html">http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/Festival/SASCIA.html</a>
FESTIVAL	Projekt Festival (Fertilizing distributEd Systems To support interactIVe AppLications) Webseite des Projekts Festival <a href="http://festival.informatik.uni-stuttgart.de/">http://festival.informatik.uni-stuttgart.de/</a>
VERTEILTE SYSTEME	Abteilung Verteilte Systeme Webseite der Abteilung <a href="http://www.informatik.uni-stuttgart.de/ipvr/vs/">http://www.informatik.uni-stuttgart.de/ipvr/vs/</a>
IPVR	IPVR (Institut für Parallele und Verteilte HöchstleistungsRechner) Webseite des IPVR <a href="http://www.informatik.uni-stuttgart.de/ipvr/ipvr.html">http://www.informatik.uni-stuttgart.de/ipvr/ipvr.html</a>
JSDT	JSDT (Java Shared Data Toolkit) <a href="http://java.sun.com/produkts/java-media/jsdt/">http://java.sun.com/produkts/java-media/jsdt/</a>
HISAP	HiSaP (Highly interactive Simulation of Algorithms and Protocols) <a href="http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/HiSAP/hisap.html">http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/HiSAP/hisap.html</a>
BAI	Xue Bai <i>„Gemeinsames Lernen von Kommunikationsprotokollen durch elektronisch unterstütztes Rollenspiel“</i> Universität Stuttgart, Fakultät Informatik, Diplomarbeit 2001
OBERPARLEITER	Peter Oberparleiter <i>„Vergabe von Zugangsberechtigungen (floor control) für die gemeinsame Nutzung von Smartboards in der Vorlesung“</i> Universität Stuttgart, Fakultät Informatik, Studienarbeit 2001

SCHMID

Andreas Schmid

*„Prototypische Erstellung einer Protokollierung für den  
Einsatz in Lehrveranstaltungen“*

Universität Stuttgart, Fakultät Informatik, Studienarbeit  
2001

### **3. Anforderungsanalyse**

Wie bereits in den vorhergehenden Kapiteln verdeutlicht, soll im Projekt SASCIA ein Konferenzsystem entwickelt werden, das auf die speziellen Gegebenheiten innerhalb Lehrveranstaltungen und Gruppenübungen zugeschnitten ist. In diesem Kapitel werden nun die dafür benötigten Anforderungen herausgestellt. Da es sich bei dieser Studienarbeit um die Sitzungsverwaltungskomponente handelt, beschränkt sich die Untersuchung der Anforderungen hier natürlich auch nur auf die Sitzungsverwaltung und deren Schnittstellen zu den anderen Komponenten im Projekt SASCIA.

Gegliedert werden die einzelnen herausgefundenen Anforderungen nach den Anforderungen, die bei der Eingabe der Sitzungsdaten entstehen, Anforderungen die bei der Benutzerverwaltung wichtig sind, Anforderungen an das Sitzungsverzeichnis und den Abschluss bilden die Anforderungen an die Sitzungsverwaltung.

#### ***3.1 Eingabe der Sitzungsdaten***

Jede Sitzung in SASCIA spiegelt entweder einen Vortrag oder eine Übung im reellen Leben wieder. Um eine Sitzung im SASCIA System starten zu können müssen verschiedene Angaben gemacht werden. Dies dient einerseits dazu, dem System die korrekten Parameter zu übermitteln, als auch dazu, eine Sitzung näher spezifizieren zu können. Die Eingabe der Sitzungsdaten und der Start einer Sitzungsverwaltung auf Serverseite soll für jeden möglich sein. Damit wird erreicht, dass nicht nur Benutzer mit hoher Priorität, wie zum Beispiel Dozenten, eine Sitzung starten können. Somit ist es möglich, dass auch Studierende einen Sitzungsserver starten und gemeinsam in der Gruppe an einem Projekt arbeiten können. Wird aber eine Veröffentlichung der Daten im Sitzungsverzeichnis gewünscht ist eine Authentifizierung nötig. Dies soll eine kontrollierte Eintragung von Sitzungen in das Sitzungsverzeichnis ermöglichen.

Zu Beginn einer Sitzung werden immer alle benötigten Sitzungsdaten abgefragt. Um dem Benutzer sowohl auf Serverseite als auch auf Clientseite diese Eingabe zu erleichtern ist es wünschenswert, dass die Eingabe mancher Sitzungsdaten durch die Festlegung von Standardwerten erleichtert wird. Es sollen hierbei die Parameter mit

Standardwerten belegt werden, die entweder für die Sitzung eine geringe Bedeutung haben oder beim Start einer Sitzung häufig mit dem gleichen Wert belegt werden.

Die Installation der Serversoftware bedeutet immer einen gewissen Aufwand. Zusätzlich kommt hinzu, dass bei Installation der Serversoftware auf mehreren verschiedenen Rechnern immer alle Rechner bei einem Softwareupdate neu installiert werden müssen. Um den Administrationsaufwand bei der Softwareinstallation deshalb minimieren zu können, soll es möglich sein, serverseitig mehrere verschiedene Sitzungen auf nur einem Rechner starten zu können. Um nun von überall den Server starten zu können, sollte die Abfrage der Sitzungsdaten textuell erfolgen. Dies erleichtert den Zugriff mittels bereits bekannter Fernwartungssoftware wie zum Beispiel TELNET.

Bei der Abfrage der Sitzungsdaten ist zu berücksichtigen, dass es die Möglichkeit gibt bestimmte Teilnehmer zu einer Sitzung zuzulassen, oder auch davon auszuschließen. Die Namen der betroffenen Teilnehmer sollten jeweils in Listen aufgeführt werden, die später von der Sitzungsverwaltung zur Überprüfung einer Teilnahmeberechtigung herangezogen werden können.

### **3.2 Sitzungsverzeichnis**

Um sich als Teilnehmer jeweils zur richtigen Sitzung anmelden zu können, werden die dazu passenden Verbindungs- und Sitzungsinformationen benötigt. Erst mit diesen Angaben ist der Teilnehmer in der Lage, sich mit seinem Client auf den richtigen Server verbinden zu können. Damit diese Informationen nicht immer von Hand an die einzelnen Teilnehmer weitergegeben werden müssen, liegt die Einrichtung eines Sitzungsverzeichnis nahe. Dieses Sitzungsverzeichnis sollte alle gestarteten und zur Veröffentlichung freigegebenen Sitzungen der einzelnen Sitzungsserver aufnehmen können. Um dies realisieren zu können, muss das Sitzungsverzeichnis für alle Sitzungsserver und Teilnehmer-Clients ständig verfügbar sein. Die Eingabe der benötigten Daten erfolgt auf dem jeweiligen Sitzungsserver selbst. Somit müssen die Sitzungsdaten nicht zweimal eingegeben werden. Wenn bei der Eingabe der Sitzungsdaten auf dem Sitzungsserver eine Veröffentlichung gewünscht wurde, müssen

die Sitzungsdaten dann auch im Sitzungsverzeichnis veröffentlicht werden. Damit eine kontrollierte Eintragung von Sitzungen auf dem Sitzungsverzeichnis zu gewährleistet werden kann, ist es auch wichtig, dass die Eintragung von Sitzungen auf dem Sitzungsverzeichnis durch Passwortschutz gesichert wird. Für die einzelnen Teilnehmer ist die Möglichkeit zu implementieren, sich eine Gesamtliste aller vorhandenen Sitzungen anzeigen zu lassen. Um den Teilnehmern das Auffinden einer bestimmten Sitzung zu vereinfachen, sollte die Anzeige der Sitzungen nach den Sitzungsnamen sortiert erfolgen. Gleiche Namen für eine Sitzung sind hier erlaubt. Die Unterscheidung erfolgt dann hier über die weiteren Angaben die zu einer Sitzung gemacht wurden. Wünschenswert wäre auch, dass der Teilnehmer die ausgegebene Liste mit Sitzungen nach anderen Kriterien sortieren kann. Denkbar wäre hier zum Beispiel zusätzlich eine Sortierung nach den Sitzungsorten oder nach Dozenten.

### **3.3 Benutzerverwaltung**

Neben den sonst üblichen Parametern wie Benutzername und Passwort, die dafür benötigt werden, um einen Teilnehmer zu authentifizieren, der an einer Sitzung teilnehmen möchte, muss zusätzlich noch eine Priorität des einzelnen Benutzers vorhanden sein. Diese Priorität dient dann der Ermittlung des Rederechts durch die Floor Control innerhalb einer Sitzung. Gibt es einen Teilnehmer innerhalb einer Sitzung der eine höhere Priorität als alle anderen besitzt, ist dieser automatisch der Vorsitzende der Sitzung und hat die Möglichkeit das Rederecht an andere zu vergeben. Weitere Funktionen in Hinblick auf die Priorität in Zusammenarbeit mit der Floor Control werden in der bereits in Kapitel zwei erwähnten Studienarbeit „Vergabe von Zugangsberechtigungen (floor control) für die gemeinsame Nutzung von Smartboards in der Vorlesung“ von Peter Oberparleiter dargestellt.

Im Alltag kommt es häufig vor, dass ein Benutzer an mehreren verschiedenen Sitzungen teilnehmen will. Dies würde bedeuten, dass der Benutzer in jeder der lokalen Benutzerdatenbanken auf den einzelnen Rechnern auf denen eine Sitzung läuft vorhanden sein muss. Um die Benutzerverwaltung hier nun ein wenig zu vereinfachen, wäre es wünschenswert, wenn die Möglichkeit besteht einen Benutzer in einer zentralen Benutzerdatenbank nur einmal eintragen zu müssen, anstatt in jeder der lokalen Benutzerdatenbanken. Dieser Eintrag ist dann automatisch bei allen verschiedenen

Sitzungen gültig. Diese zentrale Datenbank sollte von allen SASCIA Sitzungsservern aus erreichbar und rund um die Uhr verfügbar sein.

Um bei der zentralen Benutzerdatenbank den Administrationsaufwand zusätzlich zu minimieren, kann von dieser noch eine Verbindung zum fakultätsweiten Accountingsystem implementiert werden. Dieses ist ein Dienst, der an der Fakultät für die Authentifizierung der einzelnen Rechner- und Modembenutzer zuständig ist.

Da das fakultätsweite Accountingsystem aber nur die folgenden Daten:

- Benutzername
- Passwort

zur Verfügung stellt, muss in der zentralen Benutzerdatenbank noch die statische Priorität für die einzelnen Benutzer mit aufgenommen werden. Um dies zu erleichtern, sollten alle Benutzer die nur im fakultätsweiten Accountingsystem vorhanden sind, automatisch eine Standardpriorität zugewiesen bekommen. Eine Änderung der Benutzerdaten oder der standardmäßig zugeordneten Priorität sollte entweder direkt in der zentralen oder der lokalen Benutzerdatenbank durch einen speziellen Administrator möglich sein.

Die Benutzerdatenbank spielt eine zentrale Rolle. Sie dient zur Unterstützung des Anmeldevorgangs von Teilnehmern bei einer einzelnen Sitzung. Mittels der Benutzerdatenbank wird sichergestellt, dass nur berechtigte Personen zu einer Sitzung zugelassen werden. Aus diesem Grund muss die Benutzerdatenbank den anderen Komponenten des SASCIA Projekts eine Schnittstelle zur Authentifizierung von Teilnehmern bieten.

### **3.4 Sitzungsverwaltung**

Die Sitzungsverwaltung auf Serverseite dient dazu eine Sitzung zu starten. Normalerweise wird eine Sitzung auch sofort, nachdem alle Sitzungsdaten erfasst worden sind, gestartet. Wenn eine Veröffentlichung der Sitzung bei der Abfrage der

Sitzungsdaten gewünscht wurde, wird die Sitzung auch automatisch im Sitzungsverzeichnis veröffentlicht. Um nun aber auch zukünftige Sitzungen bereits vorher im Sitzungsverzeichnis ankündigen zu können wäre es wünschenswert, wenn die Sitzungsverwaltung auf Serverseite nach der Abfrage der Sitzungsdaten auch einen Start der Sitzung zu einem späteren Zeitpunkt ermöglichen könnte. Somit kann ein Rechner für eine Sitzung zu einem bestimmten Zeitpunkt reserviert werden und gleichzeitig eine Sitzung im Sitzungsverzeichnis angekündigt werden.

Eigentlich werden die Daten, die zur Verbindung eines Clients mit einem Sitzungsserver benötigt werden, über das Sitzungsverzeichnis veröffentlicht. Sofern dieses Sitzungsverzeichnis aber nicht zur Verfügung steht oder eine Veröffentlichung nicht gewünscht wird, ist es sinnvoll wenn auf Serverseite nach Start des Servers diese Daten ausgegeben werden. Somit ist es möglich, dass sich Clients mittels manueller Angabe der Verbindungsdaten auf den Server verbinden können. Dies ist vor allem dann sinnvoll, wenn Studierende in einer geschlossenen Benutzergruppe an einem Projekt zusammen arbeiten wollen und eine Veröffentlichung nicht gewünscht wird, oder keiner der Beteiligten eine Berechtigung besitzt, die zur Veröffentlichung benötigt wird,.

Neben der Veröffentlichung einer Sitzung und der Ausgabe der Serveradresse muss die Sitzungsverwaltung auch alle gewünschten Applikationen starten. Sobald nun eine Sitzung verlassen wird, müssen alle Applikationen auch wieder beendet werden. Gleichzeitig müssen auch die Protokollierungsdaten durch die Applikationen selbst unter dem beim Start angegebenen Dateinamen gespeichert werden. Aber nicht nur beim Verlassen einer Sitzung müssen Applikationen beendet werden. Auch dann, wenn dem Teilnehmer Client ein Fehler durch eine Applikation gemeldet wird, muss diese beendet und automatisch wieder neu gestartet werden können. Ein Neustart sollte hier aber nur in Erwägung gezogen werden, wenn anhand der Fehlermeldung ein Neustart überhaupt sinnvoll erscheint. Zusätzlich soll dem Teilnehmer auch die Möglichkeit gegeben werden, eventuell auch nur einige Applikationen beenden zu können, aber weiterhin an der Sitzung mit den verbleibenden Applikationen teilnehmen zu können. Dies ist zum Beispiel dann sinnvoll, wenn die Hardwarekapazität des Teilnehmerrechners nicht ausreicht alle benötigten Applikationen auch ausführen zu können.

Weiterhin ist es wünschenswert, eine Sitzung nur für einen bestimmten Personenkreis zuzulassen. Aus diesem Grund müssen von der Sitzungsverwaltung nach der Authentifizierung auch die Listen der erlaubten und nicht erlaubten Teilnehmer herangezogen werden. Erstellt wurden diese bereits bei der Abfrage der Sitzungsdaten. Dadurch kann festgestellt werden, ob die Sitzung für alle oder nur für einen bestimmten Personenkreis aus der Benutzerdatenbank zur Verfügung steht. Manchmal kommt es vor, dass interessante Vorträge sehr gut besucht werden. Aus diesem Grund ist es wichtig, dass es für den Dozenten zusätzlich noch die Möglichkeit gibt eine Sitzung generell für weitere Teilnehmer zu sperren, um genügend Übertragungskapazität für die bereits angemeldeten Teilnehmer zu haben. Bei dieser Art der Sperrung bleiben die bereits angemeldeten Teilnehmer in der Sitzung. Es gibt für neue Teilnehmer dann aber keine Möglichkeit mehr, sich zur Sitzung anzumelden, selbst wenn Sie eigentlich an der Sitzung teilnehmen dürften. Sofern sich ein Teilnehmer nicht an die Regeln hält, muss es dem Dozenten auch möglich sein, einen bestimmten Teilnehmer aus der Sitzung auszuschließen. Dieser Ausschluss des Teilnehmers muss solange anhalten, bis der Dozent den Teilnehmer entweder wieder freigibt oder die Sitzung beendet und dann wieder neu gestartet wird.

Nach dem Start der Sitzung, welcher entweder direkt am Rechner auf dem der Server installiert ist, oder auch per Fernwartung erfolgt sein kann, sitzt auch der Dozent wie alle anderen Teilnehmer einer Sitzung an seinem Teilnehmer-Client. Damit dieser nun die Verwaltung einer laufenden Sitzung nicht immer per Fernwartung wie beim Start des Servers durchführen muss, ist es wichtig, dass der Dozent die Möglichkeit erhält die Verwaltung bequem von seinem eigenen Rechner aus regeln zu können. Die Funktionen zur Regelung einer Sitzung sollten in einem speziellen Client zusammengefasst werden, der zwar für jeden Teilnehmer startbar, aber nur durch den Sitzungsvorsitzenden bedienbar ist. Damit ist sichergestellt, dass nur berechtigte Personen Zugriff auf die Verwaltung der Sitzung haben.

Durch die Client Server Architektur des System muss, bevor ein Teilnehmer an einer Sitzung bei einem Sitzungsserver teilnehmen kann, eine Verbindung zwischen Teilnehmer Client und Sitzungsserver aufgebaut werden. Diese Verbindung wird mittels des Kommunikationssystems hergestellt. Da beim Start des Servers entschieden



werden kann, welche Applikationen benutzt und somit auf Serverseite auch gestartet werden, ist es für die Sitzungsverwaltung auf Clientseite wichtig über die Verbindung eine Liste der auf Serverseite gestarteten Applikationen zu bekommen. Anhand dieser Liste werden dann die darauf angegebenen und auf Clientseite vorhandenen Applikationen zu Beginn gestartet. Somit wird gewährleistet, dass beim Benutzer immer genau die gleichen Applikationen gestartet werden wie auf dem Server. Dies bedeutet aber nicht zwangsläufig, dass auf Clientseite auch wirklich automatisch alle Applikationen laufen. Aufgrund Fehler kann es sein, dass nicht alle geforderten Applikationen gestartet werden können. Hier kommt dann die bei der Sitzungsverwaltung auf Serverseite geforderte Fehlerbehandlung zum Zug.

Zur allgemeinen Information sollte die Sitzungsverwaltung auf Clientseite dem Nutzer auch eine Liste der einzelnen Teilnehmer der Sitzung anzeigen. Diese erhält die Sitzungsverwaltung auf Clientseite direkt von der Sitzungsverwaltung auf Serverseite. Interessant ist diese Liste aus diesem Grund, weil sie alle Teilnehmer zeigt die sich zur Sitzung angemeldet haben. Im Vergleich mit den Listen innerhalb den einzelnen Applikationen lässt sich hier herausfinden wer zwar angemeldet ist, aber nicht alle Applikationen gestartet hat. In solch einem Fall kann meist auf ein technisches Problem auf Seiten des Teilnehmers geschlossen werden, da nach der Anmeldung eines Teilnehmers eigentlich alle Applikationen aufgrund der übermittelten Applikationsliste gestartet sein müssten. Somit lässt sich leicht herausfinden, welcher Teilnehmer, sofern er die Applikation nicht absichtlich beendete, noch Probleme hat an der Sitzung teilzunehmen.

Sofern die jeweiligen Applikationen dies unterstützen, werden alle Daten einer Sitzung durch die Applikationen nach Beendigung permanent in die beim Start der Sitzungsverwaltung auf Clientseite angegebene Datei abgespeichert. Um nun eine Offline Betrachtung von bereits erhaltenen Daten durchführen zu können, ist es auch wichtig, dass beim Start der Sitzungsverwaltung auf Clientseite die Möglichkeit gegeben wird, zwischen einer Online- (direkte Verbindung zu einem Sitzungsserver) und einer Offlinesitzung (keine Verbindung zu einem Sitzungsserver) auswählen zu können. Bei der Onlinesitzung muss direkt eine Verbindung mit dem Sitzungsverzeichnis aufgebaut und eine laufende Sitzung ausgewählt werden oder manuell eine Adresse des Sitzungsservers eingegeben werden. Bei einer Offlinesitzung

hingegen wird die Möglichkeit gegeben, auf bereits abgespeicherte Sitzungen zuzugreifen. Diese abgespeicherten Sitzungen können entweder aus eigenen Protokollierungen von besuchten Sitzungen oder aus einem Archiv stammen.

## **4. Architektur**

Nachdem im vorhergehenden Kapitel ausführlich die Anforderungen an die Sitzungsverwaltung herausgestellt wurde, soll nun in diesem Kapitel die Architektur der Sitzungsverwaltung auf Server- und danach auf Clientseite erläutert werden.

### **4.1 Server Architektur**

Hier wird nun näher auf die Architektur auf der Serverseite eingegangen. Die einzelnen Kapitelüberschriften geben jeweils den Namen einer logischen Programmeinheit wieder. Die im vorherigen Kapitel herausgefundenen Anforderungen werden hier in folgende Programmeinheiten aufgeteilt:

- Lokaler Administrationsclient
- Benutzerdatenbank
- Sitzungsverzeichnis
- Sitzungsverwaltung (Serverseite)
- Sitzungsverwaltung Administrationsserver
- Serverkonfiguration

#### **4.1.1 Lokaler Administrationsclient**

Um eine Sitzung im SASCIA System korrekt starten zu können, müssen beim Start des SASCIA Systems verschiedene Daten abgefragt werden, die für den reibungslosen Betrieb des Session Management nötig sind. Hier werden nun die Parameter aufgeführt die für eine Sitzung benötigt werden, und somit zu Beginn abgefragt werden müssen:

- Sitzungsname
- Dozent/Vorsitzender
- Sitzungsort
- Sitzungsdatum
- Sitzungsbeginn
- Sitzungsbeschreibung
- Soll Anmeldung am Sitzungsverzeichnis erfolgen?

- Liste von erlaubten Benutzern
- Liste von nicht erlaubten Benutzern
- Liste der zu startenden Applikationen
- Dateiname für Protokolldateien

Diese Liste der Parameter lässt sich in dringend benötigte Parameter und optionale Parameter aufteilen. Die Liste der dringend benötigten Parameter lautet wie folgt:

- Liste der zu startenden Applikationen
- Dateiname für Protokolldatei

Alle anderen Parameter sind optional und werden daher nicht unbedingt für den Start oder den Betrieb benötigt. Um eine genauere Beschreibung einer Sitzung zu erhalten, oder um zu wissen ob die Zusatzdienste wie das Sitzungsverzeichnis (Session Directory) benutzen werden sollen, wäre es gut wenn diese vorhanden sind. Die in der Diskussion herausgefundenen Standardwerte für einige der Angaben sind in folgender Tabelle aufgeführt.

Parameter	Standardwert
Sitzungsdatum	Heute
Sitzungsbeginn	Sofort
Anmeldung am Sitzungsverzeichnis?	Nein
Liste von erlaubten Benutzern	Alle
Liste von nicht erlaubten Benutzern	Leer
Liste der zu startenden Applikationen	Alle vorhandenen
Dateiname für Protokolldatei	Standardpfad und Standarddateiname

**Abbildung 3: Standardwerte für Sitzungsparameter**

Um diese Sitzungsdaten erfassen zu können wird ein Abfragemechanismus benötigt. Dieser Mechanismus wird in der Sitzungsverwaltung Local Administration Client genannt.

Da der Local Administration Client lediglich zur Erfassung der Sitzungsdaten und zum Start der Sitzungsverwaltung benötigt wird, fallen hier wenig Funktionen an. Benötigt werden hier folgende Funktionen:

- Abfrage der Sitzungsdaten
- Start der Sitzungsverwaltung

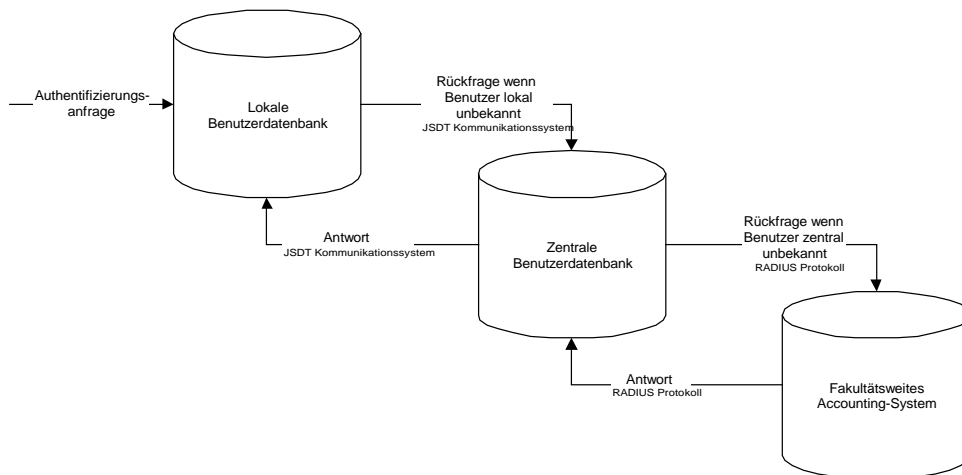
Die Sitzungsdaten werden dann per Parameter beim Start der Sitzungsverwaltung an diese übergeben. Weitere Funktionen wie zum Beispiel Änderung oder Beendigung einer Sitzung werden dann direkt mittels dem in Abschnitt 4.2.2 beschriebenen Session Management Remote Admin Client vorgenommen.

### **4.1.2 Benutzerdatenbank**

Die Benutzerdatenbank dient im Projekt SASCIA zur Authentifizierung von neuen Teilnehmern einer Sitzung und wird in drei in sich abgeschlossene Teile aufgeteilt. Folgende drei Benutzerdatenbanken existieren:

- Lokale Benutzerdatenbank auf dem Sitzungsserver
- Zentrale Benutzerdatenbank (eventuell auf dem Sitzungsverzeichnisserver)
- Fakultätsweites Accountingsystem

Wie diese Datenbanken voneinander abhängen zeigt folgendes Schaubild:



**Abbildung 4: Ablauf einer Authentifizierungsanfrage**

Wie das Schaubild verdeutlicht, handelt es sich hierbei um ein mehrstufiges Modell von Datenbanken. Von der Sitzungsverwaltung auf dem Sitzungsserver wird immer nur die lokale Benutzerdatenbank abgefragt. Wird dort ein Benutzer nicht gefunden, so fragt die lokale Benutzerdatenbank bei der zentralen Benutzerdatenbank an. Wenn auch die zentrale Benutzerdatenbank den Benutzer nicht finden kann, so fragt die zentrale Benutzerdatenbank über das [RADIUS] Protokoll das fakultätsweite Accountingsystem ab. Rückwärts werden die Antworten über die einzelnen Datenbanken zur Sitzungsverwaltung zurückgeleitet. Dieses System gleicht dem System beim DNS (Domain Name System) Dienst aus dem Internet. Hier werden Internetnamen in IP-Adressen übersetzt. Die jeweilige Übersetzung für alle Namen ist auch nicht überall auf den lokalen Nameservern vorhanden. Wenn ein lokaler Nameserver die Übersetzung nicht kennt, fragt er den nächsten Nameserver und wenn der diese auch nicht kennt, fragt der wieder einen nächsten. Sobald ein Nameserver die Übersetzung kennt, wird das Ergebnis durch die Kette der Nameserver wieder zurückgegeben. Wenn keiner die Übersetzung kennt, wird ein Fehler über die Kette der Nameserver zurückgegeben.

Damit die Benutzerdatenbanken die Aufgabe der Authentifizierung auch erfüllen können, müssen in den Benutzerdatenbanken folgende Daten abgespeichert werden können:

- Benutzername
- Passwort
- Statische Priorität

Die Benutzerdatenbank welche im SASCIA Projekt auch User Database genannt wird, bietet zum einen alle für die Benutzerverwaltung nötigen Funktionen. Zum anderen werden alle für die Authorisierung eines neuen Teilnehmers nötigen Funktionen hier angeboten. Die einzelnen Funktionen die bei der User Database benötigt werden lauten im einzelnen wie folgt:

- Anlegen einer lokalen User Database
- Eintragung neuer Benutzer
- Löschung von Benutzern
- Änderung von Benutzern
- Auflistung aller Benutzer
- Authentifizierung eines Benutzers (mittels Benutzername und Passwort)
- Abfrage der Priorität

Die oben aufgeführten Funktionen können nur über die Sitzungsverwaltung direkt aufgerufen werden. Wenn andere Teile des Projektes SASCIA Zugang zu diesen Funktionen benötigen, wird dieser nur über die Sitzungsverwaltung gewährt.

Neben diesen eher externen Funktionen benötigt die User Database noch interne Funktionen. Diese werden vor allem für die Verbindung zur zentralen User Database und deren Abfrage benötigt. Im einzelnen sind dies folgende Funktionen:

- Beitritt zum Kanal der zentralen User Database
- Rückfrage bei zentraler User Database für Authorisierung, falls User nicht lokal vorhanden ist.

Neben diesen Funktionen der lokalen User Database auf dem Sitzungsserver selbst gibt es noch die Funktionen der zentralen User Database, welche ja als Ergänzung der lokalen User Database gesehen wird. Diese zentrale Datenbank bietet prinzipiell die gleichen Funktionen wie die oben beschriebene lokale User Database auf dem Sitzungsserver. Allerdings werden hier einige Funktionen im Vergleich zu den Funktionen bei der lokalen User Database modifiziert. Diese Modifizierungen lauten:

- Anlage eines Kanals für die Kommunikation zwischen lokaler und zentraler User Database (ersetzt Funktion "Beitritt zum Channel")
- Rückfrage bei Accounting Dienst der Fakultät über Radius Protokoll (ersetzt Funktion "Rückfrage bei zentraler User Database")

Wie bereits bei den Anforderungen beschrieben, liegen beim Accounting Dienst der Fakultät zwar der Benutzername und ein Passwort vor, allerdings fehlt dort die Angabe der Priorität eines Benutzers. Aus diesem Grund ist es erforderlich, dass bei der Rückfragefunktion der zentralen User Database, beim Accounting Dienst für die Priorität ein Standardwert gesetzt wird.

### **4.1.3 Sitzungsverzeichnis**

Indem man auf verschiedenen Servern jeweils einen Sitzungsserver startet besteht die Möglichkeit eine Vielzahl von verschiedenen Sitzungen zu erzeugen. Dies kann auch gleichzeitig geschehen. Damit ein Teilnehmer die nötigen Informationen zu einer Sitzung bekommt, müssen mittels des Session Directory folgende Daten öffentlich verfügbar gemacht werden:

- Sitzungsname
- Dozent
- Sitzungsort
- Sitzungsdatum
- Sitzungsbeginn
- Sitzungsbeschreibung
- Vorsitzender
- Adresse des Sitzungsservers



Diese Parameter lassen sich auch wieder in dringend benötigte und optionale Parameter unterteilen. Zu den dringend benötigten Parametern, die für den Aufbau einer Verbindung zwischen einem Teilnehmer Client und einem Sitzungsserver wichtig sind, zählen folgende Parameter:

- Sitzungsname
- Adresse des Sitzungsserver

Der Ablauf bei einer Veröffentlichung einer Sitzung im Sitzungsverzeichnis wird in folgendem Schaubild nochmals genauer verdeutlicht.



**Abbildung 5: Ablauf einer Veröffentlichung im Sitzungsverzeichnis**

Das Sitzungsverzeichnis, in der Sitzungsverwaltung Session Directory genannt, besteht aus zwei Teilen. Der erste Teil befindet sich auf dem Rechner auf dem die Sitzung gestartet wird, der zweite Teil befindet sich auf dem zentralen Rechner auf dem der Session Directory Dienst bereitgestellt wird. Um die zwei Teile voneinander auch vom Namen her unterscheiden zu können, wird der Teil auf dem zentralen Rechner Session Directory (Server) und der Teil auf dem Sitzungsserver Session Directory (Client) genannt.

Die zwei Teile kommunizieren über einen eigenen Kanal, der vom Session Directory (Server) auf dem zentralen Rechner zur Verfügung gestellt wird. Geöffnet wird dieser Kanal beim Start des Session Directory. Für den Eintritt in den Kanal ist eine Authentifizierung erforderlich. Somit wird gewährleistet, dass nur berechtigte Personen die Möglichkeit zur Veröffentlichung einer Sitzung im Session Directory haben.

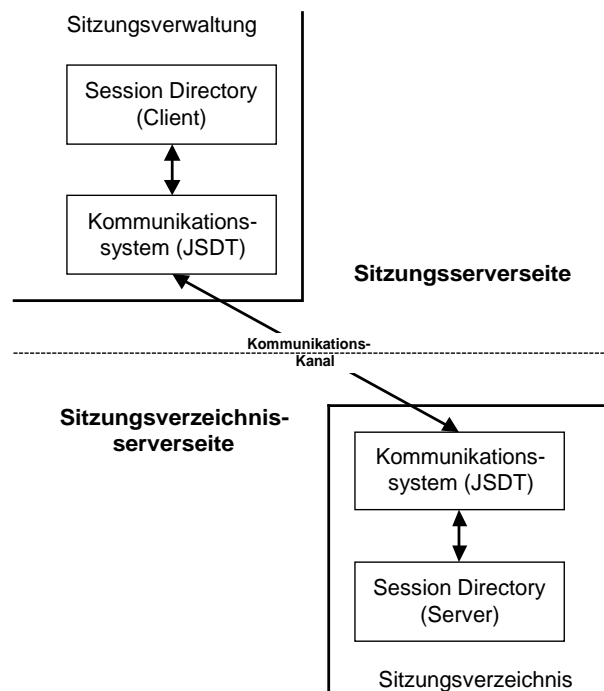
Um auf dem zentralen Rechner auf dem das Session Directory laufen soll überhaupt einen Server starten zu können, wird hier ebenfalls das Kommunikationssystem benötigt.

Die Hauptaufgabe des Session Directory (Client) besteht darin, dem vom Session Directory (Server) eröffneten Kanal beizutreten und bestimmte Funktionen des Session Directory (Server) aufzurufen. Dieser Aufruf erfolgt jeweils über einen Austausch von Nachrichten.

Um die vom Session Directory (Client) erhaltenen Anfragen bearbeiten zu können, stellt das Session Directory (Server) folgende Funktionen zur Verfügung:

- Eintragung einer Session
- Änderung von Sessiondaten
- Löschung einer Session
- Auflistung aller vorhandenen Sessions

Auch hier sind die Funktionen des Session Directory (Client), die auf der Seite des Sitzungsservers aufrufbar sind, nicht direkt von anderen Teilen des SASCIA Projekts aufrufbar. Vielmehr muss auch hier der Weg über die Sitzungsverwaltung gewählt werden.



**Abbildung 6: Komponenten des Sitzungsverzeichnisses**

#### 4.1.4 Sitzungsverwaltung Serverseite

Sobald eine Sitzung auf dem Sitzungsserver nach der Eingabe der Sitzungsdaten gestartet wird, wird auch eine, nur für diese Sitzung zuständige Sitzungsverwaltung gestartet. Informationen und Daten für diese Sitzungsverwaltung stammen entweder aus der Server Konfiguration oder aus den Eingaben der Sitzungsdaten. Die Aufgaben der Sitzungsverwaltung sind vielfältig. Die wichtigsten Aufgaben der Sitzungsverwaltung werden hier kurz aufgelistet:

- Start des Kommunikationssystems
- Verbindung mit dem Sitzungsverzeichnis (wenn gewünscht)
- Eintragung/Austragung der Sitzung im Sitzungsverzeichnis (wenn gewünscht)
- Ausgabe der Serveradresse
- Start eines Kanals für Kommunikation zwischen Sitzungsverwaltung und Teilnehmer Clients
- Start/Beenden der benötigten Applikationen
- Benutzerauthentifizierung

Wählt ein Benutzer eine bestimmte Session aus, so wird er über das Kommunikationssystem mit der Sitzungsverwaltung des zuständigen Sitzungsservers verbunden. Diese Verbindung läuft über den bereits vom Kommunikationssystem geöffneten Kommunikationskanal. Nun muss die Sitzungsverwaltung, in Zusammenarbeit mit dem Kommunikationssystem und der Benutzerdatenbank, die Berechtigung des anfragenden Teilnehmer Clients überprüfen. Dies geschieht über eine Benutzernamen- und Paßwortabfrage. Zur Verifizierung der Anfragen wird dann von der Sitzungsverwaltung die Benutzerdatenbank herangezogen.

Der Session Management Server ist die Komponente, welche die Sitzungsverwaltung auf Serverseite wiederspiegelt. Er wird als zentraler Anlaufpunkt angesehen und dient gleichzeitig als alleinige Schnittstelle zum Session Management und dessen einzelnen Funktionen im gesamten SASCIA Projekt. Vom Local Administration Client erhält der Session Management Server alle wichtigen Informationen, um eine Sitzung und die dafür benötigten Applikationsserver starten zu können. Einige wichtige Funktionen die direkt vom Session Management angeboten werden sind folgende:

- Start eines Kanals für Kommunikation zwischen Sitzungsverwaltung und Teilnehmer-Clients
- Start/Beenden des Kommunikationssystems
- Start/Beenden der Applikationsserver
- Übermittlung der gestarteten Applikationen
- Neustart von mit Fehlern beendeten Applikationen.

Neben diesen Funktionen bietet der Session Management Server noch weitere Funktionen an. Dies liegt an der Konzeption, dass das Session Management als zentrales Objekt gedacht ist. Somit haben alle anderen Teile des Projekts die Möglichkeit sich nur an ein Objekt wenden zu können, welches Ihnen alle Dienstleistungen anbietet. Um dies zu ermöglichen, greift der Session Management Server dann wiederum auf die oben beschriebenen Funktionen der einzelnen anderen Teile der Sitzungsverwaltung zu. Die Liste der weiteren Funktionen ergibt sich somit leicht aus den einzelnen Funktionen der oben beschriebenen Komponenten. Hier sind diese zur besseren Übersicht alle nochmals aufgeführt:

- Anlegen einer lokalen User Database
- Eintragung neuer Benutzer in User Database
- Löschung von Benutzern in User Database
- Änderung von Benutzern in User Database
- Auflistung aller Benutzer in User Database
- Authentifizierung eines Benutzers (mittels Benutzername und Passwort)
- Abfrage der Priorität in der User Database
- Abfrage der aktuellen Serveradresse
- Verbindung mit dem Session Directory
- Eintragung einer Session in Session Directory
- Änderung von Sessiondaten im Session Directory
- Löschung einer Session im Session Directory
- Auflistung aller vorhandenen Sessions laut Session Directory

#### **4.1.5 Sitzungsverwaltung Administrationsserver**

Nachdem eine Sitzung auf dem Sitzungsserver gestartet wurde ist mittels dem durch die Sitzungsverwaltung gestarteten Kommunikationssystem ein Zugriff von jedem Teilnehmer Client aus möglich. Aus diesem Grund besteht ab diesem Zeitpunkt die Möglichkeit eine Sitzung auch direkt von seinem Platz aus steuern zu können, ohne sich erst umständlich mittels einem Fernwartungsprogramm auf dem Sitzungsserver einwählen zu müssen. Um diese Steuerung vom Platz eines jeden Teilnehmers aus möglich zu machen bedarf es auf Serverseite der Implementierung eines Administrationsservers. Dieser Administrationsserver für die Sitzungsverwaltung, im Projekt SASCIA auch Session Management Admin Server genannt, ist als Applikationsserver implementiert und läuft wie alle anderen Applikationsserver direkt auf dem Sitzungsserver. Der Session Management Admin Server öffnet über das Kommunikationssystem einen Kanal. Über diesen kann der im Client Bereich dieses Kapitels beschriebene Session Management Remote Admin Client seine Anfragen mittels Nachrichtenaustausch an den Session Management Admin Server stellen. Durch die Realisierung als Applikation wird der Zugriff auf diesen Admin Server durch die in SASCIA implementierte Floor Control geregelt. Somit wird garantiert, dass eine Sitzung nur durch die dazu berechnigte Person verwaltet werden kann.

Da im weiteren Verlauf des Textes der Session Management Remote Admin Client noch genauer mit all seinen Funktionen vorgestellt wird, wird hier an dieser Stelle auf eine Vorstellung der einzelnen Funktionen des Session Management Admin Server verzichtet. Dies liegt daran, dass der Session Management Admin Server lediglich das ausführende Organ für die Funktionen des Remote Admin Clients ist. Der Session Management Admin Server dient sozusagen als Schnittstelle zwischen Client und Sitzungsverwaltung, um eine Sitzung vom Client aus administrieren zu können.

#### **4.1.6 Serverkonfiguration**

Die Serverkonfiguration enthält alle persistenten Konfigurationsdaten die für den Betrieb der Sitzungsverwaltung nötig sind. Dies bedeutet, dass alle Daten die in der Serverkonfiguration abgelegt werden, dauerhaften Bestand haben. Die Daten sind somit auch nach Beendigung eines Sitzungsservers noch verfügbar. Die Serverkonfiguration wird beim Start der Sitzungsverwaltung eingelesen und vor dem Ende der

Sitzungsverwaltung wieder in eine Datei zurückgeschrieben, und sollte folgende Daten abgespeichert haben:

- Adresse des Sitzungsverzeichnis
- Adresse der zentralen Benutzerdatenbank
- maximal verfügbare Applikationen
- Defaultwerte laut Abschnitt 4.1.1

Die Adresse des Sitzungsverzeichnis ist vor allem dann wichtig, wenn eine Sitzung im Sitzungsverzeichnis veröffentlicht werden soll. Wird eine Veröffentlichung gewünscht, so wird diese Adresse benötigt, um sich mit dem Sitzungsverzeichnis über das Kommunikationssystem zu verbinden. Ohne diese Angabe ist eine Veröffentlichung der Sitzung nicht möglich. Da auch die zentrale User Database auf einem separaten, immer erreichbaren, Rechner installiert ist, muss auch hier die Adresse dieses Rechners in der Serverkonfiguration abgelegt sein. So ist es möglich, beim Start der Sitzungsverwaltung der lokalen Benutzerdatenbank mitzuteilen, wo diese nach weiteren Teilnehmern suchen kann, wenn ein Teilnehmer nicht gleich in der lokalen Benutzerdatenbank gefunden wird. Neben der Adresse des Sitzungsverzeichnis und der zentralen Benutzerdatenbank sollten in der Server Konfiguration auch alle Applikationen aufgeführt werden, die auf dem Sitzungsserver maximal zur Verfügung stehen. Aus diesen Applikationen kann bei der Initialisierung einer neuen Sitzung dann ausgewählt werden, welche speziell benutzt werden sollen.

Neben diesen Werten, werden hier auch die Standardwerte für:

- Sitzungsdatum
- Sitzungsbeginn
- Anmeldung am Sitzungsverzeichnis?
- Liste von erlaubten Benutzern
- Liste von nicht erlaubten Benutzern
- Liste der zu startenden Applikationen
- Dateiname für die Protokolldatei

abgespeichert. Der Standardwert für den jeweiligen Parameter wurde ja bereits in Abschnitt 4.1.1 aufgeführt.

#### 4.1.7 Ablauf beim Start einer Sitzung auf Serverseite

Um die einzelnen Schritte beim Start einer Sitzung auf Serverseite deutlich zu machen, werden in folgendem Schaubild diese der Reihe nach aufgeführt.

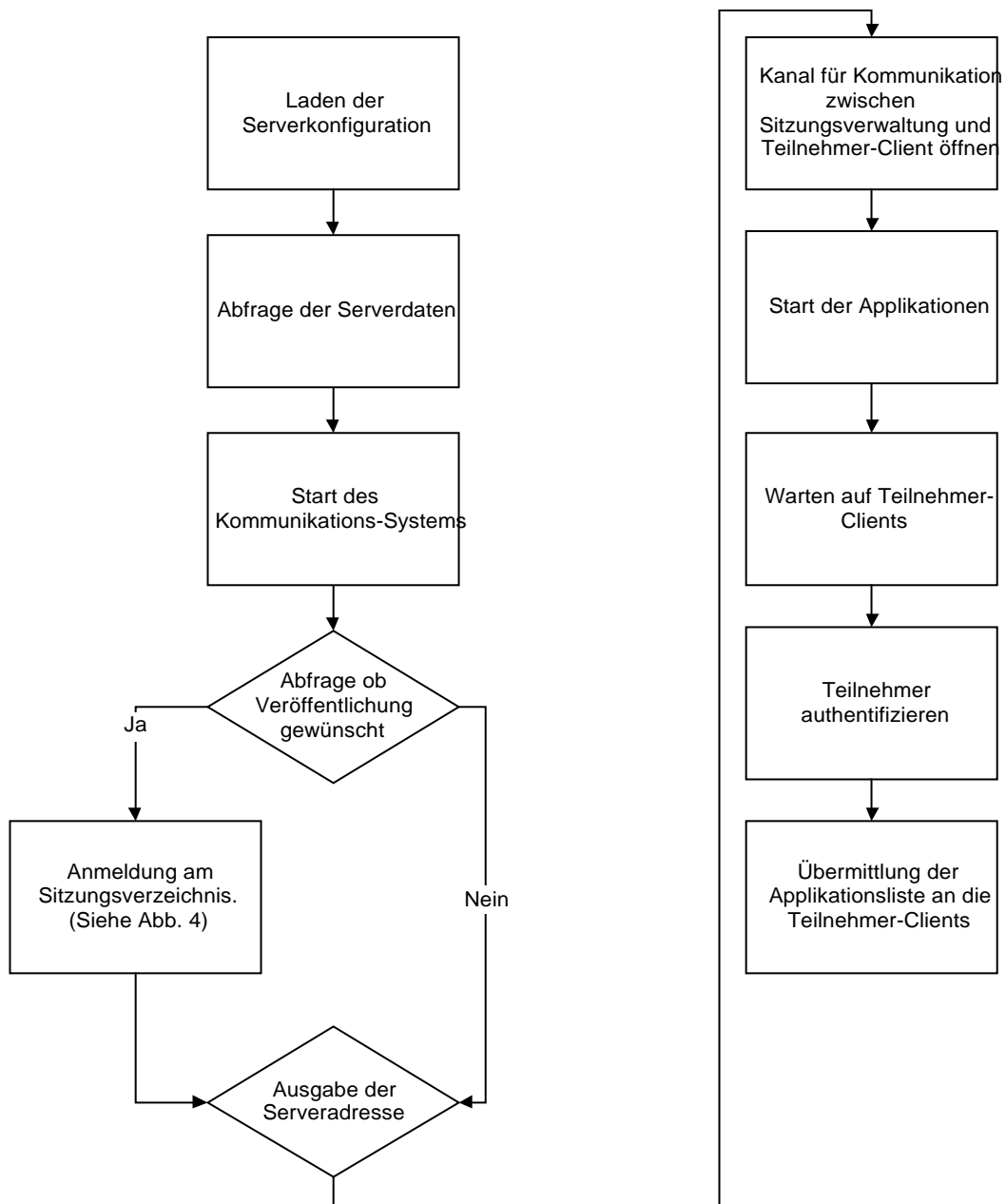


Abbildung 7: Ablauf beim Start einer Sitzung auf Serverseite

## **4.2 Client Architektur**

Nachdem im vorherigen Abschnitt die Architektur und der Aufbau der Sitzungsverwaltung auf der Serverseite am Sitzungsserver genauer erläutert wurde, wird hier nun die Architektur und der Aufbau auf Clientseite näher veranschaulicht. Hier wurden die Anforderungen in folgende logische Programmeinheiten aufgeteilt:

- Sitzungsverwaltung Clientseite
- Sitzungsverwaltung Administrationsclient

### **4.2.1 Sitzungsverwaltung Clientseite**

Um sich als Teilnehmer zu einer Sitzung anmelden zu können, müssen zuerst auf Clientseite folgende Parameter abgefragt werden:

- Benutzername
- Passwort
- manuelle Sitzungserveradresse
- Dateiname für Protokolldatei
- Pseudonym
- Sitzplatz
- Online-Präsentation oder Offline-Präsentation (nicht im Prototyp)

Wichtige und somit für eine Verbindung zum Sitzungsserver und für die Abspeicherung der Daten unbedingt benötigte Parameter sind hier folgende:

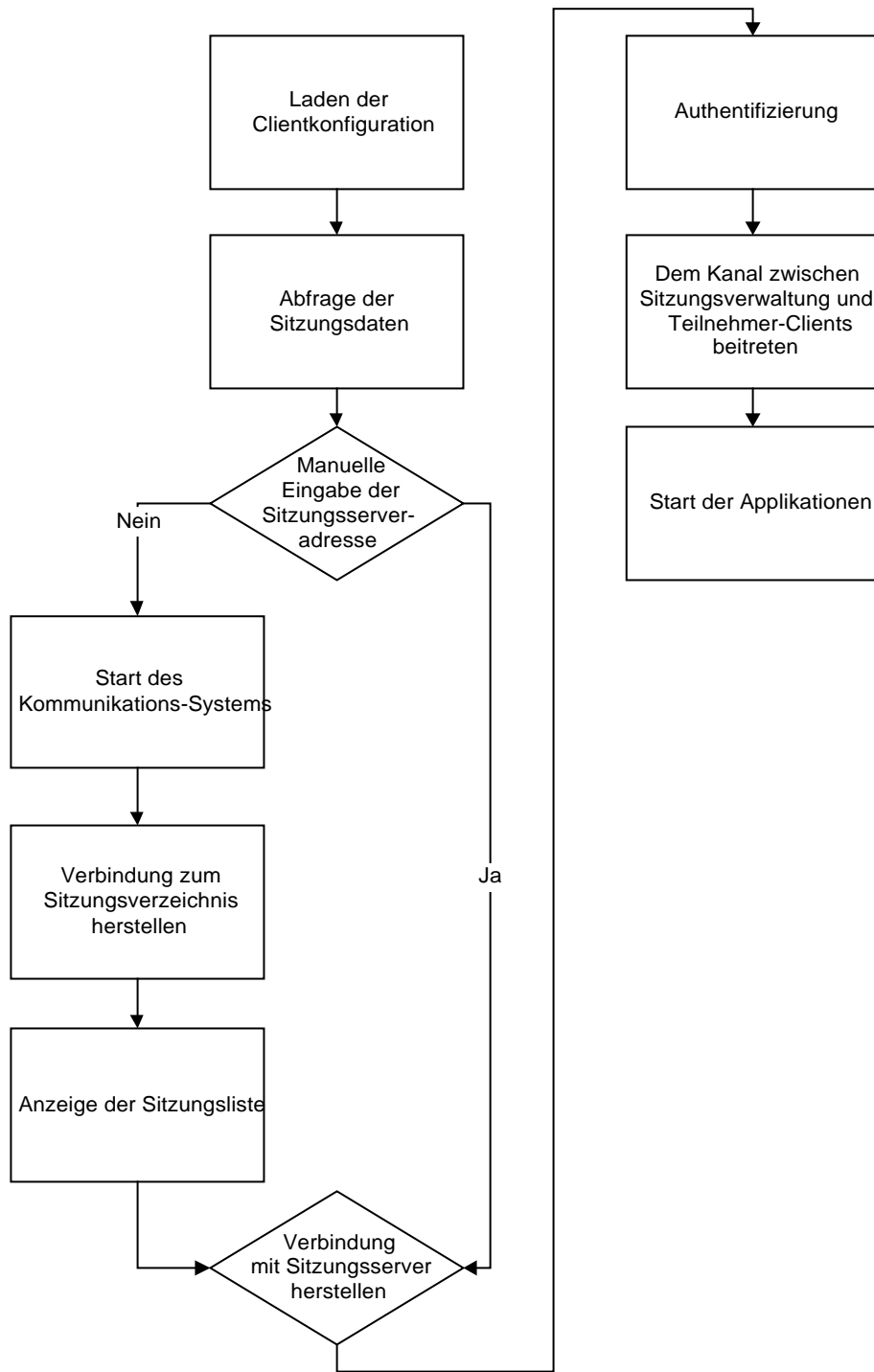
- Benutzername
- Passwort
- Sitzungserveradresse
- Dateiname für Protokolldatei

Wenn dem Teilnehmer die Sitzungsadresse nicht bereits bekannt ist, kann sich dieser mit dem Sitzungsverzeichnis verbinden. So kann dem Teilnehmer eine Liste aller



momentan laufenden Sitzungen angezeigt werden. Diese Liste wird vom Sitzungsverzeichnis aus ständig aktualisiert. Aus dieser Liste kann der Teilnehmer dann bequem auswählen, welcher Sitzung er beitreten möchte.

Sofern sich der Teilnehmer für eine Sitzung aus dem Sitzungsverzeichnis entschieden hat, oder manuell eine Sitzungsserveradresse in seinem Client eingegeben hat, muss sich der Teilnehmer Client mit der Sitzungsverwaltung auf dem jeweiligen Sitzungsrechner verbinden. Sobald die Verbindung zwischen Teilnehmer Client und Sitzungsverwaltung über das Kommunikationssystem hergestellt wurde, wird die Autorisierung des Teilnehmers mittels Benutzernamen und Passwort durchgeführt. Um das Benutzerpasswort nicht unnötigen Sicherheitsrisiken auszusetzen, wird dieses nur verschlüsselt übertragen. Die Verschlüsselung des Passwortes findet direkt nach der Eingabe im Teilnehmer Client statt. Somit wird sichergestellt, dass sein Passwort während der Übertragung zum Server geschützt bleibt. Durch die sofortige Verschlüsselung des Passwortes nach der Eingabe kann dieses auf dem Rechner der den Teilnehmer Client beherbergt auf Teilnehmerwunsch auch permanent gespeichert werden ohne ein Sicherheitsrisiko darzustellen. Fällt die Autorisierung positiv aus, müssen durch den Client die einzelnen für diese Sitzung benötigten Applikationen lokal gestartet werden. Die Informationen welche Applikationen benötigt werden erhält der Client direkt von der Sitzungsverwaltung auf dem Sitzungsserver über den zwischen Sitzungsverwaltung und Teilnehmer Client existierenden Kanal. Den einzelnen Applikationen muss beim Start zudem mitgeteilt werden, an welcher Stelle diese ihre Daten nach Beendigung permanent abspeichern können. Der Vorgang beim Start einer Sitzung auf Clientseite wird in folgender Abbildung nochmals verdeutlicht.



**Abbildung 8: Ablauf beim Start einer Sitzung auf Clientseite**

Die Sitzungsverwaltung auf Clientseite wird im Projekt Session Management Client genannt. Der Session Management Client ist jeweils lokal auf den einzelnen Teilnehmer Rechnern vorhanden. Er bietet hauptsächlich Funktionen für den Verbindungsaufbau zwischen Teilnehmer Client und dem Session Management Server bzw. dem Session Directory an. Des weiteren startet der Session Management Client die für einen Sitzung

benötigten Applikationen. Hierzu greift er auf die von den Applikationen lokal zur Verfügung gestellten Startfunktionen zu. Ebenfalls bietet der Session Management Client auch die Möglichkeit, die weiter oben bereits vorgestellten Benutzerparameter abzufragen und zu bearbeiten. Einen genauen Überblick über die vom Session Management Client zur Verfügung gestellten Funktionen bietet folgende Aufstellung:

- Abfrage/Änderung der Benutzerparameter
- Verbindung mit dem Session Directory herstellen (bei Bedarf)
- Verbindung mit dem Session Management Server
- Übermittlung von Benutzernamen und verschlüsseltem Passwort
- Abfrage der benötigten Applikationen
- Start der benötigten Applikationen
- Beendigung der laufenden Applikationen
- Neustart von abgestürzten Applikationen
- Teilnehmerliste anzeigen

### **4.2.2 Sitzungsverwaltung Administrationsclient**

Der Administrationsclient für die Sitzungsverwaltung wird Session Management Remote Admin Client genannt. Er ist das Gegenstück zum bereits weiter oben erwähnten Session Management Admin Server (Sitzungsverwaltung Administrationsserver). Der Session Management Remote Admin Client dient zur Steuerung einer bereits mit dem Local Administration Client gestarteten Sitzung. Gleichzeitig ist es möglich über den Remote Admin Client die Sitzung am Schluß zu beenden. Die vom Session Management Remote Admin Client zur Verfügung gestellten Funktionen lauten wie folgt:

- Beitritt zum Kanal zwischen Session Management Client und Session Management Server
- Sitzung sperren
- Sitzung beenden
- Benutzer entfernen
- Beantragung von Benutzerrechten
- Editierung der Liste der erlaubten Benutzer

- Editierung der Liste der nicht erlaubten Benutzer

Die Befehle und Daten werden wie bei jeder anderen Applikation auch über einen eigens für diesen Dienst gestarteten Kanal über das Kommunikationssystem hin zum Serverteil der Administrationsapplikation übertragen. Dieser wertet die Befehle und Daten aus und leitet diese dann an die Sitzungsverwaltung weiter, sofern der Absender eine Berechtigung zur Administration der Sitzung besitzt.

### 4.3 Zusammenspiel der Sitzungsverwaltungskomponenten

Nachdem nun die Aufteilung der einzelnen Anforderungen in logische Pakete in den letzten Abschnitten stattgefunden hat, werden nun in diesem Abschnitt die Zusammenhänge zwischen den einzelnen Komponenten nochmals genauer herausgestellt. Folgendes Schaubild soll den Zusammenhang der einzelnen Komponenten verdeutlichen.

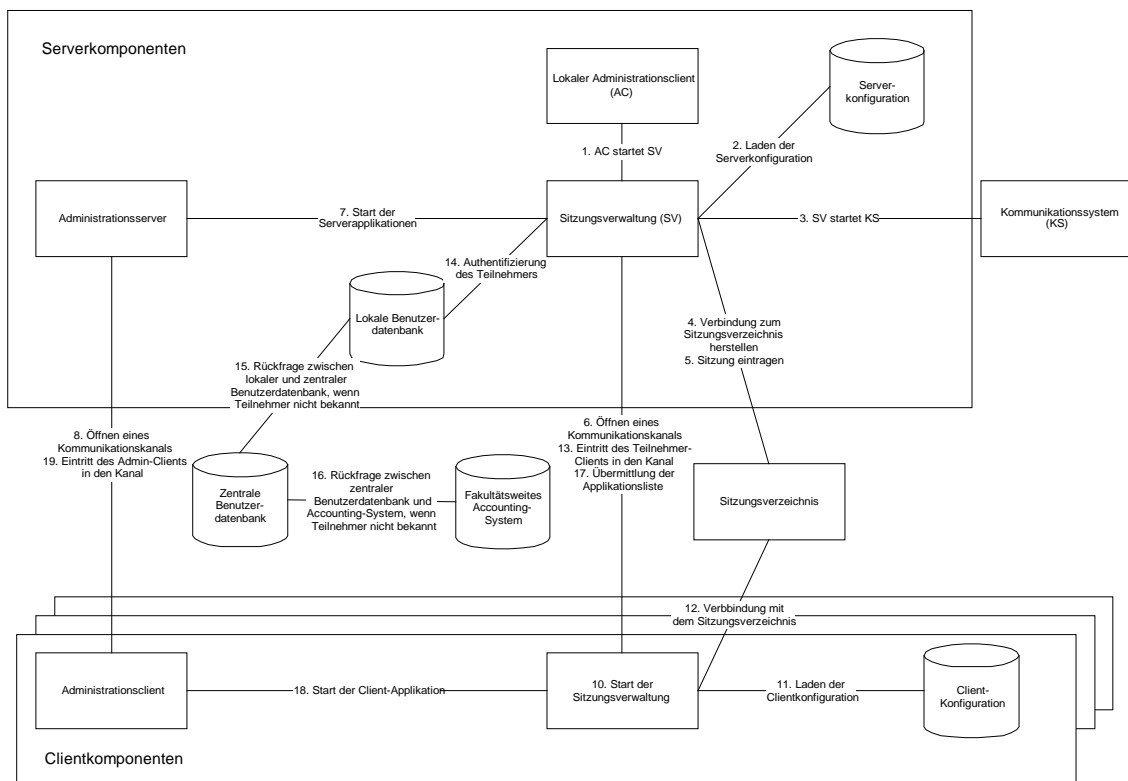


Abbildung 9: Zusammenspiel der Sitzungsverwaltungskomponenten

Wie in obigem Bild ersichtlich wird, ist die Sitzungsverwaltung der zentrale Teil. Über die Sitzungsverwaltung werden alle einzelnen Komponenten miteinander verbunden. Im folgenden wird nicht nur der Ablauf beim Starten einer Sitzung auf Serverseite beschrieben, sondern auch der Ablauf bei der Anmeldung eines Teilnehmer-Clients.

Als erstes Element startet der lokale Administrationsclient. Dieser Administrationsclient dient dazu, die einzelnen Sitzungsdaten die für den Start und Betrieb einer Sitzung nötig sind, abzufragen. Wie schon erwähnt läuft diese Abfrage der Daten nur auf textueller Basis ab. Weitere Informationen hierzu finden sich auch in Abschnitt 5.1.1. Die im lokalen Administrationsclient eingegebenen Sitzungsdaten werden dann zum Start der zentralen Sitzungsverwaltung auf Serverseite benötigt. Um weitere Standardinformationen, die für die Sitzungsverwaltung wichtig sind, zu bekommen, wird als nächstes die Serverkonfiguration geladen. Nun weiss die Sitzungsverwaltung zum Beispiel darüber Bescheid, wo sich der Sitzungsverzeichnisserver oder die zentrale Benutzerdatenbank befindet. Um die Basis für eine Kommunikation mit anderen Clients schaffen zu können, wird nun von der Sitzungsverwaltung aus das Kommunikationssystem gestartet. Über dieses Kommunikationssystem können dann später die einzelnen Teilnehmer Clients mit dem Sitzungsserver und den darauf laufenden Applikationsservern Verbindung aufnehmen. Wenn bei der Abfrage durch den lokalen Administrationsclient die Veröffentlichung im Sitzungsverzeichnis gewünscht wurde, wird nun eine Verbindung zum Sitzungsverzeichnis aufgebaut. Über diese Verbindung können die für die Veröffentlichung benötigten Sitzungsdaten dann übermittelt werden. Der genaue Ablauf wurde ja bereits weiter oben schon beschrieben. Als nächstes öffnet die Sitzungsverwaltung einen Kanal, um später mit den einzelnen Sitzungsverwaltungen auf den Teilnehmerclients kommunizieren zu können. Über diesen Kanal werden später den Sitzungsverwaltungen auch die einzelnen, zu startenden Applikationen übermittelt. Als weitere wichtige Aufgabe werden nun die Applikationsserver der einzelnen Applikationen gestartet. Welche dies genau sind hängt jeweils von den auf dem Sitzungsserver installierten Applikationen und den bei der Abfrage durch den lokalen Administrationsclient gewünschten Applikationen ab. Die einzelnen Applikationsserver eröffnen ihrerseits dann jeweils einen weiteren Kommunikationskanal, damit sich die passenden Applikationsclients dann mit dem Server über diese verbinden können. Auf Serverseite sind nun die einzelnen Aufgaben soweit erledigt. Der Sitzungsserver wartet nun auf Anfragen der einzelnen

Teilnehmerclients. Erst wenn diese eintreffen werden weitere Funktionen auf Serverseite ausgeführt.

Wie aus dem Schaubild zu entnehmen ist wird auf Clientseite zuerst die Sitzungsverwaltung gestartet. Die benötigten Sitzungsdaten werden hier grafisch, direkt durch die Sitzungsverwaltung erfasst. Der Einfachheit halber wurde hier bei der Architektur auf die Aufteilung in Administration Client und Sitzungsverwaltung wie auf Serverseite verzichtet, zumal hier auch weniger Sitzungsdaten abgefragt werden müssen. Wie auf der Serverseite wird aber auch hier im weiteren Verlauf nun die Clientkonfiguration geladen. Somit ist die Sitzungsverwaltung über die Adresse des Sitzungsverzeichnisseservers aufgeklärt. Durch die Angabe der Adresse des Sitzungsverzeichnisseservers in der Clientkonfiguration steht der Verbindung zwischen der Sitzungsverwaltung auf dem Teilnehmerclient und dem Sitzungsverzeichnis nun nichts mehr im Wege. Über die Daten aus dem Sitzungsverzeichnis kann durch den Teilnehmer nun eine passende Sitzung ausgewählt werden. Die Verbindung zur jeweiligen Sitzungsverwaltung wird nun durch den Eintritt des Teilnehmerclients in den Kanal, der durch die Sitzungsverwaltung aufgemacht wurde, realisiert. Als erstes wird nun der Teilnehmer authentifiziert. Dies erfolgt, indem das durch den Teilnehmerclient übermittelte Benutzername und Passwort Paar der Benutzerdatenbank zur Authentifizierung übergeben wird. Sofern die lokale Benutzerdatenbank den Benutzer nicht findet, erfolgt eine Rückfrage bei der zentralen Benutzerdatenbank. Wenn auch in dieser Datenbank kein Benutzer mit diesem Benutzernamen gefunden wird, erfolgt eine Rückfrage beim fakultätsweiten Accountingsystem. Erst wenn auch hier das Ergebnis negativ ausfällt wird der Teilnehmer abgewiesen. Sofern die Authentifizierung aber positiv ausfällt, wird der Teilnehmerclient in die Sitzung aufgenommen und erhält vom Sitzungsserver die Liste der benötigten Applikationen. Danach startet der Teilnehmerclient die einzelnen Applikationsclients die in dieser Applikationsliste aufgeführt werden, sofern diese lokal vorhanden sind. Die Applikationsclients verbinden sich hier nun direkt über den jeweiligen zur Applikation gehörenden Kommunikationskanal mit dem passenden Applikationsserver. Nun besteht für den Teilnehmerclient die Möglichkeit Daten zu empfangen und bei vorhandener Berechtigung auch Daten zu senden.

Die Sitzungsverwaltungen auf Server- und Clientseite warten nun jeweils auf die Beendigung der Applikationen oder auf einen Fehler bei den Applikationen.

Wenn alle Applikationen beendet wurden wird auch die Sitzungsverwaltung beendet. Bei Fehlern in einer Applikation führt die Sitzungsverwaltung einen Neustart der Applikation durch.

#### **4.4 Quellenangaben**

RADIUS                      Radius (**R**emote **A**uthentication **D**ial In **U**ser **S**ervice)  
Webseite zum Radius Protokoll (RFC 2865)  
<http://www.ietf.org/rfc.html>



## **5. Benutzungsoberfläche**

Nachdem in den vorhergehenden Abschnitten nun die Anforderungen und die Architektur der Sitzungsverwaltung ausführlich besprochen wurden, soll nun in diesem Kapitel die bereits implementierten Benutzungsoberflächen vorgestellt werden. Die Aufteilung findet hier, so wie bei der Architektur, wieder in einen Serverteil und einen Clientteil statt. Der Serverteil stellt die jeweiligen Oberflächen auf Serverseite und der Clientteil die jeweiligen Oberflächen auf Clientseite vor.

Eine weitere wichtige Eigenschaft die sich allerdings erst im Laufe des Projekts herausgestellt hat, ist die Möglichkeit die Benutzungsoberflächen mehrsprachig darstellen zu können. Realisiert wird diese Anforderung über eine eigens dafür von [JAVA] bereitgestellte Funktion. Mittels diese Funktion lassen sich alle Oberflächenpunkte in einer beliebigen Sprache darstellen. Welche Sprache jeweils dargestellt werden soll, wird von [JAVA] direkt beim Start anhand des installierten Betriebssystems herausgefunden. Im Moment existieren für die bereits entwickelten Oberflächen als Standard eine englische und eine deutsche Version. Wenn anhand des Betriebssystems nicht ermittelt werden kann, welche Sprache benutzt wird oder für die laut Betriebssystem ermittelte Sprache keine Version vorhanden ist, wird automatisch die englische Version verwendet.

### **5.1 Serveroberflächen**

In diesem Abschnitt wird nun genauer auf die Serveroberflächen eingegangen. Zu den jeweiligen Oberflächen wird anhand einzelnen Bildschirmkopien deren Handhabung erläutert. Zuerst werden aber noch einige allgemeine Bemerkungen zu den Benutzungsoberflächen auf Serverseite gemacht.

#### **5.1.1 Allgemeine Bemerkungen**

Auf Serverseite wurde bei der Benutzungsoberfläche Wert darauf gelegt, dass diese aus der Ferne zu bedienen ist. Dies ermöglicht die Installation der SASCIA Serversoftware auf einem zentralen Rechner. Auf diesem können die einzelnen Sitzungen dann bequem

vom eigenen Rechner aus gestartet und administriert werden. Somit wird der umständliche Gang zum Sitzungsrechner minimiert.

Um diese Administration des SASCIA-Systems auf der Serverseite nun gewährleisten zu können wurde bei der Erfassung der Sitzungsdaten und bei der Benutzerverwaltung auf die grafische Wiedergabe verzichtet. Vielmehr kommt hier eine textuelle Wiedergabe zum Einsatz. Dies hat einen entscheidenden Vorteil. Durch die textuelle Oberfläche ist es möglich die Sitzungsverwaltung auf dem Sitzungsserver über bekannte Fernadministrationsprogramme zu starten. Auf [UNIX] Systemen kommt hierbei das Programm [TELNET] oder [SSH] zum Einsatz. Für Windowsumgebungen empfiehlt sich das Programm [PCANYWHERE].

### **5.1.2 Erfassung der Sitzungsdaten**

Folgendes Bild stellt ein stark vereinfachtes Beispiel zur Erfassung der Sitzungsdaten dar. Erfasst werden hierbei, bis auf die Frage zur Veröffentlichung am Sitzungsverzeichnis, nur die nötigsten Daten die für den Start einer Sitzungsverwaltung benötigt werden.

```
C:\studienarbeit\sascia>run sascia.SMSessionManagementServer

*****
Herzlich Willkommen beim Sascia Session Management!

*****

Liste der installierten Applikationsserver: Chat,WB
Benotigte Applikationsserver durch (,) getrennt eingeben: Chat,WB
Server Administrator eingeben: sommer
Sitzung im Sitzungsverzeichnis veroeffentlichen? <<J>a or <N>ein [Standard: Nein]: Nein
Pfad zur Datenbank [standard: \studienarbeit\sascia\temp\server]:
Datenbankname [standard: test]: test
Moechten Sie die Benutzerdatenbank anpassen? <<J>a oder <N>ein [Standard: Nein]: Nein

*****

Server laeuft unter:
Registry-Host: laptopalt
Registry-Port: 4561
SessionURL: jsdt://laptopalt:26576/socket/Session/sascia.mainsession

*****

Create SM Channel
Enhydra InstantDB - Version 3.26
The Initial Developer of the Original Code is Lutris Technologies Inc.
Portions created by Lutris are Copyright (C) 1997-2001 Lutris Technologies, Inc.
All Rights Reserved.
```

Abbildung 10: Abfrage der Sitzungsdaten

Als erstes wird eine Liste der auf dem Sitzungsserver installierten Applikationen ausgegeben. Erzeugt wird diese Liste dynamisch. Sobald eine neue Applikation auf dem Sitzungsserver installiert wird und diese in das Konfigurationsobjekt eingetragen ist, erscheint auch diese Applikation in der Liste. Angegeben werden die Kurzformen der Applikationsnamen. Zur Zeit existiert eine Diskussionsapplikation (Chat) und eine elektronische Tafel (WB).

In der nächste Zeile werden dann die gewünschten Applikationen abgefragt. Der Benutzer hat die Möglichkeit aus allen Applikationen die in der vorherigen Liste aufgeführt worden sind auszuwählen, welche er benötigt. Die Auswahl erfolgt über die Eingabe der Kurzformen der Applikationsnamen. Wie diese lauten, ist der Liste zu entnehmen. Die einzelnen Applikationsnamen sind durch Komma getrennt anzugeben. Nachdem die Abfrage der Sitzungsdaten erfolgt ist, werden dann alle in der durch Komma getrennten Liste angegebenen Applikationen und die dazugehörigen Applikationsserver gestartet.

Weiter geht es mit der Frage, ob die Sitzung am Sitzungsverzeichnis eingetragen werden soll. Der Standardwert für diese Abfrage ist auf Nein gesetzt. Die Auswahl des Standardwertes wird erreicht indem man einfach die Eingabetaste betätigt ohne eine Auswahl anzugeben. Wird bei dieser Abfrage mit Ja geantwortet wird die Sitzung am Sitzungsverzeichnis angemeldet und veröffentlicht. Der genaue Ablauf der zur Eintragung der Sitzung im Sitzungsverzeichnis führt, wurde ja bereits weiter oben im Abschnitt 4.1.3 erklärt.

Die beiden nächsten Punkte dienen zur Festlegung des Pfades und des Dateinamens für die Protokollierungsdatenbanken. Mittels dieser Angabe wird bestimmt wo und unter welchem Namen die Applikationsdaten der einzelnen Applikationen abgespeichert werden. Der angegebene Dateiname dient hier aber nur als Präfix und wird unter diesem Namen nicht direkt auf die Platte geschrieben. Vielmehr wird pro Applikation eine eigene Datei erzeugt, damit die einzelnen Applikationen die Daten nicht alle in die Datei „test“ schreiben. Hierzu wird an den unter Dateinamen angegebenen Namen zusätzlich noch das jeweilige Kürzel der Applikation angehängt. Bei der elektronischen Tafel (WB) würde der Dateinamen dann „test\_WB“ heißen, sofern als Dateinamen test eingegeben wird. Unter diesem automatisch erzeugten Dateinamen sind dann auch die jeweiligen Daten der Applikation zu finden. Da aber jede Applikation selber entscheidet ob Ihre Daten protokolliert werden sollen oder nicht, ist es nicht sicher, dass zu jeder Applikation auch wirklich eine Datei existiert.

Der letzte Abfragepunkt ermöglicht den Zugang zur in Abschnitt 5.1.3 beschriebenen Benutzerverwaltung. Standardmäßig steht der Zugang auf Nein. Dies bedeutet soviel, dass wenn nicht explizit „Ja“ bei dieser Abfrage eingegeben wird, die Benutzerverwaltung auch nicht gestartet wird.

Sofern man bei der Abfrage der Benutzerverwaltung nichts eingegeben hat, oder man die Benutzerverwaltung wieder verlässt, wird automatisch nun das Kommunikationssystem gestartet. An der weiteren Ausgabe lässt sich dies auch erkennen. Die Daten in den angezeigten Feldern Registry-Host, Registry-Port und Session URL geben die Verbindungsdaten wieder, die von den Teilnehmerclients benötigt werden, um sich mit dieser Sitzung verbinden zu können. Sofern bei der

Abfrage zu Veröffentlichung der Sitzung mit „Ja“ geantwortet wurde, werden diese Daten über das zentrale Sitzungsverzeichnis den Teilnehmerclients zur Verfügung gestellt. Wenn aber mit Nein geantwortet wurde, müssen diese Daten manuell an den Teilnehmerclients eingetragen werden, damit eine Verbindung zustande kommt.

### 5.1.3 Benutzerverwaltung

Im vorherigen Abschnitt wurde die Benutzerverwaltung ja bereits erwähnt. Wenn auf die Abfrage nach der Anpassung der Benutzerdatenbank mit „Ja“ geantwortet wurde, wird nun die Benutzerverwaltung aufgerufen. Folgender Bildschirmausdruck zeigt die textuelle Oberfläche der Benutzerverwaltung.

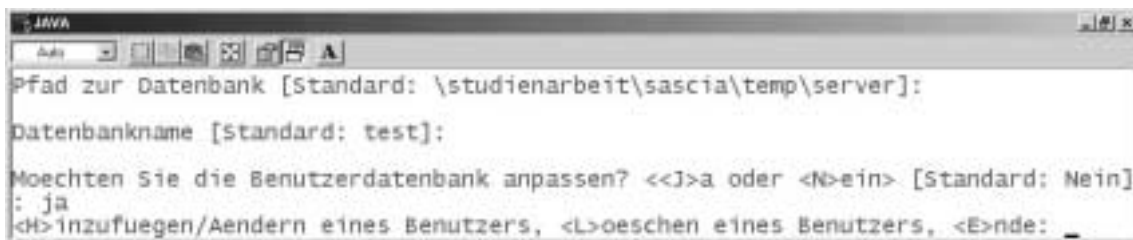


Abbildung 11: Benutzerverwaltung

Bei der Benutzerverwaltung gibt es vier wichtigen Möglichkeiten:

- Hinzufügen eines Benutzers
- Ändern eines Benutzers
- Löschen eines Benutzers
- Ende der Benutzerverwaltung

Diese vier Möglichkeiten werden auch gleich nach Start der Benutzerverwaltung im Hauptmenü zur Auswahl angeboten. Da sich die Möglichkeiten „Hinzufügen und Ändern eines Benutzers“ nur geringfügig voneinander unterscheiden, sind diese der Einfachheit halber zusammengefasst worden.

Wählt man nun den ersten Punkt „Hinzufügen / Ändern eines Benutzers“ aus, so wird man nacheinander nach:

- Benutzername
- Passwort
- Priorität

gefragt.

Sofern man unter Benutzername nun hier einen bereits existierenden Benutzer auswählt, wird dieser einfach was das Passwort und die Priorität betrifft auf die neu eingegebenen Werte geändert. Gibt man einen noch nicht existierenden Benutzername ein, so wird dieser mit dem dazugehörigen Passwort und der Priorität neu angelegt.

Die Abfolge der einzelnen Abfragen im Punkt „Hinzufügen / Ändern eines Benutzers“ zeigt folgender Bildschirmausschnitt.



**Abbildung 12: Änderung/Hinzufügen eines Benutzers**

Nach der Eingabe des Benutzernamens, des dazugehörigen Passworts und der Priorität gelangt man automatisch wieder in das Hauptmenü der Benutzerverwaltung. Dort stehen dann wieder alle vier Möglichkeiten der Benutzerverwaltung zur Auswahl.

Der zweite Punkt im Hauptmenü dient zum Löschen von Benutzern. Wählt man diesen Punkt aus wird man zur Eingabe des Benutzernamens aufgefordert. Nach Eingabe dieses Namens wird der dazugehörige Benutzer automatisch gelöscht und es erfolgt wieder die Darstellung des Hauptmenüs der Benutzerverwaltung.

Sofern alle Änderungen an den Benutzern vollzogen sind, dient der Auswahlpunkt Ende im Hauptmenü zum Verlassen der Benutzerverwaltung.

## 5.2 Clientoberflächen

In den zwei vorhergehenden Abschnitten wurden die textuellen Oberflächen auf Serverseite vorgestellt. Nun werden in den beiden folgenden Abschnitten jeweils die Oberflächen der Clientseite beschrieben.

Anders als auf der Serverseite sind diese Oberflächen grafikbasiert, da hier keine Fernwartung erfolgt und der Benutzer hier ja direkt vor dem Rechner sitzt. Aufgeteilt wurde die Vorstellung in die zwei Hauptbestandteile auf Clientseite. Zum einen wird die Oberfläche der Sitzungsverwaltung auf der Clientseite, das Gegenstück zur Sitzungsverwaltung auf der Serverseite und zum anderen wird der Administrationsclient für die Steuerung der Sitzungsverwaltung auf Serverseite vorgestellt.

### 5.2.1 Sitzungsverwaltung Clientseite

Die Sitzungsverwaltung auf Clientseite oder auch Session Management Client dient hauptsächlich zur Erfassung der Sitzungsdaten und zum Start der Applikationsclients. Folgender Bildschirmausdruck soll einen Überblick über den Session Management Client verschaffen.

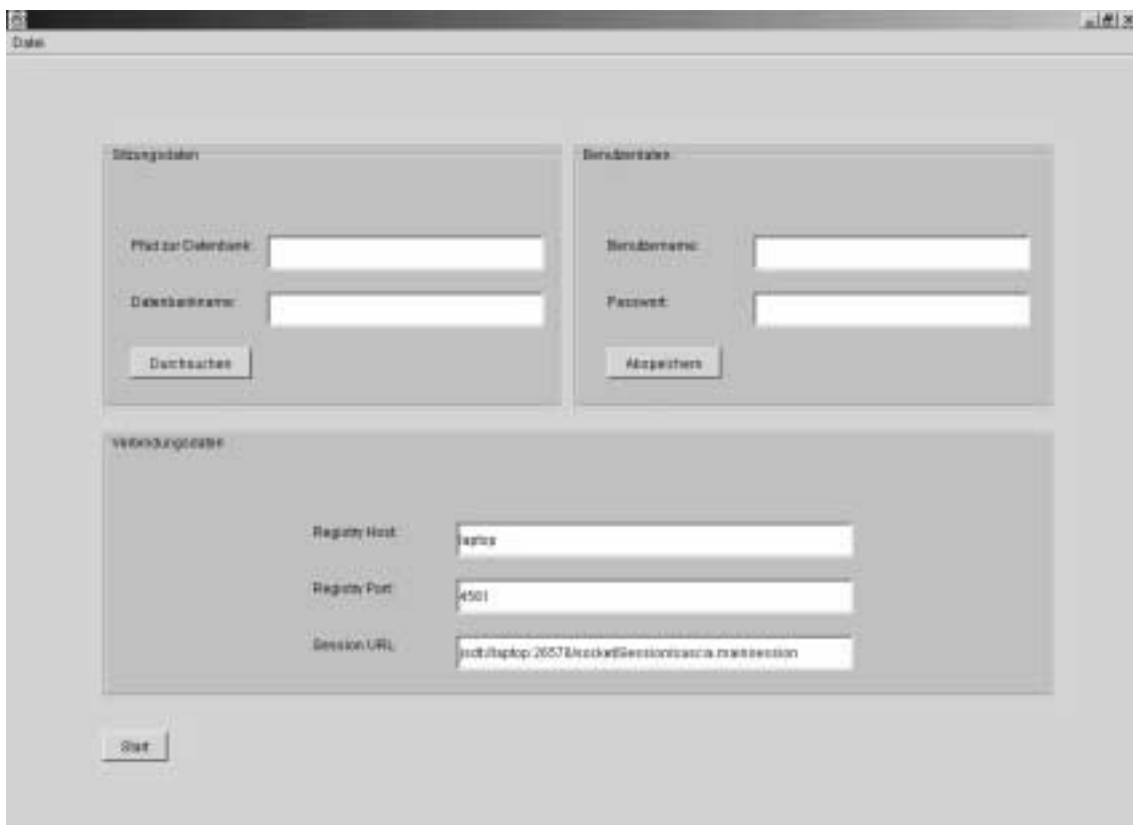


Abbildung 13: Session Management Client Oberfläche

Der Session Management Client ist einfach gehalten und ist in drei Bereiche aufgeteilt. Die drei Bereiche unterteilt nach dem Verwendungszweck der abgefragten Parameter sind:

- Sitzungsdaten
- Benutzerdaten
- Verbindungsdaten

Im Bereich Sitzungsdaten werden diese Parameter abgefragt die lokal für den Start einer Sitzung wichtig sind. Wie auf Serverseite wird auch hier der Pfad und der Dateiname für die Protokollierungsdatenbanken der einzelnen Applikationen abgefragt. Um die Eingabe zu erleichtern gibt es die Möglichkeit den Pfad und den Dateiname mittels eines Dateiauswahlfensters zu bestimmen. Geöffnet wird das Fenster über den Knopf „Durchsuchen“. Auch hier werden die wirklichen Dateinamen der Protokollierungsdatenbanken der einzelnen Applikationen durch hinzufügen eines Applikationskürzels erzeugt.

Rechts vom Bereich der Sitzungsdaten aus gesehen liegt der Bereich der Benutzerdaten. Die Eingabe der Benutzerdaten wird für die Anmeldung des Session Management Clients an der Sitzung benötigt. Der Benutzer hat hier die Möglichkeit seinen Benutzernamen und sein Passwort für die Sitzung einzugeben, an der er teilnehmen möchte. Um bei weiteren Teilnahmen bei einer Sitzung die Eingabe der Benutzerdaten zu vereinfachen, gibt es in diesem Bereich über den Knopf „Abspeichern“ noch die Möglichkeit seinen Benutzernamen und sein Passwort abzuspeichern. Bei einem erneuten Aufruf der Sitzung werden diese dann automatisch eingetragen, so dass der Teilnehmer diese nicht nochmals eintragen muss. Diese Vorgehensweise ist aber nur dann empfehlenswert, wenn kein anderer Nutzer Zugang zu dem Rechner hat. Sonst besteht die Gefahr, dass sich unberechtigte Teilnehmer über diesen Rechner an einer Sitzung beteiligen können.

Im unteren Bereich des Fensters befinden sich die Verbindungsdaten. Hier können die für eine Verbindung benötigten Daten wie Registry-Host, Registry-Port und Session-URL eingetragen werden. Um die Bedienung zu vereinfachen werden beim Start des



Session Management Client Standardwerte aus der Clientkonfiguration geladen und dort in den passenden Feldern wiedergegeben. Ein Zugriff auf das Sitzungsverzeichnis, um die Daten von dort zu bekommen ist im Prototyp noch nicht vorgesehen. Weitere Informationen hierzu gibt Abschnitt 7.

### 5.2.2 Sitzungsverwaltung Administrationsclient

Auf folgendem Bildschirmausdruck sieht man die Benutzungsoberfläche des Administrationsclients für die Sitzungsverwaltung. Im Projekt SASCIA auch Session Management Remote Admin Client genannt. Gestartet wird der Client automatisch nach dem Start des im vorherigen Abschnitt beschriebenen Session Management Client und der erfolgreichen Anmeldung bei einer vorhanden Sitzung an einem Sitzungsserver.

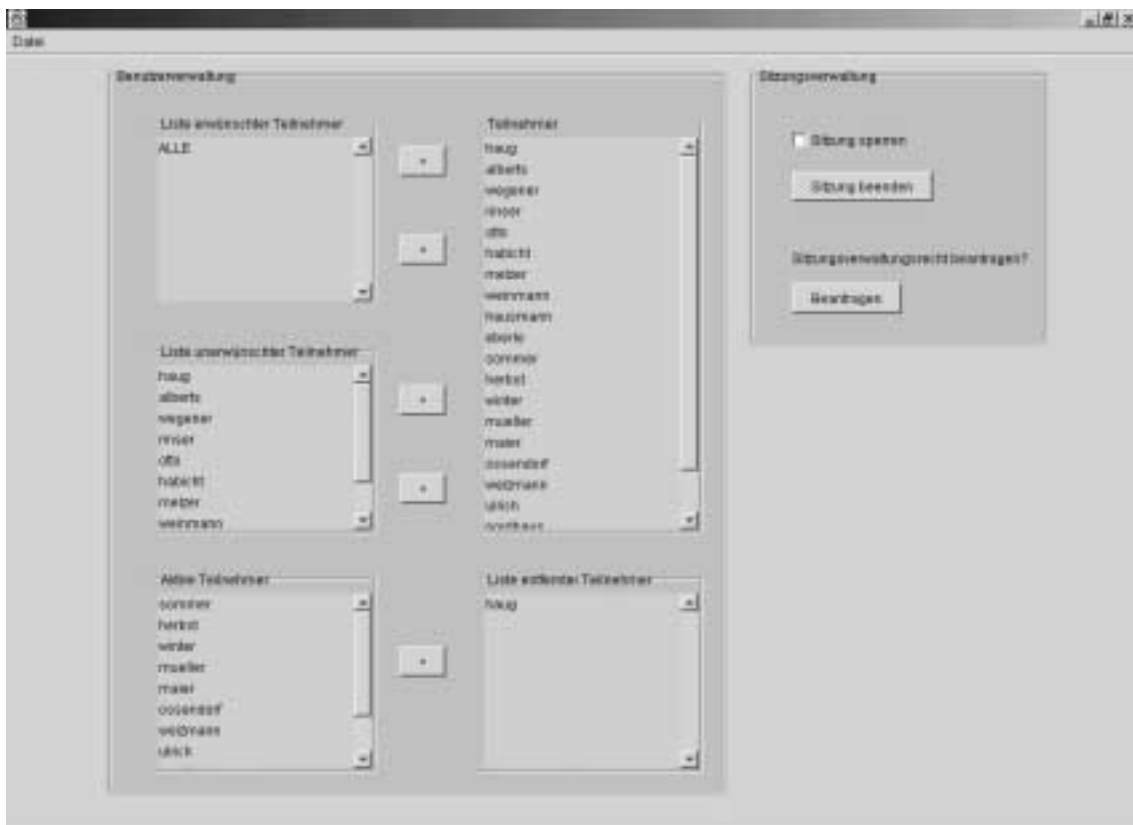


Abbildung 14: Remote Admin Client

Mit dem Session Management Remote Admin Client kann ein berechtigter Teilnehmer die Sitzungsverwaltung aus der Ferne administrieren. Diese Administration erfolgt über verschiedene Bereiche.

Der Session Management Remote Admin Client ist in zwei verschiedenen Bereiche aufgeteilt. Zum einen gibt es den Bereich der Benutzerverwaltung und zum anderen gibt es den Bereich der Sitzungsverwaltung.

Im Bereich der Benutzerverwaltung gibt es die Möglichkeit verschiedene Listen editieren zu können. Als erstes von oben links her gesehen beginnt die Liste der erwünschten Teilnehmer. Alle in dieser Liste aufgeführten Teilnehmer haben die Möglichkeit sich an der vom Remote Admin Client administrierten Sitzung anzumelden.

Das Hinzufügen und Entfernen von Teilnehmern zur Liste kann über die zwei Knöpfe rechts der Liste erfolgen. Zuerst wählt man in der großen Teilnehmerliste in der Bildmitte die einzelnen Teilnehmer durch Mausklicks aus, die an der Sitzung teilnehmen dürfen. Durch Betätigen des „< „ Knopfes werden diese Teilnehmer dann zur Liste der erwünschten Teilnehmer hinzugefügt. Das Entfernen der Teilnehmer aus der erlaubten Liste funktioniert fast genauso. Allerdings werden hier die zu entfernenden Teilnehmer in der linken, der „Liste der erlaubten Teilnehmer“, markiert und dann durch Betätigen des „>“ Knopfes aus der Liste entfernt. Sofern man die Sitzung für jeden freigeben möchte, genügt es wenn aus der Teilnehmerliste in der Bildmitte der Standardwert ALLE in die Liste der erwünschten Teilnehmer übernommen wird.

Unterhalb der Liste der erwünschten Teilnehmer befindet sich die Liste der unerwünschten Teilnehmer. Hier können die Teilnehmer aus der Teilnehmerliste eingefügt werden, die auf keinen Fall an der Sitzung teilnehmen sollen. Das Hinzufügen und Entfernen von Teilnehmern funktioniert hier genauso wie bei der vorherigen Liste. Sofern diese Liste leer ist oder der Standardwert KEINE eingefügt ist, können sich alle Teilnehmer, die in der Liste der erlaubten Teilnehmer aufgeführt sind, zu dieser Sitzung anmelden. Eine Doppeleintragung eines Namens in beide Listen (erlaubte und nicht erlaubte Liste) ist nicht möglich, da dies auch keinen Sinn ergibt.

Als nächstes kommt die Liste der aktiven Teilnehmer. Hier werden alle Teilnehmer aufgeführt, die sich bereits mit Ihren Teilnehmerclients an der Sitzungsverwaltung

angemeldet haben und erfolgreich die Authentifizierung durchlaufen haben. Diese Liste dient dazu, um aktive Teilnehmer aus der laufenden Sitzung zu entfernen. Entfernt werden kann ein Teilnehmer indem er in der Liste mittels der Maus markiert wird und dann der Knopf rechts von der Liste betätigt wird. Sobald dies durchgeführt wird, wandert der Teilnehmer aus der Liste der aktiven Teilnehmer in die Liste der entfernten Teilnehmer rechts davon. Bei dieser Aktion wird der betroffene Teilnehmerclient vom Sitzungsserver getrennt. Automatisch wandert der Teilnehmernamen auch in die Liste der unerwünschten Teilnehmer und wird falls vorhanden aus der Liste der erwünschten Teilnehmer entfernt, damit sich der Teilnehmer nicht wieder erneut an der Sitzung anmelden kann. Eine erneute Anmeldung dieses Teilnehmers ist erst dann möglich, wenn er wieder aus der Liste von nicht erlaubten Teilnehmern entfernt wurde und die Liste von erwünschten Teilnehmern entweder der Standardwert ALLE enthält oder sein Name dort eingetragen worden ist. In obigem Bildschirmausdruck wurde der Teilnehmer „haug“ entfernt und somit automatisch auch in die Liste der nicht erwünschten Teilnehmer mit aufgenommen.

Neben dem Bereich der Benutzerverwaltung liegt der Bereich für die Sitzungsverwaltung. Innerhalb der Sitzungsverwaltung gibt es bis jetzt noch wenige Funktionen. Eine davon ist die Möglichkeit, die Sitzung mittel einer Checkbox zu sperren. Dies hat zur Folge, dass sich kein neuer Teilnehmer mehr zu dieser Sitzung anmelden kann. Dies ist selbst dann der Fall wenn er als erwünschter Teilnehmer eingetragen ist.

Die nächste Funktion im Bereich der Sitzungsverwaltung ist die Möglichkeit die Sitzung zu beenden. Durch Betätigen des Knopfes „Sitzung beenden“ werden auf Serverseite automatisch die einzelnen Applikationsserver und die Sitzungsverwaltung beendet. Danach ist von den einzelnen Teilnehmerclients keine Verbindung zum Sitzungsserver mehr möglich, da das gesamte Kommunikationssystem mit beendet wurde.

Wie bereits schon erwähnt, ist die Benutzung des Remote Admin Clients von einer Berechtigung abhängig. Nicht jeder Teilnehmer hat automatisch Zugriff auf die Sitzungsverwaltung mittels dieses Clients. Der erste, sich an der Sitzung am Sitzungsserver anmeldende Teilnehmer erhält automatisch die Berechtigung zur

Benutzung des Remote Admin Clients. Diese Berechtigung bleibt aber nur solange auf der Seite dieses Teilnehmers, solange sich zur Sitzung kein weiterer Teilnehmer anmeldet, der eine höhere Priorität bei seinen Benutzerdaten in der Benutzerdatenbank vorzuweisen hat. Gibt es in der Sitzung nun aber einen Teilnehmer der eine höhere Priorität hat als der Teilnehmer, der gerade die Berechtigung besitzt den Remote Admin Client zu bedienen, so kann dieser Teilnehmer die Berechtigung anfordern. Diese Anforderung erfolgt durch die Betätigung des Knopfes „Beantragen“ bei der Frage: „Sitzungsverwaltungsrecht beantragen?“. Liegt nun beim Beantragenden tatsächlich eine höhere Priorität vor, so wird ihm die Berechtigung erteilt und dem bisherigen Rechteinhaber entzogen.

### **5.3 Quellenangaben**

JAVA	Java Webseite <a href="http://java.sun.com">http://java.sun.com</a>
UNIX	Unix Informationswebseite <a href="http://www.rvs.uni-bielefeld.de/~dstoesse/sysadmin/unix.html">http://www.rvs.uni-bielefeld.de/~dstoesse/sysadmin/unix.html</a>
TELNET	Telnet (RFC854) <a href="http://www.ietf.org">http://www.ietf.org</a>
SSH	SSH Internet-Draft <a href="http://search.ietf.org/internet-drafts/draft-ietf-secsh-filexfer-00.txt">http://search.ietf.org/internet-drafts/draft-ietf-secsh-filexfer-00.txt</a>
PCANYWHERE	pcAnywhere Webseite des Herstellers <a href="http://www.symantec.com/region/de/product/pcanywhere/index_ent.html">http://www.symantec.com/region/de/product/pcanywhere/index_ent.html</a>

## 6. Test und Bewertung

### 6.1 Test und Bewertung

Nachdem in den letzten Abschnitten die Anforderungen, Architektur und die Benutzeroberflächen der Sitzungsverwaltung vorgestellt wurden, soll nun in diesem Abschnitt ein Test und eine Bewertung der Sitzungsverwaltung durchgeführt werden.

Hierzu werden alle Einzelkomponenten der Sitzungsverwaltung unter die Lupe genommen. Selbstverständlich konnten allerdings nur die bereits fertigen Komponenten zur Begutachtung herangezogen werden.

Zu den bereits fertigen Komponenten, die zur Begutachtung herangezogen wurden, zählen folgende:

- Lokaler Administrationsclient
- Benutzerverwaltung
- Sitzungsverwaltung auf Serverseite
- Sitzungsverwaltung auf Clientseite

Zuerst werden diese Komponenten daraufhin überprüft, ob diese auch die von Ihnen geforderten Aufgaben erfüllen, um dann daraufhin überprüft zu werden, ob Sie die Aufgaben auch korrekt erfüllen.

Begonnen mit der Überprüfung wurde mit dem lokalen Administrationsclient auf Serverseite. Wie bereits in Abschnitt 4.1.1 aufgeführt, soll der lokale Administrationsclient auf Serverseite folgende Aufgaben durchführen können:

- Abfrage der Sitzungsdaten
- Start der Sitzungsverwaltung

Bei der Abfrage der Sitzungsdaten werden durch den implementierten ersten Prototypen des lokalen Administrationsclients alle wichtigen Daten abgefragt (siehe Abschnitt

4.1.1) die unmittelbar für den Start einer Sitzung benötigt werden. Hinzu kommt, dass bei der geforderten Abfrage auch die benötigten Standardwerte korrekt berücksichtigt werden, was die Abfrage für den Benutzer wesentlich erleichtert. Alle weiteren, auch von Hand eingegebenen Daten, werden dann auch richtig an die Sitzungsverwaltung weitergegeben.

Die zweite Aufgabe die an den lokalen Administrationsclient gestellt wird, wird ebenfalls korrekt durchgeführt. Nach Beendigung der Abfrage der Sitzungsdaten wird automatisch die Sitzungsverwaltung auf der Serverseite gestartet.

Als nächstes wurde die Benutzerverwaltung untersucht. Die Benutzerverwaltung hat die Aufgabe, folgende Parameter eines Benutzers abzuspeichern:

- Benutzername
- Passwort
- Statische Priorität

Dies wird auch korrekt durchgeführt. Wie im Abschnitt Benutzeroberflächen angegeben, lassen sich ohne weitere Probleme Benutzer zur Datenbank hinzufügen, ändern und auch löschen. Abgespeichert werden die Benutzerdaten direkt auf der Festplatte des jeweiligen Sitzungsservers. Allerdings werden die Passwörter zur Zeit noch im Klartext bei der Abfrage des Passworts eingegeben und auch im Klartext in der Datei der Benutzerdatenbank abgespeichert, was im Punkt Sicherheit eventuell zu Problemen führen könnte. Dies ist daher ein wichtiger Punkt für weitere Entwicklungsarbeit. Die korrekte Funktion der Benutzerverwaltung wird durch dieses Problem aber nicht beeinträchtigt.

Die Sitzungsverwaltung auf Serverseite war der nächste Punkt der einem Test unterzogen worden ist. Wie schon weiter oben erwähnt sind die Hauptaufgaben der Sitzungsverwaltung auf Serverseite folgende:

- Start eines Kanals für Kommunikation zwischen Sitzungsverwaltung und Teilnehmer-Clients
- Start/Beenden des Kommunikationssystems

- Start/Beenden der Applikationsserver
- Übermittlung der gestarteten Applikationen
- Neustart von mit Fehlern beendeten Applikationen.

Nach dem Start der Sitzungsverwaltung durch den lokalen Administrationsclient werden diese Hauptaufgaben der Sitzungsverwaltung bis auf die letzten beiden Aufgaben auch korrekt bewerkstelligt. Probleme gibt es allerdings noch bei der Übermittlung der gestarteten Applikationen zu den Teilnehmerclients. Genauer erläutert wird das Problem weiter unten beim Test der Sitzungsverwaltung auf Clientseite. Auch der Neustart von mit Fehlern beendeten Applikationen funktioniert noch nicht richtig. Dieses Problem liegt aber vor allem noch an der Schnittstelle zwischen Applikation und Sitzungsverwaltung. Beide Probleme führen aber auch hier nicht zu einer Beeinträchtigung der Funktionalität der Sitzungsverwaltung.

Von der Serverseite aus führten die Test jetzt nun zur Clientseite. Auf dieser Seite wurde nur die Hauptkomponente, die Sitzungsverwaltung auf Clientseite, zum Test herangezogen.

Anders als auf der Serverseite, die textbasierte Aus- und Eingaben hatte, erfolgt hier die Kommunikation zwischen Benutzer und Programm durch eine grafische Oberfläche. Aber auch hier wurde zuerst überprüft, ob die von der Sitzungsverwaltung geforderten Funktionen erfüllt werden. Zum besseren Verständnis hier nochmals die von der Sitzungsverwaltung auf Clientseite zu erfüllenden Funktionen:

- Abfrage/Änderung der Benutzerparameter
- Verbindung mit dem Session Directory herstellen (bei Bedarf)
- Verbindung mit dem Session Management Server
- Übermittlung von Benutzernamen und verschlüsseltem Passwort
- Abfrage der benötigten Applikationen
- Start der benötigten Applikationen
- Beendung der laufenden Applikationen
- Neustart von abgestürzten Applikationen
- Teilnehmerliste anzeigen



Bis auf die Verbindung mit dem Session Directory und die Anzeige der Teilnehmerliste waren allen Funktionen im Prototyp vorhanden und konnten einem Test unterzogen werden. Auch hier stellte sich heraus, dass die einzelnen Funktionen bis auf weiteres korrekt arbeiten. Allerdings gibt es auch hier, wie schon bei der Sitzungsverwaltung auf Serverseite, Probleme bei der Abfrage der benötigten Applikationen und beim Neustart von abgestürzten Applikationen. Das Problem beim Neustart von abgestürzten Applikationen liegt genauso wie auf Serverseite an Problemen der Schnittstelle zwischen Applikation und Sitzungsverwaltung. Die Abfrage der benötigten Applikationen und das dadurch entstehende Problem liegt eher an der Oberflächengestaltung unter Java. Immer dann, wenn die Abfrage der benötigten Applikationen nicht schnell und korrekt erledigt werden kann, zum Beispiel bei einem Netzwerkproblem, hängt sich die gesamte Oberfläche der Sitzungsverwaltung auf Clientseite auf und lässt sich nicht mehr bedienen. Um trotzdem einen lauffähigen Client zum Test zu haben, wird im momentan vorhandenen Prototyp auf die Abfrage der benötigten Applikationen verzichtet und es werden alle vorhandenen Applikationen gestartet.

Obwohl im Sitzungsverwaltungsclient nicht alle Funktionen implementiert werden und eine implementierte Funktion nicht ganz korrekt arbeitet, ist der jetzige Client für den Einsatz als Prototyp geeignet. Allerdings stehen auch hier weitere Entwicklungsarbeiten an.

Als letzter Test wurde die Server- und Clientseite einer Zeitmessung unterzogen. Hierfür ist im Programm ein Zählermechanismus implementiert worden. Dieser Zählermechanismus wertet an zwei vorher bestimmten Programmpunkten die Systemzeit aus und errechnet die Differenz in Millisekunden. Diese Messungen beruhen auf der Systemuhr des jeweiligen Computers. Da diese zum Teil allerdings nicht ganz korrekt läuft, sind die ermittelten Werte jeweils mit Vorsicht zu genießen. Sie geben aber trotzdem einen groben Überblick über die Performance der Programme.

Als Testsysteme kommen folgende Komponenten zum Einsatz:

- Laptop mit Funk-LAN Anbindung
  - Intel Pentium III 1000 MHz
  - 256 MB Hauptspeicher
  - Betriebssystem Windows ME
  - 2 Mbit/s Wireless LAN Funkkarte
- Arbeitsplatzrechner mit fester Netzwerkanbindung
  - AMD Athlon 900 MHz
  - 128 MB Hauptspeicher
  - Betriebssystem Windows 2000 Professional
  - 100 Mbit/s Netzwerkanschluss
- Wave LAN
  - [LUCENT] Technologies Wireless Access Point
  - 2 Mbit/s Wireless LAN Funkkarte
  - 10 Mbit/s Anbindung (switched) an 100 Mbit/s Backbone

Folgende Abbildung soll den Aufbau des zum Test eingesetzten Systems verdeutlichen.

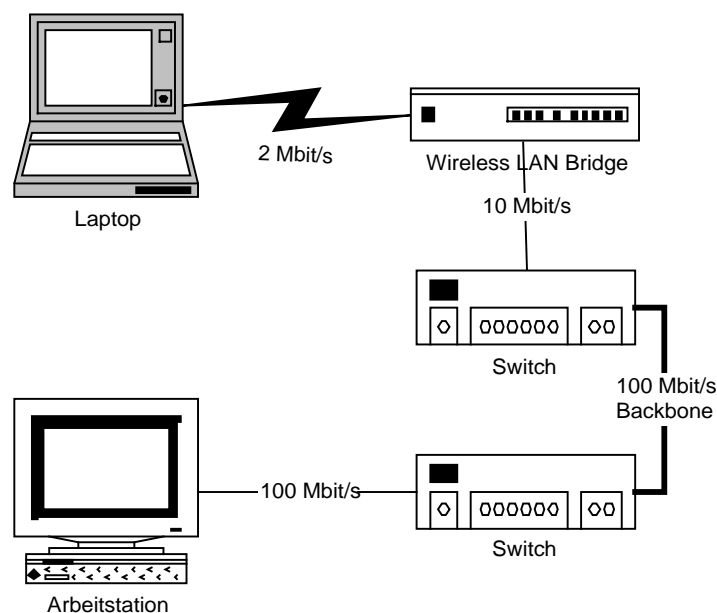


Abbildung 15: Aufbau des Testsystems

Insgesamt wurden vier verschiedene Fälle getestet. Getestet wurde jeweils wie lange es dauert bis alle Applikationen gestartet sind und wie lange es dauert bis eine Verbindung zwischen Sitzungsverwaltungsclient und Sitzungsverwaltungsserver hergestellt wurde. Als zusätzliche Variante wurde einmal der Client und der Server auf dem gleichen Rechner und beim nächsten mal auf zwei getrennten Rechnern installiert. Bei den Tests wurden jeweils zehn Durchläufe ausgeführt und ein Mittelwert der einzelnen Tests errechnet.

Als erstes wurde die Variante getestet, bei der Client und Server auf dem gleichen Rechner installiert sind. In diesem Fall waren beide auf dem oben beschriebenen Laptop installiert. Zugegeben ist es in der Realität eher selten der Fall, dass der Server und der Client auf dem gleichen Rechner laufen, aber durch diesen Test lässt sich im Vergleich mit dem Test auf getrennten Rechnern herausfinden, inwiefern sich das zusätzliche Netzwerk zwischen den Rechnern negativ auf die Performance des Systems auswirkt.

Um nun die Zeit herauszufinden wie lange der Start der Applikationen dauert, wurde die Abfrage der Systemzeit vor und nach dem Aufruf der Programmroutine, die für den Start der Applikationen zuständig ist, durchgeführt. Als Testapplikationen wurden die Chat-Applikation und die Whiteboard-Applikation herangezogen. Wie sich bei diesem Test herausstellte, dauerte der Start der Serverapplikationen mit 172 ms mehr als dreimal solange wie der Start der Clientapplikationen mit 55 ms und dies obwohl beide auf der gleichen Maschine laufen. Erklären lässt sich dies durch die umfangreicheren Arbeiten, vor allem beim Start des Kommunikationssystems auf Serverseite, wohingegen die Clientapplikationen beim Start lediglich den bereits bestehenden Kanälen beitreten müssen. Der Start des Kommunikationssystems wurde auf Clientseite bereits während dem Verbindungsaufbau zwischen Client und Server, welcher im nächsten Test untersucht wurde, erledigt.

Bei der Ermittlung des Zeitwertes für den Verbindungsaufbau zwischen Client und Server, welcher 297 ms betrug, zeigte sich wie schon beim Test des Starts von Applikationen, dass der Start des Kommunikationssystem viel Zeit verbraucht. Gemessen wurde hier die Zeit die benötigt wird, um sich mit der Sitzungsverwaltung auf Clientseite mit der Sitzungsverwaltung auf Serverseite zu verbinden. Es wurden

hierbei nicht die Zeiten erfasst, welche zusätzlich noch von den Applikationen benötigt werden, um sich mit dem Server zu verbinden.

Als nächstes wurden nun der Server und der Client auf verschiedenen Rechnern installiert und getestet. Der Server wurde auf dem oben beschriebenen Arbeitsplatzrechner mit fester Netzwerkanbindung und der Client wurde auf dem Laptop mit Funk-LAN Anbindung installiert

Beim Test für die Ermittlung der Startzeit der Applikationen wurden wieder die beiden oben schon erwähnten Applikationen benutzt. Auch hier stellte sich heraus, dass der Start der Applikationsserver mit 312 ms deutlich mehr Zeit in Anspruch nimmt, als der Start der Applikationsclients auf dem Laptop, welcher wieder 55 ms im Durchschnitt betrug. Deutlich wird hier auch, dass die Startzeit der Applikationsserver stark von der Rechnerkapazität abhängt. Im Vergleich zum Test des Starts von Applikationen auf dem Laptop, sind die Startzeiten der Applikationsserver auf dem Arbeitsplatzrechner bald doppelt so hoch. Dies rührt vor allem von der schlechteren Prozessorleistung und dem geringeren Speicher auf dem Arbeitsplatzrechner her.

Der letzte Test beschäftigte sich dann mit der Zeit für die Verbindung der Sitzungsverwaltung auf Clientseite mit der Sitzungsverwaltung auf Serverseite, wenn Client und Server auf verschiedenen Rechnern liegen. Auch hier wurde der Server auf dem oben beschriebenen Arbeitsplatzrechner und der Client auf dem Laptop gestartet. Der hier ermittelte Wert von 398 ms für die Verbindung zwischen Client und Server spiegelt hier auch deutlich das erwartete Verhalten wieder. Die mehr als 100 ms höhere Zeit, im Vergleich zur Installation von Server und Client auf dem gleichen Rechner, rührt hier vor allem durch die Netzwerkanbindung des Laptops über das oben beschriebene Funknetzwerk her. Vermutungsweise spielt hier aber auch die etwas schlechtere Leistung des Arbeitsplatzrechners, auf dem der Server läuft, eine Rolle. Dies konnte aber leider mangels Hardware nicht überprüft werden.

Folgende Abbildung gibt nochmals alle im Test ermittelten Werte wieder.

	Zeit zum Start der Applikationen (Serverseite / Clientseite)	Zeit zum Herstellen einer Verbindung zwischen Client und Server
Server/Client zusammen	172 ms / 55 ms	297 ms
Server/Client getrennt	312 ms / 55 ms	398 ms

**Abbildung 16: Testergebnisse**

In der ersten der beiden Zeilen werden jeweils die Ergebnisse wiedergegeben die gemessen wurden, wenn der Server und der Client zusammen auf dem oben beschriebenen Laptop installiert waren. Die zweite Zeile gibt die Werte wieder die ermittelt wurden, wenn der Server auf dem Arbeitsplatzrechner und der Client auf dem Laptop installiert waren.

Wie erwartet zeigten sich durch die Testdurchläufe Abhängigkeiten der Programme von der benutzten Hardware und vom Netzwerk. Wie den Werten zu entnehmen ist, gab es allerdings keine nennenswerten Probleme im Hinblick auf die Performance der Programme. Somit sollte einem Einsatz auch in Funknetzwerken nichts entgegenstehen.

## **6.2 Quellenangaben**

LUCENT

Lucent Technologies

Webseite des Herstellers

<http://www.wavelan.com>

## 7. Zusammenfassung und Ausblick

Im Gesamtprojekt SASCIA soll ein Konferenzsystem entwickelt werden, welches die Interaktivität zwischen Lehrenden und Lernenden verstärken soll. Dies wird bei SASCIA durch die Möglichkeit erreicht, dass den Zuhörern einer Lehrveranstaltung die Möglichkeit gegeben wird, sich direkt zu beteiligen. Diese Beteiligung wird dadurch erreicht, dass die Zuhörer direkt von Ihrem Platz aus zeigend oder sogar modifizierend in den Vortrag und die Anmerkungen des Vortragenden eingreifen können.

Um die Komplexität dieses Gesamtsystems zu entschärfen, wurde das gesamte System in mehrere Teile aufgeteilt und auf verschiedenen Personen zur Entwicklung verteilt. In dieser Studienarbeit wurde der Teil der Sitzungsverwaltung behandelt.

Zuerst wurden in Zusammenarbeit mit den anderen Entwicklern die Anforderungen an das Gesamtprojekt gestellt. Aus diesen Anforderungen heraus sind dann die in dieser Arbeit hervorgehobenen Anforderungen an die Sitzungsverwaltung entstanden.

Schnell wurde klar, dass die Hauptanforderungen der Sitzungsverwaltung die Herstellung der Verbindung zwischen Sitzungsserver und Teilnehmerclient und der Start der einzelnen Applikationen sind.

Aufgrund der herausgefundenen Anforderungen wurde dann eine Systemarchitektur der Sitzungsverwaltung sowohl auf Server- als auch auf Clientseite entworfen. Hierbei musste eng mit den anderen Teilnehmern des Projekts zusammengearbeitet werden, um die Schnittstellen zu den einzelnen anderen Teilen des Gesamtsystems genau festzulegen.

Nachdem nun der Entwurf der Systemarchitektur erfolgreich abgeschlossen wurde, erfolgte der Entwurf und die Implementierung der Benutzeroberflächen für die Sitzungsverwaltung. Auch hier mussten wieder zwei verschiedenen Benutzeroberflächen jeweils für die Server- und die Clientseite entworfen werden.

Wie vor allem Abschnitt 5 zeigt, wurden nicht alle in der Systemarchitektur und der Anforderungsanalyse erwähnten Komponenten und Anforderungen auch tatsächlich umgesetzt. Aufgrund der umfangreichen Anforderungen, und der zusätzlich zu

entwerfenden Benutzeroberfläche, kam es zu zeitlichen Problemen im Hinblick auf die Realisierung innerhalb der vorgegebenen Zeit.

Aus diesem Grund sind in dem Bereich Sitzungsverwaltung einige Bereiche noch nicht zur Zufriedenheit ausgebaut. Dies macht weitere Arbeiten vor allem im Bereich des Sitzungsverzeichnisses, welches aus Zeitgründen nicht implementiert wurde, und im Bereich der Anbindung der Benutzerdatenbank an eine zentrale Benutzerdatenbank und an das fakultätsweite Accountingsystem nötig.



## 8. Anhang

### 8.1 Projektplan

Titel: Prototypische Erstellung einer Sitzungsverwaltung  
für den Einsatz in Lehrveranstaltungen  
Autor: Marcus Sommer  
Datum: 01. März 2001

#### 8.1.1 Einleitung

Der Projektplan dient der/dem Bearbeiter(in) und der/dem Betreuer(in) als Grundlage für die Projektüberwachung und –steuerung. Er legt den Projektablauf aber nicht ein für allemal fest, sondern soll im Laufe der Arbeit ergänzt und angepasst werden. So ist es z.B. möglich, dass anfänglich spezifizierte Arbeitspakete weiter verfeinert oder Arbeitspakete zugunsten anderer aufgegeben werden.

Der Projektplan kann in Absprache mit der/dem Betreuer(in) in beiderseitigem Einverständnis in den folgenden Fällen angepasst werden:

- bei Verzug,
- bei vorzeitigem Abschluß eines Arbeitspakets oder
- bei Gewinn neuer Erkenntnisse.

Der aktualisierte Projektplan soll in den Anhang der Ausarbeitung einfließen.

**Projektbeschreibung:** Bei diesem Projekt handelt es sich um eine Studienarbeit an der Abteilung Verteilte Systeme des Instituts für Parallele und Verteilte Höchstleistungsrechner der Universität Stuttgart.

Zusammen mit anderen Studienarbeiten soll ein Konferenzsystem mit der entsprechenden Funktionalität erstellt werden. Im Rahmen dieser Studienarbeit sind dazu die Komponenten für einen fakultätsweiten Verzeichnisdienst und Sitzungsverwaltungen für die einzelnen Lehrveranstaltungen zu entwerfen und

prototypisch zu realisieren. Der Verzeichnisdienst verwaltet die aktuell laufenden Sitzungen. Die Verwaltung einer einzelnen Sitzung ist für die bei ihr angemeldeten Teilnehmer einschließlich ihres Sitzplatzes und die benötigten Dienste zuständig. Die Kommunikation zwischen allen beteiligten Komponenten soll drahtlos erfolgen.

Der Bearbeitungszeitraum erstreckt sich über 6 Monate, beginnend am Donnerstag, den 01. März 2001 bis zum Freitag, den 31. August 2001.

Die Studienarbeit wird betreut von Frau Dr. rer. nat. Cora Burger.

In den nachfolgenden Abschnitten werden zunächst die Arbeitspakete identifiziert und beschrieben, dann wird ein Zeitplan zu deren Durchführung festgelegt. Schließlich erfolgt die Definition der Meilensteine und der dafür zu erstellenden Dokumente.

### **8.1.2 Beschreibung der Arbeitspakete**

Dieser Abschnitt umfasst die Definition der wesentlichen Arbeitspakete, die Abhängigkeiten zwischen einzelnen Paketen und die Identifikation von kritischen Punkten, die die Durchführung des Projekts gefährden könnten.

#### **Definition der Arbeitspakete**

Die im Rahmen der Diplomarbeit durchzuführenden Tätigkeiten lassen sich in folgende, zusammenhängende Arbeitspakete aufgliedern:

- **AP1, Projektplan:** Erstellen eines initialen Projektplans  
*Status: abgeschlossen*
- **AP2, Einarbeitung:** Einarbeitung in JAVA, JSDT, existierende Architektur und Softwarewerkzeuge  
*Status: abgeschlossen*
- **AP3, Anforderungsanalyse:** Spezifizierung der Anforderungen an die Funktionalitäten einer Sitzungsverwaltung innerhalb eines Konferenzsystems  
*Status: abgeschlossen*
- **AP4, Sitzungsverwaltung:** Entwurf einer Sitzungsverwaltung aufgrund der in

AP3 herausgefundenen Anforderungen

Status: abgeschlossen

- **AP5, Verzeichnisserver:** Entwurf eines Verzeichnisservers aufgrund der in AP3 herausgefundenen Anforderungen

Status: abgeschlossen

- **AP6, Testszenarien:** Entwurf von Testszenarien zur Untersuchung der Skalierbarkeit

Status: abgeschlossen

- **AP7, Realisierung:** Prototypische Implementierung des entworfenen Systems

Status: abgebrochen (aufgrund zeitlicher Probleme keine komplette Realisierung aller Komponenten möglich)

- **AP8, Test:** Untersuchung der bereits fertigen Komponenten mittels der Testszenarien

Status: abgeschlossen

- **AP9, Ausarbeitung:** Dokumentation von Anforderungen, Architektur, Entwurf und Test

Status: abgeschlossen

- **AP10, Vortrag:** Präsentation der Ergebnisse im Rahmen des VS-Kolloquiums

Status: -

### Abhängigkeiten der Arbeitspakete

Zwischen den Arbeitspaketen der Studienarbeit bestehen die folgenden Abhängigkeiten:

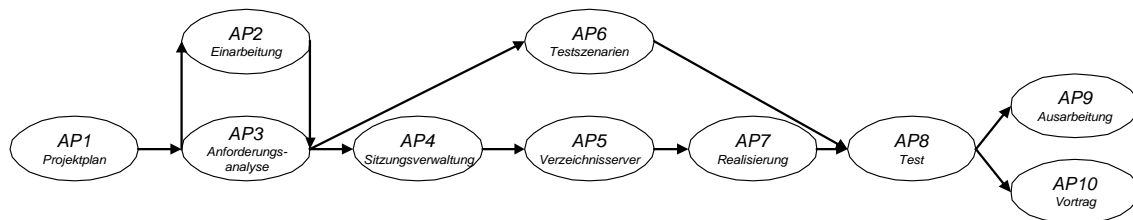


Abbildung 17: Abhängigkeiten der Arbeitspakete

Nach Aufstellen des Projektplans erfolgt die Anforderungsanalyse. Diese dient als Grundlage für den nachfolgenden Entwurf der Sitzungsverwaltung und des Verzeichnisservers. Zeitgleich mit der Anforderungsanalyse ist die Einarbeitung in die bisher aufgestellte Architektur möglich, da hierdurch weitere Anforderungen an die

Sitzungsverwaltung ersichtlich werden. Der Entwurf der Sitzungsverwaltung fördert dann wichtige Erkenntnisse zu Tage die für den Entwurf des Sitzungsverzeichnisseservers benötigt werden. Zum Beispiel stellt sich im Entwurf der Sitzungsverwaltung heraus, welche Daten im Verzeichnisseserver veröffentlicht werden müssen, um eine einfache Teilnahme an der Sitzung gewährleisten zu können. Die in den Entwürfen entwickelte Architektur hilft dann während der Realisierung des Systems weiter.

Schon während dem Entwurf der Sitzungsverwaltung und des Verzeichnisseservers und während der Realisierung treten immer wieder Fälle auf, die später zum Beispiel auf Ihre Effizienz oder Korrektheit hin getestet werden müssen. Aus diesem Grund findet die Entwicklung von Testszenarien bereits zeitgleich mit diesen Arbeitspaketen statt. Das in der Realisierung implementierte System dient dann als Grundlage für den anschließenden Test. Die herausgefundenen Anforderungen und das entworfene System werden dann schließlich in der Ausarbeitung und im Vortrag vorgestellt.

### **Kritische Punkte/Risiken**

Die Sitzungsverwaltung ist stark von den anderen Komponenten im Projekt SASCIA abhängig. Dies rührt vor allem daher, dass die Sitzungsverwaltung mit dazu benutzt wird die einzelnen Komponenten zu steuern. Das beste Beispiel sind die Applikationen welche von der Sitzungsverwaltung aus gestartet aber auch beendet werden müssen. Da diese im Projekt parallel entwickelt werden, kann es sein, dass bereits herausgefundene Anforderungen oder ein bereits bestehender Entwurf zu späteren Zeitpunkten wieder geändert werden müssen, damit die Zusammenarbeit der einzelnen Teile wirklich funktioniert.

Ein weiterer Kritischer Punkt ist der Test. Die einzelnen anderen Komponenten im Projekt SASCIA werden zum Teil nur getrennt voneinander auf ihre Funktionsfähigkeit hin getestet. Erst nach der Entwicklung der Sitzungsverwaltung ist ein Test aller Komponenten im Zusammenhang möglich. Hier ist es gut möglich, dass sich dann herausstellt, dass manche Komponenten nicht wie gedacht zusammenarbeiten. Dies würde eine Änderung der Komponenten benötigen, was wiederum eventuelle Änderungen bei der Sitzungsverwaltung nach sich ziehen könnte.

Bei der Einarbeitung in die bereits durch Peter Oberparleiter vorgegebene Architektur des Gesamtsystems kann es auch vorkommen, dass manche an eine Sitzungsverwaltung gestellten Anforderungen aufgrund der Architektur nicht verwirklicht werden können. In solch einem Fall muss überlegt werden ob die Architektur angepasst werden muss oder die Anforderung fallengelassen wird (was nur bei weniger wichtigen Anforderungen möglich ist).

### 8.1.3 Zeitplan

In diesem Abschnitt erfolgt die zeitliche Planung der Arbeitspakete und Meilensteine. Der Bearbeitungszeitraum ist 6 Monate oder 26 Wochen. Wenn sich der Zeitraum mehrerer Arbeitspakete überlappt, dann werden diese parallel bearbeitet. Am Ende des Bearbeitungszeitraums ist ein Puffer eingeplant worden, der für Verzögerungen oder andere unvorhergesehene Tätigkeiten genutzt werden soll.

geplanter Zeitraum	Aufwand	Arbeitspakete und Meilensteine	tatsächlicher Zeitraum
01.03.	1 Tag	AP 1 Projektplan	
02.03.-29.03.	4 PW	AP 2 Einarbeitung	
02.03.-29.03.	2 PW	AP 3 Anforderungsanalyse	
30.03.-26.04.	4 PW	AP 4 Sitzungsverwaltung	
27.04.-17.05.	3 PW	AP 5 Verzeichnisserver	
30.03.-19.07	1 PW	AP 6 Testszenarien	
18.05.-19.07.	9 PW	AP 7 Realisierung	
<b>Meilenstein: Prototyp</b>			
20.07.-26.07.	1 PW	AP 8 Test	
27.07.-09.08.	2 PW	AP 9 Ausarbeitung	
27.07.-09.08.	1 PW	AP 10 Vortrag	
10.08.-16.08.	1 PW	Revision/Korrekturlesen der Dokumente	
17.08.-30.08	2 PW	Puffer	
31.08.	1 Tag	Abgabe	

#### 8.1.4 Dokumente und Meilensteine

Als Ergebnis der jeweiligen Arbeitspakete soll zu den Meilensteinen folgende Dokumente entstehen, die nach diesen einer Versionskontrolle unterliegen und nur in Absprache mit der /dem Betreuer(in) geändert werden sollen.

- **Anforderungsanalyse:** Beschreibt die Anforderungen, die zur Erfüllung der Funktionalität an ein Konferenzsystem gestellt werden.
- **Entwurf:** Teilt die Anforderungen in logische Einheiten auf und gibt den Aufbau bzw. den Ablauf des Programms wieder.
- **Ausarbeitung:** Dokumentation der Anforderungen, des Entwurfs, des Prototyps und Test der vorhandenen Komponenten.