Universität Karlsruhe (TH)
Fakultät für Informatik
Institut für Telematik
Cooperation & Management


# Vereinheitlichter Zugang zu IT-Diensten am Beispiel des INFORMATIK-I-Portals

Diplomarbeit
von

**Patrick von der Hagen**

| | |
|---|---|
| Verantwortlicher Betreuer: | Prof. Dr. Sebastian Abeck |
| Betreuender Mitarbeiter: | Dipl.-Math. Klaus Scheibenberger |

Bearbeitungszeit: 01. Juli 2004 – 31. Dezember 2004

Universität Karlsruhe (TH) · Institut für Telematik · Geb. 20.20 · Zirkel 2 · 76128 Karlsruhe

# Ehrenwörtliche Erklärung

Ich erkläre hiermit, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben.

Karlsruhe, den 31. Dezember 2004

Patrick von der Hagen

# Dank

Schon Sir Isaac Newton stellte fest „*if I have seen farther it is by standing on the shoulders of giants*“ und so bin auch ich vielen Personen und Gruppen zu Dank für ihre vielfältige Unterstützung verpflichtet.

Diese Diplomarbeit wurde in der Forschungsgruppe Cooperation & Management (C&M) am Institut für Telematik der Fakultät für Informatik an der Universität Karlsruhe (TH) verfasst.

Zuallererst möchte ich mich bei Herrn Professor Sebastian Abeck bedanken, der diese Arbeit ermöglicht und durch seine Anregungen und Unterstützung zu Ihrem Gelingen beigetragen hat.

Mein besonderer Dank gilt aber auch Herrn Klaus Scheibenberger, der mich während meiner Diplomarbeit vorbildlich unterstützt und intensiv betreut hat.

Danken möchte ich auch dem IPO-Team, insbesondere Karsten Krutz und Niko Schmid, für die gute Zusammenarbeit.

Auf Seiten der ATIS möchte ich mich vor allem bei Olaf Hopp und Thomas Poisl bedanken, die auch die Ergebnisse meiner Arbeit in den Wirkbetrieb überführen werden. Sie haben meine Arbeit kritisch begleitet und wertvolle Anregungen gegeben. Auch bei der Erstellung von Machbarkeitsstudien und der Durchführung von Tests haben sie mich tatkräftig unterstützt.

Nicht unerwähnt bleiben dürfen außerdem Herr Dr. Oliver Rabe vom Zentrum für Angewandte Rechtswissenschaft, Herrn Wilhelm Sievers von der Verwaltung der Universität Karlsruhe (TH) und Herr Thomas Griesbaum von der Geschäftsführung der Fakultät für Informatik, die mich bei den Fragestellungen des *Federated Identity Management* unterstützt haben.

# Inhaltsverzeichnis

# 1 EINLEITUNG

Im Allgemeinen benötigen Dienste aller Art zur korrekten Diensterbringung gegenüber einem Nutzer eine Prüfung, ob dieser Nutzer tatsächlich zur Nutzung berechtigt ist. Diese Prüfung besteht in der Regel aus zwei Phasen. Zuerst wird der Nutzer vom Dienst identifiziert (Authentifikation) und abhängig von dieser Identifikation gewährt oder verweigert der Dienst Zugang zu bestimmten Funktionen (Autorisierung). Dies kann auch anonym erfolgen, beispielsweise indem eine Zugangskontrolle nur Ticketbesitzern den Zutritt gewährt, während ein Zutritt ohne Ticket verweigert wird. Nutzeridentifikation und eine davon abhängige Rechtevergabe sind also zentrale Bestandteile eines jeden Dienstes, sodass jeder Dienst Mechanismen benötigt, um die dazu benötigten Daten verwalten und speichern zu können.
Im Fall von Diensten, die durch den Einsatz von IT-Systemen realisiert werden (IT-Dienste), liegen die notwendigen Nutzer- bzw. Zugangsdaten in Form von Daten vor, für die jeder IT-Dienst somit entsprechende Mechanismen benötigt, um diese Art von Daten verwalten, speichern und nutzen zu können. Im Fall kollaborierender IT-Dienste ist es allerdings wahrscheinlich, dass verschiedene IT-Dienste dieselben Daten benötigen, sodass unterschiedliche Verwaltungs- und Ablagestrukturen für die verschiedenen IT-Dienste nicht nur technisch ineffizient erscheinen, sondern auch aus Betreibersicht unnötig aufwendige Verwaltungsprozesse erfordern.
Dieses Problem kann bereits im Rahmen eines einzigen IT-Dienstes auftreten, sofern dieser IT-Dienst auf kollaborierenden IT-Systemen beruht.
Im Folgenden sollen nur IT-Dienste betrachtet werden, die Bezeichnung „Dienst“ steht dementsprechend synonym für IT-Dienst.

## 1.1 Motivation

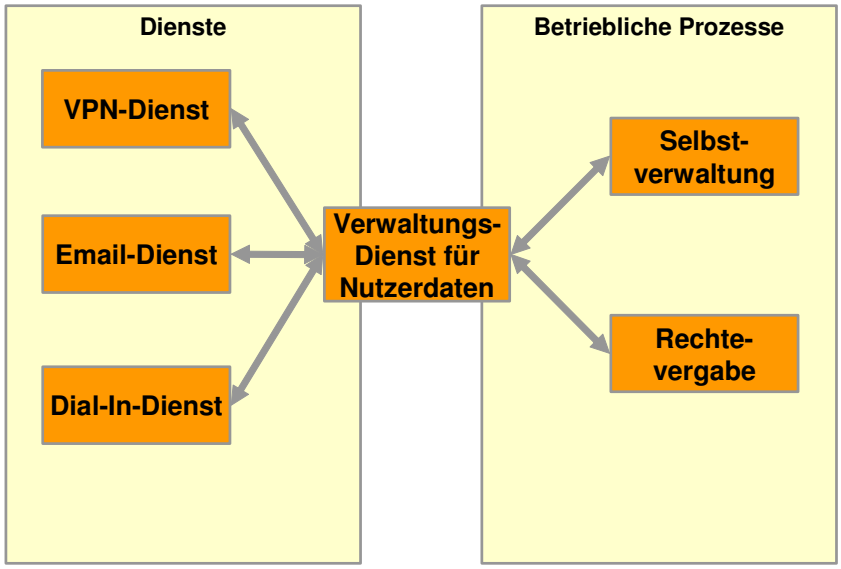
An der Fakultät für Informatik der Universität Karlsruhe (TH) werden derzeit, historisch bedingt, Nutzer- und Zugangsdaten für bestehende IT-Systeme in unterschiedlichen Datenbeständen gehalten und mit unterschiedlichen Verfahren verwaltet. Als Beispiel kann der Bereich des Studentenpools (siehe auch 6.3.4) dienen. Unter diesem Begriff werden die Dienste zusammengefasst, die den Studierenden die Möglichkeit geben, das in den Vorlesungen erworbene Wissen aufzuarbeiten und zu vertiefen. Beispielsweise gehört dazu die Bereitstellung von Computerarbeitsplätzen.
Ein Student muss zuerst im Useradm, einem an der Fakultät für Informatik eingesetzten System zur Verwaltung der Nutzerdaten für den Maildienst (E-Mail-Adresse, Login zum Abruf von Emails) registriert werden. Zur Nutzung seines Zugangs müssen jedoch weitere Einträge im Studierenden-LDAP-Server (Lightweight Directory Access Protokoll) und im Microsoft Active Directory erzeugt werden. Das *Accounting*-System für den Druckdienst der Studierenden ist bereits die vierte Stelle, die eine Registrierung der Mitglieder des Studentenpools erfordert. Bedingt durch die unkoordinierte, verteilte Datenhaltung sind daher schon in diesem vergleichsweise kleinen Szenario aufwendige und redundante Betriebsprozesse erforderlich, um die Konsistenz der Daten über die unterschiedlichen Systemgrenzen hinweg zu gewährleisten.
Aber auch bei der Verwaltung anderer Dienste der Fakultät, die den Mitarbeitern bereitgestellt werden, wie beispielsweise Mail, Dial-In und VPN (Virtual Private Network), existieren derzeit parallele Nutzerdatenbestände, woraus sich für den Betreiber verschiedene, schwer wiegende Probleme bzgl. Konsistenz und Synchronisation der Daten ergeben. Diese Schwierigkeiten sind natürlich nicht nur auf das Umfeld der Fakultät für Informatik beschränkt, sondern treten immer dort auf, wo ein Nutzer personalisierten Zugang zu heterogenen Systemen unterschiedlicher Hersteller

erhalten soll, die jeweils über eigene Verwaltungsmechanismen verfügen. Nach Untersuchungen von META Group [Me02] können die Daten eines einzelnen Mitarbeiters inzwischen in über 22 verschiedenen Datenbeständen innerhalb eines Unternehmens erfasst und gepflegt werden.
Werden die Nutzerdaten direkt an den einzelnen Systemen, die zur Diensterbringung eingesetzt werden, eingepflegt, reichen einfache Mechanismen zur Synchronisation häufig nicht aus, da unterschiedliche Bedienoberflächen und die Verwendung unterschiedlicher Datenformate zusätzliche Anforderungen an die Synchronisationsprozesse stellen. Beispielsweise können zur Nutzerverwaltung Weboberflächen, spezielle GUIs oder einfache Konfigurationsdateien verwendet werden und die jeweils zu verwaltenden Nutzerdaten können unterschiedlichen und sogar widersprüchlichen Formaten bzw. Einschränkungen unterliegen. Während ein System nur Nutzernamen mit ASCII-Zeichen erlaubt, kann ein anderes System stattdessen Unicode gestatten. Während ein System nur kurze Passwörter mit 8 Zeichen verwenden kann, hat ein anderes System keine derartigen Einschränkungen. Während ein System Zugriffsrechte über *Capabilities* verwaltet, setzt ein anderes stattdessen *Access Control Lists* ein.
Die Nutzerverwaltung dieser Systeme kann durch ein manuelles Vorgehen nicht effizient durchgeführt werden, während gleichzeitig die unterschiedlichen Schnittstellen und Datenformate die Automatisierung sehr erschweren. Dennoch muss schon allein aus Produktivitätsgründen eine effektivere Lösung gefunden werden, die konsistente Daten garantieren kann und die unterschiedlichen Systeme und Konzepte gegenüber der Verwaltung verschattet.
Insbesondere durch eine fehlende zentrale Sicht auf das Gesamtsystem können inkonsistente Daten entstehen, die dem Betreiber vielfältige Probleme verursachen können. So könnte einem neuen Nutzer von einem bereits aktualisierten System aus eine Begrüßungsnachricht gesendet werden, die der mit veralteten Daten operierende Mail-Dienst ablehnt.
Gravierender können durch Inkonsistenzen verursachte Sicherheitsprobleme sein, da aufgrund einer fehlenden zentralen Sicht auch kein zentrales Audit geführt werden kann, aus dem ersichtlich wäre, welchem Nutzer für welche Systeme Rechte zugewiesen wurden und über welche Rechte ein Nutzer gegenwärtig verfügt. Als Folge wäre es möglich, dass ein Nutzer, der aufgrund einer Exmatrikulation aus dem Studentenpool ausscheidet, zwar aus dem Active Directory entfernt wird, nicht aber im Mailsystem deaktiviert wird, sodass er diesen Dienst noch längere Zeit unberechtigt nutzen kann. Allgemein stellt die Verwaltung von Zugriffsrechten und Nutzerrollen hohe Anforderungen an die Verwaltungsprozesse des Betreibers.
Es ist daher zu überlegen, ob ein zentraler Verwaltungsdienst für Nutzerdaten entwickelt werden kann, der sowohl den technischen als auch den betrieblichen Erfordernissen gerecht wird.
Zu diesen Anforderungen an den Verwaltungsdienst gehören aus technischer Sicht die Kompatibilität zu existierenden Diensten, Zuverlässigkeit und Performance, die betrieblichen Anforderungen umfassen einfache und effiziente Schnittstellen, die Vermeidung von redundanten Aufgaben innerhalb der (Betriebs-) Prozesse (ein User soll nur einmal zentral für alle Dienste angelegt werden) und ausreichende Mächtigkeit, um alle Verwaltungsanforderungen abbilden zu können.
Dabei darf nicht übersehen werden, dass durch eine Zentralisierung von Nutzerdaten auch neue Probleme entstehen können. Während sich zum einen die derzeit auf verschiedene Dienste verteilte Last auf nur noch einen einzelnen Dienst konzentrieren würde, käme es gleichzeitig zur Abhängigkeit aller Dienste von der zentralen

Verwaltung und ein *Single-Point of Failure* (SPoF) würde entstehen. Diese Probleme sind daher zu berücksichtigen und gegen die erwarteten Vorteile abzuwägen.
Im Folgenden werden wir unter der Bezeichnung „Nutzerverwaltung“ die Nutzerdatenbestände und die zugehörigen Verwaltungswerkzeuge zusammenfassen.

## 1.2 Ziel der Arbeit

Die in 1.1 angesprochenen Probleme im Bereich der Nutzer- und Zugangsdatenverwaltung treten bereits im Umfeld kleiner Szenarien, wie beispielsweise einzelnen Webportalen, auf, ebenso aber auch an der Fakultät für Informatik und allgemein an allen größeren Institutionen, die heterogene Systeme mit jeweils eigenen Nutzerverwaltungen einsetzten.
Das Ziel der Arbeit ist es daher, eine vereinfachte Verwaltung der Nutzerdaten zu erarbeiten, sodass diese als ein Dienst über Standardschnittstellen bereitgestellt werden kann. Dabei sollen die zum Betrieb erforderlichen Prozesse vereinfacht und die Qualität der verwalteten Daten bzgl. Konsistenz und Aktualität verbessert werden.



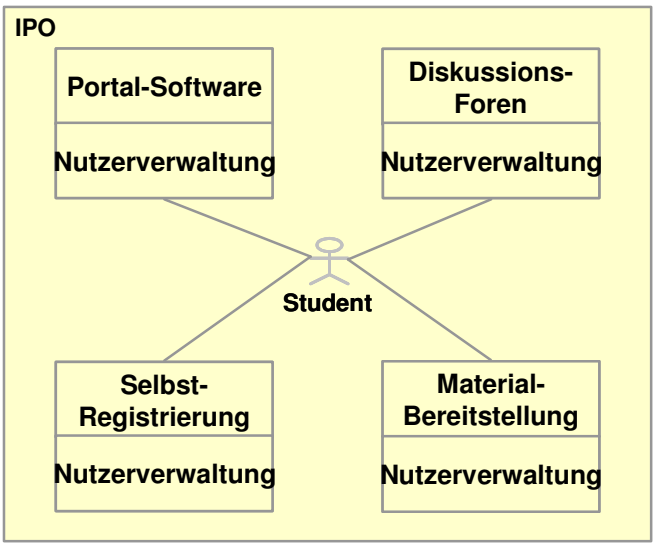

**Information 1: Technische vs. betriebliche Sicht**

Um diese Zielstellung zu erreichen, soll exemplarisch das INFORMATIK-I-Portal (IPO) untersucht werden, da es sich hier um ein eng umrissenes und klar definiertes Beispiel im Rahmen der Fakultät für Informatik handelt. Da es sich noch in der Entwicklung befindet, existieren hier insbesondere keine historisch bedingten Datenbestände, auf die Rücksicht genommen werden müsste.
Am Beispiel eines Angehörigen der Fakultät soll später untersucht werden, ob dieser aus vergleichsweise einfachen Randbedingungen entstandene IPO-Verwaltungsdienst auch in größeren Szenarien an der Fakultät für Informatik eingesetzt werden kann bzw. welche Anpassungen und Erweiterungen erforderlich werden. Der Schwerpunkt liegt dabei sowohl auf technischen Aspekten der bereitgestellten Dienste, wie der Integration mit bestehenden Systemen und Datenbeständen, als auch auf Anforderungen, die aus den betrieblichen Prozessen resultieren. Daher kann der Verwaltungsdienst als Schnittstelle zwischen den Diensten und den betrieblichen Prozessen eingeordnet werden. Ziel ist ein an Diensten orientiertes Konzept zur Verwaltung von Nutzerdaten durch den Betreiber.

Dazu ist zu untersuchen, in welchen Diensten Nutzer- und Zugangsdaten benötigt werden, mit welchen Systemen die Verwaltung erfolgt und wie die vorgefundenen Systeme bzgl. der Nutzerdaten miteinander kooperieren. Des Weiteren müssen die existierenden Dienste, die diese Daten nutzen, identifiziert und bezüglich ihrer spezifischen Anforderungen analysiert werden. Auf dieser Grundlage lässt sich dann ein koordinierter Verwaltungsdienst entwickeln, um die Nutzerdaten effizient verwalten und sämtlichen Diensten verfügbar machen zu können. Insgesamt ist diese Untersuchung in das Problemfeld des *Identity Management* einzuordnen.

### 1.2.1 Beispielszenario INFORMATIK-I-Portal

Im Rahmen des IPO sollen verschiedene Anwendungen zusammen eine Dienstleistung erbringen. Das konkrete Ziel ist es, die INFORMATIK-I-Veranstaltung zusammen mit der zugehörigen Übung und den Tutorien durch das IPO zu unterstützen. Eine detaillierte Untersuchung des IPO findet sich in [Sc04].
Das IPO soll von verschiedenen Nutzergruppen wie z. B. „Mitarbeiter“, „Tutoren“ und „Studenten“ genutzt werden, die für die unterschiedlichen Komponenten unterschiedliche Rechte erhalten werden.
Durch eine Komponente zur Materialbereitstellung sollen vorlesungsunterstützende Materialien (Ldocs, Referenzen, etc.), von den Mitarbeitern veröffentlicht und von den Studierenden abgerufen werden können.
Da diese bereitgestellten Materialien auch diskutiert werden sollen, ist eine weitere Komponente zur Kommunikation erforderlich. Das betrifft sowohl die Kommunikation der Mitarbeiter untereinander, als auch die Diskussion innerhalb der Tutorien oder unter den Studierenden. Dazu werden Diskussionsforen benötigt, die aus Gründen der Benutzerführung in den Portalkontext einzubinden sind.
Um schließlich die unterschiedlichen Nutzergruppen effektiv unterstützen zu können, ist eine geeignete Portal-Software erforderlich, die eine personalisierte Nutzerführung bereitstellen kann.
Aufgrund der hohen Studierendenzahlen ist es aus betrieblicher Sicht außerdem erforderlich, das sich die Studierenden zur Nutzung des Portals selbstständig unter Angabe persönlicher Daten registrieren können, sodass auch für diese Funktionalität eine Komponente erforderlich ist.

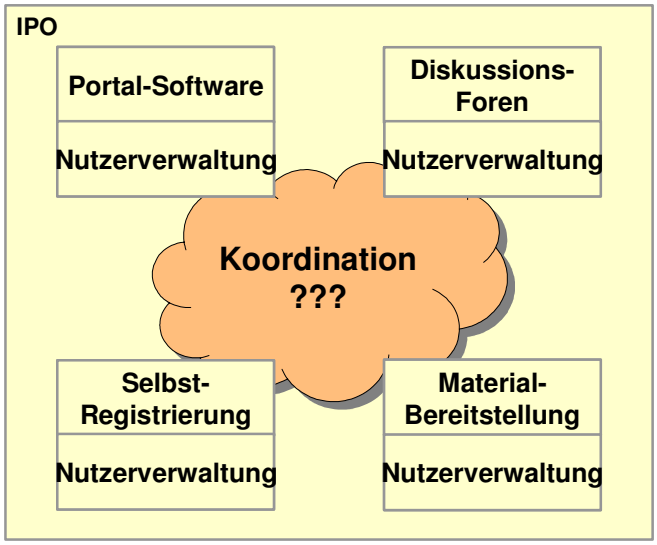



**Information 2: IPO: Getrennte Nutzerverwaltung**

Da die Studierenden die zahlenmäßig stärkste Nutzergruppe stellen werden, sollen sie hier als Beispiel dienen. Ein Studierender kann sich selbst registrieren und soll daraufhin vom Portal als Student erkannt werden. Im Forum für Studierende oder im Forum für sein Tutorium soll er schreibend teilnehmen können, während er auf andere Foren keinerlei Zugriff erhalten wird. Einen Teil der Materialien darf er lesen, weitergehende Rechte innerhalb der Materialbereitstellung werden ihm nicht gegeben.
Wie aus Information 2 ersichtlich ist, verfügt jeder der zuvor genannten Bestandteile des IPO über eigene Verwaltungswerkzeuge und eigene Bestände an Nutzerdaten, um die Nutzer personalisiert unterstützen zu können. Dabei muss jeder Nutzer, der auf das Portal zugreifen können soll, in jeder Nutzerverwaltung mit Informationen zur Identifikation und den zugewiesenen Zugriffsrechten eingetragen werden. Da die einzelnen Dienstkomponenten kollaborieren werden, ist eine Koordination dieser Nutzerverwaltungen im Rahmen von Betriebsprozessen nötig, die durch die Verwendung einheitlicher standarisierter Schnittstellen unterstützt werden soll.

### 1.2.2 Problemstellung am Beispiel IPO

Schon an dem vergleichsweise einfachen Beispiel des IPO wird deutlich, dass durch die getrennten Nutzerverwaltungen der eingesetzten Komponenten eine redundante Datenhaltung erfolgt, die durch Betriebsprozesse koordiniert werden muss. Bei der Selbst-Registrierung eines Nutzers etwa muss automatisch für jede der beteiligten Komponenten ein neuer Nutzer angelegt werden. Auch in anderen Anwendungsfällen wie beispielsweise einer Passwortänderung oder der Sperrung eines Nutzers muss diese Änderung schnell und zuverlässig an alle beteiligten Komponenten propagiert werden.




**Information 3: IPO: Nutzerverwaltung**

Der durch die unterschiedlichen, parallelen Nutzerverwaltungen der verschiedenen Komponenten entstehende Koordinationsaufwand ist erheblich und muss durch den Betreiber in Form von individuell angepassten und in der Regel aufwendigen Betriebsprozessen bewältigt werden muss.
Schon die Koordination von nur zwei Nutzerverwaltungen ist sehr aufwendig, da auch bei einer vorübergehenden Nichtverfügbarkeit einer der beteiligten Nutzerverwaltungen die Propagation von Änderungen garantiert werden muss. Um Datenverluste zu vermeiden und Konflikte erkennen und beheben zu können, müssen daher Mechanismen wie Warteschlangen und Transaktionsschutz implementiert werden.

Da bei einer Änderung der Nutzerdaten jede der eingesetzten Nutzerverwaltungen aktualisiert werden muss, steigt die Komplexität dieser Koordination dabei mindestens linear mit der Zahl der eingesetzten Nutzerverwaltungen. Die Änderungen von Nutzerdaten müssen jedoch nicht zwangsläufig von einer speziellen Komponente durchgeführt werden, von der aus alle übrigen Nutzerverwaltungen aktualisiert werden können, sodass der tatsächlich zu erwartende Aufwand deutlich höher liegt. Fehlende Standardschnittstellen zur Nutzerverwaltung sowie unterschiedliche Verwaltungsmechanismen können die Komplexität weiter erhöhen. Im Extremfall kann in jeder beteiligten Komponente eine Änderung auftreten, die an jede andere Komponente propagiert werden muss, sodass der zur Koordination erforderliche Aufwand quadratisch mit der Anzahl der verwendeten Nutzerverwaltungen wachsen kann.
Ein solcher Aufwand auf Betreiberebene ist offensichtlich ineffizient und daher nach Möglichkeit zu vermeiden. Dies kann zum einen durch eine Optimierung der Synchronisationsprozesse realisiert werden, zum anderen ist auch eine Minimierung der eingesetzten Nutzerverwaltungen denkbar, indem beispielsweise mehrere Komponenten eine gemeinsame Nutzerverwaltung verwenden.
Es wird daher für das IPO eine Nutzerverwaltung benötigt, die die angesprochenen Probleme verteilter Nutzerverwaltungen vermeiden bzw. durch Koordinationsmechanismen umgehen kann. Die Nutzerverwaltung soll zudem eine feingranulare, nutzerbezogene Rechteverwaltung ermöglichen, bei der den unterschiedlichen Anwendern für die verschiedenen Komponenten unterschiedliche Rechte zugewiesen werden können. Beispielsweise soll Tutoren und Mitarbeitern der Zugriff auf mehr Materialien möglich sein, als einem Studierenden.

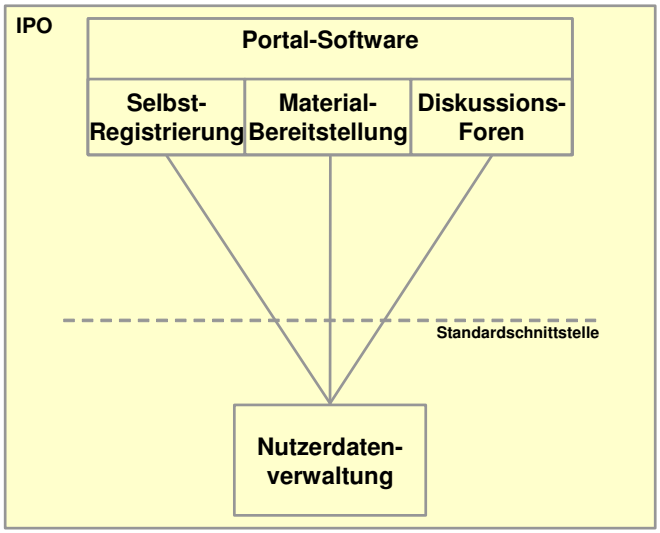
## 1.3 Die Aufgabe: Identity Management

Die Problematik der effizienten Nutzerverwaltung ist sicherlich nicht neu, doch hat sie sich durch den fortschreitenden Einsatz von IT-Systemen im Laufe der Zeit erheblich verschärft. Nach einer Untersuchung von META Group [Me02] werden die Daten eines einzelnen Mitarbeiters in großen US-Unternehmen (über 500 Millionen Dollar Umsatz) im Durchschnitt in 22 verschiedenen Datenbeständen erfasst und gepflegt. Um in einem solchen Umfeld eine effiziente und konsistente Verwaltung realisieren zu können, reichten manuelle Synchronisationsansätze nicht mehr aus und es wurde im Zusammenhang mit dieser Problematik der Begriff *Identity Management* geprägt. Dabei bezeichnet *Identity Management* die Aufgabe, verteilte Nutzerdaten konsistent zu pflegen. Microsoft [HB+04] definiert die Aufgaben des *Identity Management* wie folg: *Identity management covers information that relates to individuals. Identity management includes the management of computer user accounts, the contact details of those user accounts, door entry system user accounts, application user accounts, e-mail system user addresses and accounts, and much more.* Nach dieser Definition steht also der einzelne Nutzer als Individuum im Mittelpunkt des Interesses.
Spencer Lee [Le03] legt einen Schwerpunkt auf *Identity Management* als Prozess: *Identity management refers to the process of employing emerging technologies to manage information about the identity of users and control access to company resources. The goal of identity management is to improve productivity and security while lowering costs associated with managing users and their identities, attributes, and credentials.* Dabei umfasst der Prozess des *Identity Management* sowohl das Einrichten von Nutzern, als auch die kontinuierliche Pflege der Nutzerdaten. Das *Identity Management* endet nicht notwendigerweise mit der Deaktivierung der Nutzeraccounts, da die Nutzerdaten, beispielsweise die Tatsache, dass die zugehörigen Account deaktiviert worden sind, auch weiterhin von Interesse sein können.

Besonders mit der Idee des Single-Sign-On (SSO), bei dem sich ein Nutzer mit einer Nutzerkennung an allen eingesetzten Systemen anmelden können soll, gewann das *Identity Management* als unverzichtbare Grundlage an Bedeutung.

## 1.4 Lösungsansatz

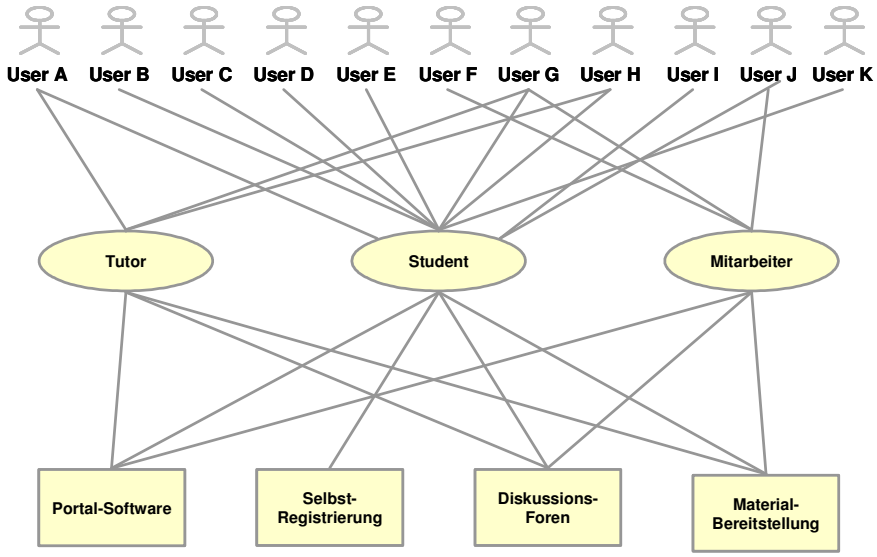
Beim Einsatz heterogener Systeme müssen häufig parallele und redundante Nutzer- und Zugangsverwaltungen betrieben und durch den Betreiber gepflegt werden, da effiziente Koordinations- oder Synchronisationsmechanismen nicht verfügbar sind. Da diese unterschiedlichen Nutzer- und Zugangsverwaltungen über keine gemeinsamen Standardschnittstellen verfügen, sind automatische Verfahren nur individuell und damit sehr aufwendig zu entwickeln. Die Folge sind ein hoher Aufwand und hohe Kosten für den Betreiber. Insbesondere wenn zu einem späteren Zeitpunkt weitere Komponenten integriert werden sollen, kann ein hoher Aufwand entstehen, der von der Anzahl der eingesetzten Nutzerverwaltungen abhängt.
Es ergibt sich daher die Überlegung, ob der Betreiber durch eine gemeinsame, zentrale Nutzerdatenverwaltung und eine einheitliche, standarisierte Schnittstelle zwischen den jeweiligen Anwendungen und dem Verwaltungsdienst entlastet werden kann. Damit sollten auch die Beziehungen zwischen den einzelnen Anwendungen vereinfacht und die Betriebsprozesse verbessert werden können.
Dieser Ansatz ist in Information 4 skizziert und soll im Folgenden untersucht werden.




**Information 4: IPO: gemeinsame Nutzerverwaltung**

Als Folge würde sich die Skalierbarkeit durch die Reduzierung der Schnittstellen gegenüber den bisherigen Ansätzen deutlich verbessern und eine redundante Datenspeicherung würde nicht nur koordiniert, sondern nach Möglichkeit vermieden.
Allerdings muss trotz der zunehmenden Zentralisierung eine optimale Zuverlässigkeit und Verfügbarkeit gewährleistet werden, beispielsweise durch eine redundante Auslegung der Datenspeicherung. In diesem Bereich finden derzeit Verzeichnisdienste unter Nutzung von Replikationsmechanismen Verwendung.
Besondere Vorteile entstehen, wenn etwa zusätzliche Komponenten an das zentrale Nutzerdatenverzeichnis angebunden werden sollen, da die Anbindung über standardisierte Schnittstellen und völlig unabhängig von den bereits eingesetzten Komponenten erfolgen kann. Der Aufwand zur Koordinierung ist also unabhängig von der Anzahl der bereits eingesetzten Anwendungen.

Zur Vereinfachung der Verwaltung bietet es sich außerdem an, nach Möglichkeit nicht den einzelnen Nutzern direkt die Rechte für die einzelnen Komponenten zuzuordnen, sondern von den Nutzern zu abstrahieren, indem man zuerst die Rollen zu identifiziert, unter denen die Nutzer gegenüber dem Portal auftreten und diesen Rollen die nötigen Rechte zuweist.




**Information 5: IPO: Rollen-Abstraktion**

Die einzelnen Anwendungen sollten dann diese Rolleninformationen nutzen, um anwendungsspezifisch Rechte zu vergeben. Da eine solche Abstraktion den Verwaltungsaufwand auf Betreiberseite weiter reduziert kann, sollen geeignete Mechanismen von der zu entwickelnden Nutzerverwaltung unterstützt werden.

## 1.5 Erzielte Ergebnisse

Im Rahmen dieser Diplomarbeit wurden Fragestellungen im Rahmen des *Identity Management* im Umfeld der Fakultät für Informatik untersucht. Ein erster Schwerpunkt lag auf der Realisierung einer integrierten Nutzerverwaltung für das IPO, die sowohl die Anpassung der verwendeten Portalsoftware Jetspeed, als auch die Entwicklung einer eigenständigen Registrierungskomponente umfasste, die in [Ha04] dokumentiert ist. Über 650 Nutzer haben sich inzwischen erfolgreich zur IPO-Nutzung registriert, sodass von einer erfolgreichen Überführung in den Wirkbetrieb gesprochen werden kann.
Für die Abteilung Technische Infrastruktur (ATIS) wurde ein Konzept zur Einführung einer an Diensten orientierten Nutzerverwaltung erarbeitet. Erste Schritte zur Implementierung der erforderlichen Werkzeuge wurden bereits, basierend auf dem entwickelten Konzept, von den Mitarbeitern der ATIS erfolgreich umgesetzt, sodass nicht nur ein Nachweis der Tragfähigkeit erbracht wurde, sondern die Umsetzung bereits begonnen hat.
Auch für weitere Entwicklungsschritte im Rahmen eines Federated Identity Management wurden viel versprechende Fälle identifiziert und analysiert. Auf Basis einer neuen Nutzerverwaltung liegen auch hier die erforderlichen Konzepte vor, um eine sinnvolle Koppelung durchzuführen. Dabei hat sich die Betrachtung nicht nur auf technische oder betriebliche Aspekte beschränkt, sondern auch Fragestellungen des Datenschutzes wurden berücksichtigt.
Der Bereich des *Identity Management* konnte im Rahmen dieser Diplomarbeit sicherlich nicht abschließend untersucht werden, doch wurde eine sinnvolle Basis

geschaffen, von der aus die Untersuchungen vertieft werden können. Das gilt auch in besonderem Maße für Fragestellungen des *Federated Identity Management*, beispielsweise in Bezug auf die *Security Assertion Markup Language* (SAML) oder die *Web Service Federation Language* (WSFL).

## 1.6 Gliederung der Arbeit

Nach der allgemeinen Einführung in Problemstellung und Motivation durch Kapitel 1 folgt in Kapitel 2 ein Überblick über mögliche Lösungsmöglichkeiten, die in den letzten 20 Jahren entwickelt wurden und ihre Anwendbarkeit auf die gegebene Problemstellung. In diesem Zusammenhang erfolgt auch eine tiefere Einführung in existierende Verzeichnisdienste. Besonders Verzeichnisdienste nach LDAP-Standard werden wir in Kapitel 3 intensiv untersuchen.
Kapitel 4 gibt eine detaillierte Einführung in Problematiken im Zusammenhang mit dem INFORMATIK-I-Portal und erörtert eine mögliche Lösung. Aus den gewonnenen Erkenntnissen leiten wir in Kapitel 5 einen nutzerzentrierten Verwaltungsansatz ab, den wir in Kapitel 6 mit Szenarien für *Identity Management* an der Fakultät für Informatik überprüfen. Die in Kapitel 4 gewonnenen Erkenntnisse sollen dabei angewandt und weiter entwickelt werden. In Kapitel 7 gehen wir über die Grenzen der Fakultät für Informatik hinaus und untersuchen Fälle des *Federated Identity Management* zwischen der Fakultät und anderen Geschäftsbereichen an der Universität Karlsruhe.
In Kapitel 8 fassen wir zusammen, welche Änderungen sich an den Betriebsabläufen ergeben und wagen schließlich in Kapitel 9 einen Ausblick.

# 2 BESTEHENDE LÖSUNGSANSÄTZE

## 2.1 Überblick

Um zentrale Verwaltungsstrukturen von Nutzer- und Zugangsdatenbeständen zu realisieren, werden schon seit vielen Jahren Verzeichnisdienste eingesetzt. Im Unix-Umfeld wurde das von Sun Microsystems entwickelte und 1985 veröffentlichte Network Information System (kurz NIS) zum Standard, der auch von den Produkten anderer Hersteller unterstützt wird. Microsoft veröffentlichte 1987 unter der Bezeichnung LAN Manager einen ähnlichen Dienst für Microsoft-basierte Netzwerke, der erst mit der Entwicklung und Veröffentlichung von Microsoft Windows 2000 durch das Active Directory ersetzt wurde. Eine Unterstützung des LAN Manager auf anderen Plattformen wie Unix oder Apple Macintosh wurde nicht entwickelt.

Ein Standard, der systemübergreifend akzeptiert und unterstützt wird, konnte sich lange Zeit nicht entwickeln. Diese Situation hat sich jedoch spätestens 1997 mit der Verabschiedung der Version 3 des *Lightweight Directory Access Protocols* (kurz LDAP) geändert. Dabei wurde LDAP nicht mit dem Ziel entwickelt, ein Zugangssystem zu ermöglichen, sondern ursprünglich als Zugangsprotokoll zu existierenden Verzeichnisdiensten nach X.500-Standard konzipiert. Erst später wurden LDAP-Server entwickelt, die selbst universelle Verzeichnisdienste bereitstellten.

LDAP hat sich in den vergangenen Jahren weit gehend unbemerkt als zentrale Komponente von Systemen zur Nutzer- und Zugangsverwaltung etabliert. Beispielsweise wären Microsoft Active Directory, Novell Netware oder Sun Portal Server ohne LDAP-Verzeichnisdienst kaum denkbar. Auch die Produkte zur Lösung der Problematik des „*Identity Management*“ (siehe 1.3), beispielsweise von Sun Microsystems und Novell entwickelt, verwenden LDAP-Verzeichnisdienste zur Datenhaltung.

Damit existiert nun erstmals ein offenes, standarisiertes Protokoll, das inzwischen von allen etablierten Betriebssystemen und nahezu allen Programmiersprachen verwendet werden kann, um auf Verzeichnisdaten zuzugreifen.

Leider wird die verfügbare Dokumentation der Entwicklung im LDAP-Bereich kaum gerecht. Die in den letzten Jahren erschienenen Fachbücher behandeln überwiegend spezifische Produkte oder die LDAP-Programmierschnittstellen (beispielsweise [KL03] und [Ca03]), doch insbesondere zu Themen aus Sicht von Verwaltung, Betrieb und Entwurf von LDAP-Verzeichnissen existieren nur wenige Veröffentlichungen. Dennoch sind gerade diese Fragestellungen für eine erfolgreiche Einführung und einen sinnvollen Einsatz von Verzeichnisdiensten wesentlich. Eine empfehlenswerte Ausnahme bildet [HS+03].

Die aufgeführten Ansätze sollen im Folgenden untersucht werden, um Anregungen für eine eigene Lösung erarbeiten zu können. Betrachtet werden dabei die Zukunftssicherheit und die Eignung für einen Einsatz im IPO-Szenario oder an der Fakultät für Informatik.

Ein Schwerpunkt der Untersuchung wird dabei auf dem Einsatz von Verzeichnisdiensten nach dem LDAP-Standard liegen, die in Kapitel 3 ausführlich analysiert werden.

Eine gute Einführung in die Probleme der Nutzerdatenverwaltung unter Microsoft Windows und Unix findet man beispielsweise in [HB+04]. Dort werden beispielsweise proprietäre Ansätze, NIS und LDAP-Verzeichnisdienste als Lösungsmöglichkeiten vorgestellt.

## 2.2 Proprietäre Synchronisation

Existieren bereits verschiedene Datenbestände zur Nutzerverwaltung, so ist es möglich, individuelle Synchronisationsverfahren für genau diese Datenbestände im speziellen Umfeld zu entwickeln und einzusetzen. Dabei wird die Existenz verschiedener Datenbestände akzeptiert und lediglich um Mechanismen zur Koordination und Synchronisation erweitert.
Ein solches Vorgehen ist schon bei wenigen Diensten sehr aufwendig zu realisieren. Da für jedes einzubindende System aufgrund fehlender Standardisierung eine neue Schnittstelle entwickelt werden muss, fehlen Skalierbarkeit und Zukunftssicherheit.
Insbesondere wird es bei mehreren, verteilten Datenbeständen schwierig, Zuverlässigkeit und Transaktionsschutz zu gewährleisten. Unterschiedliche Semantiken oder Zeichenkodierungen sind weitere Probleme, die zu hohem Aufwand führen. Insbesondere wenn zwei Systeme unterschiedliche Einwege-Hash-Funktionen zur Vermeidung der Speicherung von Passwörtern im Klartext verwenden, können hier große, nahezu unlösbare Probleme entstehen.
Im IPO wären proprietäre Synchronisationsmechanismen zwar aufgrund der geringen Anzahl beteiligter Systeme noch umsetzbar, aber bei möglichen Erweiterungen aufgrund fehlender Standarisierung und Schnittstellen nicht auf Dauer tragfähig. An der Fakultät für Informatik werden derzeit verschiedene proprietäre Synchronisationsmechanismen, beispielsweise im Rahmen der Abteilung Technische Infrastruktur (ATIS), eingesetzt. Aufgrund der hohen Komplexität und der fehlenden Skalierbarkeit wird jedoch aktuell überlegt, ob und wie diese Systeme durch standardisierte Mechanismen ersetzt werden können.

## 2.3 Verzeichnisdienste

Unter einem Verzeichnisdienst versteht man einen Dienst, der über ein standardisiertes Zugriffsprotokoll Daten bereitstellt. Dabei geht man von weit gehend statischen Daten aus, d. h., es werden wesentlich mehr Lesezugriffe als Schreiboperationen (hinzufügen, ändern oder löschen von Daten) erwartet. Obwohl Verzeichnisdienste beliebige Daten speichern können, werden wir im Folgenden primär die Verwaltung von Nutzerdaten betrachten. Dazu konzentrieren wir uns auf die universell einsetzbaren Verzeichnisdienste und werden Spezialfälle wie beispielsweise DNS nicht weiter betrachten.
Zur Organisation der gespeicherten Daten in einem Verzeichnisdienst werden vergleichsweise einfachen Strukturen verwendet. Bei NIS (siehe 2.4) wird auf unverknüpfte Tabellen zurückgegriffen, bei LDAP (siehe 3) werden die Daten als Baumstruktur organisiert. Aufwendigere Strukturen wie beispielsweise Beziehungen zwischen Tabellen, Fremdschlüsseldefinitionen oder Verknüpfungen bei Abfragen (Join-Operation) etc. existieren in einem Verzeichnisdienst in der Regel nicht.
Um die Zuverlässigkeit und die Zugriffsgeschwindigkeit zu erhöhen, werden in der Regel mehrere Verzeichnisserver eingesetzt, die Änderungen ihrer Datenbestände durch Nutzung von Replikationsmechanismen abgleichen. Anders als bei Datenbanken wird bei dieser Replikation ein Transaktionsschutz nicht als wesentlich angesehen. Eine einzelne Operation auf einem einzelnen Verzeichnis und einem einzelnen Eintrag wird atomar ausgeführt, doch sind keine weitergehenden Zusicherungen durch den Verzeichnisdienst vorgesehen. Daher können vorübergehend Inkonsistenzen zwischen den einzelnen Datenbeständen entstehen, was bei Verzeichnisdiensten in der Regel akzeptiert wird, sofern Mechanismen existieren, die dazu führen, dass schließlich eine Synchronisation erfolgt und ein identischer Zustand erreicht wird.

Anders als bei Datenbanken entfällt bei Verzeichnisdiensten durch standarisierte Zugriffsprotokolle die Notwendigkeit, spezielle Treiber über Bibliotheken auf den verwendeten Clienten zu installieren und zu pflegen.
Im Gegensatz zu den Verzeichnisdiensten unterstützen die gängigen SQL-Datenbanken Tabellenstrukturen mit aufwendigen Beziehungen, automatischen Konsistenzprüfungen und komplexe, verknüpfte Abfragen. Transaktionen können garantieren, dass auch aufwendige Änderungen, die mehrere Einträge betreffen, vollständig oder überhaupt nicht durchgeführt werden. Spezielle Annahmen über Lese- u. Schreiboperationen sind dabei nicht möglich, daher geht man bei der Entwicklung von Datenbanken in der Regel davon aus, dass diese Operationen gleich gewichtet werden können.
Vergleicht man die Annahmen, die einem Verzeichnisdienst zugrunde liegen, mit denen einer SQL-Datenbank, so ergeben sich verschiedene Vereinfachungen und Einschränkungen, die die Hersteller von Verzeichnisdiensten nutzen können, um Optimierungen durchzuführen, die bei SQL-Datenbanken nicht möglich wären. Sofern diese Annahmen tatsächlich zutreffen (insbesondere das Ungleichverhältnis von Lese- zu Schreiboperationen), können Verzeichnisdienste deutliche Leistungsvorteile erreichen.
Im Bereich des *Identity Management* (siehe 1.3) kann man davon ausgehen, dass Änderungen für ein spezielles Attribut nur von einem einzelnen, autoritativen System ausgehen und auch jeweils nur eine einzelne Identität betreffen, sodass aufwendige Mechanismen zum Transaktionsschutz nicht erforderlich sind. Wichtig ist jedoch eine sehr hohe Geschwindigkeit und Zuverlässigkeit des Dienstes. In großen Szenarien, etwa bei Unternehmen mit verteilten Standorten, ist es erforderlich, an jedem Standort zuverlässigen und schnellen Zugriff auf die Daten zu erhalten, sodass es sich anbietet, an jedem Standort die zum *Identity Management* erforderlichen Daten vorzuhalten. In der Praxis werden dazu häufig Verzeichnisserver verwendet, die über Replikationsmechanismen aktualisieren die Datenbestände aktualisieren können.
Schließlich ist damit zu rechnen, dass die zu erfassenden Informationen tatsächlich in der Regel statischer Natur sind und sich nur selten ändern (etwa Mitarbeiternummer, Raumzuordnung, Abteilungszugehörigkeit, etc.). Setzt man dies in Bezug zu den diskutierten Unterschieden zwischen allgemeinen Datenbanken und Verzeichnisdiensten, so erkennt man, dass Verzeichnisdienste tatsächlich besser in der Lage sind, die Anforderungen des *Identity Management* zu erfüllen.

## 2.4 Network Information System (NIS)

Beim Network Information System handelt es sich um einen von Sun Microsystems entwickelten Standard, um die Nutzer- und Zugangsverwaltung im Unix-Bereich für unterschiedliche Dienste zu zentralisieren. Es baut dazu auf den 1985 von Sun Microsystems eingeführten „Remote Procedure Calls“ (RPC) auf. Die Bezeichnung lautete ursprünglich „*Yellow Pages*“ bzw. „YP“ und wurde später in „*Network Information System*“ bzw. „NIS“ umbenannt. Kern von NIS sind so genannte „*Maps*“, die eine Abbildung von einem Schlüssel auf die zugehörigen Daten realisieren. Dabei werden für jeden erwarteten Anfragetyp eigene *Maps* gepflegt.
Auf einem dedizierten „*Master*-Server“ können Änderungen an den Datentabellen durchgeführt werden. Diese Änderungen können vom *Master*-Server auf mehrere *Slave*-Server repliziert werden, die zur Lastverteilung und Sicherung der Verfügbarkeit genutzt werden können. Änderungen werden nach Möglichkeit sofort an die *Slave*-Server propagiert. Ist ein *Slave*-Server nicht erreichbar, so erhält er die Änderungen erst bei einer der nächsten periodischen Synchronisationen, es liegt hier also eine „schwache Konsistenz“ vor (siehe auch 3.4.6).

Leider werden bei dieser Synchronisation nicht nur inkrementelle Änderungen innerhalb einer *Map* repliziert, sondern immer der komplette Datenbestand, sodass dieses Protokoll bei großen *Maps* ineffizient ist. Außerdem erfolgt diese Übertragung aufgrund der Einschränkungen durch den zugrunde liegenden RPC-Mechanismus unverschlüsselt und ohne Authentifikation der Kommunikationspartner, sodass einfache Angriffe, z. B. passiv durch Abhören des Netzwerkverkehrs oder aktiv durch *IP-Spoofing*, möglich werden.
Generell werden Zugriffsrechte vom NIS-System nur Rechner-basiert vergeben, ohne zwischen den einzelnen Nutzern zu differenzieren. Es ist ebenfalls nicht möglich, den Zugriff nur auf bestimmte Attribute einer *Map* oder auf bestimmte Einträge zu beschränken.
Die verwendeten *Maps* werden aus ASCII-Quelldateien erzeugt, wobei für jeden gewünschten Schlüssel eine eigene Indexdatei generiert werden muss. Eine Verknüpfung der einzelnen *Maps* existiert nicht, sodass dem Verzeichnisdienst nur ein flacher Namensraum zugrunde liegt. Suchanfragen über mehrere Attribute können die Indizes der einzelnen *Maps* nicht nutzen und auch Abbildungen zwischen verschiedenen *Maps* (vergleichbar zu SQL-Join-Operationen) werden von NIS nicht unterstützt. Derartige Abfragen müssen von den darauf zugreifenden Applikationen selbst implementiert werden. Auch wird in diesen *Maps* nur zwischen „Schlüssel“ und „Wert“ unterschieden, weitergehende Typ- oder Syntaxinformationen wie beispielsweise Datentypen sind leider nicht vorhanden. Eine Abfrage liefert also zunächst nur eine Zeichenkette zurück und die zugreifende Anwendung muss die nötige Logik implementieren, um diese Zeichenkette interpretieren zu können. Das Format der Zeichenkette unterscheidet sich dabei je nach Anwendung und *Map*.
Da NIS keine Mechanismen oder Schnittstellen zum Erzeugen oder Pflegen von Daten bereitstellt, ist es erforderlich, neben einem NIS weitere externe Systeme zur Erzeugung der Quelldaten zu betreiben, wodurch zusätzlicher Aufwand für den Betreiber entsteht.
NIS ist eng an das Unix-Umfeld gekoppelt und obwohl der Zugriff beispielsweise per JAVA über JNDI möglich ist, hat sich NIS außerhalb des Unix-Umfelds kaum verbreitet.
Neben fehlenden Verwaltungsschnittstellen haben die unflexiblen *Maps* und die fehlenden Sicherheitsmechanismen dazu geführt, dass inzwischen von Einsatz von NIS in Neuinstallationen abgeraten wird. Ein weiterer Grund ist die fehlende NIS-Unterstützung für aktuelle Microsoft-Betriebssysteme.
Auch Sun Microsystems hat inzwischen ein „*End-of-Feature (EOF) Announcement*“ veröffentlicht und erklärt, die Unterstützung von NIS mittelfristig zu beenden. Stattdessen empfiehlt Sun Microsystems als Alternative die Migration auf LDAP und stellt dazu einen eigenen LDAP-Server, Anleitungen und Migrationswerkzeuge zur Verfügung.
Für das IPO-Szenario wäre von einer NIS-basierten Lösung sicherlich abzuraten. Neben den Sicherheitsaspekten (fehlende Verschlüsselung, nur rudimentäre Zugriffsrechte) stellen insbesondere die fehlenden Verwaltungsmechanismen ein Problem dar. Diese müssten vollständig selbst entwickelt werden und würde angesichts der übrigen von NIS implizierten Einschränkungen einen unangemessen hohen Aufwand darstellen.
Im Szenario der Fakultät für Informatik ist NIS bereits seit Langem im Einsatz, doch da von Sun Microsystems ein Ende der NIS-Unterstützung angekündigt wurde, stellt die fehlende Zukunftssicherheit ein großes Risiko dar. Mittelfristig muss daher auch hier, aufgrund der Supportsituation und des schlechten Sicherheitskonzepts, über eine Alternative nachgedacht werden muss. Da insbesondere von Sun Microsystems eine Migration auf einen LDAP-Verzeichnisdienst empfohlen wird, sollte diese Empfehlung zuerst untersucht werden. Weitere Informationen zu NIS findet man in [SE+01].

## 2.5 Microsoft Active Directory

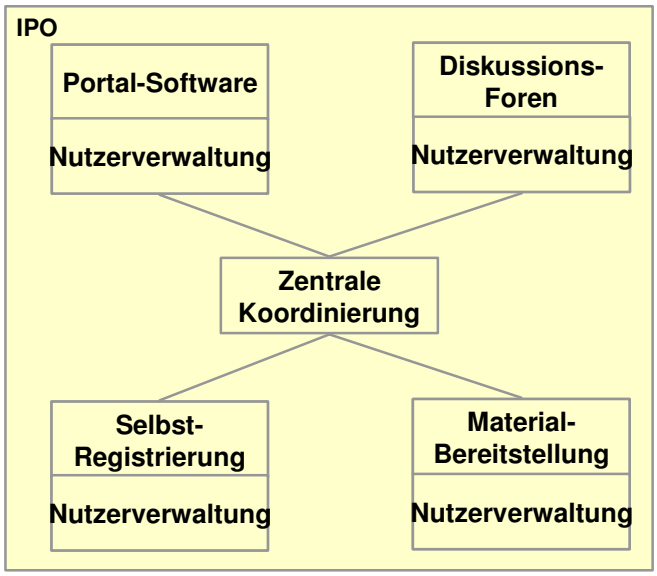
Microsoft hat von 1987 bis 1999 ein eigenes, weit gehend undokumentiertes Protokoll unter der Bezeichnung „LAN Manager“, später „NT Domain“, eingesetzt, um eine zentrale Nutzer-, Gruppen- und Druckerverwaltung zu realisieren. Dieses Protokoll war nicht für eine Anwendung außerhalb des Microsoft-Umfelds vorgesehen und wurde inzwischen durch das „Active Directory“ abgelöst, sodass hier auf eine genauere Betrachtung des LAN Manager verzichtet werden kann.
Das Active Directory (kurz AD) wurde 1999 zusammen mit „Windows 2000“ eingeführt und später in der Version „Windows 2003“ erweitert. Das Active Directory basiert auf mehreren Standardprotokollen, dem DNS-Dienst zum Auffinden von Diensten, Kerberos zur sicheren Übertragung von Anmeldeinformationen und einem Verzeichnisserver nach LDAPv3-Standard zur Verwaltung von Konfigurationsdaten, Nutzerdaten, Rollen und Zugriffsrechten.
Microsoft selbst definiert die Aufgaben eines Verzeichnisdienstes wie folgt: *„[A directory service] provides a consistent method for naming, describing, locating, accessing, managing, and securing information about the resources.“*
Ein Active Directory kann tatsächlich gemäß LDAP-Standard untersucht und individuell erweitert werden. Beispielsweise wird es von Microsoft Exchange erweitert, um die bereits existierenden Nutzerdaten um Exchange-spezifische Informationen zu ergänzen.
Als Verzeichnis bietet das Active Directory einige fortgeschrittene Funktionen wie beispielsweise eine Multi-Master-Replikation (siehe 3.4.6). Eine Besonderheit ist der Global Catalog, der einen Ausschnitt des vollständigen Verzeichnisses enthält und besonders weit repliziert wird. Dieser Global Catalog enthält diejenigen Informationen, auf die besonders häufig zugegriffen wird und entlastet somit die Instanzen, die den vollständigen Verzeichnisinhalt bereitstellen.
Da die vollständige Datenspeicherung in einem LDAP-Verzeichnis erfolgt, soll im Folgenden nicht ein spezifisches Active Directory untersucht werden, sondern es werden lediglich allgemeine Eigenschaften von LDAP-Verzeichnissen betrachtet.

## 2.6 Novell Netware und eDirectory

Nachdem Novell für sein Produkt Netware ursprünglich einen proprietären Verzeichnisdienst zur Nutzerdatenverwaltung eingesetzt hatte, wurde die Nutzerverwaltung mit der Einführung von Netware 4 auf ein LDAP-Verzeichnis umgestellt. Dieses LDAP-Verzeichnis wird zur Verwaltung sämtlicher Netzwerkobjekte wie Nutzer, Gruppen, Server, Drucker, Zugriffsrechte etc. genutzt. Der von Novell eingesetzte LDAP-Server ist wurde zuerst als Bestandteil Novell Netware verwendet und erst später unter der Bezeichnung Novell Directory Service eigenständig vermarktet. Die Bezeichnung wurde später zu eDirectory geändert. eDirectory bezeichnet also einen aktuellen LDAP-Server von Novell und beherrscht fortgeschrittene Funktionen wie beispielsweise die *Multi-Master-Replikation*.
Heute ist das eDirectory die zentrale Basis für nahezu alle aktuellen Novell-Produkte. Dazu gehören beispielsweise Netware, die Mail-Anwendungen GroupWise und NetMail, das zentrale Konfigurations- und Verwaltungswerkzeug ZenWorks sowie insbesondere die Nsure-Produktreihe zum Identitätsmanagement. Dabei dient das eDirectory jeweils als Speicher für sämtliche Konfigurations- und Account-Daten der erwähnten Produkte.

## 2.7 Identity Management Systeme

Nachdem wir das *Identity Management* als Aufgabe und Prozess innerhalb einer Organisation bereits eingeführt haben (siehe 1.3), betrachten wir nur die existierenden Produkte, die eine Lösung für dieses Problem versprechen.
Das Problem ist nach wie vor die unkoordinierte, redundante Nutzerverwaltung in parallelen Verwaltungsstrukturen. Von der Industrie wurden daher Systeme entwickelt, die die mit dem *Identity Management* verbundenen Probleme systematisch lösen sollen.
Der von den aktuellen Werkzeugen aus dem Bereich des *Identity Management* verfolgte Ansatz verwendet eine zentrale Koordinierungsinstanz, an die die existierenden Nutzerverwaltungen über individuelle Schnittstellen angebunden werden. Es entsteht so eine sternförmige Struktur.




**Information 6: Portal: Zentrale Koordinierungsinstanz**

Diese zentrale Koordinierungsinstanz wird von den Herstellern derartiger Systeme als *Identity Management*-Dienst vermarktet und bietet i. d. R. neben der reinen Koordinierung weitergehende Funktionen wie z. B. die Einbindung in Workflows, die Protokollierung der durchgeführten Aktionen und die Bereitstellung einer globalen Sicht auf die angebundenen Systeme.
Dieser Dienst wird über seine Schnittstellen über Änderungen in den angeschlossenen Systemen informiert und kann dort auch selbst Änderungen veranlassen. Damit wird es möglich, Änderungen in einem angebundenen System zu erkennen und, anhand interner Regeln, diese Änderungen soweit nötig an die übrigen Systeme zu propagieren. Nach Durchführung der erforderlichen Änderungen liegt schließlich ein konsistentes Gesamtsystem vor. Alternativ können die angebundenen Systeme den *Identity Management*-Dienst auch direkt befragen, wozu vorzugsweise Standardprotokolle wie z. B. LDAP eingesetzt werden sollten.
Bei diesem Koordinierungsprozess durch den *Identity Management*-Dienst wird in der Regel vorausgesetzt, dass für eine spezielle Information jeweils nur ein autoritatives System existiert, von dem eine Änderung ausgehen kann. Damit wird eine aufwendige Ausnahmebehandlung für den Fall, dass eine Information zeitgleich in zwei Systemen verändert wird, vermieden.
Um Änderungen in den Systemen erkennen und auch selbst Änderungen durchführen zu können, müssen die jeweils anzubindenden Anwendungen um geeignete Schnittstellen

erweitert werden, wobei entsprechende Implementierungen für verbreitete Produkte von den Herstellern der *Identity Management*-Systeme selbst bereitgestellt oder von Drittanbietern erworben werden können. Diese Schnittstellen sind dabei nicht standarisiert, sondern unterscheiden sich je nach verwendetem *Identity Management*-System.
Für viele *Identity Management*-System ist auch die Einbindung in Workflows möglich, sodass ein *Identity Management*-Dienst beispielsweise erst eine Bestätigung von einem zuständigen Mitarbeiter anfordern kann, bevor eine Aktion seitens des *Identity Management*-Diensts durchgeführt wird. Auch Anbindungen an Audit- oder *Passwort-Policy*-Dienste sind in der Regel problemlos durchführbar.
Ein Beispiel für ein solches Werkzeug ist der Nsure Identity Manager von Novell, der bereits Schnittstellen für verbreitete Anwendungen wie Novell Netware, Microsoft Active Directory oder verschiedene SAP-Komponenten enthält und die Nutzerverwaltungen dieser Anwendungen synchronisieren kann. Weitere Schnittstellen können individuell durch den Betreiber entwickelt und ergänzt werden. Dabei bleibt die eigenständige Nutzerdatenhaltung der einzelnen Komponenten erhalten und der Nsure Identity Manager übernimmt die Rolle einer zentralen Koordinierungsinstanz, wie in Information 6 dargestellt.
Als Beispiel könnte in einem System zur Personalverwaltung ein neuer Mitarbeiter aufgenommen werden, der in der Abteilung Marketing arbeiten soll. Der *Identity Management*-Dienst wird daraufhin automatisch über diesen neuen Mitarbeiter informiert und erhält automatisch die Daten, die von allgemeinem Interesse sind und auch in anderen Diensten benötigt werden. Der *Identity Management*-Dienst gewährt daraufhin diesem neuen Mitarbeiter den Zugriff zum Unternehmensportal, veranlasst die Erzeugung eines Nutzereintrags im Active Directory und weist dem Mitarbeiter die speziellen Rechte zu, die zur Arbeit in der Marketingabteilung erforderlich sind. Außerdem wird das Mail-System zur Anlage einer neuen Mailauslieferung veranlasst, der Mitarbeiter wird automatisch in die relevanten Mailinglisten eingetragen und erscheint sofort mit einem Eintrag im globalen Adressbuch. Die durchgeführten Aktionen werden in einem angeschlossenen Audit-Dienst protokolliert, um beispielsweise gesetzlichen Nachweispflichten nachkommen zu können und der Password-Policy-Dienst veranlasst den Mitarbeiter bei der ersten Anmeldung, zum Setzen eines sicheren Passworts.
Eine sehr gute Einführung in die Thematik biete beispielsweise [Le03].
Auch ein *Identity Management*-System benötigt eine eigene Datenspeicherung, um eine globale Sicht auf die verwalteten Identitäten erzeugen zu können. Außerdem muss hier eine zuverlässige Zustandsverwaltung erfolgen können, aus der hervorgeht, welche Änderungen aufgetreten sind und an welche Systeme diese Änderungen noch propagiert werden müssen. Bei vielen Produkten wie beispielsweise der Novells Nsure-Produktreihe oder dem Sun Java System Identity Manager von Sun Microsystems werden für diesen Zweck Verzeichnisdienste auf LDAP-Basis zur Datenspeicherung verwendet.
Da jedoch trotz der Datenspeicherung in standarisierten Verzeichnisdiensten für jede der anzubindenden Komponenten eine spezielle Schnittstelle zwischen der zentralen Koordinationsinstanz und der einzelnen Komponente erforderlich wird, ist die Anzahl der zu implementierenden Schnittstellen linear von der Zahl der anzubindenden Komponenten abhängig. Damit bietet die Koordination durch eine zentrale Instanz bereits eine Verbesserung gegenüber proprietären Mechanismen (siehe 2.2), die zu einem höheren Aufwand führen könnten. Außerdem werden durch die zentrale Synchronisation Abhängigkeiten zwischen den einzelnen Anwendungen vermieden. Auch die spätere Anbindung weiterer Komponenten könnte durch klar spezifizierte

Schnittstellen der zentralen Nutzerverwaltung vereinfacht werden und würde durch die bereits existierenden Komponenten nicht beeinflusst.
Nach wie vor ist jedoch eine Vielzahl von Schnittstellen erforderlich, da für jede Komponente eine eigene Schnittstelle und eigene Mechanismen spezifiziert und implementiert werden müssen. Darin liegt der Unterschied zu unserem Lösungsansatz, der durch die Verwendung von Standardschnittstellen die Komplexität zu reduzieren versucht.

## 2.8 Zusammenfassung

Bei einer zusammenfassenden Betrachtung der bisher existierenden und etablierten Verfahren muss leider festgestellt werden, dass die proprietären Verfahren (siehe 2.2) oder NIS (siehe 2.4) insbesondere aufgrund fehlender Zukunftssicherheit keine überzeugenden Lösungsmöglichkeiten darstellen.
Auch aktuelle Systeme wie das Active Directory bilden eher in sich abgeschlossene Lösungen, die an bestimmte Plattformen gebunden sind und nicht die erwünschte Heterogenität unterstützen.
Viel versprechend erscheint hingegen der Ansatz des *Identity Management*, das bestehende, heterogene Systeme über spezielle Schnittstellen anbindet, um ein zentrales System zur Nutzerverwaltung, Authentifikation und Autorisation zu entwickeln. Allerdings entsteht hier nach wie vor eine hohe Komplexität, da für alle anzubindenden Systeme individuelle Adapter entwickelt werden müssen. Vor diesem Hintergrund ist die Einführung von *Identity Management*-Diensten nach wie vor ein aufwendiger und kostenintensiver Prozess.
Im Folgenden werden wir daher unser Augenmerk auf ein *Identity Management* über Standardschnittstellen legen und diese Fragestellung im Zusammenhang mit dem IPO (siehe Kapitel 4) und mit Fragestellungen an der Fakultät für Informatik (siehe Kapitel 6 und Kapitel 7) näher untersuchen.
Da bisher LDAP den einzigen offenen und standarisierten Verzeichnisdienst darstellt, der auch von vielen Produkten des *Identity Management* genutzt wird, werden wir diesen Standard im Folgenden näher einführen (siehe Kapitel 3) und als Basis für unsere Untersuchung verwenden.

# 3 LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP)

Die Bezeichnung *Lightweight Directory Access Protocol* wird für einen speziellen Verzeichnisdienst (siehe 2.3) verwendet. Verzeichnisdienste nach LDAP-Standard (aktuell ist Version 3 des Standards) werden derzeit sowohl von Sun Microsystems als auch von Microsoft als Nachfolger für die älteren Dienste NIS (siehe 2.4) und LAN Manager (siehe 2.5) verwendet. Auch in anderen Bereichen wie etwa dem *Identity Management* (siehe 1.3 und 2.7) werden LDAP-basierte Verzeichnisse zur Verwaltung der Nutzer- und Zugangsdaten eingesetzt. Da auch die Problemstellungen im IPO-Szenario (siehe Kapitel 4) und an der Fakultät für Informatik (siehe Kapitel 6 und Kapitel 7) dem Bereich „*Identity Management*“ angehören, erscheint es sinnvoll, zuerst die generellen Eigenschaften LDAP-basierter Verzeichnisdienste zu untersuchen, um in den folgenden Kapiteln darauf aufbauen zu können. Leser, denen LDAP bereits vertraut ist, können die folgenden Erläuterungen überspringen und ab 3.5 weiter lesen.

## 3.1 Kurze Geschichte von LDAP

Das LDAP-Protokoll wurde 1993 an der University of Michigan entwickelt, um basierend auf diesem Protokoll ein *Gateway* zwischen den existierenden X.500-Servern auf OSI-Basis und TCP/IP-basierten Clienten entwickeln zu können. Verglichen mit dem X.500-DAP (Directory Access Protocol) war das resultierende TCP/IP-basierte Protokoll deutlich einfacher (nur noch neun verschiedene Operationen, eine vereinfachte Codierung, etc.), woraus sich die Bezeichnung Lightweight-DAP bzw. LDAP ergeben hat. Die Spezifikation von LDAPv1 wurde 1993 als RFC 1487 veröffentlicht. RFC bezeichnet als Abkürzung für *Request For Comments* einen Standard der *Internet Engineering Task Force*, kurz IETF. 1995 wurde die zweite Version des LDAP-Protokolls unter der Bezeichnung LDAPv2 in RFC 1777ff. verabschiedet. Schon 1997 kam mit LDAPv3 (RFC 2251ff.) die nächste Version, die bis heute unverändert geblieben ist. Deshalb soll im Folgenden lediglich LDAPv3 betrachtet werden. Sofern nichts anderes angegeben, ist also unter LDAP immer LDAPv3 zu verstehen.

## 3.2 Informelle Definition

Im Allgemeinen versteht man unter LDAP einen Verzeichnisdienst, der die LDAP-Standards verwendet, was sich insbesondere auf das LDAP-Zugriffsprotokoll, den LDAP-Namensraum und die standardisierten LDAP-Schemata bezieht.

## 3.3 Formale Definition

Obwohl der Begriff *Lightweight Directory Access Protocol* ein spezifisches Protokoll nahe legt, versteht man unter LDAP mehrere Standards, die in Form von RFCs von der IETF verabschiedet wurden. Anders als viele andere Systeme (NIS, LAN Manager, Active Directory) ist LDAP damit kein proprietäres System sondern ein offener Standard.
Ursprünglich wurden lediglich die RFC 2251 bis RFC 2256 spezifiziert, noch ohne Angaben über zu implementierende sicherere Authentifizierungs- und Verschlüsselungsmechanismen. Diese wurden später in RFC 2289f. definiert, woraufhin die Standarisierung in RFC 3377 abgeschlossen werden konnte.

| | |
|---|---|
| RFC 2251 | Das LDAP-Netzwerkprotokoll |
| RFC 2252 | Attribut-Syntax-Definitionen, Vergleichsregeln |
| RFC 2253 | UTF-8-Darstellung der „Distinguished Names“ (DN) |
| RFC 2254 | String-Repräsentation von Suchfiltern |
| RFC 2255 | URL-Format für LDAP |
| RFC 2256 | Verwendung von X.500-User-Schemata mit LDAP |
| RFC 2829 | Authentifizierungsmechanismen für LDAP |
| RFC 2830 | Erweiterung für TLS-Unterstützung |
| RFC 3377 | Spezifikation, welche RFCs offiziell das LDAPv3-Protokoll definieren |

**Tabelle 1: LDAPv3 RFC**

Neben dieser „Kern“-LDAP-Spezifikation existieren weitere Vorschläge, beispielsweise LDAP-Erweiterungen wie z. B. die serverseitige Sortierung von Such-Ergebnissen (RFC 2891) oder neue Schemadefinitionen wie etwa „*inetOrgPerson*“ (RFC 2798). Die existierenden LDAP-Server-Implementierungen gehen in der Regel über die Kernspezifikation hinaus.

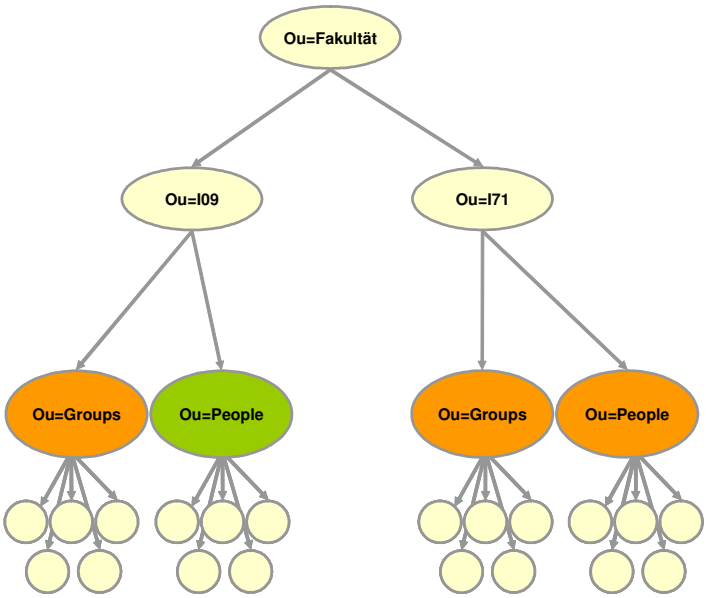
## 3.4 LDAP-Modelle

Die LDAP-Spezifikation beschreibt vier verschiedenen Gesichtspunkten, von denen aus LDAP definiert werden kann. Dies sind die so genannten LDAP-Modelle, zum einen das *Naming-Model* (wie wird ein Eintrag benannt und welche Struktur erhält das Verzeichnis), das *Infomation-Model* (wie wird ein Eintrag konstruiert), das *Functional-Model* (welche Operationen stehen zur Verfügung) und das *Security-Model* (Authentifizierung und Verschlüsselung). Die in Tabelle 1 angegebenen RFC können thematisch in diese Modelle einordnen.
Im Folgenden werden diese Modelle vor einem eher technischen Hintergrund detailliert untersucht. Leser, denen diese Details bereits bekannt sind oder die die technischen Aspekte später vertiefen möchten, können an dieser Stelle auch zu 3.5 übergehen. Dort wird der Einsatz von LDAP im Rahmen der Nutzerverwaltung untersucht.

### 3.4.1 LDAP-Naming-Model

Die Benennung einzelner LDAP-Einträge ist das Thema des *Naming-Model*. Dieses umfasst zusätzlich die Struktur eines LDAP-Verzeichnisses, die mit dieser Benennung zusammenhängt.
Grundsätzlich ist das LDAP-Verzeichnis in einer Baumstruktur angeordnet, d. h. jeder Eintrag, mit Ausnahme der Wurzel, hat einen eindeutigen Vatereintrag und jeder Eintrag kann beliebig viele Kindereinträge haben. Bezogen auf den jeweiligen Vater hat jeder LDAP-Eintrag einen eindeutigen Namen, der sich aus einem Attributtyp und einem zugehörigen Attributwert ergibt, beispielsweise „mail=hagen@ira.uka.de“. Das einen LDAP-Eintrag identifizierende Attribut muss konform zu den angegebenen Objektklassen des Eintrags sein, kann aber ansonsten frei gewählt werden. Aus Gründen der Übersicht wird man aber für gleichartige Einträge jeweils dasselbe, eindeutige Attribut zur Identifikation auswählen.
Technisch gesehen ist es auch möglich, einen aus mehreren Attributen zusammengesetzten Namen zu wählen, jedoch werden diese in der Praxis selten eingesetzt. Bezogen auf den jeweiligen Vaterknoten muss der gewählte Name eindeutig sein, allerdings kann in einem anderen Bereich des LDAP-Verzeichnisses ein anderer Eintrag mit dem gleichen Namen existieren. Da dieser Name immer auf den Vaterknoten bezogen wird, spricht man von einem „*Relative Distinguished Name*“,

(kurz RDN). Einen vollständigen Bezeichner für einen LDAP-Eintrag, einen „*Distinguished Name*“ (DN), erhält man durch Verkettung des RDN mit den Namen aller Vaterknoten. Da der Name eines Eintrags nur bezogen auf den Vater eindeutig ist, kann dieser Name nicht als global eindeutig eingesetzt werden.

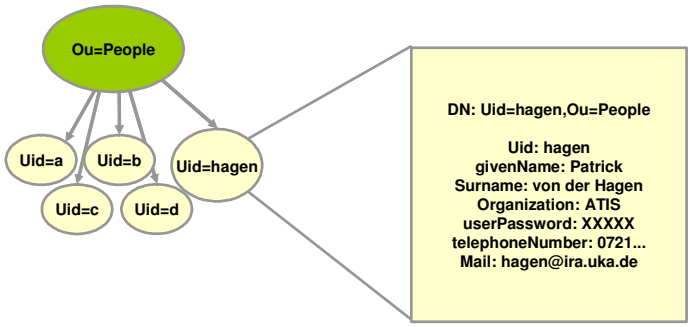



**Information 7: Beispiel für LDAP-Struktur**

In Information 7 existiert eine LDAP-Wurzel mit der Bezeichnung „Ou=Fakultät“, an der die Institute „I09“ und „I71“ als eigenständige Bereiche verwaltet werden. Jedes Institut unterscheidet Einträge für Gruppen und Personen in den Bereichen „*Groups*“ bzw. „*People*“. Jeder Knoten ist mit seinem RDN gekennzeichnet, der vollständige DN des grün markierten Knotens lautet in diesem Beispiel „Ou=People, Ou=I09, Ou=Fakultät“. „Ou“ ist die Abkürzung für die Objektklasse „*Organizational Unit*“ und wird häufig zur Strukturierung von LDAP-Verzeichnissen verwendet.
Um global eindeutige Einträge gewährleisten, ist die durch den LDAP-Standard spezifizierte Funktionalität nicht ausreichend. Sind global eindeutige Attribute erforderlich, so muss dies vom Betreiber durch angepasste Betriebsprozesse gewährleistet werden, beispielsweise indem vor Änderungsoperationen durch eine Suchanfrage geprüft wird, dass das zu vergebende Attribut tatsächlich im gesamten Verzeichnis eindeutig ist. Je nach eingesetzter LDAP-Implementierung kann es auch sein, dass der Hersteller in Ergänzung zum LDAP-Standard Mechanismen vorgesehen hat, die global eindeutige Attribute gewährleisten können. Nachteilig wäre in diesem Fall die Abhängigkeit von einer konkreten Implementierung.
RFC 2253 und RFC 2255 stehen in Bezug zum *Naming-Model*.

### 3.4.2 LDAP-Information-Model

Das *Information-Model* beschreibt die Bestandteile, aus denen sich die Einträge im LDAP-Verzeichnis zusammensetzen können. Es wird vorausgesetzt, dass jeder Eintrag über einen *Distinguished Name* verfügen muss, über den der Eintrag innerhalb des Verzeichnisses eindeutig zu identifizieren ist. Der Eintrag selbst besteht aus Attributen, wobei jedem Attribut ein Attributtyp (Name, Telefonnummer, E-Mail-Adresse, etc.) und einer oder mehrere Werte zugeordnet sind.




**Information 8: LDAP-Information-Model**

In Information 8 wird der Bezug zwischen der LDAP-Baumstruktur und den Attributen eines einzelnen Eintrags hergestellt. Einem Attributtypen wiederum sind Syntax-Regeln und (möglicherweise) Vergleichsregeln zugewiesen. Damit kann der Verzeichnisserver prüfen, ob einem Attribut tatsächlich ein gültiger Wert zugewiesen wird (Syntax) und kann gegebenenfalls Vergleichsoperationen durchführen. Beispielsweise hängt die Sortierung von „123“ und „11111“ davon ab, ob Zeichenketten oder Zahlenwerte verglichen werden soll. Dabei muss jeder LDAP-Server zumindest die in RFC 2252 definierten Syntax- und Vergleichsregeln implementieren, Erweiterungen sind jedoch möglich.
Auch das Konzept der „Objektklassen“ ist Bestandteil des *Information-Model*. Eine Objektklasse spezifiziert eine Menge von Attributen, die ein Eintrag haben kann („*May*“-Attribute) oder haben muss („*Must*“-Attribute), wobei eine Klasse in der Regel einem definierten Einsatzzweck entspricht. Einem Eintrag werden zuerst eine oder mehrere Objektklassen zugewiesen, woraus sich die möglichen und die verpflichtenden Attribute für den konkreten Eintrag ergeben. Beispielsweise muss ein Eintrag, der der Objektklasse „*Person*“ entspricht, einen Vornamen und einen Nachnamen enthalten, und kann eine Beschreibung und ein Passwort angeben. Diese Objektklasse wird in RFC 2256 definiert.
Kommen mehrere Objektklassen für einen Eintrag zum Einsatz, so unterliegt der resultierende Eintrag den Einschränkungen, die sich durch die Vereinigung der „*Must*“- und der „*May*“-Definitionen ergeben.
Auch bei den Objektklassen existiert eine wichtige Klassifizierung und zwar die Unterscheidung zwischen „strukturellen Objektklassen“ und „Hilfs-Objektklassen“. Dazu existiert die Einschränkung, dass jeder Eintrag eines LDAP-Verzeichnisses genau einer strukturellen Objektklasse angehören muss und beliebig vielen Hilfs-Objektklassen angehören kann. Veröffentlicht ein Hersteller eine spezielle Objektklasse, so ist diese Unterscheidung mitunter jedoch recht willkürlich und kann daher in der Praxis teilweise zu Problemen führen, die sich jedoch in der Regel einfach dadurch auflösen lassen, dass anhand der vorliegenden Objektklasse eine nahezu identische Objektklasse des jeweils anderen Typs definiert wird. Aus Kompatiblitätsgründen sollte die Änderung einer existierenden Objektklasse unbedingt vermieden werden.
Die Definitionen von Objektklassen und Attributen werden auch allgemein als „Schema“ bezeichnet.
Jedes der bisher genannten Elemente (Objektklasse, Attributtyp, Syntaxregel) wird LDAP-intern über eine global eindeutige Objekt-ID (kurz OID) identifiziert. Um eigene Elemente zu spezifizieren, ist es daher erforderlich, bei der *Internet Assigned Numbers Authority* (kurz IANA) einen eigenen Nummernbereich zu registrieren. Dieser Bereich besteht aus einem Präfix, das beliebig erweitert werden kann. Jeder zugewiesene

Bereich hat also prinzipiell eine unendliche Größe. Dabei ist eine LDAP-OID identisch zu den beispielsweise bei SNMP verwendeten OIDs, für die die gleichen Regeln gelten. Es existiert auch keine Unterscheidung zwischen SNMP- und LDAP-OID, etwa anhand unterschiedlicher Präfix-Räume. Die konkrete Verwendung bleibt der einzelnen Organisation überlassen, die einen zugewiesenen Bereich verwaltet. Details zur Beziehung zwischen IANA-OIDs und LDAP sowie Hinweise zur Vergabe finden sich in RFC 3383.
Insgesamt spezifiziert das Information-Model also grundlegende Klassen, Attribute, Attributtypen und Vergleichsregeln für LDAP-Verzeichnisdienste, sodass daraus eine spezielle Semantik für LDAP resultiert. Dies stellt einen wesentlichen Unterschied zu den universellen Datenbanken dar, die über keine gemeinsam definierte Strukturen und nur rudimentäre Datentypen verfügen. Durch die Spezifikation einer Objektklasse „*Person*“ mit Namen und Passwort werden alle LDAP-Nutzer und LDAP-Betreiber angehalten, nach Möglichkeit genau diese Objektklasse zur Repräsentation von Personen zu verwenden und von dieser Konvention soll nur in Ausnahmefällen abgewichen werden. Damit können Anwendungen entwickelt werden, die berechtigterweise die Verwendung der „*Person*“-Objektklasse voraussetzen und trotzdem mit den meisten LDAP-Verzeichnissen kompatibel sind. Da für solche Zwecke keine allgemein akzeptierten Konventionen etwa für SQL-Datenbanken existieren, bedeutet dies eine wesentliche Vereinfachung gegenüber anderen Ansätzen.
Durch die spezifizierte Semantik der existierenden Attribute wird die Integration verschiedener Datenbestände in einem einzelnen, zentralen Verzeichnis wesentlich vereinfacht und fortgeschrittene Anwendungen des *Identity Management* werden erst durch diese Semantik effizient ermöglicht. Damit bildet die gemeinsame, standarisierte Semantik einen wesentlichen Vorteil gegenüber allen anderen existierenden Verzeichnissen oder Datenbanken, die bisher auf eine solche Semantik verzichtet haben.

### 3.4.3 LDAP-Functional-Model

Das funktionale Modell spezifiziert die Operationen, die gegenüber einem LDAP-Verzeichnis ausgeführt werden können. Es stehen neun Funktionen zur Verfügung, beispielsweise zum An- und Abmelden, zum Hinzufügen, Löschen, Ändern und Umbenennen von Einträgen und zur Durchführung von Suchoperationen. Auch die möglichen Suchfilter, die bei LDAP-Anfragen eingesetzt werden können, zählen zu diesem Modell. RFC 2251 und RFC 2254 spezifizieren die wesentlichen Aspekte.

### 3.4.4 LDAP-Security-Model

Die Sicherheitsaspekte des LDAP-Modells definiert ein verbindungsorientiertes Protokoll. Solange nach dem Verbindungsaufbau keine Anmeldung durchgeführt wurde, gilt die Verbindung als „anonym“, nach der Anmeldung hat die Verbindung die Rechte des Users, mit dem die Anmeldung durchgeführt wurde. Der jeweils aktive Nutzer kann innerhalb einer LDAP-Verbindung gewechselt werden, indem eine neue Authentifikationsanforderung gesendet wird. Die Anmeldung selbst kann unverschlüsselt, verschlüsselt nach TLS-Standard (spezifiziert in RFC 2830 und RFC 2830) oder verschlüsselt nach SSL-Standard (nicht in Form von RFCs spezifiziert, aber üblich) erfolgen. Dabei kommen verschiedenen Mechanismen zum Einsatz. Neben der „simple-Authentication“, bei der lediglich Anmeldename und Passwort im Klartext (ggf. über eine zuvor verschlüsselte Verbindung) übermittelt werden, gibt es über die SASL-Schnittstelle auch die Möglichkeit, andere Verfahren wie beispielsweise Kerberos oder Sicherheitszertifikate zur Authentifikation einzusetzen.

Auch Anwendungen müssen sich gegenüber einem LDAP-Verzeichnis authentifizieren, um darauf zugreifen zu können. Dazu wird im LDAP-Verzeichnis ein spezieller Nutzer angelegt, der die jeweilige Anwendung repräsentiert. Dieser Nutzer wird in der Literatur als „*Proxy-User*“, „*System DN*“ oder „*Service DN*“ bezeichnet, da er stellvertretend für die Anwendung bzw. ein System genutzt wird. Indem diesem speziellen Nutzer Zugriffsrechte zugewiesen werden, kann der LDAP-Dienst die Zugriffsrechte jeder einzelnen Anwendung präzise steuern.
Eine Spezifikation, welche Zugriffsrechte innerhalb eines LDAP-Verzeichnisses vergeben werden können, ist nicht Bestandteil der LDAPv3-Spezifikation. In der Regel werden verschiedene Stufen wie „Lesen“, „Schreiben“, „Vergleichen“ unterschieden und können auf Attributebene spezifiziert werden. Beispielsweise ist vorstellbar, dass ein Eintrag für einen Mitarbeiter aus E-Mail-Adresse, Mitarbeiternummer und Passwort besteht, wobei die E-Mail-Adresse nur von den Mitarbeitern der IT-Abteilung verändert werden kann, die Mitarbeiternummer nur von der Verwaltung und das Passwort nur vom Mitarbeiter selbst. Fehlen Leserechte für einen Eintrag oder ein Attribut, so wird dieser Eintrag oder dieses Attribut verborgen.
Aus Betriebssicht kann hier also sehr feinkörnig für jedes Attribut definiert werden, welche Datenquelle für ein Attribut als „führendes System“ auftritt, indem nur der jeweiligen Quelle ermöglicht wird, dieses Attribut zu verändern. Auch der Datenschutz kann von der feingranularen Rechtevergabe profitieren.
Passwörter können in der Regel vom LDAP-Server vor der Speicherung mit einer Einwege-Hash-Funktion geschützt werden. Erfolgt ein Angriff auf den Server oder werden die Daten durch einen Bedienfehler kompromittiert, so bedeutet es einen erheblichen Aufwand, die den gehashten Passwörtern zugehörigen Klartextpasswörter zu ermitteln, sodass der Betreiber mehr Zeit hat, um den Angriff zu entdecken und geeignet zu reagieren.
Dies erhöht die Sicherheit, ohne dass bei der Anmeldung für einen Clienten ein zusätzlicher Aufwand entsteht, da die Passwortprüfung komplett vom LDAP-Server durchgeführt wird. Deshalb benötigen Clienten keine Kenntnis der verwendeten Hash-Funktion und auch keine Leseberechtigung für das Passwortattribut.

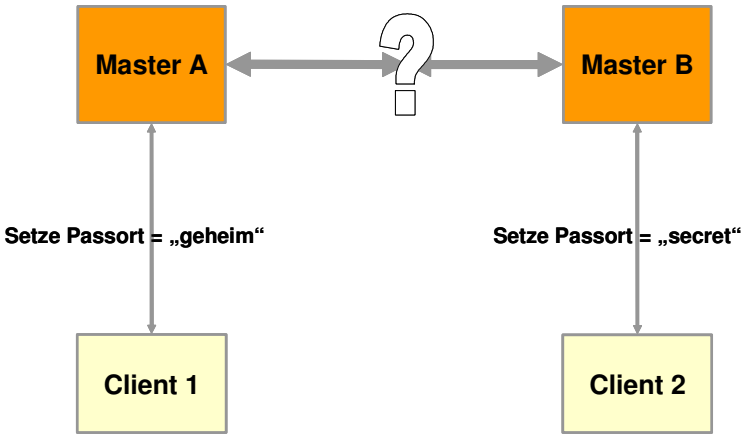
### 3.4.5 LDAP Erweiterungen

Der LDAP-Standard in LDAPv3 ist von erweiterbar ausgelegt, um eine stabile Basis zu etablieren und Änderungen oder Erweiterungen am Kernstandard weit gehend zu vermeiden. Dazu wird spezifiziert, dass LDAPv3-kompatible Server Erweiterungen implementieren und über eine spezielle Erweiterungsschnittstelle bekannt geben können. Die Clienten können dann über eine ebenfalls noch im LDAPv3 spezifizierte Erweiterungsschnittstelle die im konkreten Fall vorhandenen Erweiterungen abfragen und verwenden. Die genaue Spezifikation findet sich in RFC 2251, dort werden die Erweiterungen als „*Controls*“ bezeichnet.
Genutzt wird diese Schnittstelle beispielsweise zur Einführung zusätzlicher Sicherheitsmechanismen oder um Sortierfunktionen im Server bereitzustellen. Ein Beispiel ist die Unterstützung von TLS-verschlüsselten (*Transport Layer Security*) Verbindungen, die in RFC 2830 in Form einer Erweiterung spezifiziert ist und selbst zum Kernstandard gehört.

### 3.4.6 Replikation und Konsistenz

Zur Sicherung der Verfügbarkeit sind bei LDAP-Verzeichnisdiensten Replikations-Mechanismen vorgesehen, die sicher stellen sollen, dass mehrere LDAP-Server über den gleichen Datenbestand verfügen, sodass trotz Ausfalls eines Servers die Daten für

die LDAP-Clienten weiterhin verfügbar bleiben. Diese Replikation kann außerdem genutzt werden, um die Antwortzeiten bei der Durchführung von LDAP-Anfragen zu verbessern, indem etwa eine Lastverteilung über mehrere LDAP-Server durchgeführt werden kann oder bei ungünstigen Netzwerkverbindungen LDAP-Slave-Server so verteilt werden, dass jeder Netzbereich über einen eigenen LDAP-Server verfügt und langsame Verbindungen vermieden werden. Die Platzierung und Anzahl der sinnvoll einzusetzenden LDAP-Server hängt dabei von der konkreten Netzwerktopologie ab.
Diese Replikation bringt jedoch auch ein Problem mit sich: Wie können die Daten der verschiedenen, beteiligten LDAP-Server konsistent gehalten werden bzw. was genau versteht man im LDAP-Umfeld unter dem Begriff der Konsistenz? Aus der Datenbanktheorie kennt man die „verteilte Transaktion“, die garantieren kann, dass alle beteiligten Server zu jedem Zeitpunkt einen identischen Datenbestand haben. Mit der verteilten Transaktion sind jedoch erhebliche Anforderungen an die beteiligten Server verbunden, denn Änderungen können nur durchgeführt werden, wenn alle beteiligten Server verfügbar sind. Andernfalls würden Inkonsistenzen zwischen den Datenbeständen entstehen, da die Änderungen nur auf einem Teil der Server wirksam werden könnten. Alternativ ließe sich die Wirksamkeit der Transaktion auch bis zum nächsten Zeitpunkt, zu dem abermals alle Server verfügbar sind, verschieben, was allerdings nur in seltenen Fällen eine akzeptable Lösung darstellt.
Wie auch bei anderen Verzeichnisdiensten wurde daher beim Entwurf des LDAP-Standards eine deutlich schwächere Konsistenz vorgesehen. Nach einer Datenänderung wird diese Änderung vom Server, der sie entgegen genommen hat, an die anderen LDAP-Server weiter gegeben. Ist ein Server vorübergehend nicht erreichbar, so wird periodisch versucht, die Änderung zu propagieren, bis der Server wieder verfügbar wird und die Änderung annimmt. Folglich werden vorübergehend voneinander abweichende Datenbestände akzeptiert, solange garantiert ist, das zu einem späteren Zeitpunkt ein Abgleich erfolgt. Da die zu erwartenden Propagationszeiträume, von dem Zeitpunkt der Änderung bis zur Wirksamkeit im gesamten Verzeichnis, von der Verfügbarkeit der beteiligten Server abhängig sind, können diese Zeiträume vom Betreiber beeinflusst werden. Die im Einzelfall akzeptablen Propagationszeiträume müssen daher jeweils von Betreiber spezifiziert und durch Betriebsmaßnahmen unterstützt werden.
Unterschieden wird zwischen *Master-Slave-* und *Multi-Master*-Replikationsverfahren. Das *Master-Slave*-Verfahren stellt die einfachere Lösung dar, die auch bei NIS (siehe 2.4) zum Einsatz kommt. Nur ein speziell ausgezeichneter *Master*-Server darf Änderungsoperationen entgegen nehmen, die anderen LDAP-Server werden als *Slave* bezeichnet und verweigern Modifikationen, die nicht vom Master übermittelt werden. In diesem Fall wird dem Client über das LDAP-Protokoll eine Referenz auf den *Master*-Server gesendet, die der Client dann verwendet, um die Datenänderung direkt am *Master*-Server durchzuführen. Der Vorteil dieser Lösung liegt darin, dass dem *Master*-Server immer der aktuellste Datenbestand zur Änderung vorliegt und Konflikte durch konkurrierende Änderungen vermieden werden.
Bei der *Multi-Master*-Replikation existieren mehrere LDAP-Server, die Änderungsanforderungen akzeptieren und durchführen. Diese Änderungen werden dem Client sofort bestätigt und dann an die anderen LDAP-Server propagiert.




**Information 9: Multi-Master-Replikation**

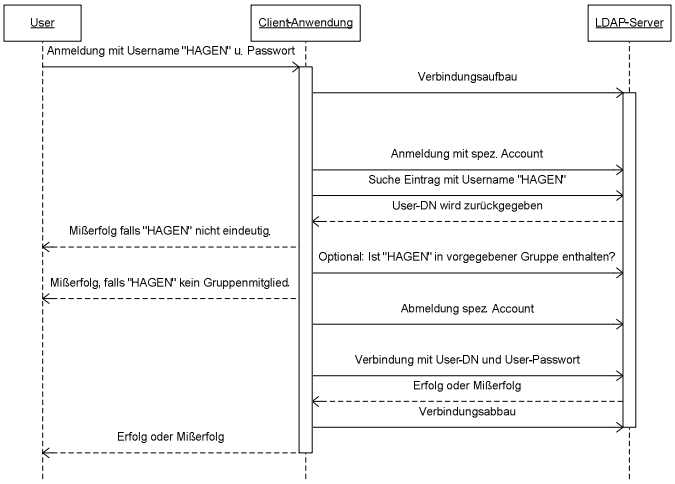
Werden jedoch zwei widersprüchliche Änderungen durchgeführt, so kommt es zu einem unlösbaren Konflikt, bei dem eine der beiden Änderungen verworfen werden muss. Beispielsweise kann jeder Änderung ein Zeitstempel zugeordnet werden und die jeweils aktuellste Änderung wird durchgeführt, während die ältere Änderung überschrieben wird. Bei gleichen Zeitstempeln greifen weitere Mechanismen und entscheiden deterministisch darüber, welche Änderung verworfen wird. Beispielsweise kann die „höhere IP-Adresse“ darüber entscheiden, welcher Master seine Änderung durchsetzen kann. Weitergehende Maßnahmen wie beispielsweise synchronisierte Serveruhren sind beim *Multi-Master*-Einsatz nötig, um einen problemlosen Betrieb gewährleisten zu können.

## 3.5 Nutzerverwaltung mit LDAP

Hier wird die rein technische Ebene verlassen und LDAP in Beziehung zu einer betrieblichen Aufgabe, der „Nutzerverwaltung“, betrachtet. Es werden derzeit in der Praxis mehrere Ansätze verfolgt, die hier nach steigender Komplexität unterschieden werden sollen.

### 3.5.1 LDAP wird ausschließlich zur Passwort-Prüfung eingesetzt

Eine beliebige Anwendung kann mit geringem Aufwand um eine LDAP-Schnittstelle erweitert werden, sodass die Anwendung einen Login und ein Passwort gegenüber einem LDAP-Verzeichnis verifizieren kann. Dazu muss diese Schnittstelle im ersten Schritt aus dem Login ein *Distinguished Name* (siehe 3.4) generieren (entweder über eine Suche im LDAP-Verzeichnis oder nach einem vorgegebenen Schema) und dann eine Anmeldung mit diesem *Distinguished Name* und dem angegebenen Passwort durchführen. Dabei tritt diese Anwendung dem LDAP-Server als Client gegenüber.




**Information 10: LDAP-Nutzung Stufe 1**

Ist diese LDAP-Anmeldung erfolgreich, so wird das Passwort akzeptiert, andernfalls wird die Anmeldung verworfen. Da dazu die LDAP-*Bind*-Funktion genutzt wird, wird dieses Verfahren als „*Authentication Bind*“ bezeichnet. Zu beachten ist dabei, dass das Passwort von der Client-Anwendung an den LDAP-Server übermittelt wird und dieser lediglich mit „Erfolg“ oder „Misserfolg“ antwortet. Daher ist es weder erforderlich, der Client-Anwendung Leserechte auf das tatsächlich im LDAP-Server gespeicherte Passwort zu geben, noch das Passwort im Klartext im LDAP-Verzeichnis zu hinterlegen. Dies ist aus Sicherheitsgründen zu begrüßen, da ein Angreifer auf den LDAP-Server keine Klartextpasswörter auslesen kann. Allerdings müssen zusätzliche Mechanismen, wie beispielsweise TLS-Verschlüsselung, SASL, implementiert werden, um auch die Passwortübertragung zwischen Client und Server zu schützen.
Optional kann noch überprüft werden, ob der angegebene Nutzer einer bestimmten, im LDAP-Verzeichnis hinterlegten, Gruppe angehört, wodurch eine einfache Rollenprüfung durchgeführt werden kann. Dies wird ausführlicher in 3.6 untersucht.
Dieses Vorgehen ist sehr einfach zu realisieren und stellt insbesondere seitens der Client-Anwendung minimale Erwartungen an den LDAP-Server. Beispielsweise kann seitens des Servers ein beliebiges Schema eingesetzt werden. Allerdings werden dabei die erweiterten LDAP-Möglichkeiten kaum genutzt und Daten wie E-Mail-Adresse, Name, etc. müssen bei Bedarf redundant in die Client-Anwendung eingepflegt werden.

### 3.5.2 LDAP als führendes System zur Verwaltung von Nutzerdaten

Da zu einer Person häufig weitere Daten als lediglich ein Passwort im LDAP-Verzeichnis gespeichert werden, liegt es nahe, diese Daten auch in den Client-Anwendungen zu nutzen. Die in 3.4 erwähnte Objektklasse „*Person*“ biete sich dazu an und wird beispielsweise in der Anwendung BSCW (siehe 4.5) verwendet. Aus dieser Objektklasse lassen sich bereits Vorname und Nachname abfragen, davon abgeleitete Objektklassen wie „*organizationalPerson*“ (RFC 2256) oder „*inetOrgPerson*“ (RFC 2798) bieten auch E-Mail-Adresse, Anschrift und ähnliche Attribute zur allgemeinen Nutzung.
Damit kann eine LDAP-fähige Anwendung versuchen, diese Information aus einem LDAP-Verzeichnis abzufragen. Natürlich sollte diese Abfrage nicht einmalig bei der ersten Anmeldung sondern bei jedem Zugriff auf eine dieser Informationen erfolgen, um Aktualisierungen der Daten zu erkennen. Lediglich Informationen, die vom Verzeichnisdienst nicht bereitgestellt werden, müssen dann noch manuell vom

Anwender oder dem Betreiber eingepflegt werden. Die auf diese Art angebundenen Anwendungen werden das Verzeichnis in der Regel lediglich zum Lesen allgemeiner Attribute verwendet, während nur die für ein spezifisches Attribut führenden Systeme die Berechtigung erhalten, diese Attribute zu verändern bzw. zu verwalten.
Dieses Verfahren stellt leicht höhere Anforderungen als lediglich das Prüfen von Passwörtern, doch können diese Anforderungen, sofern sie den üblichen Objektklassen (siehe insbesondere RFC 2256) entsprechen, in der Regel ohne größere Probleme von einem LDAP-Verzeichnis erfüllt werden. Sinnvollerweise sollte die jeweilige Anwendung jedoch flexibel konfigurierbar sein, sodass sie auch an LDAP-Verzeichnisse angepasst werden können, die von den LDAP-Standards abweichen. Ein Beispiel für eine solche Anwendung ist das in 4.5 diskutierte BSCW.

### 3.5.3 Anwendungsinformationen werden in LDAP gespeichert

Nachdem die bisher betrachteten LDAP-Nutzungsfälle lediglich existierende LDAP-Datenbestände passiv genutzt werden, wird nun der Fall betrachtet, dass eine Anwendung selbst aktiv Daten in einem LDAP-Verzeichnis speichert und pflegt.
In einem einfachen Fall wird die Anwendung lediglich eigenständige Daten pflegen, die nicht von anderen Anwendungen genutzt werden und auch nicht in Bezug zu bereits im LDAP-Verzeichnis erfassten Daten stehen. In einem komplexen Fall müssen die Anwendungsdaten mit bereits im LDAP-Verzeichnis erfassten Daten koexistieren oder es werden existierende LDAP-Einträge um neue Anwendungsinformationen erweitert.
Der einfache Fall kann gelöst werden, indem ein spezielles LDAP-Verzeichnis für den konkreten Dienst eingerichtet oder ein spezieller Bereich in einem existierenden LDAP-Verzeichnis ausschließlich für den konkreten Dienst vorgesehen wird. Der komplexere Fall soll im Folgenden genauer untersucht werden.
Werden Informationen unterschiedlicher Quellen gemeinsam verwaltet, so ist es generell sinnvoll, für jedes Attribut ein führendes System zu identifizieren und nur diesem System die erforderlichen Verwaltungsrechte zu gewähren. Konflikte, bei denen mehrere Systeme als führend für ein Attribut angesehen werden, sollten unbedingt vermieden werden. Insbesondere die Verwaltung von Attributen, die von mehreren Anwendungen gemeinsam genutzt werden sollen, erfordert besondere Aufmerksamkeit, hingegen ist die Verwaltung von Attributen, die lediglich von einer einzigen Anwendung genutzt werden, vergleichsweise unkritisch. Eine umfassende Analyse der verwalteten Attribute und der verwaltenden bzw. nutzenden Anwendungen bietet dabei die erforderliche Basis, um geeignete Verwaltungsprozesse und Zugriffsrechte für ein LDAP-Verzeichnis festzulegen. Dabei sollten die vergebenen Rechte minimiert und auf das unbedingt Erforderliche eingeschränkt werden.
Für jede Datenverwaltung muss zunächst untersucht werden, ob ein LDAP-Verzeichnis eine angemessene Verwaltung ermöglicht. In 2.3 wurden die speziellen Eigenschaften von Verzeichnisdiensten und die Unterschiede zu Datenbanken bereits eingeführt. Es sei daran erinnert, dass aufwendige Beziehungen (z. B. m:n-Beziehungen), Schlüsselbedingungen oder Transaktionsschutz von LDAP-Verzeichnissen i. d. R. nicht bereitgestellt werden.
Beispielsweise würde eine *Groupware*-Anwendung zur Terminverwaltung, die auch Termine mit mehreren Teilnehmern ermöglicht, aufwendige m:n-Beziehungen erfordern, die mit einem LDAP-Verzeichnis zur Datenspeicherung nur vergleichsweise aufwendig zu realisieren wären. Daher wäre eine Realisierung mit einer Datenbank vermutlich vorteilhaft, die dann nach 3.5.1 oder 3.5.2 ein LDAP-Verzeichnis ergänzend als führendes System für einen Teil der Nutzerdaten verwenden könnte.

Lediglich für Anwendungen, die tatsächlich sämtliche relevanten Informationen in einem LDAP-Verzeichnis ablegen können, ohne durch die angesprochenen Einschränkungen behindert zu werden, bietet sich hingegen eine aktive Verwaltung von Daten im LDAP-Verzeichnis an.
Dabei ist es wichtig, die standarisierten Objektklassen (siehe beispielsweise RFC 2256) zu beachten und zu unterstützen. Sollte etwa eine Klasse „*PersonWithBirthday*“ eingeführt werden, die im Konflikt zur üblichen Klasse „*Person*“ steht, würde vermutlich ein recht großer Aufwand entstehen, da andere Anwendungen die Verwendung der Klasse „*Person*“ erfordern könnten (siehe z. B. 4.5.3). In diesem Fall wäre es Aufgabe des Betreibers, die Anforderungen der einzelnen Komponenten zu koordinieren und eine effektive Kooperation zu ermöglichen. Besser wäre in diesem Beispiel die Einführung einer Objektklasse „*BirthdayExtension*“, die einem Eintrag zusätzlich zur Objektklasse „*Person*“ zugeordnet werden kann und damit die gewünschte Kompatibilität nicht verletzt.
Ein Beispiel für Anwendungen, die besonders von einer Datenspeicherung in einem LDAP-Verzeichnis profitieren, sind Serveranwendungen aus dem Mail-Bereich wie beispielsweise Microsoft Exchange. Werden beispielsweise aus Redundanzgründen mehrere Mail-Server zur Mail-Verarbeitung eingesetzt, die einen hochverfügbaren, konsistenten Datenbestand bzgl. Existenz und Verarbeitung spezifischer E-Mail-Adressen benötigen, dann bieten sich häufig LDAP-Server zur Speicherung dieser Informationen an, die hier durch ihre automatischen Replikationsmechanismen hohe Verfügbarkeiten und gute Lastverteilung ermöglichen.

## 3.6 Rollenverwaltung mit LDAP

Im Szenario des IPO sollen neben den einzelnen Nutzern auch Rollen in einem zentralen Datenspeicher verwaltet werden, um durch diese Abstraktion die Verwaltung vereinfachen zu können.
Abstrakt betrachtet handelt es sich bei einer Rollenverwaltung um die Verwaltung verschiedener Gruppen, wobei den Mitgliedern der einzelnen Gruppen aufgrund ihrer Gruppenmitgliedschaft besondere Rechte innerhalb einer Anwendung zugewiesen werden können.
LDAP bietet verschiedenen Möglichkeiten zur Verwaltung von Gruppen, die von Anwendungen als Basis zur Rollenverwaltung genutzt werden können. Die Vergabe von Rechten an die jeweiligen Gruppen bleibt jedoch den konkreten Anwendungen überlassen, die zur Verwaltung dieser Rechte allerdings das LDAP-Verzeichnis nutzen können.
Die Möglichkeiten zur Definition und Verwaltung von Gruppen lassen sich in vier Bereiche unterteilen, die statischen Gruppen, dynamische Gruppen, Gruppenzugehörigkeit durch Vorwärtsreferenz oder Gruppenbildung durch die Strukturierung des LDAP-Verzeichnisses.

### 3.6.1 Statische Gruppen

Es kann im LDAP-Verzeichnis ein spezielles Gruppenobjekt angelegt werden, dem über ein mehrwertiges Attribut eine Liste von Mitgliedern übergeben wird. Anwendungen kann nun diese Gruppe über ihren Distinguished Name bekannt gemacht werden.
Diese Gruppen können mitunter einen hohen Verwaltungsaufwand erfordern, da bei Änderungen am LDAP-Datenbestand Inkonsistenzen entstehen können, wenn etwa der Eintrag für ein Gruppenmitglied gelöscht wurde. Daher ist eine regelmäßige Konsistenzprüfung der statischen Gruppen erforderlich, die durch Betriebsprozesse entweder regelmäßig durchgeführt oder nach jeder Datenänderung automatisch aufgerufen werden. Die Zugriffsrechte zu statischen Gruppen können recht einfach

verwaltet werden, da außerhalb dieser Gruppe keine Rechte vergeben werden müssen. Standardmäßig sind beispielsweise die „*groupOfNames*“ (RFC 2256) oder die „*posixGroup*“ (RFC 2307) zur Konstruktion statischer Gruppen vorgesehen.

### 3.6.2 Dynamische Gruppen

Anders als bei statischen Gruppen verwendet eine dynamische Gruppe keine Liste von Gruppenmitgliedern, sondern eine LDAP-Abfrage, deren Ergebnis die Mitglieder der Gruppe enthält. So könnte eine Gruppe aller wissenschaftlichen Mitarbeiter beispielsweise aus allen LDAP-Einträgen bestehen, die vom Typ „*OrganizationalPerson*“ sind und im Attribut „*employeeType*“ einen bestimmten Wert beinhalten. Vorteil dieses Ansatzes ist die Aktualität der Gruppe, die bei jedem Zugriff neu erzeugt wird, Nachteil ist die vergleichsweise hohe Belastung des LDAP-Servers, da die nötige Suche im LDAP-Verzeichnis sehr umfangreich werden kann.

### 3.6.3 Gruppenbildung durch Vorwärtsreferenz

Eine Gruppe kann auch gebildet werden, indem kein spezielles Gruppenobjekt angelegt wird, das die Gruppenmitglieder aufführt (statisch oder dynamisch), sondern indem jedem LDAP-Eintrag, der einer Gruppe zugehören soll, ein spezielles Attribut hinzugefügt wird, das diese Mitgliedschaft kennzeichnet. Beispielsweise könnte ein Attribut „*memberOf*“ die Gruppen auflisten, in denen der Eintrag Mitglied ist. In diesem Fall ist die Vergabe von Zugriffsrechten schwieriger bzw. grobkörniger, als im Fall statischer Definitionen, da bei nur einem „*memberOf*“-Attribut, das von mehreren Gruppen verwendet wird, verschiedene Applikationen Informationen über Gruppenmitgliedschaften erhalten können, die für die Funktion der Anwendung nicht erforderlich sind.
Alternativ kann man für verschiedene Anwendungen verschiedene Vorwärtsreferenzattribute definieren, beispielsweise Wahrheitswerte „*memberOfGroupA*“ und „*memberOfGroupB*“, die sich zugriffstechnisch besser schützen lassen, jedoch vergleichsweise umfangreiche Anpassungen am LDAP-Schema erfordern können.
Vorwärtsreferenzen stehen in Bezug zu dynamischen Gruppen, da das Vorhandensein von Vorwärtsreferenzen Bedingung im Filter einer dynamischen Gruppe sein kann. Wie bei dynamischen Gruppen kann auch hier die Belastung des LDAP-Servers höher sein, wenn beispielsweise eine Liste aller Gruppenmitglieder erstellt werden soll.

### 3.6.4 Räumliche Gruppen

Auch basierend auf der Struktur eines LDAP-Verzeichnisses lassen sich Gruppen bilden. Beispielsweise ist es möglich, in der Baumstruktur disjunkte Bereiche für einzelne Geschäftsbereiche einzurichten und die zugehörigen Objekte im jeweiligen Geschäftsbereich anzulegen. Allerdings kann eine solche Gruppenbildung nicht effizient skalieren, da eine Zuordnung der Einträge zu mehreren Gruppen nicht möglich ist.
Wenn räumlich unterschiedliche Geschäftsbereiche definiert werden, denen jeweils Mitarbeiter und Abteilungsleiter angehören, kann zwar räumlich zwischen den Geschäftsbereichen unterschieden werden, jedoch kann auf diese Weise keine Gruppe „alle Abteilungsleiter“ gebildet werden. Anders als die übrigen Gruppierungsmaßnahmen sind räumliche Gruppen nur in bestimmten, eingeschränkten Fällen einsetzbar.

### 3.6.5 Fazit

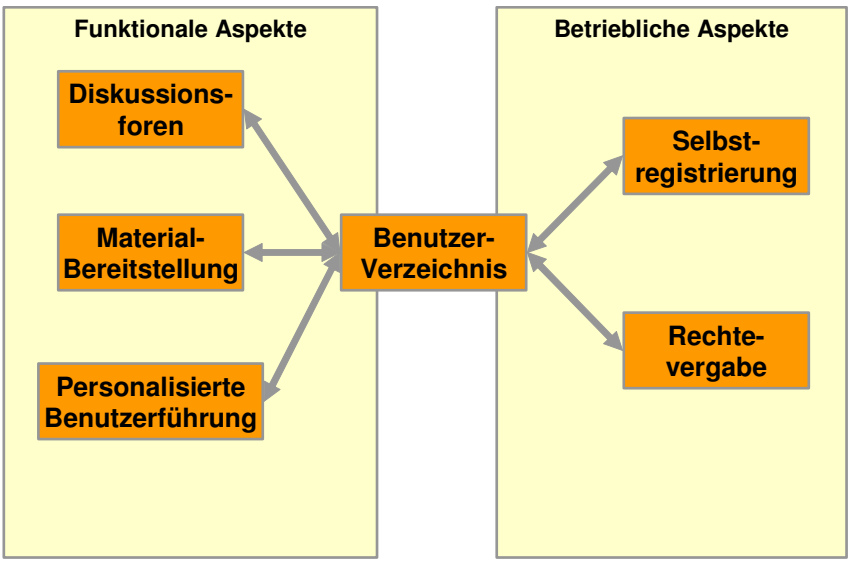
Wir haben unterschiedliche Möglichkeiten betrachtet, um in einem LDAP-Verzeichnis Gruppen definieren zu können. Je nach konkreten Anforderungen und Zugriffscharakteristiken bieten sich unterschiedliche Gruppen zur Nutzung an. Beispielsweise haben statische Gruppen den Vorteil, schnell und effizient eine Liste alle Gruppenmitglieder zu erstellen, während dynamische Gruppen für diesen Anwendungsfall nicht effizient sind, im Gegenzug aber keine Probleme bzgl. Aktualität und Konsistenz haben.
Damit ist also die Definition von Rollenzugehörigkeiten möglich, die von LDAP-fähigen Anwendungen abgefragt werden können. Beispielsweise wäre es ohne weiteres möglich, anhand einer Gruppenzugehörigkeit nur den Mitgliedern der Gruppe „Mitarbeiter“ Zugriff auf eine spezielle Webseite zu geben.
Es ist allerdings zu beachten, dass die Definition der Rechte, die einer speziellen Gruppe zugewiesen werden sollen, jeweils in den Client-Anwendungen durchgeführt werden muss und in der Regel nicht im LDAP-Verzeichnis durchgeführt wird. Das hat unter anderem den Vorteil, dass sich die Rechte einer Rolle je nach Anwendung unterscheiden können. Eine Rolle kann in einer Anwendung Änderungsrechte haben, während in einer anderen Anwendung nur Leseoperationen erlaubt sind.
Für eine tiefere Diskussion siehe beispielsweise [Ba02].

## 3.7 Zusammenfassung

Bei LDAP handelt es sich um einen universellen Verzeichnisdienst, der im Umfeld von X.500 entstanden ist, sich aber inzwischen unabhängig davon weiter entwickelt hat. Da es sich bei LDAP um einen offenen IETF-Standard handelt, konnte LDAP sich schnell sehr weit verbreiten und LDAP-Verzeichnisse können gegenwärtig von einem Großteil der existierenden Betriebssysteme und Programmiersprachen verwendet werden.
Da sich das LDAP-Schema flexibel anpassen lässt, bietet LDAP nicht nur die Möglichkeit, bestehende Verzeichnisdienste wie NIS zu ersetzen, sondern aufgrund der standarisierten Semantik zu Speicherung und Verwaltung von Informationen bietet LDAP darüber hinaus als erster Verzeichnisdienst überhaupt die Möglichkeit, effizient verschiedene, bisher nebeneinander existierende Datenbestände, zu integrieren und über Standardschnittstellen verfügbar zu machen.
Insgesamt hat sich LDAP gegenwärtig im Feld der universell eingesetzten Verzeichnisdienste durchgesetzt und in der Version LDAPv3 weite Verbreitung gefunden. Die integrativen Fähigkeiten werden in zunehmendem Maße genutzt und finden insbesondere im Bereich *Identity Management* Verwendung.

# 4 ANALYSE DES IPO-PORTALS

In der ersten Fassung wird das INFORMATIK-I-Portal, kurz IPO, aus Komponenten bestehen, die zur Materialbereitstellung und zur Realisierung von Diskussionsforen dienen. Außerdem soll eine personalisierte Benutzerführung ermöglicht werden (siehe 1.2.1). Eine detaillierte Analyse des IPO findet sich in [Sc04]. Anhand der im Betrieb gewonnenen Erfahrungen soll das IPO kontinuierlich verbessert und erweitert werden. Deshalb müssen schon jetzt Schnittstellen geschaffen und identifiziert werden, über die spätere Erweiterungen ermöglicht werden können.
Da das IPO von bis zu 700 Studierenden genutzt werden soll, ist es insbesondere erforderlich, eine effiziente Verwaltung und Registrierung dieser Nutzer zu ermöglichen. Manuelle Eingriffe der betreuenden Mitarbeiter sollen die Ausnahme bilden.
Aus Sicht der Nutzerverwaltung stellt sich also vor allem die Aufgabe, eine effiziente Registrierung der Portal-Nutzer ohne manuelle Eingriffe der betreuenden Mitarbeiter zu ermöglichen und einen Datenbestand zu schaffen, der von aktuellen und auch von zukünftigen Komponenten effizient genutzt werden kann. Deshalb ist zum einen die Realisierung einer gemeinsamen Nutzerverwaltung für die beteiligten Komponenten zu untersuchen, zum anderen ist zu prüfen, ob eine Komponente zur Selbst-Registrierung der Nutzer entwickelt werden kann, die den speziellen Anforderungen im IPO-Kontext entspricht.
So wird beispielsweise erwartet, dass die Nutzer dem Portal in den Rollen „Mitarbeiter“, „Tutoren“, „Studierende“ und „Gäste“ gegenübertreten, wobei möglicherweise weitere Differenzierungen nötig werden könnten, z. B. die Aufschlüsselung der „Mitarbeiter“ in „Dozenten“ und „Betreuer“. Anhand dieser Rollen ist eine personalisierte Nutzerführung vorgesehen, sodass für diese Informationen eine effiziente Erfassung und Speicherung innerhalb der Nutzerverwaltung erforderlich ist.
Die gegenwärtig für das IPO entwickelte Monitoring-Anbindung an MCon hat keinerlei Bezug zu den Nutzerdatenbeständen der Studierenden und muss deshalb an dieser Stelle nicht weiter betrachtet werden.




**Information 11: Funktionale vs. betriebliche Aspekte (IPO)**

Insgesamt steht die Nutzerverwaltung zwischen den funktionalen Anforderungen auf der einen und den betrieblichen Anforderungen auf der anderen Seite und hat daher eine Schnittstellenfunktion, über die die einzelnen Seiten voneinander abstrahiert werden können.

## 4.1 Funktionale Anforderungen

### 4.1.1 Personalisierter Zugang

Das Portal muss die Nutzer identifizieren und gemäß ihrer Rollen eine personalisierte Benutzerführung bereitstellen können. Beispielsweise sollen den Nutzern nur die Funktionen angeboten werden, zu deren Durchführen die jeweiligen Nutzer tatsächlich berechtigt sind. Untersucht wird derzeit die Verwendung von Jetspeed, einer Portalsoftware der Apache Software Foundation. Jetspeed kann zur Nutzerdatenverwaltung auf verschiedene SQL-Datenbanken oder auch auf LDAP-Verzeichnisdienste zugreifen.

### 4.1.2 Materialien-Verwaltung

Die mit der Durchführung der INFORMATIK-I-Veranstaltung beauftragten Mitarbeiter sollen gemeinsam für unterschiedliche Zwecke (z. B. Vorlesung, Übung, Tutorium) unterschiedliche Materialien (z. B. Word-Dokumente, Powerpoint-Präsentationen) erstellen, pflegen und über das Portal veröffentlichen. Zur Unterstützung ist der Einsatz einer Kollaborationsumgebung vorgesehen.
Werden Materialien mithilfe dieser Kollaborationsplattform veröffentlicht, so sollen die Nutzer, die den Rollen „Gäste“, „Studierende“ oder „Tutoren“ angehören werden, gemäß ihren individuellen Rollen und den damit verbundenen Rechten auf die Materialien zugreifen können.
BSCW ist ein System, das diese Anforderungen erfüllen kann und bereits in einer Studienarbeit und einer Pilotanwendung im Rahmen der Vorlesung „Kommunikation und Datenhaltung“ (kurz K&D) näher untersucht wurde (siehe [Po04]). Neben der Portalsoftware soll BSCW den zweiten zentralen Bestandteil des IPO-Szenarios bilden. Als Schnittstellen unterstützt BSCW XML-RPC und LDAP.

### 4.1.3 Foren zur Kommunikation

Es ist vorgesehen, Teile der Kommunikation der Mitarbeiter mit den Tutoren, innerhalb der jeweiligen Tutorien und unter den Studierenden mit Hilfe von Diskussionsforen zu unterstützen. Diese Diskussionsforen sollen daher ebenfalls im Portalkontext eingebunden werden.
Im IPO-Kontext ist es dabei erforderlich, den Zugriff auf die verschiedenen Foren anhand der Rollenzugehörigkeiten zu regeln. Beispielsweise sollen Mitarbeitern und Tutoren interne Foren vorbehalten sein, die den Studierenden oder gar Gästen nicht zur Verfügung stehen. Für diese Gruppen werden weitere Foren eingerichtet, die von bestimmten Mitarbeitern moderiert werden können.
Auch für die Bereitstellung von Foren kann BSCW genutzt werden, was Vorteile wie z. B. die einfache Referenz der verwalteten Materialien aus den Foren heraus ermöglichen kann. Außerdem reduziert es die Anzahl der im Gesamtsystem zu integrierenden Komponenten und minimiert den Einarbeitungsaufwand für die Mitarbeiter, da diese Foren bereits zur Unterstützung anderer Veranstaltungen (z. B. K&D) erfolgreich eingesetzt wurden. Ein Einsatz alternativer Lösungen ist ebenfalls denkbar, sofern eine einfache Integration in die Portalumgebung und in die Nutzerverwaltung möglich ist.

### 4.1.4 Rollen und Gruppen auf IPO-Ebene

Die im IPO vorgesehenen Rollen und damit verbundene Berechtigungen (siehe 4.1) sind, bezogen auf die einzelnen Komponenten, in der folgenden Tabelle aufgeführt.

| | **Diskussionsforum** | **Materialablage** | **Portal** |
|---|---|---|---|
| **Anonym** | Keine Rechte. | Abrufen von Materialien im öffentlichen Bereich. | Kein Zugriff auf das Forum und nur Zugriff auf Materialien im öffentlichen Bereich. |
| **Registrierter, externer Nutzer** | Darf Beiträge im öffentlichen Forum lesen. | Darf auch im geschützten Bereich Materialien abrufen. | Zugriff auf öffentliches Forum, und geschützten Material-Bereich, Lesen der FAQ. |
| **Registrierte Studierende oder Angehörige der Universität** | Darf im öffentlichen Forum lesen und schreiben. | Darf auch im geschützten Bereich Materialien abrufen. | Zugriff auf Forum und geschützten Material-Bereich, Lesen der FAQ. |
| **Tutor** | Darf im öffentlichen Forum lesen und schreiben, ebenso in bestimmten internen Foren. | Darf auf alle Materialien für Studierende sowie auf spezielle Materialien zur Tutorienvorbereitung zugreifen. | Wie Studenten, zusätzlich weitere Foren und weitere Material-Bereiche. |
| **Mitarbeiter** | Darf in allen Foren lesen und schreiben. | Darf in allen Bereichen Materialen abrufen, veröffentlichen und ändern. | Zugriff auf alle Foren und Bereiche, Aktualisierung aller Daten (z. B. FAQ). |

**Tabelle 2: Nutzerrechte im IPO-Kontext**

## 4.2 Betriebliche Anforderungen

Insgesamt werden als Nutzer des IPO 500-700 Studierende und registrierte Angehörige der Universität erwartet, außerdem 24 Tutoren und fünf bis zehn Mitarbeiter. Während die Tutoren und Mitarbeiter bereits vor Vorlesungsbeginn bekannt sind, können die Studierenden erst nach Vorlesungsbeginn erfasst und registriert werden. Um eine möglichst effektive Unterstützung der Vorlesung durch das Portal zu gewährleisten, muss die Nutzung des Portals durch die Studierenden unmittelbar nach Vorlesungsbeginn möglich sein und daher die Registrierung der Studierenden innerhalb kürzester Zeit erfolgen.

Angesichts dieser Zahlen und auch der zeitlichen Einschränkungen ist klar, dass für die Studierenden und Universitätsangehörigen eine Lösung zur automatisierten Registrierung und Verwaltung gefunden werden muss, da eine manuelle Bearbeitung nicht effizient zu realisieren ist. Gleichzeitig ist dies für die Tutoren und Mitarbeiter nicht unbedingt erforderlich, da diese bereits vor Beginn der Lehrveranstaltung registriert werden können und auch die Gesamtzahl eine manuelle Verarbeitung ermöglicht.
Die Anmeldung und Registrierung der Studierenden und Universitätsangehörigen als IPO-Nutzer soll so weit wie möglich eigenständig und ohne administrative Eingriffe der betreuenden Mitarbeiter am Portal durchgeführt werden können. Auch im laufenden Betrieb sollen Standardfälle wie beispielsweise vergessene Passwörter automatisch vom System behandelt werden, wozu eine entsprechende Komponente im System vorzusehen ist. Daher muss untersucht werden, ob für diese Aufgabe die vorhandenen Verwaltungswerkzeuge der anderen Komponenten verwendet bzw. angepasst werden können oder ob die Entwicklung einer zusätzlichen, eingeständigen Komponente erforderlich wird.
Es ist auch zu beachten, dass die Entwicklung des IPO nicht als abgeschlossen betrachtet werden kann und spätere Erweiterungen um neue Komponenten möglich sein sollen. Deshalb ist es wesentlich, zur Nutzerverwaltung existierende Standards zu verwenden, um so eine einfache Integration künftiger Komponenten zu ermöglichen.
Dem Portal müssen verschiedene Daten bekannt sein, um die Nutzer individuell bzw. personalisiert zu unterstützen und die deshalb von der Nutzerverwaltung erfasst werden müssen. Dies umfasst sowohl die technisch erforderlichen Daten zum Betrieb der einzelnen Komponenten, als auch durch den Betreiber gestellte Anforderungen.
Aus Sicht des Portals umfassen diese Daten minimal einen Login, ein Passwort sowie die einem Nutzer zugewiesenen Rollen. Da unterschiedliche Rollen wie „extern“ und „Studierender“ automatisch zugeteilt werden sollen, ist es erforderlich, diese Rollenzugehörigkeiten automatisch verifizieren zu können. Dies kann durch Daten erfolgen, die nur einem Angehörigen der jeweiligen Rolle bekannt sind und die bei der Registrierung abgefragt werden können, bei einem Studierenden beispielsweise durch Angabe einer gültigen Matrikelnummer oder Fricardnummer. Eine Matrikelnummer dient der eindeutigen Identifikation eines Studierenden und wird bei der Immatrikulation vergeben. Die Fricardnummer ist eine Identifikationsnummer für die Fricard, den Studentenausweis der Universität. Diese Nummer kann zur eindeutigen Identifikation von Studierenden verwendet werden, kann sich aber verändern, beispielsweise wenn nach dem Verlust einer Fricard ein neuer Ausweis ausgestellt wird. Daher wäre die Fricardnummer über einen längeren Zeitraum nur schlecht geeignet, kann aber dennoch im zeitlich begrenzt nutzbaren IPO verwendet werden.
Um bei Verwaltungsvorfällen wie etwa einem vergessenen Passwort automatisiert reagieren oder die Nutzer in Problemfällen persönlich kontaktieren zu können, ist schließlich die Erfassung einer gültigen E-Mail-Adresse notwendig. Zusätzlich sollen als Betreiberanforderung Vorname und Nachname als weitere Nutzerdaten gespeichert werden. Auch vom System generierte Informationen wie beispielsweise der Zeitpunkt der Registrierung können für die Nutzerverwaltung sinnvoll sein.

## 4.3 Zusammenfassung der Anforderungen

Das IPO wird zunächst aus den Komponenten Jetspeed und BSCW sowie möglicherweise einer Komponente zur Registrierung und LDAP-basierten Verwaltung der Nutzer bestehen. Hauptanforderungen an die Schnittstelle zur Nutzerverwaltung sind die Möglichkeit zur Selbstverwaltung und –registrierung von 700 Nutzern ohne administrative Eingriffe der betreuenden Mitarbeiter, sowie die einfache Integration in

die vorgesehenen Komponenten. Auch zukünftige Erweiterungen des Portals durch zusätzliche Komponenten sollten sich in die Nutzerverwaltung integrieren lassen, sodass die Verwendung von Standardschnittstellen erforderlich ist.

- Das IPO besteht aus verschiedenen Komponenten
  - Jetspeed
  - BSCW
  - Modul zur Registrierung der Studenten
- LDAP existiert als gemeinsame Schnittstelle von Jetspeed und BSCW
- Spätere Erweiterungen sind möglich
  - Webinscribe
  - Andere Foren mit LDAP-Support
  - Etc.

**Information 12: Überblick**

Da sowohl Jetspeed als auch BSCW eine LDAP-Schnittstelle zur Nutzerverwaltung bieten, sollen zuerst diese LDAP-Schnittstellen näher untersucht werden, um festzustellen, ob eine Datenintegration über einen gemeinsamen LDAP-Datenbestand mit geringem Aufwand möglich ist. Um künftige Erweiterungen zu unterstützen, sind Abweichungen von den existierenden Standards für LDAP-Strukturen (beispielsweise RFC 2256 oder RFC 2307) soweit möglich zu vermeiden. Eine nähere Untersuchung möglicher Erweiterungen ist an dieser Stelle jedoch nicht vorgesehen.

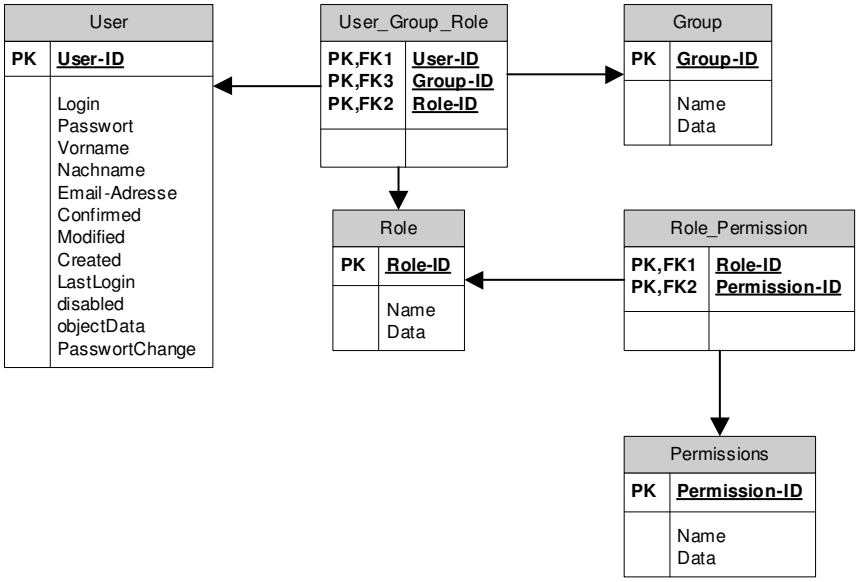
## 4.4 Jetspeed Analyse

### 4.4.1 Überblick

Jetspeed ist eine Portalsoftware auf Java-Basis, die als Projekt der Apache Software Foundation entwickelt wird. Wie in [Sc04] motiviert, soll Jetspeed als Integrations-Plattform für alle weiteren Komponenten soll Jetspeed in der Version 1.4 dienen. In dieser Version wird der Java-Portlet-Standard JSR-168 noch nicht unterstützt, dies ist jedoch für die Version „Jetspeed 2“ geplant, die gegenwärtig entwickelt wird.
Bei der Entwicklung von Jetspeed wurde die Datenspeicherung und Nutzerverwaltung durch die Anbindung an verschiedene SQL-Datenbanken unter Verwendung des JDBC-Standards realisiert. Um die Nutzung unterschiedlicher Datenbanken zu ermöglichen bzw. zu vereinfachen, wurde dazu eine Schnittstelle definiert, der die jeweiligen Datenbank-Konnektoren entsprechen müssen. Später wurde eine LDAP-Anbindung gemäß dieser Schnittstellendefinition hinzugefügt, die die bisherigen SQL-Anbindungen nicht ergänzt, sondern vollständig ersetzt. Die verwendete LDAP-Struktur orientiert sich dabei sehr eng an den älteren SQL-Tabellen und ist damit nicht konform zu den üblicherweise eingesetzten LDAP-Strukturen (siehe 3.4). Dies kann leider bei der Integration weiterer LDAP-fähiger Anwendungen Probleme verursachen. Darauf wird im Folgenden im Detail eingegangen.

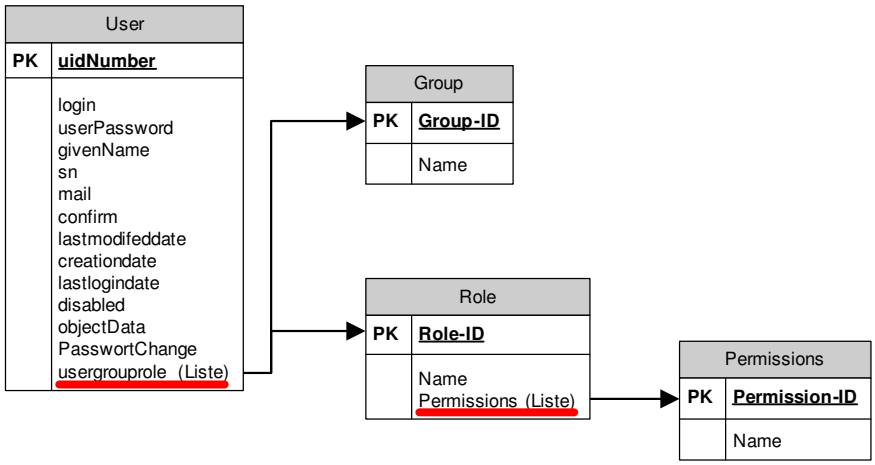
### 4.4.2 Übergang von SQL zu LDAP

Zum besseren Verständnis soll kurz betrachtet werden, wie die Jetspeed-Daten ursprünglich in Form einer SQL-Datenbank strukturiert waren und wie die Transformation dieser Daten zu LDAP erfolgte. Anschließend lassen sich die speziellen Eigenschaften der realisierten LDAP-Unterstützung besser untersuchen.



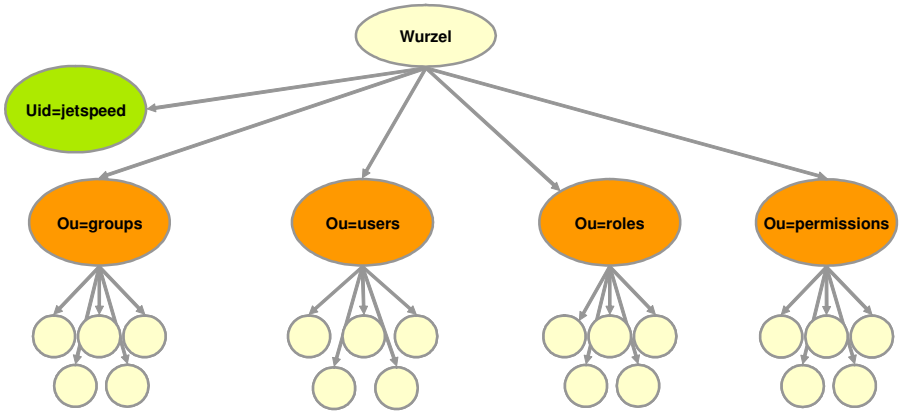

**Information 13: Jetspeed-SQL**

Wie in Information 13 ersichtlich, speichert Jetspeed Informationen über „*User*“, „Rollen“, „Gruppen“ und „Rechte“. Zwei weitere Tabellen „*User_Group_Role*“ und „*Role_Permission*“ sind nötig, um die Beziehungen dieser Daten abbilden zu können, enthalten aber keine eigenständigen Attribute.




**Information 14: Jetspeed-LDAP: Transformation**

Bei der Transformation in einen LDAP-Baum wurden die Tabellen „*User_Group_Role*“ und „*Role_Permission*“ aufgelöst und durch Listen ersetzt, die den Einträgen „*User*“ und „*Role*“ direkt in Form von mehrwertigen Attributen (siehe 3.4) zugeordnet werden konnten. Die ternäre Relation „*User_Group_Role*“ konnte allerdings nicht durch eine einfache Liste aufgelöst werden, sondern die Elemente der Liste haben die Form „*GROUP;ROLE*“. Eine solche Liste zur Rollen- bzw. Gruppenabbildung ist leider nicht LDAP-typisch und lässt sich von anderen Anwendungen nicht verwenden (siehe auch 3.5 und 3.6).
Aus jeder der vier noch verbleibenden Tabellen wurde ein eigener Unterbaum gleichen Namens abgeleitet. Alle Elemente im jeweiligen Unterbaum entsprechen dann einem Eintrag der ehemaligen SQL-Tabelle.




**Information 15: Jetspeed-LDAP: Baum-Struktur (Standard)**

Außerdem wurde ein LDAP-User „Jetspeed“ eingeführt, der volle Rechte auf sämtliche Jetspeed-Bereiche hat. Dieser Nutzer wird von Jetspeed als *Proxy User* (siehe 3.4.4) verwendet, um auf das LDAP-Verzeichnis zugreifen zu können.

### 4.4.3 Schwächen der LDAP-Umsetzung

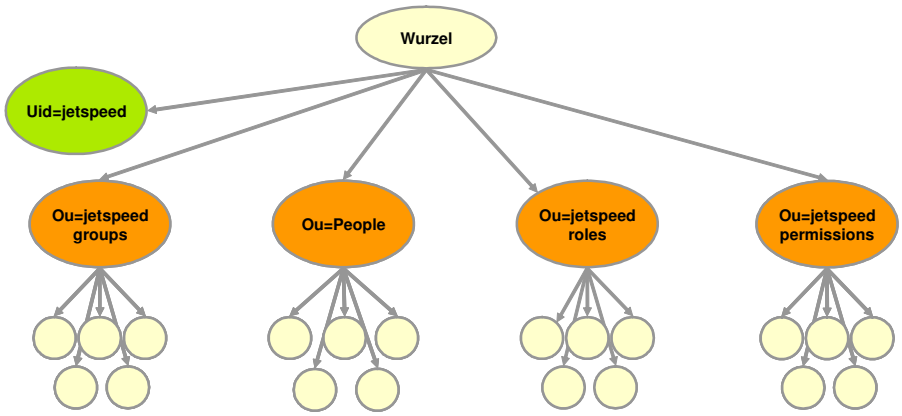
- LDAP-Struktur fest vorgegeben
- Verwendet ungewöhnliche Bezeichner für Baumstruktur
- Um die vorgesehene LDAP-Struktur zu verändern sind Quellcode-Anpassungen nötig
- Definiert proprietäre ObjectClass „jetspeeduser“
- Jetspeed benötigt Lesezugriff auf User-Passwort, aus LDAP-Sicht ist das unnötig
- Passwörter im Klartext gespeichert, für Crypt-Passwörter sind kleine Quellcode-Anpassungen nötig
- Rollen und Rechte werden in einem proprietären Format im LDAP-Verzeichnis gespeichert
- Häufige Aktualisierung von LDAP-Einträgen im Verzeichnisserver, unabhängig von Änderungen

**Information 16: Schwächen von Jetspeed-LDAP**

Leider werden in der LDAP-Unterstützung von Jetspeed existierende Standards zur Nutzerdatenspeicherung in LDAP nicht beachtet. So wird die Struktur des LDAP-Verzeichnisses fest vorgegeben und Standard-Objektklassen wie beispielsweise „*Person*“ werden nicht verwendet, sondern durch eigene, nicht kompatible Objektklassen ersetzt. Die widerspricht beispielsweise RFC 2256 oder RFC 2307, die von vielen Anwendungen beachtet werden und könnte daher die Integration mit anderen Komponenten erschweren.
Auch die Speicherung der Passwörter im Klartext ist unnötig und aus Sicherheitsgründen unerwünscht. Hingegen ist die mögliche Selbst-Registrierung und Verwaltung der Nutzer über die Jetspeed-Oberfläche sinnvoll, die genaue Realisierung und die Eignung im IPO-Kontext müssen jedoch auf ihre Eignung untersucht werden.

### Bezeichnungen der Unterbäume

Es haben sich bereits Bezeichnungen für Unterbäume entwickelt (im Unix-Umfeld siehe RFC 2307), die von vielen LDAP-Anwendungen genutzt werden. Beispielsweise wird für die Speicherung von Gruppen häufig der Bereich „*Ou=group*“ verwendet, statt wie bei Jetspeed „*Ou=groups*“. Da die von Jetspeed in „*Ou=groups*“ verwendete LDAP-Objektklasse nicht der üblicherweise in „*Ou=group*“ verwendeten und erwarteten LDAP-Objektklasse entspricht (siehe auch „Ungewöhnliche Konstruktion der Gruppen“), ist eine solche Trennung in diesem Fall zwar sinnvoll und vertretbar, jedoch sollte aufgrund der Verwechslungsgefahr zwischen „*group*“ und „*groups*“ ein eindeutiger Bezeichner wie z. B. „*Ou=Jetspeedgroups*“ verwendet werden. Auch die Bäume „*Ou=roles*“ und „*Ou=permissions*“ werden ausschließlich von Jetspeed verwendet und sollten deshalb besser gekennzeichnet werden.
In dem Bereich „Ou=users“ werden die einzelnen Nutzereinträge gespeichert. Da zur Verwaltung von Nutzereinträgen üblicherweise in LDAP-Anwendungen der Bereich „*Ou=people*“ verwendet wird, erscheint auch hier eine Umbenennung sinnvoll. Allerdings sollte dann sichergestellt werden, dass die in diesem Bereich gespeicherten Einträge auch den üblichen Konventionen entsprechen und die Objektklasse „*Person*“ verwenden, was gegenwärtig nicht der Fall ist. Diese Problematik wird im Folgenden unter Verwendung der Klasse „JetspeedUser“ näher diskutiert.
Leider können die Namen der einzelnen Bereiche nicht per Konfigurationsanweisung verändert werden, weshalb Quellcodeanpassungen erforderlich wären. Da es sich bei den Bezeichnern allerdings lediglich um Konstanten handelt, sollten diese Anpassungen einfach zu realisieren sein.



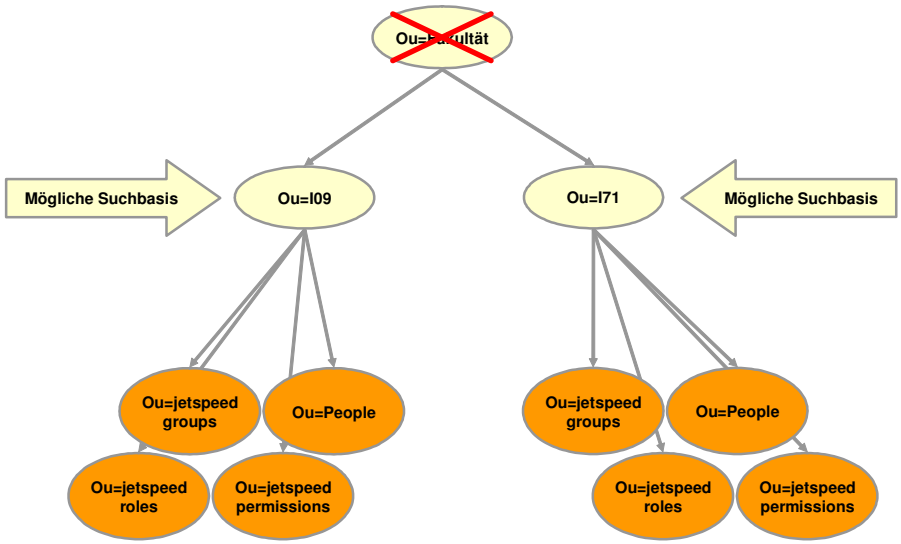

**Information 17: Jetspeed-LDAP: Baum-Struktur (verbessert)**

Information 17 zeigt die vorgeschlagenen Verbesserungen der LDAP-Struktur. In den weiteren Grafiken werden wir diese Struktur weiter verwenden.

### LDAP-Suchbasis nicht konfigurierbar

Ausgehend von einem gemeinsamen Wurzelknoten erwartet Jetspeed für User, Gruppen, Rollen und Rechte jeweils einen eigenen Unterbaum mit von Jetspeed vorgegebener Bezeichnung (siehe „Bezeichnungen der Unterbäume“) und startet die Suche im jeweiligen Unterbaum. So startet die Suche nach einem User immer in einem

Knoten mit Bezeichnung „*Ou=People*“, die Suche nach einer Gruppe immer in „*Ou=jetspeedgroups*“.




**Information 18: LDAP-Suche mit Jetspeed**

Bei größeren LDAP-Bäumen ist diese Forderung mitunter schlecht zu realisieren und wird von anderen LDAP-Anwendungen (beispielsweise BSCW, siehe 4.5) nicht erhoben. So kann oft eine beliebige Wurzel angegeben werden, von der aus über mehrere, beliebig bezeichnete Unterbäume hinweg nach einem Eintrag gesucht werden kann (siehe Information 20), was dem Betreiber eine wesentlich flexiblere Organisation und Strukturierung des Verzeichnisses ermöglicht.

### Ungewöhnliche Konstruktion der Gruppen

Es gibt verschiedene Möglichkeiten, wie Gruppen- und Rollenzugehörigkeiten gespeichert werden können (siehe 3.6). Man kann der Gruppe die Information zuweisen, welche Nutzer ihr angehören oder aber man kann dem Nutzer die Information zuweisen, welchen Gruppen er angehört. Der erste Ansatz findet bei LDAP-Anwendungen insbesondere in Form der Klasse „*groupOfNames*“ Verwendung und wird beispielsweise von der Unix-Nutzerverwaltung, dem Apache-Webserver, Samba und verschiedenen Groupware-Anwendungen unterstützt. Damit kann ein Anwender durch eine zentrale Änderung an seiner Gruppenzugehörigkeit auf zusätzliche Dateien, zusätzliche Webseiten und zusätzliche *Groupware*-Funktionen zugreifen. Auch der zweite Ansatz wird in der Praxis eingesetzt, ein Beispiel ist das Microsoft Active Directory. Eine Gruppenbildung durch räumliche Gruppen (siehe 3.6.4) kann nicht verwendet werden, wenn ein Nutzer mehreren Gruppen angehören soll, und wird daher an dieser Stelle nicht weiter betrachtet.
Leider nutzt Jetspeed keinen der diskutierten Ansätze, da es keine einfache „Nutzer-Gruppen-Abbildung“ verwendet, sondern den Nutzern innerhalb der Gruppe zusätzlich unterschiedliche Rollen zuweist. Diese beiden Informationen „Gruppe“ und „Rolle innerhalb dieser Gruppe“ werden zu einem einzigen Attribut verschmolzen und im Verzeichnis gespeichert. Damit ist die LDAP-übliche Gruppenbildung an dieser Stelle nicht möglich. Zum Gründen dieser Konstruktion siehe 4.4.2.
Vor dem Hintergrund der gewünschten Integration im Rahmen des IPO ist dieses Verhalten leider hinderlich. Ist eine Verwendung des LDAP-Verzeichnisses in anderen Anwendungen geplant, die die übliche Gruppenabbildung verwenden, so entsteht ein

Problem, dass nur durch eine doppelte Darstellung der Gruppen im LDAP-Verzeichnis gelöst werden, indem sowohl Einträge gemäß Jetspeed-Schema als auch weitere Einträge entsprechend der üblichen LDAP-Schemata generiert werden. Diese redundante Datenpflege könnte im Rahmen der Betriebsprozesse gelöst und automatisiert werden.
Eine Anpassung von Jetspeed an eine üblichere Gruppenverwaltung scheint ohne Aufgabe des Jetspeed-Rollenkonzeptes nicht möglich zu sein. Allenfalls könnte Jetspeed derart erweitert werden, dass zusätzlich zur Jetspeed-Gruppendarstellung auch eine zusätzliche Gruppendarstellung erzeugt wird, sodass anderen Anwendungen eine kompatible Darstellung zur Verfügung steht. Die Frage, ob die Jetspeed-eigene Nutzer- und Gruppenverwaltung überhaupt genutzt oder ein alternatives Werkzeug entwickelt werden soll, wird an anderer Stelle diskutiert (siehe 4.4.4).

### Verwendung der Klasse „JetspeedUser“

Jetspeed verwendet zur Verwaltung der Nutzer ausschließlich die eigene Klasse „*JetspeedUser*“, obwohl die Speicherung der erforderlichen Daten auch als Erweiterung der standarisierten LDAP-Objektklassen „*Person*“ bzw. „*inetOrgPerson*“ möglich gewesen wäre (siehe RFC 2256 und RFC 2798). Da genau diese Klassen von einigen anderen LDAP-Anwendungen wie z. B. BSCW (siehe 4.5) verwendet bzw. sogar vorausgesetzt werden, ergibt sich hier durch fehlende Konformität abermals ein Integrationsproblem. Man kann die Jetspeed-Nutzerverwaltung jedoch einfach erweitern, um dieses Problem zu lösen. Die zur Verwendung der Objektklasse „*Person*“ nötigen Erweiterungen sind sehr überschaubar, da schon jetzt von Jetspeed alle nötigen Attribute erfasst werden und daher vorliegen. Es müsste lediglich ein zusätzliches, automatisch generierbares Attribut gespeichert und eine zusätzliche Objektklasse angegeben werden. Außerdem wäre eine minimale Änderung der JetspeedUser-Objektklasse nötig, die statt als „strukturelle Klasse“ als „Hilfsklasse“ deklariert werden müsste (siehe 3.4).
Eine Alternative zur Erweiterung bzw. Modifikation von Jetspeed wäre die Entwicklung einer eigenen Komponente zur Nutzerverwaltung, die dann die Verwendung beider Objektklassen gewährleisten könnte, womit das Problem in den Bereich der Betriebsprozesse verlagert würde. Die bereits angesprochenen Änderungen an der JetspeedUser-Objektklasse wären auch in diesem Fall erforderlich.

### User-Authentifikation gegenüber LDAP

Prinzipiell kann sich jeder Eintrag in einem LDAP-Server, dem ein Passwort zugewiesen wurde, gegenüber dem LDAP-Server anmelden. Dazu wird im LDAP-Verzeichnis anhand des Nutzernamens nach dem Nutzereintrag gesucht. Wird ein passender Eintrag gefunden, so wird anschließend eine weitere LDAP-Verbindung aufgebaut, die den LDAP-Eintrag und das übergebene Passwort zur Authentifizierung verwendet. Akzeptiert der LDAP-Server diese Anmeldung, so ist das Passwort ist korrekt (siehe 3.5).
Dies ist das Standard-LDAP-Vorgehen und ermöglicht es, die Nutzerpasswörter effektiv zu schützen. Das Passwort muss von den Client-Anwendungen nicht gelesen werden können, sodass eine durch einen Angreifer kompromittierte Client-Anwendungen keine Gefahr für das LDAP-Verzeichnis darstellt. Ein weiterer Vorteil ist die Tatsache, dass im LDAP-Server eine beliebige Einwege-Hash-Funktion verwendet werden kann, um die Speicherung der Passwörter im Klartext zu vermeiden.
Im Gegensatz zum üblichen Vorgehen fragt Jetspeed jedoch das LDAP-Passwort ab, wozu Leserechte benötigt werden. Außerdem kann Jetspeed lediglich die „Crypt“-Einwege-Hash-Funktion anwenden, um geschützte Passwörter zu vergleichen. Diese

Funktion genügt aktuellen Sicherheitsansprüchen jedoch nicht mehr, da die Crypt-Funktion auf dem *Data Encryption Standard*, kurz DES, basiert, das bereits 1998 durch *Brute-Force*-Angriffe kompromittiert werden konnte. Statt dessen werden daher heute Verfahren wie MD5 oder SHA verwendet.
Eine Änderung dieser Nutzer-Authentifikation wäre wünschenswert, würde aber einen vergleichsweise großen Programmieraufwand erfordern. Vorteile wären die größere Flexibilität, da Jetspeed nicht mehr auf eine spezielle Konfiguration des verwendeten Verzeichnisdienstes angewiesen wäre. Außerdem würde Jetspeed weniger Rechte auf das kritische Passwortattribut benötigen, womit eine bessere Sicherheit des LDAP-Verzeichnisses erreicht werden könnte.

#### Pflege unnötiger Attribute

Die Speicherung der Zeitstempel, wann ein Eintrag angelegt oder verändert wurde, wird vom Jetspeed selbst in den Attributen „*creationdate*“ und „*lastmodifieddate*“ erfasst. Dies ist jedoch insofern unnötig, als das diese Informationen in der Regel automatisch vom verwendeten LDAP-Server gepflegt werden und in den Attributen „*createtimestamp*“ oder „*modifytimestamp*“ verfügbar sind.

#### Häufige Datenspeicherung aller Attribute

Jetspeed speichert in mehreren Situationen den vollständigen, aktuellen Datensatz im LDAP-Server, beispielsweise bei jeder Abmeldung eines Nutzers vom System. Dieses Verhalten ist sowohl aus Geschwindigkeits- als auch aus Sicherheitsgründen unerwünscht.
Da Jetspeed sämtliche Attribute im Verzeichnis aktualisiert, unabhängig davon, ob sie überhaupt verändert worden sind, wird eine unnötig hohe Belastung des LDAP-Servers erzeugt. Da bei der Implementierung von Verzeichnisdiensten in der Regel davon ausgegangen wird, dass es nur wenige Schreiboperationen auf die einzelnen Datensätze geben wird (siehe 2.3), ist dies ungünstig. Bei einem dedizierten LDAP-Server für eine Jetspeed-Installation stellt dies zwar noch kein Problem dar, kann aber bei LDAP-Servern, die auch für andere Zwecke eingesetzt werden, spürbare Auswirkungen haben
Die Speicherung aller Jetspeed bekannten Attribute erfordert auch Schreibrechte auf alle Attribute, was kein Problem darstellt, wenn ein dedizierter LDAP-Server für eine Jetspeed-Installation verfügbar ist und Jetspeed das führende System darstellt. Im aktuellen Szenario sind diese Voraussetzungen jedoch nicht notwendigerweise gegeben. Zum einen kann die Nutzung des LDAP-Verzeichnisses auch für andere Zwecke sinnvoll sein, zum anderen kann ein anderes System als führendes System gewählt werden, beispielsweise eine externe Nutzerverwaltung (siehe auch 4.7). Anhand des Quelltextes lässt sich die Funktion, die diese Speicherung durchführt, leicht finden und dort die Menge der zu speichernden Attribute einschränken. Da diese Funktion jedoch auch zum Anlegen von Accounts verwendet wird, kann eine Einschränkung nur durchgeführt werden, wenn Jetspeed selbst keine Accountverwaltung durchführt. Diese Frage wird jedoch erst später diskutiert, sodass hier noch keine Entscheidung getroffen werden kann, ob eine solche Modifikation im konkreten Fall sinnvoll wäre.

### 4.4.4 Nutzerverwaltung durch Jetspeed

Jetspeed ist selbst in der Lage, neue Nutzer im LDAP-Verzeichnis anzulegen und bietet Möglichkeiten zur Selbst-Registrierung und Selbst-Verwaltung. Leider gelten für diese Funktion alle bereits angeführten Nachteile bzgl. der LDAP-Unterstützung von Jetspeed.

Insbesondere wäre von Jetspeed nicht vorgesehen, das ein Teil der Attribute auch von anderen führenden Systemen verwaltet werden kann. Da eine Konfiguration, bei der mehrere führende Systeme existieren, nicht ausgeschlossen werden kann, stellt dies leider einen Nachteil dar. Außerdem sollten im IPO-Kontext zusätzliche Attribute gespeichert werden (Matrikelnummer, Account-Typ). Die dazu erforderlichen Änderungen würden einen vergleichsweise hohen Aufwand darstellen.
Die Frage der Nutzerverwaltung im IPO-Kontext wird in 4.7und 4.8 näher untersucht.

### 4.4.5 LDAP-Ausblick

Jetspeed liegt aktuell in Version 1.5 vor. Gegenwärtig wird die Version 2 entwickelt, für die eine neue LDAP-Anbindung angekündigt ist. Details sind noch nicht bekannt, es ist aber geplant, die angesprochenen Schwächen in der nächsten Version zu beheben.

### 4.4.6 Fazit

Ein Teil der Schwächen kann mit vergleichsweise geringem Aufwand behoben werden. Die Bezeichner der Unterbäume können durch Quellcodeanpassungen geändert werden und auch die zusätzliche Verwendung der Objektklasse „*Person*“ stellt für Jetspeed kein Problem dar.
Eine Änderung der Gruppenverwaltung oder der Authentifikation würde hingegen deutlich mehr Aufwand erzeugen und ist mit Blick auf die angekündigte Version 2 vermutlich nicht mehr sinnvoll.

## 4.5 BSCW Analyse

### 4.5.1 Überblick

BSCW ist eine in Python entwickelte Anwendung, die über eine HTML-Oberfläche ein Kollaborationssystem realisiert. Dabei umfasst BSCW eine umfangreiche Materialablage und -verwaltung, Diskussionsforen, einen Kalender und ein Adressbuch.
Als weitere Schnittstellen unterstützt BSCW WebDAV zum Zugriff auf die Dateiablage, eine XML-RPC-Schnittstelle zum externen Zugriff auf alle Funktionen sowie eine LDAP-Schnittstelle. Die in einem LDAP-Server verfügbaren Daten können dabei als Adressbuch und zur Nutzerauthentifikation verwendet werden. Da BSCW eine eigene, nicht-relationale Datenbank verwendet, fehlen entsprechende Schnittstellen für externe Anwendungen, um direkt auf den BSCW-Datenbestand zugreifen zu können.
Die Authentifizierung von Nutzern kann entweder von BSCW selbst durchgeführt werden, oder aber über die eingesetzte Webserver-Software, beispielsweise Microsoft IIS oder Apache Webserver. In beiden Fällen können externe Quellen zur Authentifizierung herangezogen werden, beispielsweise ein LDAP-Server oder ein Active Directory. Damit kann jedoch nicht die Nutzerverwaltung innerhalb von BSCW ersetzt werden, die nach wie vor zur Rollenverwaltung und Rechtevergabe benötigt wird. Damit dient LDAP hier lediglich als Ergänzung im Bereich der Nutzerverwaltung und Authentifikation (siehe 3.5.2), nicht aber zum Ersatz der eigenen Datenspeicherung, wie das beispielsweise bei Jetspeed der Fall ist (siehe 4.4.4).
BSCW wurde bereits im Rahmen der Vorlesung Kommunikation und Datenhaltung (kurz K&D) als vorlesungsbegleitendes System zur Materialablage und zur Bereitstellung von Diskussionsforen eingesetzt. Aufgrund der dabei erworbenen Erfahrungen soll BSCW mit diesen Funktionen nun auch im Rahmen des IPO verwendet werden. Weitergehende Informationen finden sich in [Po04].

### 4.5.2 BSCW: interne Nutzerverwaltung

Neben den Begriff des „Nutzers“, der sich über einen Namen und ein Passwort gegenüber dem System authentifizieren kann, kennt BSCW noch die „Arbeitsbereiche“ und ein Rechte- und Rollenkonzept.
Ein Arbeitsbereich ist eine Menge von Objekten innerhalb des BSCW-Systems, beispielsweise Dateien, Verzeichnisse und Diskussionsforen. Zu einem solchen Bereich kann der authentifizierte Nutzer Zugriff erhalten, wobei ihm eine bestimmte Rolle innerhalb des Arbeitsbereichs zugewiesen wird. Dabei kann und wird ein Nutzer in unterschiedlichen Arbeitsbereichen unterschiedliche Rollen verwenden. Mögliche, standardmäßig eingerichtete Rollen sind beispielsweise „Anonym“, Mitglied“ oder „Manager“. Jeder Rolle ist schließlich eine Menge von Rechten zugewiesen, beispielsweise „lesen“ oder „ändern“.
Sollte ein Zugriff auf diese Nutzerverwaltung nötig werden, so BSCW drei mögliche Schnittstellen an, zum einen über die HTML-Oberfläche, über ein Script „bsadmin“ und über eine XML-RPC-Schnittstelle. Zur Automatisierung eignet sich die Verwendung der XML-RPC-Schnittstelle aller Voraussicht nach am besten.
Intern identifiziert BSCW die Nutzer über deren E-Mail-Adresse, und betrachtet daher die Änderung der E-Mail-Adresse als privilegierte Aufgabe, die nur der Administrator durchführen kann.
Als besondere Funktion kann speziellen Nutzern die Möglichkeit gegeben werden, weitere Anwender zur Nutzung von BSCW einzuladen. Diese Nutzer erhalten dann automatisch einen BSCW-Zugang und können über einen gemeinsamen Kalender oder gemeinsame Dateibereiche miteinander kollaborieren. Wird diese Funktion verwendet, so haben die kann eine dezentrale Selbstverwaltung der Nutzer entstehen, die große Flexibilität ermöglicht und den Betreiber entlastet. Jedoch existiert in diesem Fall keine zentrale Verwaltung der Nutzer.

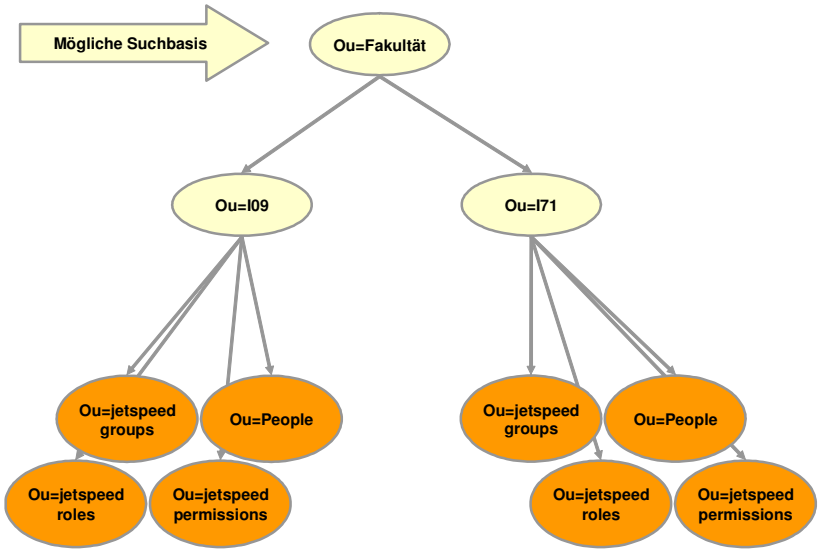
### 4.5.3 BSCW und LDAP im Detail

- LDAP ergänzt die eigene Datenspeicherung
- BSCW setzt die Objektklasse „Person“ voraus
- LDAP-Struktur beliebig
  - „Suche ab Wurzel nach Eintrag mit Login s_hagen und verwende diesen“
- Benötigte Attribute
  - „Common Name“ = vollständiger Name
  - „surname“ = Familienname
  - Username
  - Passwort
  - Email-Adresse
- BSCW legt LDAP-Nutzer beim ersten Login mit default-Rollen in der eigenen Datenverwaltung an
- BSCW ermöglicht es dem Nutzer, Daten im LDAP zu verändern (Passwort, etc.)

**Information 19: BSCW und LDAP**

BSCW setzt die Verwendung der LDAP-Objektklasse „*Person*“ voraus, was LDAP-seitig die Attribute „*Common Name*“ (vollständiger Name, z. B. „Max Mustermann“) und „*Surname*“ (Nachname) erzwingt. Zusätzlich benötigt BSCW noch die Speicherung

einer E-Mail-Adresse, eines Nutzernamens und eines Passworts. Als Nutzername kann ein beliebiges Attribut eingesetzt werden, also beispielsweise die ohnehin benötigte E-Mail-Adresse. In der Regel wird jedoch ein zusätzliches Attribut „*uid*“ verwendet, das einen speziellen Nutzernamen enthält. Informationen zur Objektklasse „*Person*“ sowie den genannten Attributen findet man insbesondere in RFC 2256.
Anders als Jetspeed macht BSCW keine speziellen Vorgaben zur LDAP-Struktur. Ausgehend von einer beliebigen Wurzel ist die Suche nach LDAP-Einträgen mit passendem Nutzernamen möglich. Bei dieser Suche können auch weitere Bedingungen angegeben werden, beispielsweise „Suche nur Einträge mit BSCW-Berechtigung = WAHR“.
Außerdem ist es möglich, eine BSCW-Instanz für den Zugriff auf verschiedene LDAP-Server zu konfigurieren oder auch die BSCW-Nutzerverwaltung und einen LDAP-Server parallel einzusetzen, sodass die LDAP-Nutzerverwaltung durch BSCW insgesamt sehr flexibel einsetzbar ist.



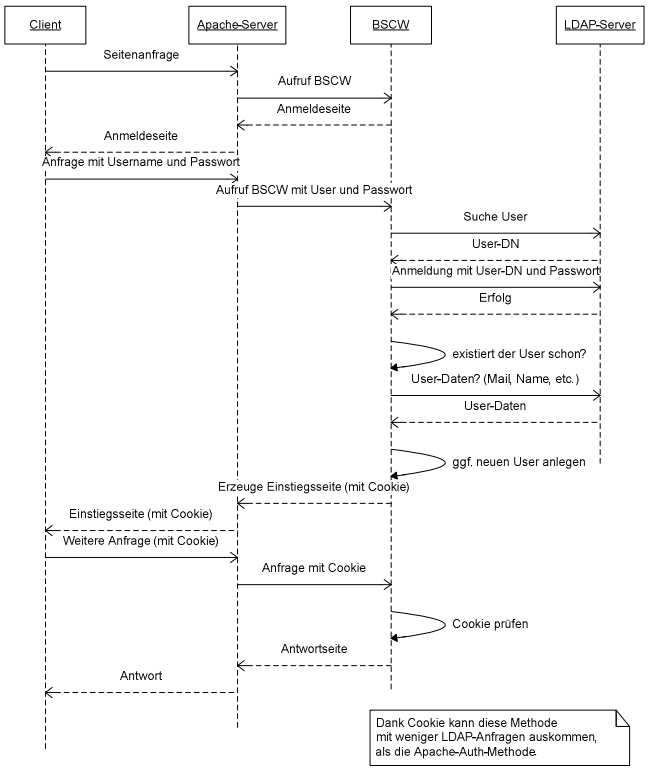

**Information 20: LDAP-Suche mit BSCW**

Nutzer, die sich erfolgreich gegenüber LDAP authentifizieren, können von BSCW automatisch mit Default-Rechten angelegt werden, sofern sie noch nicht im BSCW-Datenbestand existieren. Damit ist die Nutzung von LDAP als führendes System für die Nutzerverwaltung möglich. Reicht die Vergabe von Default-Rollen für alle Einträge jedoch nicht aus, so muss ein Administrator eingreifen, um über eine der anderen Schnittstellen die im Einzelfall erforderlichen Arbeitsbereiche und Rollen zuzuweisen.
Im BSCW-eigenen Datenbestand können auf Wunsch vom Administrator parallel weitere Nutzer eingerichtet werden, die nicht im LDAP-Verzeichnis existieren müssen. In diesem Fall dient ein LDAP-Verzeichnis also nicht als alleinige Datenquelle, allerdings besteht die Gefahr von Konflikten. Im Mischbetrieb wäre es möglich, dass ein Nutzer durch den Administrator oder durch eine Einladung im BSCW-Datenbestand angelegt wird, während ein zweiter, gleichnamiger Account auch im LDAP-Verzeichnis eingetragen existiert. Es ist hier die Aufgabe des Betreibers, durch geeignete Maßnahmen das Auftreten solcher Konflikte zu verhindern.
Zum Zugriff auf den LDAP-Server benötigt BSCW selbst wenig Rechte. Wesentlich sind Such- und Leserechte auf die Attribute für *Common Name*, *Surname*, Email und Nutzername. Weitere im LDAP-Verzeichnis gespeicherte Daten sind auch als Adressbuch nutzbar. Ist diese Nutzung erwünscht, sind hier ebenfalls die

entsprechenden Leserechte vorzusehen. Allerdings werden leider nicht alle Änderungen im LDAP-Datenbestand auch von BSCW übernommen, denn Änderungen einer im LDAP-Verzeichnis hinterlegen E-Mail-Adresse werden ignoriert.
Hintergrund ist die Verwendung der E-Mail-Adresse als Schlüssel innerhalb der BSCW-Nutzerverwaltung (siehe 4.5.2).

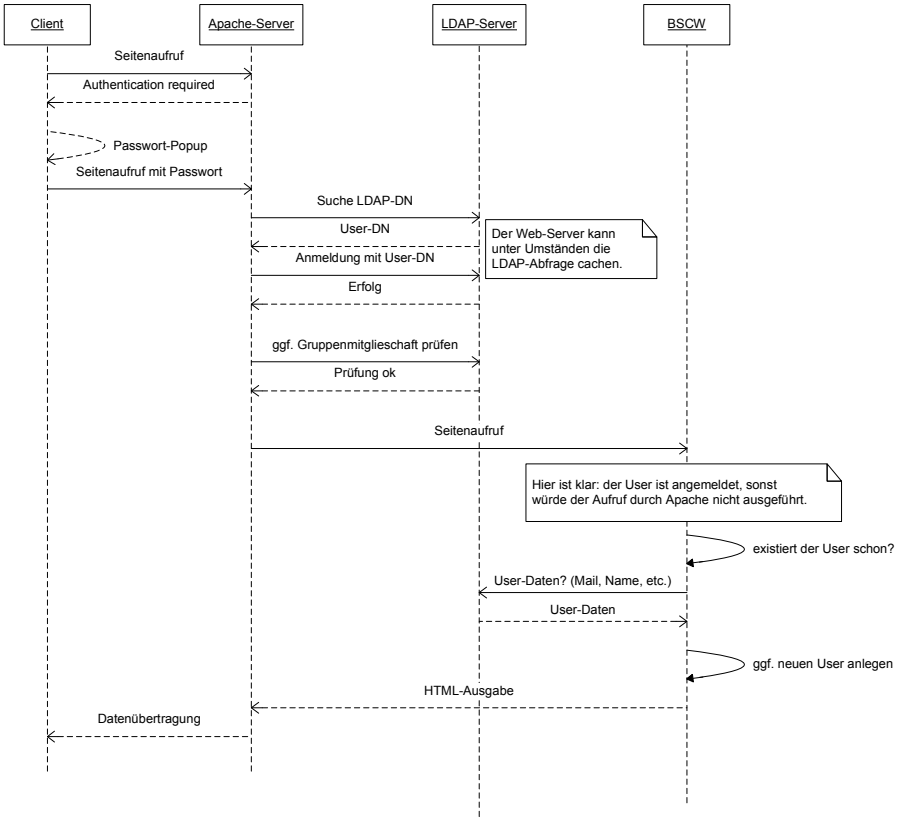
### 4.5.4 BSCW: LDAP-Authentifikation

BSCW bietet mehrere Möglichkeiten zur Einbindung von LDAP zur Nutzerauthentifikation. Zum einen kann die Nutzeranmeldung von BSCW selbst durchgeführt werden, wobei wahlweise die eigene Nutzerverwaltung, ein LDAP-Verzeichnis oder beide Quellen gleichzeitig zur Authentifikation herangezogen werden können.




**Information 21: LDAP-Authentifikation durch BSCW**

Zum anderen lässt sich der Anmeldevorgang statt von BSCW durch den Webserver durchführen, der seinerseits an LDAP angebunden sein kann. Dabei können Geschwindigkeitsvorteile realisiert werden, da die Anmeldung bereits geprüft werden kann, ohne den vergleichsweise langsamen, interpretierten BSCW-Prozess ausführen zu müssen. Die parallele Verwendung der BSCW-Nutzerverwaltung zur Authentifizierung ist in diesem Fall jedoch nicht möglich.




**Information 22: LDAP-Authentifikation durch Webserver**

In beiden Fällen gibt es die Möglichkeit, Nutzer, die nicht in der BSCW-Nutzerverwaltung existieren, aber im LDAP-Server vorgefunden werden, automatisch in der BSCW-Nutzerverwaltung anzulegen. Für Details siehe 4.5.3.

### 4.5.5 Selbstverwaltung durch BSCW

Die bereits angelegten Nutzer haben die Möglichkeit, mit Hilfe von BSCW ihre eigenen Einträge im LDAP-Verzeichnis zu verändern. Es ist beispielsweise möglich, die eigene Telefonnummer oder Adresse im LDAP-Verzeichnis einzutragen bzw. zu verändern, sofern dem Nutzer durch den LDAP-Server die nötigen Rechte eingeräumt werden. Diese Informationen stehen dann allen BSCW-Nutzern durch die eingebaute Adressbuchfunktionalität zur Verfügung.
Leider ist es nicht möglich, dass sich Nutzer gegenüber BSCW derart selbst registrieren, dass sie automatisch in einem LDAP-Verzeichnis angelegt werden. Entsprechende Funktionen sind nur zum Anlegen von Nutzern in der BSCW-eigenen Datenverwaltung vorgesehen, ein LDAP-Server wird prinzipiell als führendes System verwendet, dessen Daten nicht automatisch, sondern allenfalls durch die Selbstverwaltungsfunktionen verändert werden.
Damit sind die BSCW-Funktionen zur Selbstverwaltung der Nutzer nicht geeignet, um im IPO-Szenario die Datenhaltung im LDAP-Server vollständig zu verwalten.

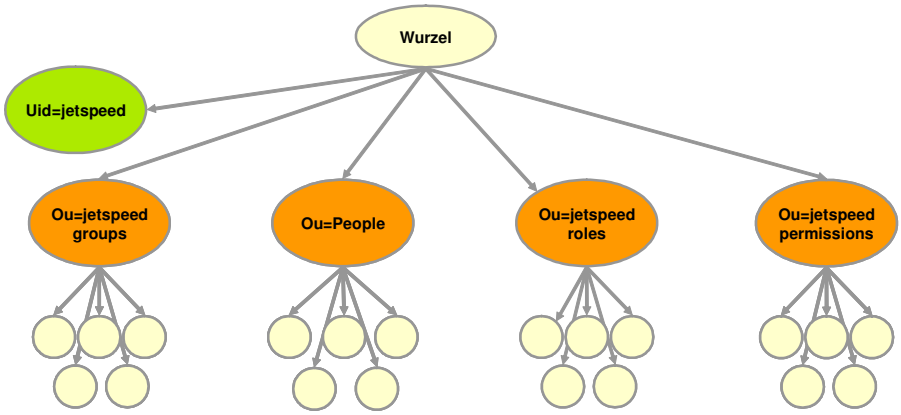
### 4.5.6 Fazit

BSCW ist ein mächtiges Werkzeug zur Materialverwaltung, das im IPO-Kontext eingesetzt werden soll. Da ein Zugriff auf die eigene Datenhaltung von weiteren Systemen nicht direkt möglich ist, kann das Problem der Nutzerverwaltung nur über die

XML-RPC-Schnittstelle oder durch die Einführung eines zusätzlichen LDAP-Servers, der als führendes System zur Nutzerverwaltung fungieren würde, gelöst werden.
BSCW bietet eine Standardkonforme LDAP-Schnittstelle, für deren Einsatz minimale Zugriffsrechte ausreichen. Außerdem lässt sich BSCW sehr einfach und ohne Änderungen am Quellcode an unterschiedlichste LDAP-Strukturen anpassen. Insofern erscheint es im Regelfall unnötig, direkte Zugriffe auf die BSCW-Nutzerverwaltung durchzuführen.

## 4.6 Erster Entwurf der LDAP-Struktur bzgl. IPO

Nachdem wir gesehen haben, dass Jetspeed eine sehr starre LDAP-Struktur voraussetzt während BSCW deutlich flexiblere Möglichkeiten zur Anpassung an unterschiedliche Strukturen bietet, soll hier eine gemeinsam nutzbare Struktur entwickelt werden.
Da BSCW lediglich Nutzer im LDAP verwaltet, sind Anpassungen der Rollen-, Gruppen- und Rechteverwaltung von Jetspeed in Bezug auf BSCW nicht erforderlich. Es werden daher an dieser Stelle lediglich Fragestellungen bzgl. der Nutzerverwaltung betrachtet.
Da die Jetspeed-LDAP-Struktur recht starr erscheint, soll zuerst versucht werden, die BSCW-LDAP-Schnittstelle an diese gegebene Struktur anzupassen. Als Ausgangspunkt dient dazu die bereits verbesserte Jetspeed-LDAP-Struktur.




**Information 23: Jetspeed-LDAP: Baum-Struktur (verbessert)**

### 4.6.1 Attribute für Personen

Sämtliche Attribute, die von einem der beiden Dienste BSCW und Jetspeed benötigt werden, müssen im LDAP-Verzeichnis vorliegen. Zusätzlich kann es erforderlich werden, weitere Verwaltungsinformationen zu erfassen. Ziel ist die Definition eines LDAP-Eintrags, der alle erforderlichen Attribute enthält und von allen beteiligten Diensten genutzt werden kann.
Zuerst sollen die technisch für BSCW und Jetspeed erforderlichen Attribute betrachtet werden, die in Information 24 gegenübergestellt sind.

| BSCW | JETSPEED |
|---|---|
| Surname, Common Name | Surname, givenName |
| Uid | Uid |
| userPassword | userPassword |
| Mail | Mail |
| | Disabled, confirmed |
| | Uidnumber |
| | Lastlogindate, lastmodifieddate, creationdate |
| | objectdata |
| | Usergrouprole |

**Information 24: Kombination BSCW und JETSPEED**

Es kann zuerst festgestellt werden, das, mit Ausnahme von „Common Name“, alle von BSCW benötigten Attribute auch von Jetspeed verwendet werden. Dank LDAP-Standard wird eine gemeinsame Semantik der gleich benannten Attribute garantiert (siehe RFC 2252).
Während Jetspeed die Verwendung der Objektklasse „*JetspeedUser*“ benötigt, erfordert BSCW die Verwendung der Objektklasse „*Person*“. Diese Objektklasse verpflichtet zur Angabe der Attribute „*Common Name*“ und „*Surname*“ im LDAP-Eintrag. Da „*Surname*“ bereits durch Jetspeed vorhanden ist, muss nur noch geklärt werden, wie der „*Common Name*“ erfasst werden kann.
Der „*Common Name*“ umfasst nach LDAP-Standard (siehe RFC 2256) den vollständigen Namen einer Person, im amerikanischen häufig „Vorname Mittelinitial Nachname“. Im Rahmen des IPO kann aber die Vereinfachung verwendet werden, dass dem „*Common Name*“ in Europa in der Regel „Vorname Nachname“ entspricht und diese beiden Attribute werden im Jetspeed-Schema bereits durch „*givenName*“ und „*Surname*“ erfasst. Damit lässt sich das von BSCW benötigte „*Common Name*“-Attribut leicht aus den von Jetspeed ohnehin erfassten Attributen generieren.
Damit ist es also möglich, aus den ohnehin für Jetspeed erfassten Attributen automatisch die nötigen Angaben für BSCW zu erzeugen. Ob dies durch eine Jetspeed-Erweiterung (siehe 4.4.4) oder durch eine eigenständige Verwaltungskomponente erfolgt, wird später diskutiert (siehe 4.7).

### 4.6.2 LDAP-Struktur

Die Baumstruktur des LDAP-Verzeichnisses wird hier maßgeblich durch die eingeschränkten Möglichkeiten von Jetspeed beeinflusst. Da BSCW sehr flexibel konfiguriert werden kann, ist es ohne weitere Änderungen möglich, auch die durch Jetspeed vorgegebene Struktur durch geeignete Konfiguration zu verwenden.

### 4.6.3 Unterschiedliche Rollen in BSCW

Es ist vorgesehen, dass die Nutzer gegenüber dem IPO in unterschiedlichen Rollen auftreten. Da jedoch BSCW für neue Nutzer nur Default-Rollen vergeben kann, entsteht hier ein Problem, das nicht allein durch LDAP gelöst werden kann. Es wird daher erforderlich, für jeden Nutzer bei der Registrierung eine Rollenvergabe durchzuführen, die auf der BSCW-Nutzerverwaltung operiert. Dazu bietet sich der Einsatz der XML-RPC-Schnittstelle an.

### 4.6.4 Fazit

Eine Lösung für das Problem der unterschiedlichen Verwendung von Attributen und Objektklassen konnte bereits aufgezeigt werden (siehe 4.6.1). Auch die von Jetspeed verwendete Strukturierung des LDAP-Verzeichnisses stellt aufgrund der Flexibilität von BSCW derzeit kein Problem dar.
Insgesamt gesehen wird so eine gemeinsame LDAP-Nutzung von BSCW und Jetspeed einfach zu realisieren sein. Es ist lediglich ein geringer Programmieraufwand erforderlich, um das von BSCW benötigte Attribut *Common Name* zu erfassen und die Objektklasse „*Person*“ parallel zu „*JetspeedUser*“ verwenden zu können.

- Die Jetspeed-LDAP-Struktur kann übernommen weren
- Die Einträge müssen jeweils leicht erweitert werden
- Sowohl BSCW als auch Jetspeed geben dem User die Möglichkeit, Daten wie Name oder Passwort zu verändern
  - Beide Möglichkeiten müssten an die aus der Verschmelzung entstehenden, zusätzlichen Anforderungen angepasst werden
  - Alternativ: zusätzliches Modul einführen
- Registrierung der User
  - Derzeit nur mit Jetspeed möglich, BSCW bietet keine Mechanismen
  - Jetspeed müsste erweitert werden
  - Alternativ: zusätzliches Modul zur Registrierung einfügen

**Information 25: Fazit aus Kombination**

Leider zeigt sich an dieser Stelle auch, dass nicht alle Probleme durch den Einsatz eines Verzeichnisdienstes gelöst werden können, da etwa BSCW keine Rolleninformationen von externen Quellen abrufen kann. Daher sind zur Rollenvergabe im BSCW-System Szenario weitere Maßnahmen des Betreibers erforderlich, die im Rahmen der Betriebs- und Verwaltungsprozesse durchgeführt werden müssen.
Außerdem zu beachten, dass sowohl BSCW als auch Jetspeed Funktionen bieten, mit denen die Nutzer eigene Attribute im LDAP-Verzeichnis verändern können. Ein Parallelbetrieb wäre nicht sinnvoll, da dies die Anwender verwirren könnte und da die beiden Anwendungen jeweils nur auf einer Untermenge der insgesamt verfügbaren Attribute operieren. Das Problem der Selbst-Verwaltung wird deshalb im Folgenden separat betrachtet.

## 4.7 Realisierung der Selbstverwaltung und -registrierung

Die Notwendigkeit einer Selbstregistrierung und Selbstverwaltung der Nutzer des IPO resultiert allein aus der hohen Nutzerzahl, die manuell nicht bewältigt werden kann (siehe 4). Durch die Selbstverwaltung der IPO-Nutzer müssen sowohl die technischen Anforderungen (resultierend aus den verwendeten Komponenten Jetspeed und BSCW), als auch Anforderungen aus dem Verwaltungsbereich erfüllt werden. Die technischen Anforderungen der eingesetzten Software wurden bereits in 4.4, 4.5 und 4.6 untersucht, sodass hier auf eine Wiederholung verzichtet werden kann.

### 4.7.1 Attribute zur Unterstützung des Betriebs

Neben den technischen Erfordernissen gibt es weitere Daten, die aus Betreibersicht zur Verwaltung des IPO erforderlich sind. Beispielsweise soll zu jedem Nutzer seine

Fricardnummer und die Zugehörigkeit zu den Gruppen „extern“, „Universitätsmitglied“ und „Student“ gespeichert werden können.
Um diese Informationen im LDAP-Verzeichnis speichern zu können, müssen zwei zusätzliche Attribute eingeführt werden. Denkbar wären Attributnamen wie beispielsweise „*fricardNummer*“ oder „*rolle*“. Da auch für andere Anwendungen Attribute mit ähnlichen Namen definiert sein können, sollte hier zur Vermeidung von Konflikten ein eindeutiges Präfix wie „*cm-tm*“ oder „*ipo*“ verwenden, also etwa „*ipoRole*“ statt „*rolle*“ und „*ipoFricardNumber*“ statt „*fricardNummer*“.
Da diese neuen Attribute sonst keine Verwendung finden, sind keine Konflikte zu befürchten und die Einführung ist unkritisch. Es muss allerdings noch geklärt werden, an welcher Stelle diese Attribute innerhalb des Registrierungsprozesses erfasst und verwaltet werden sollen.
Weitere aus Sicht der Verwaltung interessante Informationen sind die Zeitstempel, zu denen ein Nutzer angelegt oder verändert wurde, sowie die Information, unter welcher Nutzerkennung das Anlegen bzw. die letzte Datenänderung durchgeführt wurde. Da diese Informationen jedoch automatisch vom LDAP-Server gepflegt werden können, ist eine weitere Betrachtung an dieser Stelle nicht erforderlich.

### 4.7.2 Umsetzung der Verwaltungswerkzeuge

Die Nutzer des IPO sollen sich selbst über eine Weboberfläche am System registrieren und später nötige Verwaltungsinformationen selbstständig aktualisieren können. Beispielsweise soll ein Nutzer ohne administrative Eingriffe ein neues Passwort erhalten können.

- Überlegungen
  - Die Funktionen zur Selbstverwaltung sind aktuell weder bei Jetspeed noch bei BSCW ausreichend und müssten angepasst werden
  - Eine Auslagerung dieser Funktionalität in ein separates Modul wäre eine Alternative
- Vorteile der Auslagerung
  - Erhöhte Modularität
  - Unabhängigkeit von konkreten Einsatzszenario (IPO)
  - Bessere Wiederverwendbarkeit
  - Spätere Erweiterungen der Selbstverwaltung können einfach und zentral durchgeführt werden

**Information 26: Modul zur Selbstverwaltung**

Jetspeed stellt zur Registrierung einen Mechanismus bereit, der nicht ohne Anpassungen genutzt werden könnte (siehe 4.6.1). Von Jetspeed werden bei der Selbstregistrierung der Nutzer lediglich Vorname, Nachname, E-Mail-Adresse und das gewünschte Passwort erfasst. Der Aufwand, um zusätzliche Attribute abzufragen und zu verwalten, wäre, unter anderem aufgrund der Schwächen der LDAP-Unterstützung (siehe 4.4.3), vermutlich deutlich höher. Außerdem werden Passwörter derzeit nicht befriedigend gespeichert, weshalb diese Funktion verbessert werden müsste.
Zur Verwaltung der eigenen Daten stellen sowohl Jetspeed als auch BSCW Funktionen zur Verfügung, die jedoch einen unterschiedlichen Umfang haben. So kann BSCW neben den zur Nutzung unbedingt erforderlichen Daten auch Attribute verwalten, die von der Adressbuchfunktion genutzt werden. Jetspeed hingegen kann nur die unbedingt erforderlichen Nutzerdaten wie Vorname, Nachname und Passwort verändern.

Allerdings bietet Jetspeed Hilfe, wenn ein Nutzer sein Passwort vergessen hat, was bei BSCW nicht vorgesehen ist. Aus Gründen der Benutzerführung sollte jedoch nur eine der beiden Schnittstellen zur Selbstverwaltung aktiv sein, da eine Kombination die Nutzer verwirren könnte und so zusätzlichen Support-Aufwand seitens des Betreibers nach sich ziehen würde.
Daher müssen wir leider feststellen, dass die existierenden Verwaltungswerkzeuge den im IPO-Szenario gestellten Anforderungen nicht gerecht werden können und deshalb eine alternative Lösung gefunden werden muss.
Es ist zu überlegen, ob eine zusätzliche Komponente zur Nutzerverwaltung entwickelt werden kann. Diese Komponente sollte es dem Nutzer ermöglichen, sich selbst am IPO zu registrieren und später ein neues Passwort anfordern bzw. sein altes Passwort verändern zu können.
Bei der Registrierung sollen Fricardnummer und E-Mail-Adresse abgefragt und die E-Mail-Adresse über eine Test-E-Mail verifiziert werden. Erst nach erfolgreicher Verifikation über eine spezielle URL wird der Account aktiviert und anhand der Fricardnummer wird zwischen Universitätsangehörigen und externen Nutzern unterschieden. Die einzelnen Aktionen sollen außerdem automatisch protokolliert werden, um eine Nachverfolgung und die Analyse im Fehlerfall zu unterstützen und so auch in Ausnahmesituationen eine hohe Qualität zu gewährleisten.

- Nutzer ruft Registrierungsfunktion auf
- Es werden folgende Daten abgefragt
  - Vorname, Nachname
  - Fricardnummer, falls vorhanden
  - E-Mail-Adresse
  - Passwort (zweimal, um Tipp-Fehler zu vermeiden)
- Das Registrierungsmodul generiert einen Login und prüft, dass dieser Login im LDAP-Verzeichnis und in der BSCW-Nutzerverwaltung noch nicht vergeben ist
- Ein LDAP-Eintrag wird erzeugt
- Das Registrierungsmodul sendet eine E-Mail mit dem generierten Login an die angegebene Adresse
- Der Nutzer ruf eine in der E-Mail angegebene URL auf und verifiziert damit die Gültigkeit der E-Mail-Adresse
- Das Registrierungsmodul aktiviert den Account.

**Information 27: Ablauf der Selbstregistrierung**

Leider unterstützt Jetspeed nicht den JSR-168-Standard für Portlets, sodass eine Verwaltungsportlet nicht zwischen verschiedenen Portlet-Containern portabel wäre. Außerdem könne ein solches Portlet außerhalb eines Portlet-Containers nicht genutzt werden. Als Alternative bietet sich daher die Entwicklung einer eigenständigen Webanwendung an, deren Aufgabe die Registrierung von Nutzern sowie deren Selbst-Verwaltung wäre.

### 4.7.3 Fazit

Insgesamt erscheint es einfacher, eine neue Registrierungsanwendung zu entwickeln, als die bestehende Jetspeed-Funktionalität zu nutzen. Gründe sind insbesondere die Verwaltung zusätzlicher Attribute sowie die angesprochenen Schwächen der Jetspeed-LDAP-Unterstützung. In diese neue Registrierungsanwendung lassen sich auch

Selbstverwaltungsfunktionen integrieren, wie beispielsweise das Ändern des Passworts oder Hilfestellungen, falls ein Nutzer sein Passwort vergessen hat.
Die Realisierung dieser Anwendung soll als eigenständige Webanwendung erfolgen, um eine Abhängigkeit vom IPO-Kontext zu vermeiden und die Wiederverwendbarkeit zu verbessern. Eine genauere Analyse der Registrierungskomponente findet sich in [Ha04].

## 4.8 Zugriffsrechte

LDAP selbst biete eine sehr feingranulare Steuerung von Zugriffsrechten. Ziel ist die Minimierung der von jeder Anwendung benötigten Zugriffsrechte, um die Sicherheit des Gesamtsystems zu erhöhen.

| Attribut | BSCW | JETSPEED |
|---|---|---|
| Uid (=Login) | READ | READ |
| CN | READ | |
| SN | READ | READ |
| givenName | | READ |
| userPassword | AUTH | READ |
| Mail | READ | READ |
| objectClass | READ | READ |
| Disabled, confirmed,Lastlogindate, lastmodifieddate, createiondate, usergrouprole | | WRITE |
| objectData | | WRITE |

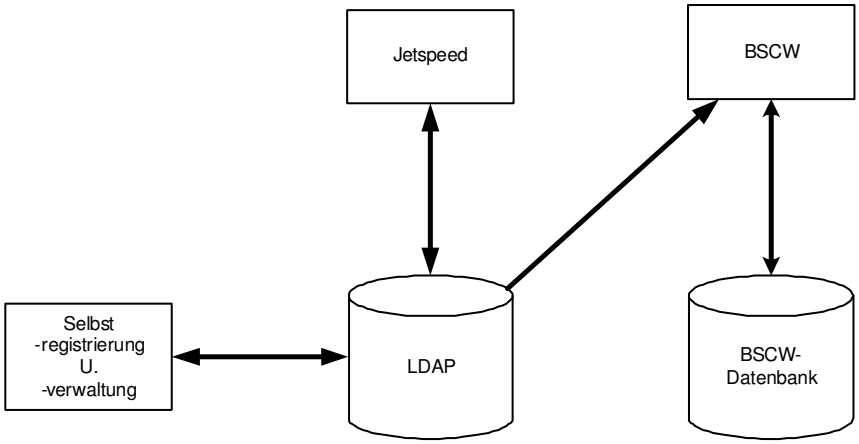
Voraussetzung: Keine Verwaltung der Userdaten durch Jetspeed

**Information 28: Zugriffsrechte auf Einträge**

Information 28 zeigt, welche Zugriffsrechte im IPO-Kontext von den einzelnen Anwendungen minimal benötigt werden, wenn eine zusätzliche Komponente zur Nutzerverwaltung eingesetzt wird. Dabei sind überwiegend Leserechte erforderlich, nur für wenige spezifische Attribute benötigt Jetspeed Schreibrechte, beispielsweise für die eigene Rechteverwaltung in „usergrouprole“. Tatsächlich erfordert Jetspeed Schreibrechte für alle von Jetspeed genutzten Attribute (siehe 4.4.3). Nachdem jedoch die Entscheidung gefallen ist, eine eigenständige Nutzerverwaltung zu entwickeln können diese Rechte durch Änderungen am Jetspeed-Quellcode anpasst werden. Die eigenständige Nutzerverwaltung wird Schreibrechte für jedes LDAP-Attribut benötigen, da die Einträge sonst nicht angelegt werden könnten.
Im IPO-Kontext sollten zusätzliche Proxy-Nutzer für BSCW und Jetspeed definiert werden. Diese Nutzer werden von den Anwendungen zum Zugriff auf das Verzeichnis verwendet und mit den konkreten Rechten versehen, mit denen die Anwendung auf das Verzeichnis zugreifen können soll (siehe 3.4.4).

## 4.9 Datenbestände

Es sei ausdrücklich darauf hingewiesen, dass LDAP nicht als alleiniger Datenbestand zu verstehen ist. Zwar speichert Jetspeed die benötigten Daten vollständig im LDAP-Verzeichnis, doch pflegt BSCW parallel dazu eine eigene Datenverwaltung.
Die beteiligten Anwendungen, die Datenbestände und die Beziehungen untereinander sind in Information 29 zusammengefasst.




**Information 29: Datenbestände**

Es sei daran erinnert, dass von BSCW eine eigenständige Nutzerverwaltung parallel zur LDAP-Nutzerverwaltung durchgeführt wird, sodass Synchronisationsprobleme auftreten können. Diese können gelöst werden, indem auf eine parallele Nutzerverwaltung verzichtet wird, oder indem geeignete Betriebsprozesse eingeführt werden, um Konflikte aufzulösen.

## 4.10 Ausblick

Das INFORMATIK-I-Portal bildet derzeit eine stabile Basis, bei der durch den Einsatz eines LDAP-Verzeichnisdienstes eine effektive, gemeinsame Verwaltung der erforderlichen Nutzerdaten realisiert werden konnte.
Dabei wurde, beispielsweise durch Verwendung der Standard-Objektklasse „*Person*“, darauf geachtet, dass auch künftige Erweiterungen ein standardkonformes Verzeichnis vorfinden und nutzen können. Insbesondere zur Authentifikation gegenüber weiteren Diensten stehen bereits alle erforderlichen Informationen zur Verfügung, sodass Anwendungen, die keine personalisierte Datenspeicherung benötigen (etwa ein geschützter Bereich einer Homepage) oder die wie BSCW eine eigene Datenspeicherung betreiben und ein LDAP-Verzeichnis nur zu Authentifikation und einem initialen Anlegen von Nutzern verwenden, ohne weitere Probleme integriert werden können.
Lediglich bei Anwendungen, die eigene Informationen in einem LDAP-Verzeichnis verwalten, ist mit einem Integrationsaufwand zu rechnen. Anhand der aktuellen Struktur können Forderungen an solche Anwendungen gestellt werden, die erfüllt sein müssen, um eine einfache Integration zu ermöglichen.
Die Verwendung standarisierter Objektklassen und Attribute ist soweit möglich zu bevorzugen, die Einführung eigener Attribute ist zu minimieren.
Für zusätzlich geforderte Attribute soll jede Anwendung eine Schemadefinition zur Verfügung stellen, in der eine geeignete Objektklasse und die nötigen Attribute definieren sind, sodass das gegenwärtige LDAP-Schema einfach erweitert werden kann. Da gemäß LDAP-Spezifikation jeder Eintrag über eine strukturelle Objektklasse verfügen muss und keine weiteren strukturellen Objektklassen zulässig sind (siehe 3.4.2), ist eine Erweiterung um eine strukturelle Objektklasse nicht möglich. Daher müssen künftige Objektklassen als Hilfsklassen definiert sein. Anders als bei Jetspeed sollten zukünftige Erweiterungen keine speziellen Anforderungen an die Verzeichnisstruktur stellen und über konfigurierbare Suchfilter an beliebige Verzeichnisse anpassbar sein. Auch die Verwendung von LDAP-Attributen sollte konfigurierbar sein, sodass der Betreiber beispielsweise festlegen kann, ob die

Nutzeranmeldung über ein spezielles Login-Attribut (z. B. „*uid*“) erfolgen soll oder ob ein anderes eindeutiges Attribut wie eine Matrikelnummer („*ipoFricardNumber*“) oder die E-Mail-Adresse („*mail*“) diese Funktion übernehmen soll.

# 5 DIENSTORIENTIERTE NUTZERVERWALTUNG

## 5.1 Erkenntnis aus IPO

Im vorhergehenden Kapitel wurde ausführlich analysiert, wie ein LDAP-Verzeichnisdienst innerhalb eines Webportals genutzt werden kann, um ein *Identity Management* für die beteiligten Komponenten zu realisieren.
Bei den betrachteten Komponenten handelte es sich um Jetspeed, eine Portalsoftware der Apache Software Foundation und um BSCW, eine Kollaborationsumgebung. Auch weitere Anforderungen, die durch künftige Erweiterungen zu erwarten sind, wurden bereits analysiert.
Im Ergebnis konnte festgestellt werden, dass zwar bei Jetspeed kleinere Anpassungen erforderlich waren, dass aber LDAP-konforme und flexibel zu konfigurierende Anwendungen wie BSCW ohne großen Aufwand in ein gemeinsames LDAP-Verzeichnis eingebunden werden können.
Wenn man die resultierenden Daten und Strukturen genauer betrachtet, so kann man die LDAP-Attribute nach verschiedenen Kriterien klassifizieren. Ein Teil der Attribute wird dienstübergreifend zur Identifikation und Verwaltung der Nutzer genutzt und ist daher von keinem speziellen Dienst abhängig. Dazu gehören beispielsweise der Name des Nutzers, Abteilungszugehörigkeiten, Kontaktdaten sowie weitere allgemeine Attribute. Diese Attribute sind in der Regel nicht nur durch technische Aspekte, sondern auch durch Verwaltungsanforderungen seitens des Betreibers motiviert.
Ein anderer Teil der Informationen kann einem spezifischen Dienst zugeordnet werden und wäre ohne diesen konkreten Dienst nicht erforderlich. Beim Dienst „Jetspeed“ wären beispielsweise „*usergrouprole*“ oder *„creationdate*“ derartige Attribute. Anders als bei den Attributen zur Nutzerverwaltung liegen hier also überwiegend technische Gründe vor, die diese Attribute erfordern.
Man kann daher zwischen einer nutzerzentrierten Identitätsverwaltung und einer dienstspezifischen Informationsverwaltung unterscheiden. Diese können jedoch nicht vollständig getrennt betrachtet werden, da die einzelnen Dienste von den Nutzern verwendet werden sollen und daher von ihnen abhängig sind. Deshalb werden beide Aspekte im Folgenden näher untersucht.

## 5.2 Verwaltung der Nutzer

Im Fokus des *Identity Management* steht die effiziente Verwaltung existierender Nutzer. Die konkreten Anforderungen werden dabei durch den Betreiber spezifiziert und sind nur selten technischer Natur.
Dabei muss ein Nutzer zunächst im System eindeutig identifiziert werden. Dazu ist ein spezieller Login erforderlich, der mit einem realen Namen und weiteren, den Nutzer eindeutig identifizierenden, Attributen verknüpft sein muss. Denkbar wären Mitarbeiternummern, Matrikelnummern oder ähnliche Surrogate. Kontaktdaten wie beispielsweise E-Mail-Adresse oder Telefonnummer stellen weitere sinnvolle, nutzerbezogene Informationen dar. Auch zusätzliche Verwaltungsinformationen wie beispielsweise die Abteilungs- oder Rollenzugehörigkeiten (Direktor, Assistent, Hilfskraft, etc.) können je nach individuellen Anforderungen nötig sein.
Da der Nutzereintrag später zur Nutzung verschiedener Dienste verwendet werden soll, sind grundlegende Informationen zur Authentifikation erforderlich, wozu in der Regel ein Login und ein Passwort verwendet werden. Diese Informationen gehören zwar nicht mehr zu den Attributen, die im engeren Sinn zur Nutzerverwaltung benötigt werden,

doch da sie dienstübergreifend benötigt werden, ist es sinnvoll, sie dennoch den zu einer effizienten Nutzerverwaltung erforderlichen Attributen zuzurechnen.

## 5.3 Beispiel: Nutzer im IPO

Im IPO-Kontext wird jeder Nutzer innerhalb des Datenbestands durch einen eindeutigen, systemgenerierten Login identifiziert. Zusätzlich wird der Name, nach Möglichkeit zusammen mit einer Fricardnummer, als identifizierendes Attribut gespeichert, sowie die Rolle des Nutzers innerhalb des IPO zugewiesen, die später bei der Rechtevergabe für zusätzliche Dienste verwendet werden soll.
Schließlich ist noch eine E-Mail-Adresse erforderlich, um bei Problemen eine Kontaktaufnahme zu ermöglichen. Die Information, ob die angegebene E-Mail-Adresse verifiziert werden konnte, kann sowohl als Bestandteil der Nutzerdaten als auch als Bestandteil eines eigenständigen Dienstes zur Nutzerverifikation gesehen werden. Da diese Verifikation jedoch als zentrale Voraussetzung für die Nutzung von Diensten wie Diskussionsforen oder Materialzugriff genutzt werden soll, erscheint es sinnvoll, sie als Bestandteil der Nutzerdaten einzuordnen, sodass sämtliche Dienste lediglich zentral von den Nutzerdaten abhängen sind und keine Abhängigkeiten zwischen verschiedenen Diensten entstehen.

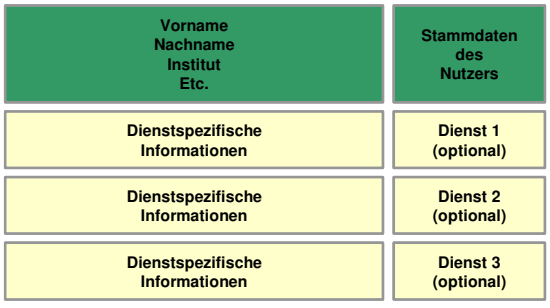
## 5.4 Verwaltung der Dienste

Um den Datenbestand zur Nutzerverwaltung möglichst universell einsetzen zu können und gleichzeitig den Erfordernissen dienstspezifischer Erweiterungen gerecht zu werden, benötigen wir eine dynamische Erweiterbarkeit der verwalteten Nutzerdaten. Der technische Rahmen für die Umsetzung wird durch die speziellen Eigenschaften eines LDAP-Verzeichnisses gegeben.
Die bereits zur Nutzerverwaltung erfassten Daten bilden unsere Basis, an die wir dienstspezifischen Erweiterungen anbinden wollen. Die benötigen Nutzerattribute wurden bereits von uns betrachtet (siehe 5.2).
Für jeden zu unterstützenden Dienst können wir die jeweils benötigten, dienstspezifischen Attribute identifizieren, sodass wir eine Abbildung vom betrachteten Dienst auf die erforderlichen Attribute erhalten. Diese Attributmenge kann nun von uns mit dem Dienst identifiziert und als LDAP-Objektklasse spezifiziert werden. Schließlich erhalten wir so für jeden Dienst eine spezifische LDAP-Objektklasse, die die Dienstnutzung ermöglicht. Es ist dann Aufgabe des Betreibers, den Eintrag eines einzelnen Nutzers um die nötige Dienst-Objektklasse und die erforderlichen Attribute zu erweitern, um dem Nutzer den Zugriff auf den Dienst zu ermöglichen. Der Betreiber kann dem einzelnen Nutzer die Zugriffsberechtigung auch wieder entziehen, indem die durchgeführten, dienstspezifischen Erweiterungen wieder vom Nutzereintrag entfernt werden.
Die spezielle Erweiterbarkeit existierender Einträge um zusätzliche Objektklassen ist ein zentrales Element in diesem Vorgehen und nur beim Einsatz eines LDAP-Verzeichnisdienstes möglich. Beim Einsatz von Datenbanken wäre dieser Ansatz nur mit wesentlich höherem Aufwand realisierbar.

## 5.5 Abstraktes Vorgehen zur Dienstunterstützung

Wird ein neuer Dienst implementiert, der an eine zentrale LDAP-Nutzerverwaltung angebunden werden soll, so werden unabhängig von den konkreten Anforderungen zuerst eine neue Objektklasse und ein zusätzlicher *Proxy-User* (in der Literatur auch „*Service DN*“ genannt, siehe 3.4.4) für den neuen Dienst benötigt. Die Verwendung existierender Objektklassen für den jeweiligen Dienst ist vorzuziehen, ist dies nicht

möglich, so muss eine eigene Objektklasse definiert werden. Die Namen der Objektklasse und des Proxy-Eintrags sollen so gewählt werden, dass sie den zu implementierenden Dienst eindeutig kennzeichnen.
Im nächsten Schritt werden die Attribute identifiziert, die vom Dienst genutzt werden müssen oder genutzt werden können. Diese Attribute werden entsprechend nach „verpflichtend“ und „möglich“ klassifiziert und in der Dienst-Objektklasse definiert. Dabei müssen auch die Attribute, die bereits aufgrund der allgemeinen Nutzerverwaltung oder aufgrund anderer Dienste spezifiziert sind, explizit in der Dienst-Objektklasse definiert werden, damit mögliche Seiteneffekte vermieden und Abhängigkeiten zwischen verschiedenen Diensten minimiert werden. Als Attribute sind dabei möglichst jene Attribute zu verwenden, die ohnehin im LDAP-Schema spezifiziert sind. Falls neue, dienstspezifische Attribute erforderlich sind, so sind diese hinsichtlich Attributtyp und Vergleichsregeln möglichst präzise zu definieren. Dienstspezifische Attribute sind außerdem durch ein dienstidentifizierendes Präfix zu kennzeichnen, etwa „*dialinTelefonnummer*“ für eine vom Dial-In-Dienst genutzte Telefonnummer.
Schließlich werden dem Proxy-Eintrag für den konkreten Dienst die erforderlichen Rechte zugewiesen, wobei der Grundsatz der Rechteminimierung zu befolgen ist. In der Regel wird ein spezieller Dienst nur auf einen kleinen Teil der im LDAP-Verzeichnis vorhandenen Attribute Leserechte benötigen.
Um den Dienst einrichten und verwalten zu können, ist außerdem ein weiterer LDAP-Nutzer erforderlich, der zu den erforderlichen Modifikationen berechtigt ist. Aus Gründen der Sicherheit sollte dieser LDAP-Nutzer jedoch von dem Nutzer getrennt werden, über den der Dienst Leserechte auf das Verzeichnis erhält.
Es ist zu beachten, dass es für jedes Attribut nur genau ein führendes System geben sollte, um Verwaltungskonflikte zu vermeiden. Die Frage, wie genau Attribute verwaltet werden sollen, die von mehreren Diensten verwendet werden, etwa ein Nutzer-Zertifikat nach X.509-Standard, ist nicht technisch zu klären sondern eine betriebliche Entscheidung. Diese Entscheidung mag zwar im Einzelfall willkürlich erscheinen, doch kann es zu unerwarteten Nebeneffekten kommen, wenn ein von mehreren Systemen verwendetes Attribut unkoordiniert verändert wird, sodass hier klare Geschäftsabläufe und Verantwortlichkeiten für einzelne Attribute definiert werden müssen.
Um eine Selbst-Verwaltung durch den Nutzer zu ermöglichen, ist zu prüfen, welche Attribute der Nutzer selber pflegen kann oder pflegen soll, um den Dienst effizient nutzen zu können. Dazu sind dem Nutzer die geeigneten Zugriffsrechte einzuräumen.
Der Dienst ist nun so zu konfigurieren, dass er mit dem Zugang des Proxy-Eintrags auf den Verzeichnisdienst zugreift und bei Suchanfragen durch den Einsatz von Filtern nur Einträge berücksichtigt, die die zugehörige Objektklasse verwenden.

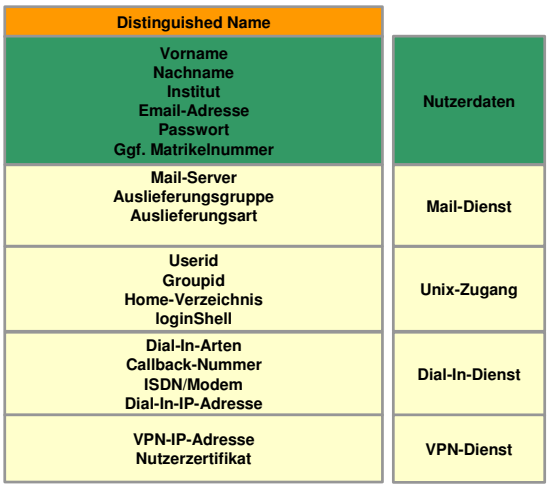



**Information 30: Dienstorientierter Ansatz**

# 6 IDENTITY MANAGEMENT AN DER FAKULTÄT

## 6.1 Überblick

An der Fakultät für Informatik werden verschiedene Dienste für die Mitarbeiter und Studierenden bereitgestellt. Dazu gehören seitens der Abteilung Technische Infrastruktur (kurz ATIS) der zentrale Maildienst, die zentrale Nutzerverwaltung, Rechnerzugang für die Studierenden („Studentenpool“), sowie Dial-In-Zugang und VPN-Zugang für Angehörige der Fakultät. Seitens der Geschäftsführung und einzelner Institute werden weitere Systeme zur Studienunterstützung betreut, die ebenfalls eine Nutzerverwaltung benötigen, beispielsweise das bereits diskutierte IPO zur Vorlesungsunterstützung oder Webinscribe zur Vergabe der Tutorien. Exemplarisch werden im Folgenden die eng zusammen gehörenden Dienste betrachtet, die von der ATIS betrieben werden. Dabei geht das Verwaltungskonzept von einem einzelnen Nutzer aus, dem durch den Betreiber der Zugriff auf die einzelnen Dienste ermöglicht werden soll.



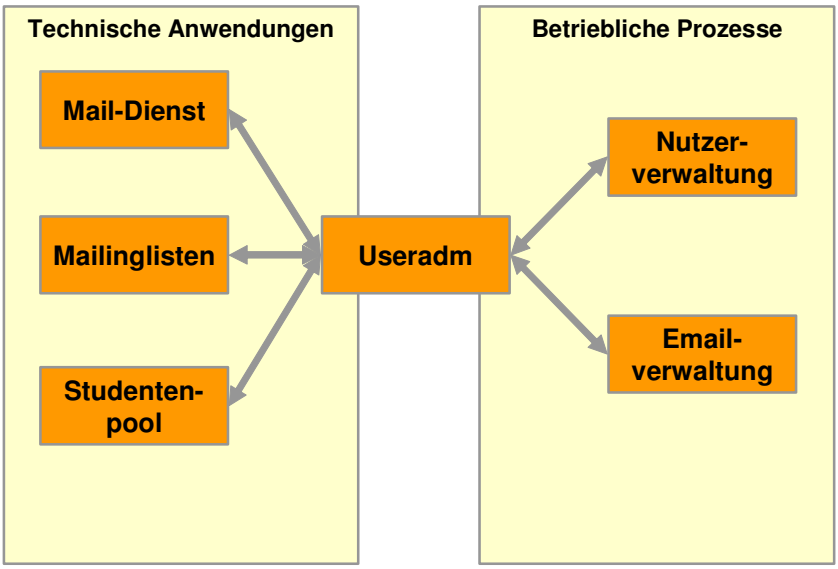

**Information 31: Aggregation von Diensten**

## 6.2 Nutzerverwaltung durch Useradm

### 6.2.1 Aufgaben

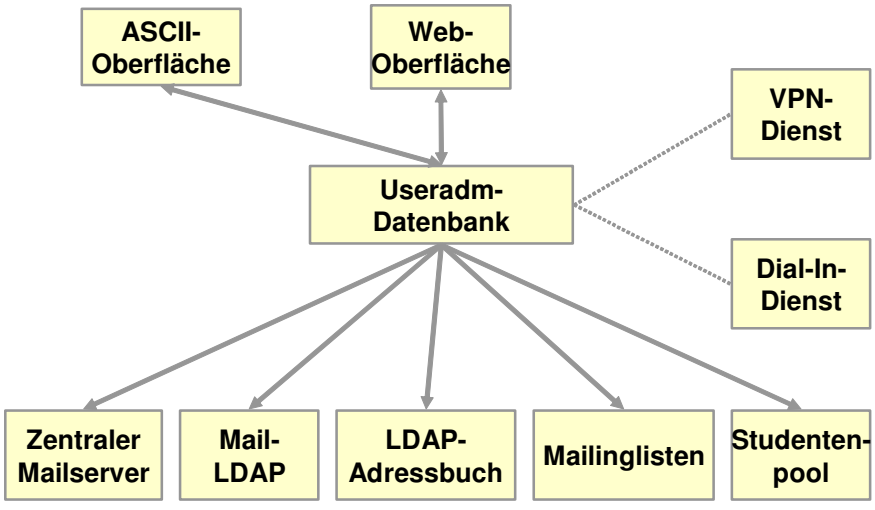
Das zentrale Nutzer-Verwaltungswerkzeug an der Fakultät für Informatik stellt der von der ATIS betriebene Useradm dar, der in [HK01] dokumentiert wird. Es handelt sich dabei um eine zentrale SQL-Datenbank, auf die von den Administratoren der einzelnen Organisationsbereiche (Institute, Forschungsgruppen, Geschäftsführung, etc.) über eine Web-Oberfläche oder eine ASCII-Oberfläche zugegriffen werden kann, um Nutzernamen zentral zu registrieren und um die Verwaltung des Mail-Dienstes durchzuführen.

Dazu werden zuerst die Nutzer registriert, wozu Vorname, Nachname und Institutszugehörigkeit angegeben werden müssen. Anschließend kann einem solchen Nutzer ein eindeutiger Nutzername zugewiesen werden, der gleichzeitig als Email-Adresse für den Nutzer verwendet wird. Auch die Angabe weiterer Email-Adressen (so genannte Mail-Aliase) ist möglich. Aus den Institutszugehörigkeiten und den erfassten Email-Adressen können auf Wunsch automatisch Mailinglisten generiert werden, über die etwa alle Mitarbeiter eines Instituts erreicht werden können.




**Information 32:Useradm**

Andere Dienste wurden später an den Useradm angebunden, teilweise über automatische Synchronisationsmechanismen, teilweise über Betriebsprozesse.
Beispielsweise wird der Studentenpool (siehe 6.3.4) direkt an den Useradm angebunden und automatisch synchronisiert, während für andere Dienste zwar eine Registrierung der Nutzer im Useradm vorausgesetzt wird, bevor Zugangsberechtigungen eingerichtet werden können, gegenwärtig jedoch keine automatische Überwachung dieser Koppelung durchgeführt wird. Beispiele sind der VPN-Dienst oder der Dial-In-Dienst, deren Anbindung in Information 33 daher besonders gekennzeichnet ist. Damit ist die Koppelung verschiedener Dienste an den Useradm derzeit nicht technisch sondern nur durch Betriebsprozesse gewährleistet, obwohl eine technische Koppelung angestrebt wird, um eine konsistente Datenhaltung über alle betriebenen Dienste hinweg zu gewährleisten.




**Information 33: Useradm und abhängige Dienste**

Insgesamt realisiert der Useradm also derzeit die für alle Dienste vorgesehene Nutzerverwaltung und stellt gleichzeitig selbst einen Dienst dar, über den das Mail-System der Fakultät gesteuert werden kann. Fehlende Synchronisationsmechanismen

zwischen den unterschiedlichen Datenbeständen erschweren jedoch gegenwärtig den Betrieb.

### 6.2.2 Verwaltete Informationen

Der Useradm verwaltet Nutzerdaten, die mehrere unterschiedliche Anwendungen betreffen. Zum einen werden originäre Nutzerinformationen wie Vorname, Nachname, Telefonnummer und Institutszugehörigkeit erfasst, zum anderen werden aber auch Daten aus den Bereichen Unix-Accounts und Mail-Verwaltung.


**Information 34: Nutzerdaten im Useradm**

Diese Vermischung unterschiedlicher Dienste ist historisch bedingt, jedoch sollte bei einer Weiterentwicklung der Nutzerverwaltung angestrebt werden, diese Dienste gemäß unseres Verwaltungsansatzes (siehe 5.4) voneinander zu trennen. Damit würde es möglich, die Verwaltung der einzelnen Dienste künftig besser und effektiver an die sich verändernden Anforderungen anzupassen.

### 6.2.3 Vergleich mit Nutzerverwaltungsansatz

In Kapitel 5 haben wir einen neuen Ansatz zur Nutzerverwaltung untersucht, bei dem eine LDAP-basierte Nutzerverwaltung als Basis genutzt wird, um eine effiziente Verwaltung von Diensten zu ermöglichen. Diesem Ansatz kann der Useradm in seiner derzeitigen Form nicht gerecht werden, da er einerseits auf einer Datenbank basiert, wodurch wichtige Erweiterungsmöglichkeiten fehlen, und neben der reinen Nutzerverwaltung auch eine Dienstverwaltung des Mailsystems durchführt.
Um unseren Ansatz zu unterstützen, wird es daher im ersten Schritt erforderlich, den bisherigen Useradm effizient an eine LDAP-Struktur anzubinden. Darauf aufbauend kann man dann die weiteren, an der Fakultät für Informatik bereit gestellten Dienste gemäß unserem dargestellten Ansatz verwalten.
Dabei erscheint es sinnvoll, die Verknüpfung von Nutzerverwaltung und Steuerung des Mail-Systems aufzulösen und sauber voneinander zu trennen.

### 6.2.4 Abbildung Useradm auf LDAP-Verzeichnis

Die Informationen, die derzeit in der Nutzerwaltung erfasst werden, sollen in einem ersten Schritt durch Synchronisationsprozesse vollständig auf ein LDAP-Verzeichnis abgebildet werden. In der Folge können die existierenden Dienste sukzessive auf eine LDAP-Nutzung umgestellt und neue Dienste direkt LDAP-basiert eingeführt und betrieben werden. Nachdem diese Umstellungen erfolgreich abgeschlossen sind, kann

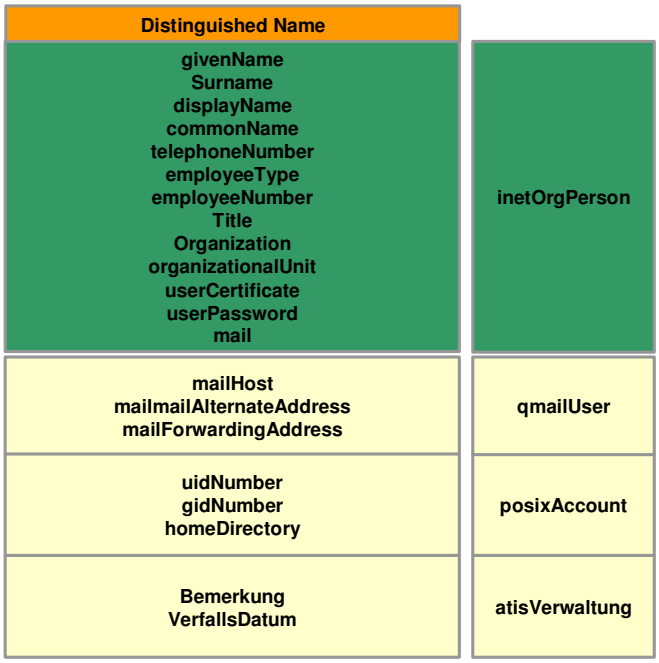
schließlich auch die Synchronisation zwischen Datenbank und LDAP-Verzeichnis deaktiviert werden, sodass die Nutzerverwaltung schließlich vollständig LDAP-basiert durchgeführt werden kann.
Da wir vom Nutzer als Person ausgehen werden, fällt der erste Blick auf die dazu vorgesehenen Objektklassen „*Person*“, „*OrganizationalPerson*“ und „*InetOrgPerson*“ (RFC 2798). Da wir unter anderen einen Nutzer-Login und eine Email-Adresse speichern wollen, scheiden „*Person*“ und „*OrganizationalPerson*“ aus, sodass lediglich „*InetOrgPerson*“ für uns in Frage kommt. Alternativ könnten wir alternative, standarisierte Basisklassen wie beispielsweise „*Account*“ untersuchen, doch haben diese eine Funktion, die über die reine Repräsentation eines Nutzers hinausgeht, indem etwa durch „*Account*“ auch ein Dienst impliziert wird. Nach den Erfahrungen mit dem bisherigen Useradm soll jedoch eine solche Verknüpfung unterschiedlicher Dienste vermieden werden (siehe 6.2.2).
Da die bisher existierenden Useradm-Daten zum Anlegen von Unix-Accounts genutzt wurden, existieren auch spezielle Attribute für diesen speziellen Anwendungsbereich. Auch wenn diese Verknüpfung in Zukunft vielleicht nicht mehr sinnvoll sein sollte und Nutzerdaten existieren könnten, denen kein Unix-Account zugeordnet ist, so müssen diese Informationen bis auf Weiteres im Rahmen der Synchronisation gepflegt werden. Die für Unix-Accounts vorgesehene Objektklasse hat den Namen „*posixAccount*“ und kann als Hilfsklasse mit der strukturellen Objektklasse „*inetOrgPerson*“ kombiniert werden.

| Bedeutung | Useradm-Attribut | LDAP-Attribut |
|---|---|---|
| Vorname | „Vorname“ | „givenName“ |
| Nachname | „Name“ | „Surname“ |
| Vollständiger Name | Generierbar aus „Vorname“ und „Nachname“ | „commonName“ |
| Anzuzeigender Name | Generierbar aus „Vorname“ und „Nachname“ | „displayName“ |
| Diensttelefonnummer | „Diensttel“ | „telephoneNumber“ |
| Mitarbeiter-Typ | „Klasse“ | „employeeType“ |
| Mitarbeiter-Nummer | Für Studierende: „Matrikelnummer“ | „employeeNumber“ |
| Titel | „Title“ | „Title“ |
| Institut | Ableitbar aus „Auslieferungsgruppe“ | „organziation“ bzw. „organizational unit“ |
| Nutzerlogin | „Login“ | „Uid“ |
| Unix-User-Nummer | „Uid“ | „uidNumber“ |
| Unix-Gruppen-Nummer | „Gid“ | „gidNumber“ |
| Email-Adresse | Entspricht „Login“ | „Mail“ |
| Verfallsdatum | „Verfallsdatum“ | |
| Bemerkungen | „Bemerkungen“ | |
| Email-Aliase | Ableitbar aus „mailalias“-Relation | |
| Ziel-Mailserver | Ableitbar aus „Auslieferungsgruppe“ | |
| Email-Auslieferung | „Auslieferungsgruppe“ | |
| X.509-Zertifikat | | „userCertificate“ |
| Passwort | | „userPassword“ |
| Heimat-Verzeichnis | | „homeDirectory“, generierbar aus Login und Institut |

**Tabelle 3: Abbildung von Useradm auf LDAP**

Tabelle 3 zeigt die benötigten Nutzerinformationen und stellt die im Useradm vorhandenen Attribute und ihre Abbildung auf Standard-LDAP-Attribute gegenüber. Bei dieser Abbildung wurden ausschließlich Standard-LDAP-Attribute verwendet, die durch die LDAP-RFC spezifiziert sind und durch die Objektklassen „*inetOrgPerson*“ bzw. „*posixAccount*“ verwendet werden können. Man erkennt jedoch Lücken, zu denen keine sinnvollen Standard-Attribute existieren, sodass eine Erweiterung des LDAP-Schemas erforderlich wird. Dies bezieht sich auf die Bereiche „Verwaltung“, in dem ein Verfallsdatum für die Accounts gepflegt und ein Bemerkungs-Feld verwendet werden, sowie auf den Email-Bereich, in dem Mail-Aliase und Auslieferungsinformationen erfasst werden müssen.
Für diese Zwecke müssen von uns neue Attribute definiert und geeignete Objektklassen eingesetzt werden. Alternativ kann eine eigene Erweiterung der „*Person*“- oder „*inetOrgPerson*“-Objektklassen spezifiziert werden oder es können neue Hilfsklassen erzeugt werden. Da die Nutzerdaten nicht dauerhaft mit dem Mail-Dienst verknüpft werden sollen (siehe 6.2.2), ist zumindest für die Verwaltung der Mail-Daten eine Hilfsklasse vorzusehen. In der Praxis existieren für diesen Zweck bereits Schema-Definitionen, die allerdings nicht in RFC-Form standarisiert sind. Häufig findet das

„Qmail-Schema“ Anwendung, das auch unsere konkreten Anforderungen erfüllen kann. Deshalb werden wir im Folgenden die Objektklasse „*qmailUser*“ verwenden.
Zur Unterstützung der Verwaltungsinformationen wäre alternativ eine Erweiterung der existierenden Objektklassen möglich, doch entscheiden wir uns aus Flexibilitätsgründen auch hier für die Einführung einer Hilfsklasse „*atisVerwaltung*“. In der Tabelle sind mit „*homeDirectory*“, „*userPassword*“ und „*userCertificate*“ auch Attribute aufgeführt, die im Useradm nicht erfasst werden, für unsere Nutzerverwaltung jedoch sinnvoll sind. „*userPassword*“ und „*userCertificate*“ bieten dem Nutzer zwei unterschiedliche Möglichkeiten, sich gegenüber den Diensten zu authentifizieren, während „*homeDirectory*“ von der Objektklasse „*posixAccount*“ erzwungen wird („*Must*“-Attribut) und sich leicht aus den existierenden Informationen generieren lässt.




**Information 35: Verwendete Objektklassen**

Information 35 zeigt einen der resultierenden Einträge, die im LDAP-Verzeichnis erzeugt werden müssen, um den Datenbestand des Useradm vollständig und verlustlos übernehmen zu können. Die verwendeten Objektklassen sind zur Veranschaulichung angegeben.

## 6.3 Dienste innerhalb der ATIS

Die Abteilung Technische Infrastruktur (ATIS) betreibt im Auftrag der Fakultät für Informatik neben dem Useradm weiter Dienste für Mitarbeiter und Studierende. Dazu gehören beispielsweise die unterschiedlichen Netzzugangsdienste per Dial-In, VPN oder 802.1x-Standard, sowie der Studentenpool. Diese Dienste sollen im Folgenden näher analysiert werden.

### 6.3.1 Dial-In an der Fakultät

Neben dem Useradm gehört auch der Dial-In-Dienst der Fakultät für Informatik zu den von der ATIS betriebenen Diensten. Der Zugang selbst erfolgt über das PPP-Protokoll, die Authentifizierung und Autorisierung erfolgt durch die Verwendung eines Radius-Servers (*Remote Authentication Dial In User Service*), der die erforderlichen

Informationen aus einem LDAP-Verzeichnis bezieht. Dieser Dienst steht allen Mitarbeitern der Fakultät für Informatik offen und ermöglicht derzeit einen sicheren und privilegierten Zugang zum allgemeinen Fakultätsnetz.
Dabei müssen für jeden Nutzer eine Identifikation, ein Passwort, die beim Dial-In zuzuweisende IP-Adresse sowie die individuellen Nutzungsrechte hinterlegt werden. Obwohl derzeit kein technischer Abgleich existiert ist es vorgesehen, nur im Useradm erfassten Mitgliedern der Fakultät auch einen Dial-In-Zugang zu ermöglichen. Deshalb ist eine manuelle Überprüfung im Rahmen der Verwaltungsprozesse erforderlich.

### 6.3.2 VPN an der Fakultät

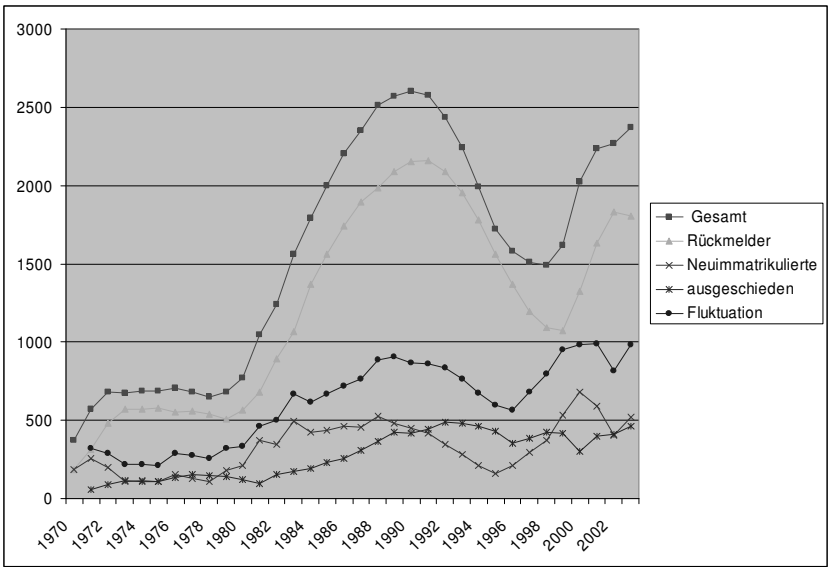
Die ATIS realisiert neben dem Dial-In-Dienst auch einen VPN-Dienst (*Virtual Private Network*), über den Angehörige der Fakultät für Informatik auch über das Internet einen sicheren, verschlüsselten Zugriff auf interne Ressourcen der Fakultät erhalten können.
Ähnlich dem Dial-In-Dienst wird auch für diesen Dienst ein Radius-Server eingesetzt, der die erforderlichen Zugangsdaten zur Authentifizierung und Autorisierung aus einem LDAP-Verzeichnis bezieht. Als Daten werden zusätzlich zur Nutzeridentifikation eine spezielle VPN-IP-Adresse sowie ein Nutzer-Zertifikat nach X.509-Standard zur Authentifikation verwaltet. Auch für diesen Dienst ist ein Abgleich mit dem Useradm erforderlich, der derzeit noch manuell durchgeführt werden muss.
Da zwischen diesem Netz und den einzelnen Instituten *Firewalls* existieren, die den Zugang zu den einzelnen Instituts-Netzwerken einschränken, ist geplant, auch den Instituten den Betrieb von VPN-Zugangsservern zu ermöglichen, die sich in die bestehende Verwaltung der ATIS einfügen.

### 6.3.3 802.1x an der Fakultät

Um den Mitarbeitern an der Fakultät für Informatik einen vollwertigen Netzwerkzugang von wechselnden Arbeitsplätzen aus zu ermöglichen, wird gegenwärtig der Einsatz eines Netzzugangssystem nach Standard 802.1x vorbereitet. Wird eine Netzwerkkomponente, i. d. R. ein Notebook, aktiv, so fordert ein Authentifikationsserver eine Anmeldung an, bevor die Komponente das Netzwerk nutzen kann. Über das *PPP Extensible Authentication Protocol* (kurz EAP) stehen dazu verschiedene Mechanismen zur Verfügung wie beispielsweise einfache Passwörter, Tickets nach Kerberos-Standard oder eine Zertifikats-basierte Authentifikation. Nach einer erfolgreichen Anmeldung wird für die Netzwerkkomponente ein vollwertiger Netzzugang in einem speziellen Netzbereich (*Virtual Local Area Network*, kurz VLAN) aktiviert.
An der Fakultät für Informatik soll eine Authentifikation auf Basis von Zertifikaten nach X.509-Standard durchgeführt werden.

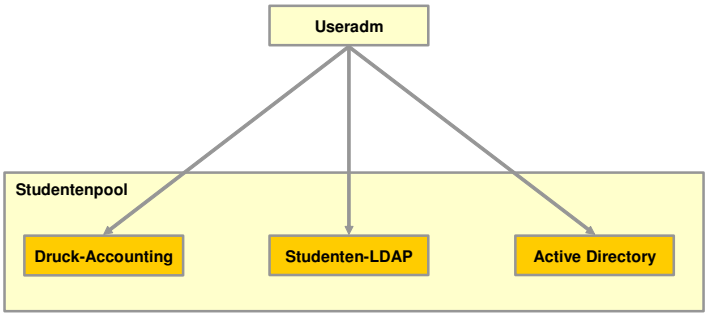
### 6.3.4 Studentenpool

Die ATIS stellt den Studierenden im Auftrag der Fakultät für Informatik Computerarbeitsplätze und weitere Ressourcen (Email, Homepage und Drucker) zur Verfügung, damit alle Studierenden die Möglichkeit haben, das in den Vorlesungen erworbene Wissen aufzuarbeiten und zu vertiefen. Dieser Dienst wird im Allgemeinen als Studentenpool bezeichnet.
Der Betriebsaufwand, der durch die ATIS bewältigt werden muss, ist für diesen Dienst vergleichsweise hoch, da gerade die Studierenden nicht nur die größte Nutzergruppe an der Fakultät darstellen, sondern auch eine besonders hohe Fluktuation haben.




**Information 36: Entwicklung der Studierendenzahlen**

Wie Information 36 zeigt, beträgt die Fluktuation schon seit mehreren Jahren über 900 Studierende pro Jahr. Da sich der Anstieg der Neuimmatrikulationen der vergangenen Jahre bisher noch nicht vollständig auf die Zahl der Exmatrikulationen ausgewirkt hat, ist mit einem weiteren Anstieg der Fluktuation zu rechnen.
Der Studentenpool wird von einem überwiegenden Teil der Studierenden genutzt, sodass man sicher von einer Fluktuation von 800 Zugängen pro Jahr ausgehen muss, die von den Mitarbeitern bearbeitet werden müssen. Dabei muss berücksichtigt werden, dass die Zugangsanträge der Studierenden in der Regel kurz nach Semesteranfang gestellt werden, sodass die Anträge nicht gleichmäßig verteilt sondern gebündelt in den Monaten Oktober und November verarbeitet werden müssen.
Eine effiziente Verwaltung ist daher wesentlich, um den Studierenden trotz dieser Belastung zeitnah die zum Studium nötigen Zugangsberechtigungen erteilen zu können.
Eine Automatisierung der Nutzerverwaltung ist als Reaktion auf die Veränderungen der letzten Jahre unumgänglich. Gegenwärtig müssen die Studierenden in verschiedenen Systemen registriert und verwaltet werden, beispielsweise im Useradm, im System zur Abrechnung der Druckernutzung, im LDAP-basierten Unix-Zugangs-Dienst und im Active Directory. Eine Reduzierung auf weniger Systeme würde die Verwaltungsschnittstellen wesentlich vereinfachen.



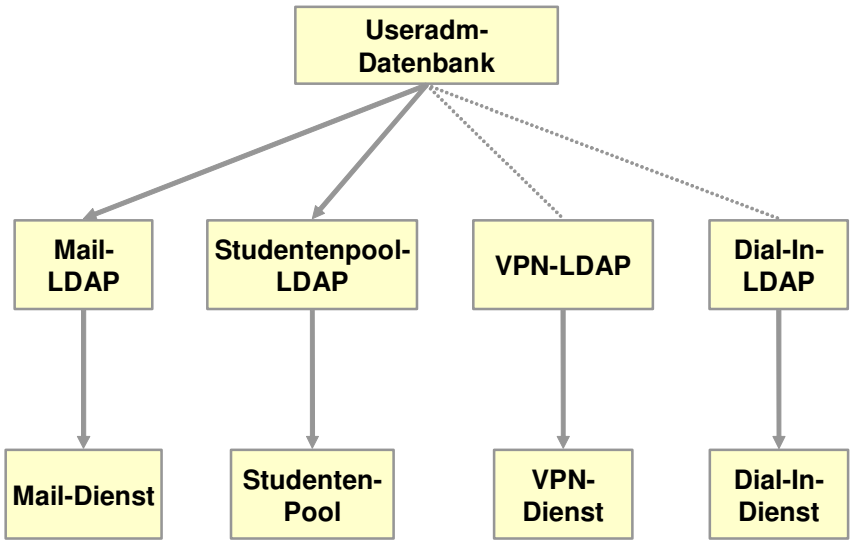

**Information 37: Studentenpool**

Wie aus Information 37 ersichtlich ist der Useradm ein führendes System, aus dem heraus die nötigen Informationen auf die anderen Systeme übertragen werden.

Als erste Maßnahme steht durch die Unterstützung der Fakultätsleitung eine Liste der gegenwärtig an der Fakultät für Informatik eingeschriebenen Studierenden zur Verfügung. Diese Liste besteht zwar lediglich aus den Matrikelnummern der Studierenden und enthält aus Gründen des Datenschutzes keine Namen, doch da bei der Anmeldung der Studierenden diese Matrikelnummer erfasst wird, ist ein automatischer Abgleich der vorhandenen Zugangsberechtigungen mit den immatrikulierten Studierenden möglich, sodass die Sperrung und Löschung der exmatrikulierten Studierenden automatisch durchgeführt werden kann.
Nach wie vor besteht jedoch das Problem, bis zu 500 neu immatrikulierten Studierenden zeitnah einen Zugang zu gewähren. Ein geeigneter Lösungsvorschlag folgt in Kapitel 7.

## 6.4 Maßnahmen

Betrachtet man die aufgeführten Dienste, so stellt man fest, dass bereits in fast allen Bereichen ein LDAP-basierter Ansatz zur Nutzerverwaltung verfolgt wird, wobei diese LDAP-Verzeichnisse derzeit jedoch noch Insellösungen bilden, die teilweise manuell mit dem Useradm abgeglichen werden müssen.




**Information 38: Gegenwärtiger Einsatz von LDAP-Verzeichnissen**

Es scheint daher sinnvoll, die existierenden LDAP-Verzeichnisse zu untersuchen und sukzessive in ein einziges Verzeichnis als zentrale Schnittstelle zu integrieren. Nach dem in 5.2 vorgestellten Ansatz ist dazu zuerst eine gemeinsame Nutzerverwaltung aller Dienste erforderlich, die in diesem Fall auf der bisherigen Nutzerverwaltung durch den Useradm basieren kann. Die bisherige, logische Sicht zeigt Information 33, bei der der Dial-In-Dienst und der VPN-Dienst lediglich logisch von der Nutzerverwaltung abhängen, während für die anderen Dienste bereits ein automatischer Abgleich durchgeführt wird.

### 6.4.1 LDAP-basierte Nutzerverwaltung

Um eine zentrale Verwaltung der unterstützten Dienste zu ermöglichen, ist eine gemeinsame Nutzerverwaltung eine unverzichtbare Voraussetzung. Da bereits bisher keinem Nutzer, der nicht zuvor im Useradm registriert wurde, eine Dienstnutzung ermöglicht werden sollte, bilden die existierenden Nutzer der einzelnen Dienste jeweils eine Untermenge der im Useradm registrierten Nutzer, sodass bei einer Verschmelzung

dieser Datenbestände auf Basis des existierenden Useradm keine Probleme zu erwarten sind.
Die derzeit in der Nutzerverwaltung erfassten Daten könnten, bei Definition einer entsprechenden Objektklasse, ohne Verluste auch in einem LDAP-Verzeichnis erfasst und verwaltet werden. Allerdings werden derzeit Daten unterschiedlicher Dienste gemeinsam im Useradm verwaltet, was unserem dienstorientierten Ansatz und der damit verbundenen klaren Diensttrennung widersprechen würde. Deshalb bietet es sich in diesem Schritt an, mehrere Objektklassen zu definieren und gemeinsam zu nutzen, um die Dienste voneinander abzugrenzen und spätere Anpassungen einzelner Dienste zu erleichtern. Gemäß Information 34 werden also vier Bereiche unterschieden, die Nutzerdaten, Verwaltungsdaten, Unixdaten und Maildaten. Zur Speicherung der Nutzerdaten kann eine der üblichen, im LDAP-Standard definierten Objektklassen verwendet werden, und zwar „*Person*“ bzw. „*OrganizationalPerson*“, falls auf die Erfassung eines Logins verzichtet werden kann, oder „*inetOrgPerson*“, wenn diese Information erhalten werden soll. Da wir als Ziel eine Dienstunterstützung vorsehen und Dienste i. d. R. eine Form von Login erfordern, werden wir also aller Voraussicht nach die Objektklasse „*inetOrgPerson*“ verwenden.
In einem ersten Schritt kann eine Synchronisation der bestehenden, datenbankbasierten Datenhaltung mit einer LDAP-basierten Datenhaltung vorgesehen werden, um eine sukzessive Entwicklung und Umstellung auf die neue Datenhaltung zu ermöglichen. Um diese Synchronisation zu unterstützen, sollten in dieser Phase auch die im bisherigen Useradm verwendeten Schlüsselwerte im LDAP-Verzeichnis gespeichert werden, um über beide Datenbestände einen gemeinsamen Schlüssel einsetzen zu können. Dazu wird eine weitere Objektklasse benötigt, die nach der Übergangsphase wieder deaktiviert werden kann.

### 6.4.2 Verifiziere Dienstansatz mit Dial-In

Die gegenwärtig verwendete LDAP-Struktur setzt für den Dial-In-Dienst eine spezielle Radius-Objektklasse ein, die vom Hersteller des Radius-Servers definiert und zur Verfügung gestellt wurde.
Diese Objektklasse „*RadiusProfile*“ ist als „strukturell“ definiert und kollidiert daher mit bereits existierenden LDAP-Einträgen, die ebenfalls eine „strukturelle“ Objektklasse verwenden müssen (gemäß LDAP-Standard darf ein LDAP-Eintrag nur genau eine strukturelle Objektklasse verwenden, siehe 3.4). Damit ist es unter Verwendung des „*RadiusProfile*“ nicht möglich, einem bereits existierenden Nutzer zu einem späteren Zeitpunkt den Zugang zu einem Dial-In-Dienst einzurichten.
Daher müssen also auch für diesen Dienst eine neue Objektklasse und ein Dienst-Nutzer eingerichtet werden, die wir als „*atisDialinDienst*“ bzw. „*atisDialinUser*“ bezeichnen werden. Allerdings können wir hier die bereits für die Objektklasse „*RadiusProfile*“ existierenden Definitionen der erforderlichen Attribute nutzen.
Die neue als „unterstützend“ definierte Objektklasse „*atisDialinDienst*“ benötigt einen Login und ein Passwort zur Authentifikation der Nutzer sowie eine IP-Adresse, die bei der Einwahl zugewiesen werden soll. Da mit einer Einwahl unterschiedliche Rechte verbunden sein können, etwa die Nutzung unterschiedlicher Protokolle, muss außerdem erfasst werden, zu welcher Nutzergruppe der jeweilige Nutzer gehört. In einem zweiten Schritt werden dann den einzelnen Gruppen spezifische Rechte zugewiesen.
Die bereits existierenden Attribute „*uid*“ (für Login) und „*userPassword*“ (für das Passwort) können auch vom Dial-In-Dienst verwendet werden. Neu ist „*RadiusGroupName*“, ein mehrwertiges Attribut das die benötigten Dial-In-Nutzergruppen spezifiziert, denen der jeweilige Nutzer zugeordnet wird. Schließlich

wird noch „*RadiusFramedIpAddress*“ für die feste IP-Adresse, die bei der Einwahl zugewiesen werden soll, benötigt. Als zusätzliche Information wünscht der Betreiber ein „*mail*“-Attribut, in dem eine Email-Adresse zur Kontaktaufnahme mit dem Nutzer hinterlegt wird. Soll ein Nutzer auch das Recht erhalten, einen Zugang per *Callback* aufzubauen, d. h., eine Verbindung wird von der Fakultät zum Nutzer hin aufgebaut und die Kommunikationskosten werden von der Fakultät getragen, dann wird auch ein Attribut benötigt, über das die zu wählende Telefonnummer angegeben werden kann. Dieses Attribut ist daher optional. Prinzipiell existieren noch weitere Radius-Attribute, die im LDAP-Verzeichnis gepflegt werden könnten, allerdings mit Standard-Werten vorbelegt sind und in der Regel nicht verändert werden müssen. Daher kann man auf die Unterstützung dieser Attribute im Schema verzichten oder sie als „optional“ kennzeichnen.
Die Zugriffsrechte für den „*atisDialinUser*“ können auf die genannten Attribute beschränkt werden. Da zur Passwort-Prüfung ein *Authentication Bind* (siehe 3.4) eingesetzt werden kann, werden vom „*atisDialinUser*“ keine Rechte für das Nutzerpasswort benötigt, für die übrigen Attribute reichen Lese-Rechte aus.
Im Rahmen einer Selbstverwaltung kann dem Nutzer das Recht eingeräumt werden, die vom *Callback* genutzte Telefonnummer eigenständig und ohne Eingriff des Betreibers zu verändern. Daneben benötigt der Nutzer lediglich das Recht, sein eigenes Passwort zu verändern. Die verbleibenden Attribute werden ausschließlich durch Verwaltungsprozesse des Betreibers verwaltet, die an dieser Stelle jedoch nicht weiter betrachtet werden sollen.
Eine genauere Analyse des gegenwärtig verwendeten Dial-In-Systems findet sich in [We04].

### 6.4.3 Verifiziere Dienstansatz mit VPN

Da der „ATIS-VPN-Dienst“ betrachtet wird, muss nach unserem Ansatz zuerst eine korrespondierende Objektklasse „*atisVpnDienst*“ erzeugt werden, ebenso nach unserem Ansatz ein Nutzereintrag „*atisVpnUser*“.
Die unbedingt erforderlichen Daten sind ein Login zur Identifizierung, ein hinterlegtes Zertifikat nach X.509-Standard zur Autorisierung und eine feste IP-Adresse, die der Verbindung bei erfolgreicher Anmeldung zugewiesen werden soll. Sowohl für den Login mit „*uid*“ und das Zertifikat mit „*userCertificate*“ existieren passende LDAP-Attribute, lediglich die IP-Adresse muss vom Dienstbetreiber als zusätzliches Attribut neu definiert werden. Da es sich um den „ATIS-VPN-Dienst“ handelt, erhält das neue Attribut den Bezeichner „*atisVpnIp*“. Neben diesen technisch erforderlichen Daten ist vom Betreiber auch die Erfassung einer Kontakt-E-Mail-Adresse vorgesehen. Dazu wird das übliche LDAP-Attribut „*mail*“ verwendet.
Damit werden von der Objektklasse „*atisVpnDienst*“ die Attribute „*atisVpnIp*“, „*uid*“, „*mail*“, und „*userCertificate*“ verpflichtend gefordert.
Die Systeme, die den Dienst bereitstellen und über den Proxy-User „*atisVpnUser*“ auf das Verzeichnis zugreifen, benötigen lediglich lesenden Zugriff auf die Attribute „*uid*“, „*atisVpnIp*“ und „*userCertificate*“. Die E-Mail-Adresse wird ausschließlich im Verwaltungsbereich genutzt, sodass der eigentliche Dienst hier keine Berechtigungen erfordert.

### 6.4.4 Verifiziere Dienstansatz mit 802.1x

Der Zugangsdienst nach 802.1x ähnelt sehr dem Netzwerkzugang mittels VPN. Auch in diesem Fall beginnen wir, indem wir eine geeignete Objektklasse „*atis8021xDienst*“ und einen Nutzereintrag „*atis8021xUser*“ erzeugen. Auch in diesem Fall werden ein Login und ein Zertifikat nach X.509-Standard benötigt, um die Nutzer eindeutig

identifizieren und autorisieren zu können. Da den Nutzern spezielle VLANs zugewiesen werden sollen, muss auch die Identifikationsnummer (VLAN-ID) des zuzuweisenden VLAN gespeichert werden. Diese VLAN-ID muss allerdings auch technischen Gründen in zwei verschiedenen Darstellungen hinterlegt werden, sodass zwei Attribute erforderlich sind. Daher definieren wir zwei neue Attribute mit den Bezeichnungen „*atisVlanID1*“ und „*atisVlanID2*“ und nehmen sie, zusammen mit „*uid*“ und „*userCertificate*“, als „*Must*“-Attribute in die Objektklasse „*atis8021xDienst*“ auf. Nur der Betreiber erhält das Recht, diese Attribute zu schreiben, außerdem benötigt der Dienst, repräsentiert durch „*atis8021xUser*“, Leserechte auf die genannten Attribute.

### 6.4.5 Verifiziere Dienstansatz mit LDAP-Adressbuch

LDAP hat sich auch als ein Standard durchgesetzt, mit dem zentrale Adressbücher gepflegt und von geeigneten Clienten wie beispielsweise Microsoft Outlook, Netscape Communicator, Eudora Mail, etc. abgefragt werden können. Ein solches Adressbuch wird schon jetzt an der Fakultät für Informatik betrieben, jedoch nur mit minimalen Informationen. Derzeit können nur Vorname, Nachname und Email-Adresse eines Fakultätsangehörigen abgefragt werden. Es kann zwar auch eine Telefonnummer angegeben werden, jedoch ist diese Information augrund des damit verbundenen Aufwands vergleichsweise schlecht gepflegt und unzuverlässig.
Die existierenden Schnittstellen für LDAP-Adressbücher unterscheiden sich zwar in Details, jedoch hat sich die Verwendung der in der Objektklasse „*inetOrgPerson*“ definierten Attribute als Kern durchgesetzt, den alle Clienten unterstützen. Da diese Objektklasse ohnehin in der zentralen Nutzerverwaltung eingesetzt werden soll, entsteht für den Betreiber kein wesentlicher Aufwand, um auch ein Adressbuch bereitzustellen. Es ist lediglich erforderlich, geeignete Zugriffsrechte zu definieren, die es dem jeweiligen Nutzer ermöglichen, seine persönlichen Informationen wie z. B. Raumnummer, Telefonnummer, etc. selbst zu pflegen. Dabei bleibt es den Nutzer überlassen, welche Informationen in welcher Art veröffentlicht werden sollen, womit auch die Bedürfnisse des Datenschutzes berücksichtigt werden. Allerdings ist es nicht möglich, dem Nutzer Änderungsrechte für von der Verwaltung benötigte Attribute wie beispielsweise Vorname, Nachname oder Institutszugehörigkeit einzuräumen. Das würde dem Grundsatz widersprechen, dass es für jedes Attribut nach Möglichkeit nur genau ein führendes System geben soll, um Konflikte zu vermeiden.
Es ist auch möglich, ein zusätzliches LDAP-Attribut „*SichtbarkeitAdressbuch*“ zu definieren, das der Nutzer selbst verwalten kann. Durch eine geschickte Vergabe von Leserechten auf das LDAP-Verzeichnis könnten dann im Adressbuch lediglich Einträge angezeigt werden, deren Veröffentlichung der Nutzer zuvor explizit zugestimmt hat. Dabei würde es allerdings erforderlich, eine neue Objektklasse speziell für dieses Attribut zu definieren und zu verwenden oder aber eine bereits existierende Objektklasse wie z. B. „*atisVerwaltung*“ (siehe 6.2.4) um dieses Attribut zu erweitern.
Damit demonstriert dieses Beispiel, anders als die zuvor genannten, den besonderen Vorteil für den Nutzer, der hier auch ohne Betreibermitwirkung eine eigenständige Datenverwaltung sinnvoll nutzen kann. Gleichzeitig kann der Betreiber seinen Kunden ohne eigenen personellen Aufwand einen erheblichen Zusatznutzen und erweiterten Service bieten.

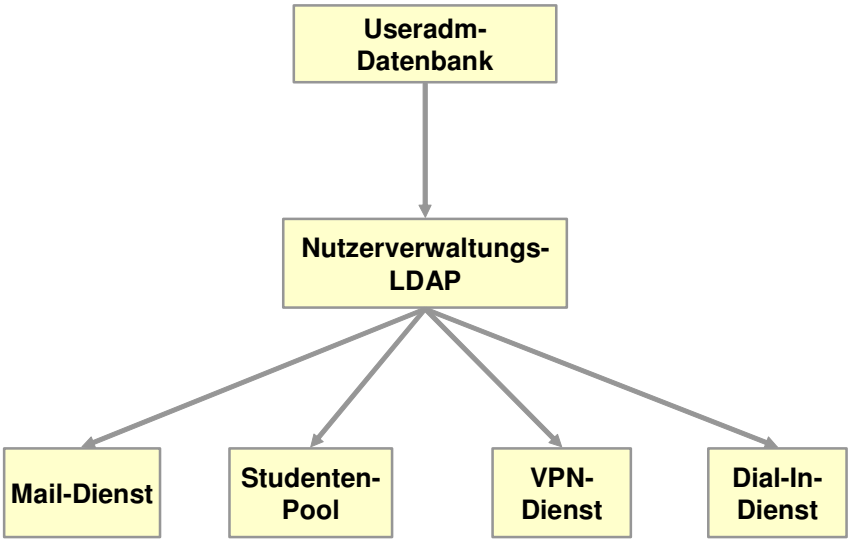
### 6.4.6 Verifiziere Dienstansatz am Beispiel Studentenpool

Den Studierenden an der Fakultät für Informatik werden durch die ATIS Computerarbeitsplätze, Netzwerkzugang und Drucker zur Verfügung gestellt. Dazu ist lediglich eine einmalige Anmeldung der Studierenden erforderlich.

Dabei wird bereits seit zwei Jahren erfolgreich ein standardkonformes LDAP-Verzeichnis zur Verwaltung der Zugangsdaten verwendet. Dieses LDAP-Verzeichnis wird derzeit durch einen Synchronisationsmechanismus an den Useradm angebunden, der allerdings unvollständig ist und nur neue Daten anlegt und keine Lösch-Operationen durchführt. Vor diesem Hintergrund sind zum einen keine prinzipiellen Probleme zu erwarten, wenn das existierende LDAP-Verzeichnis für die Studierenden in eine fakultätsweite Nutzerverwaltung (siehe 6.4.1) eingebunden werden soll, die ebenfalls den LDAP-Standards entspricht. Zum anderen entfällt durch eine solche Einbindung die bisherige, unvollständige Synchronisation, was die Verwaltungsprozesse erheblich verbessern würde.
Der existierende LDAP-Datenbestand kann problemlos über den Login oder die Unix-Nutzernummer als eindeutige Schlüssel mit der neuen, LDAP-basierten Nutzerverwaltung verknüpft werden. Allerdings werden dazu zusätzliche Attribute benötigt, die bisher nur im LDAP-Verzeichnis für den Studentenpool verwaltet werden und die bisher in der neuen Nutzerverwaltung nicht vollständig erfasst werden.
Zur Verwaltung der Unix-Zugänge werden Attribute aus den Objektklassen „*posixAccount*“ und „*shadowAccount*“ verwendet. Die Objektklasse „*posixAccount*“ wird zwar auch bei der neuen Nutzerverwaltung eingesetzt, jedoch werden dort nicht alle erforderlichen Attribute gepflegt, beispielsweise fehlen „*loginShell*“ und „*gecos*“. Andererseits sind alle anderen Attribute, die sowohl im Studentenpool als auch in der zentralen Nutzerverwaltung gepflegt werden, synchronisiert, sodass bei einer Zusammenführung keinerlei Konflikte zu erwarten sind.
Die Objektklasse „*shadowAccount*“ wird bisher noch nicht verwendet und enthält insbesondere Informationen zur Passwortverwaltung, beispielsweise das Änderungsdatum des Passworts.
Zusätzlich wird die Nutzung der Drucker mithilfe des LDAP-Verzeichnisses erfasst bzw. abgerechnet. Dazu wurde die Objektklasse „*printerAccounting*“ mit den erforderlichen Attributen definiert.
Folglich können die Objektklassen „*shadowAccount*“ und „*printerAccounting*“ ohne Probleme in der zentralen Nutzerverwaltung für die Verwaltung des Studentenpools verwendet werden. Lediglich bei der Objektklasse „*posixAccount*“, die sowohl von der Nutzerverwaltung als auch von dem Studentenpool verwendet wird, ist besondere Sorgfalt bei der Verwaltung erforderlich.
Leider verwenden die Nutzerverwaltung und der Studentenpool derzeit unterschiedliche strukturelle Objektklassen. Während die Nutzerverwaltung den Nutzer als „*Person*“ verwalten wird, setzt der Studentenpool den „*Account*“ in den Mittelpunkt. Es wird daher erforderlich, bei der Umstellung auf eine gemeinsame Nutzerverwaltung die existierenden LDAP-Einträge des Studentenpools geeignet anzupassen und nach Möglichkeit die Objektklasse „*Account*“ durch die Objektklasse „*Person*“ zu ersetzen. Da bisher von der Objektklasse „*Account*“ lediglich das Attribut „*uid*“ verwendet wird, das auch in den Objektklassen „*posixAccount*“ oder „*inetOrgPerson*“ definiert wird, sind dabei keine Probleme zu erwarten.
Insgesamt bleibt festzustellen, dass in der Migrationsphase durch die unterschiedlichen strukturellen Objektklassen und die Synchronisation der von „*posixAccount*“ erfassten Attribute ein gewisser Aufwand entstehen wird, wobei die Migration allerdings mithilfe gemeinsamer Schlüssel automatisiert werden kann. Die zu erwartenden Vorteile durch verbesserte Verwaltungsprozesse und den Wegfall von Synchronisationsprozessen rechtfertigen diesen Aufwand.

## 6.5 Fazit

Wir haben Fälle des *Identity Management* im Rahmen der Fakultät für Informatik betrachtet. Dabei haben wir die gegenwärtige Situation analysiert und erkannt, dass zwar alle betrachteten Dienste einen Bezug zur zentralen Nutzerverwaltung Useradm der Fakultät für Informatik haben, aber das diese Verbindung nicht in jedem Fall durch technische Synchronisationsprozesse, sondern oftmals lediglich durch fehleranfällige Betriebsprozesse erfolgt. Dies gilt insbesondere für die Netzwerkzugangsdienste Dial-In, VPN und 802.1x. Da aber schon jetzt von einem Großteil der existierenden Dienste LDAP-Verzeichnisse zur Datenhaltung eingesetzt werden, konnte darauf aufbauend die Datenhaltung verbessert und in nur noch einen LDAP-Datenbestand integriert werden. Der Parallelbetrieb verschiedener LDAP-Verzeichnisse (siehe Information 38) kann daher vermieden und, wie in Information 39 dargestellt, verbessert werden.




**Information 39: Künftiger LDAP-Einsatz**

Dabei wird im Migrationsprozess zuerst eine Synchronisation zwischen der existierenden Nutzerverwaltungs-Datenbank Useradm und einem LDAP-Verzeichnis implementiert. Die existierenden Dienste, von denen derzeit ein Teil direkt auf die Datenbank zugreift, können nun sukzessive an das zentrale LDAP-Verzeichnis angepasst werden. Erst nach der Migration aller Dienste wird es möglich, auf die Useradm-Datenbank zu verzichten und auch die Nutzerverwaltung ausschließlich LDAP-basiert durchzuführen.
Die ersten Migrationsschritte wurden bereits exemplarisch durchgeführt, indem ein LDAP-basiertes Verwaltungswerkzeuge für den VPN-Dienst nach den Vorgaben dieser Arbeit entwickelt wurde. Aufgrund der großen Ähnlichkeit können darauf basierend auch die erforderlichen Verwaltungswerkzeuge für den Dial-In-Dienst und 802.1x implementiert werden. Damit steht lediglich die Integration des Mail-Dienstes und des Studentenpools aus, die aber bereits jetzt LDAP-Verzeichnisse nutzen und deren Integration bereits diskutiert wurde.

# 7 EXTERNE QUELLEN FÜR BENÖTIGTE NUTZERDATEN

## 7.1 Federated Identity Management

Im Folgenden sollen Fälle von *Identity Management* betrachtet werden, bei denen wir Anwendungen analysieren, die Organisationsgrenzen überschreiten. Zur gemeinsamen Nutzung von Daten aus verschiedenen Organisationen ist insbesondere der Aufbau von Beziehungen erforderlich, weshalb für diesen Bereich der Begriff „*Federated Identity Management*“ geprägt wurde. Dabei ergeben sich besondere Fragestellungen und Probleme, von denen wir besonders rechtliche Aspekte wie z. B. den Datenschutz untersuchen wollen. Auch die technische Anbindung wird bei Fällen des „*Federated Identity Management*“ zunehmend komplexer. Beispielsweise werden hier Ansätze wie die „*Security Assertion Markup Language*“ (SAML) oder die „*Web Services Federation Language*“ eingesetzt, die feingranulare *Policy*-Entscheidungen ermöglichen, für die einfache LDAP-basierte Ansätze häufig nicht mehr ausreichen. Eine Betrachtung der technischen Protokolle würde jedoch den gegebenen Rahmen überschreiten, deshalb wird auf geeignete Literatur wie z. B. [PW04] oder [HR04] verwiesen.
In der folgenden Diskussion werden wir uns auf drei Dienste beschränken: das neu einzuführende Webinscribe, das IPO (siehe 4) und den Studentenpool (siehe 6.3.4). Diese Dienste richten sich an die Studierenden an der Fakultät für Informatik und müssen regelmäßig in einem kurzen Zeitraum die Registrierung großer Studentenzahlen bewältigen. Deshalb besteht die Möglichkeit, gerade bei diesen Diensten Gemeinsamkeiten und Überschneidungen zu identifizieren, die Optimierungen in den Abläufen und der Datenhaltung ermöglichen.

## 7.2 Webinscribe

Webinscribe wurde von der Forschungsgruppe Cooperation & Management für die Fakultät für Informatik entwickelt und wird im Rahmen der Tutorien-Zuteilung von der Geschäftsführung der Fakultät für Informatik eingesetzt.
Dazu erfasst ein Mitarbeiter zunächst die zu berücksichtigenden Tutorien der Fakultäten für Informatik und Mathematik und pflegt diese in das System ein. Die Tutorienvergabe erfolgt zu jedem Semester und betrifft jeweils über 1000 Studenten, die die angebotenen Tutorien im Haupt- oder im Nebenfach belegen möchten.
Die Studierenden können mittels Webinscribe ihre Präferenzen für die zu einer Vorlesung angebotenen Tutorien zu einer Vorlesung erfassen und werden schließlich, unter Berücksichtigung dieser Präferenzen, auf die Tutorien verteilt.

### 7.2.1 Nutzung durch die Studenten

Derzeit greifen die Studierenden über das Internet auf die Webinscribe-Anwendung zu und melden sich dort an. Als Login wird die Matrikelnummer verwendet, das Passwort wird durch einen Algorithmus aus der Matrikelnummer generiert. Der Algorithmus und damit das Passwort werden jedes Semester geändert.
Da die Studierenden Ihr eigenes Passwort nicht kennen, müssen sie zuvor zu den bekannt gegebenen Terminen die betreuenden Mitarbeiter bzw. Tutoren aufsuchen uns sich ausweisen. Daraufhin wird von den Mitarbeitern aus der Matrikelnummer ein Passwort generiert und dem Studierenden mitgeteilt. Dieses Verfahren der Passwort-Vergabe ist also sehr aufwändig und personalintensiv.
Zusätzlich zu Matrikelnummer und Passwort werden von Webinscribe auch Vorname und Nachname abgefragt, diese Daten dienen jedoch nicht der Authentifizierung.

Anschließend können die Studierenden ggf. Lerngruppen bilden und Präferenzen zu den angebotenen Tutorien angeben.
Nach Ablauf der Anmeldefrist werden die Anmeldungen von den betreuenden Mitarbeitern ausgewertet und die Tutorienzuteilung wird durchgeführt.

### 7.2.2 Zu berücksichtigende Nutzergruppen

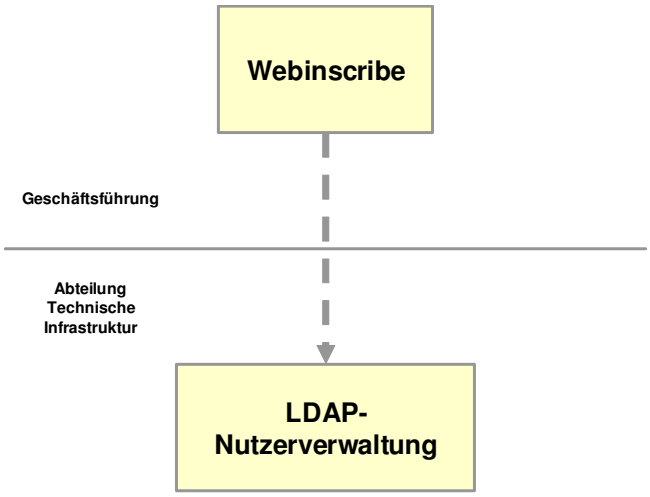
Webinscribe wird nicht nur von der Fakultät für Informatik zur Tutorienvergabe eingesetzt, sondern auch für die Vergabe von Tutorien zu Vorlesungen der Fakultät für Mathematik. Die Tutorien beider Fakultäten stehen aber auch jenen Studierenden offen, die nicht selbst an diesen Fakultäten eingeschrieben sind, sondern nur im Rahmen ihres Nebenfachs an den angebotenen Vorlesungen teilnehmen wollen. Betroffen sind jeweils Studierende mehrerer Semester, wobei leider nicht alle neu immatrikulierten Studierenden zum Zeitpunkt der Tutorien-Vergabe bereits über einen endgültigen Studierendenausweis verfügen und stattdessen nur eine vorläufige Bescheinigung vorweisen können. Es gibt auch eine kleine Gruppe von Schülern, die als Vorbereitung auf ein Studium an der Informatik-1-Vorlesung teilnehmen und die derzeit gesondert behandelt werden.

### 7.2.3 Problem

Da jedes Semester ein neues Zugangspasswort erzeugt wird und die Anmeldung zu den Tutorien innerhalb von weniger als einer Woche zwischen der ersten Vorlesung und dem Beginn der Durchführung der ersten Tutorien erfolgen muss, entsteht in jedem Semester ein sehr hoher Aufwand der nach dem derzeit eingesetzten Verfahren durch einen hohen Personaleinsatz bewältigt werden muss.
Deshalb ist es aus organisatorischen Gründen erforderlich, den durch die Zugangsvergabe an die Studierenden entstehenden Aufwand so weit wie möglich zu reduzieren. Da der wesentliche Aufwand durch die individuelle Vergabe von Zugangspasswörtern entsteht, bietet dieser Ansatzpunkt aller Voraussicht nach die größten Optimierungspotentiale.
Die Gruppe der betroffenen Studierenden ist zwar heterogen und besteht aus Studierenden verschiedenster Fakultäten, doch gehört der überwiegende Teil der Studierenden der Fakultät für Informatik an. Daher lässt sich insgesamt schon dann ein wesentlicher Fortschritt erzielen, wenn es zumindest für diesen Teil der Studierenden gelingt, eine Vereinfachung der Anmeldung zu ermöglichen.

### 7.2.4 Lösungsansatz durch Federated Identity Management

Da ein sehr großer Anteil der Studierenden an der Fakultät für Informatik bereits über einen Zugang zum Studentenpool verfügt oder aber einen solchen Zugang erhalten kann, wäre es denkbar, die im Studentenpool vorhandenen Zugangsdaten auch zur Anmeldung bei Webinscribe zu verwenden.




**Information 40: Überschreitung der Geschäftsbereiche**

In diesem Fall könnte für die Studierenden das persönliche Erscheinen bei der Passwortvergabe entfallen, während die betreuenden Mitarbeiter gleichzeitig wesentlich entlastet würden.
Da durch ein solches Vorgehen jedoch nur ein Teil der betroffenen Nutzergruppen abgedeckt werden kann, könnte damit das bisherige Verfahren allerdings nicht vollständig abgelöst sondern nur ergänzet werden. Die übrigen Nutzergruppen könnten dann weiterhin mit dem derzeit eingesetzten Verfahren behandelt werden.
Der Aufwand für eine solche Veränderung wäre aus Sicht der Betreiber von Webinscribe vergleichsweise gering. Die Einrichtung einer alternativen Passwortprüfung gegenüber den Daten des Studentenpools wäre die einzige erforderliche Veränderung und sollte mit vertretbarem Aufwand durchführbar sein.

### 7.2.5 Juristische Einschränkungen

Leider entstehen bei dem vorgeschlagenen Lösungsweg juristische Probleme, die eine sorgfältige Betrachtung erfordern. Die Tutorienvergabe wird von Mitarbeitern der Geschäftsleitung der Fakultät für Informatik durchgeführt, während der Studentenpool von der ATIS als einem Geschäftsbereich der Fakultät für Informatik durchgeführt wird.
Obwohl beide Bereiche Teile der Fakultät für Informatik sind, ist eine Weitergabe der Studierendendaten aus dem Studentenpool an die Betreiber von Webinscribe juristisch nicht zulässig, da diese Daten bei der Anmeldung der Studierenden zweckgebunden erhoben worden sind. Gemäß §4 Abs. 1 Teledienstedatenschutzgesetz (TDDSG) spezifiziert die Fakultät für Informatik als Betreiber des Studentenpools in §4 Absatz 3 der „Benutzerordnung für die zentralen Rechenanlagen und das Datennetz der Fakultät für Informatik an der Universität Karlsruhe“ die geplante Verwendung der erfassten Nutzerdaten.
Da die erfassten Daten als Bestandsdaten einer strengen Zweckbindung unterliegen, ist eine explizite Einverständniserklärung der betroffenen Studierenden eine unverzichtbare Voraussetzung für die Nutzung dieser Daten im Rahmen von Webinscribe. Diese Einwilligung der Studierenden muss so gestaltet sein, dass daraus keine neue Belastung für die betreuenden Mitarbeiter entsteht, deren Entlastung durch den veränderten Ablauf erzielt werden soll. Andererseits muss die Erfassung einer Einwilligung selbstverständlich den gesetzlichen Anforderungen genügen.
Da die Bearbeitung einer schriftlichen Zustimmung zu dieser Übertragung den insgesamt erforderlichen, personellen Aufwand zur Durchführung der Tutorienvergabe nicht verringern würde, soll als Alternative eine elektronische Einwilligung der Nutzer

in die Weitergabe der Nutzerdaten gemäß §3 Abs. 3 bzw. §4 Abs. 2 TDDSG betrachtet werden. Wesentlich für eine gültige, elektronische Einwilligung ist gemäß §4 Abs. 2 TDDSG, dass
1. sie nur durch eine eindeutige und bewusste Handlung des Nutzers erfolgen kann,
2. die Einwilligung protokolliert wird und
3. der Inhalt der Einwilligung jederzeit vom Nutzer abgerufen werden kann.
Des Weiteren muss der Studierende vor seiner Einwilligung durch den Betreiber über die konkreten Auswirkungen und sein Widerrufsrecht informiert werden. Es muss klar ersichtlich sein, welche Daten von der Einwilligung betroffen sind, welche Informationen übertragen werden sollen und zu welchem Zweck diese Informationen vom Empfänger verwendet werden sollen.
Diese Anforderungen können im Rahmen des Studentenpools vergleichsweise einfach erfüllt werden. Es wäre ausreichend, dem Studierenden eine Webseite zur Verfügung zu stellen, mit der der Studierende nach einer Anmeldung durch die Zugangsdaten des Studentenpools eine Einwilligung erteilen oder aber widerrufen kann. Dabei würde es allerdings nicht ausreichen, für den jeweiligen Studierenden zu speichern, ob eine gültige Einwilligung vorliegt, sondern es müsste auch ein Protokoll erstellt werden, aus dem genau hervorgeht, wann welcher Student eine Einwilligung erteilt oder widerrufen hat. Die kann jedoch in unterschiedlichen Systemen erfolgen, etwa indem der aktuelle Einwilligungsstatus „erteilt“ bzw. „nicht erteilt“ in einer Datenbank oder einem Verzeichnisdienst dem Nutzer zugeordnet wird und parallel eine Protokoll-Datei erstellt wird.
Von einem Widerruf der Einwilligung wären Datenübertragungen, die zwischen Einwilligung und Widerruf erfolgt sind, nicht betroffen, spätere Übertragungen dürften jedoch aufgrund des vorliegenden Widerspruchs nicht mehr durchgeführt werden.
Bei der Einholung einer Einwilligung zur Datenverarbeitung oder Datenweitergabe muss außerdem sichergestellt werden, dass das „Koppelungsverbot“ eingehalten wird, das einen „faktischen Zwang“ vermeiden soll. Dementsprechend wäre die „Koppelung der Erbringung einer Leistung an die Einwilligung in eine ansonsten unzulässige Datenverarbeitung“ ein Verstoß gegen das Koppelungsverbot und damit unzulässig. Jedoch findet das Koppelungsverbot bei Vorliegen einer alternativen, zumutbaren Zugangsmöglichkeit zum jeweiligen Dienst (vgl. §3 Abs. 4 TDDSG) keine Anwendung. Da neben einer Einwilligung in die Datenübermittelung von Studentenpool zu Webinscribe auch die persönliche Anmeldung zur Nutzung von Webinscribe bei den betreuenden Mitarbeitern weiterhin möglich sein muss, um andere Nutzergruppen zu unterstützen, ist das Koppelungsverbot derzeit nicht anwendbar. Da dieses Verfahren bereits seit mehreren Jahren eingesetzt wurde, kann davon ausgegangen werden, dass es für den Studierenden auch zumutbar ist.
Jedoch müsste dieser Aspekt genauer analysiert werden, falls zu einem späteren Zeitpunkt darüber nachgedacht wird, auf die alternative Anmeldung zur Webinscribe-Nutzung zu verzichten.

### 7.2.6 Technische Umsetzung

Wie bereits diskutiert, ist zusätzlich zu den Nutzerdaten, die im Studentenpool in einem LDAP-Verzeichnis erfasst sind und von Webinscribe verwendet werden können, ein Attribut erforderlich, in dem ein Status der Einwilligung erfasst wird. Man kann prinzipiell unterscheiden zwischen „keine Aussage“, „Einwilligung erteilt“ oder „Einwilligung widerrufen“. Da die Fälle „keine Aussage“ und „Einwilligung widerrufen“ jedoch keine Unterschiede in der Funktion bewirken, können diese Fälle auch zusammengefasst werden, sodass lediglich zwei Zustände existieren, die durch

einen einfachen Wahrheitswert ausgedrückt werden können. Neben diesen Informationen ist es erforderlich, die Erteilung oder den Widerruf zu protokollieren, was mit einer einfachen Protokolldatei geschehen kann.
Es muss sichergestellt werden, dass die Webinscribe-Anwendung lediglich auf LDAP-Einträge zugreifen kann, für die eine Einwilligung vorliegt. Aus Datenschutzgründen darf die Webinscribe-Anwendung nicht feststellen können, ob Einträge existieren, für die eine Zustimmung fehlt. Diese Einschränkungen sind jedoch mit den Konfigurations-Möglichkeiten üblicher LDAP-Server einfach zu realisieren, indem ein spezieller Webinscribe-Nutzer angelegt wird, der lediglich Zugriff auf bestimmte, vorher festgelegte Attribute von Einträgen erhält, bei denen das Attribut „WebinscribeEinwilligungErteilt“ den Wert „wahr“ beinhaltet.

### 7.2.7 Parallele Nutzerverwaltung

Da nicht alle Nutzer über den Studentenpool einen Zugriff auf Webinscribe erhalten können, muss parallel eine zweite Nutzerverwaltung für die weiteren Nutzer verwendet werden. Daraus können sich Probleme ergeben, solange keine Synchronisation dieser Nutzerverwaltungen durchgeführt wird. Sollte ein Nutzer sowohl ein Zugangspasswort von den Webinscribe betreuenden Mitarbeitern erhalten, als auch über einen Zugang zum Studentenpool verfügen, so könnte sich dieser Nutzer auf unterschiedliche Weisen gegenüber dem System anmelden. Damit wäre es dem Studierenden möglich, seine Präferenzen doppelt zu erfassen und so seine Chancen auf eine aus seiner Sicht „gute“ Tutorienzuteilung zu verbessern.
Da die Studierenden von Webinscribe über die Matrikelnummer als Schlüssel identifiziert werden, und diese Information auch vom Studentenpool übermittelt werden kann, besteht die Möglichkeit, durch zusätzliche Prüfungen derartige Missbrauchsversuche zu erkennen und zu unterbinden. Dies bedeutet jedoch einen zusätzlichen Implementierungsaufwand. Dieser Fall ist bei der Umsetzung besonders zu untersuchen, um eine geeignete Behandlung durch die Webinscribe-Anwendung zu garantieren. Es wäre möglich, die unterschiedlichen Anmeldevorgänge zu detektieren und ggf. eine Fehlermeldung auszulösen oder aber beide Anmeldevorgänge transparent zu behandeln und über beide Zugangsarten den richtigen Datensatz zuzuordnen. Prinzipielle Schwierigkeiten sind jedoch aufgrund des gemeinsamen Schlüssels nicht zu erwarten.

### 7.2.8 Fazit Webinscribe

Webinscribe ist unser erstes Beispiel für eine Anwendung von *Federated Identity Management*, bei der mehrere Organisationseinheiten involviert sind. Die in unserem Szenario neuen Probleme beziehen sich insbesondere auf Fragestellungen des Datenschutzes und der Zustimmung der Nutzer. Diese konnten jedoch befriedigend gelöst werden, sodass eine Umsetzung möglich ist.
Der personelle Aufwand zur Durchführung der Tutorien-Vergabe ließe sich durch eine Anbindung an eine zentrale Nutzerverwaltung, beispielsweise die des Studentenpools, massiv reduzieren, da für einen Großteil der Studierenden auf eine Passwort-Ausgabe verzichtet werden könnte. Als einziges Authentifizierungssystem scheinen Systeme der Fakultät für Informatik jedoch nicht geeignet, da ein Teil der Nutzer von Webinscribe keinen Zugang zum Studentenpool hat und einen alternativen Mechanismus erfordert.
Daher ist die parallele Nutzung zweier Authentifikationssysteme erforderlich, um alle Nutzergruppen unterstützen zu können, wodurch neue Synchronisationsprobleme entstehen.

## 7.3 IPO

Das IPO (siehe 1.2.1 und 4) ähnelt Webinscribe dahingehend, dass auch das IPO ein Dienst für die Studierenden an der Fakultät für Informatik bereitgestellt wird, zu dem unterschiedliche Nutzergruppen, überwiegend Studierenden an der Fakultät für Informatik, zeitnah zu Beginn der Vorlesungen Zugriff erhalten müssen. Da die Studierenden einen Zugang zum Studentenpool erhalten können, liegt es nahe, auch in diesem Fall über eine Anbindung an den Studentenpool nachzudenken.
Im Gegensatz zu Webinscribe richtet sich der Dienst jedoch primär an Studierende im ersten Semester und es erfolgt hier eine automatische Registrierung zur Nutzung des IPO, die in der Regel ohne Eingriffe der betreuenden Mitarbeiter durchgeführt werden kann. Damit sind die Optimierungsmöglichkeiten von vornherein geringer, als im Fall von Webinscribe.

### 7.3.1 Zu berücksichtigende Nutzergruppen

Die zur Nutzung des IPO erforderliche Differenzierung nach Nutzergruppen wurde bereits in Kapitel 4 detailliert untersucht. Zur Selbstregistrierung am IPO kommen externe Nutzer und Universitätsangehörige bzw. Studierende in Frage, die sich durch Angabe einer Fricardnummer legitimieren. Das hat einerseits zur Folge, dass ein Teil der Nutzer weder einen Bezug zur Fakultät, noch zur Universität hat, andererseits kann jeder Nutzer durch Angabe einer beliebigen, gültigen Fricardnummer einen privilegierten Zugang erhalten. Es existieren weder Abfragen, noch Prüfsummen, mit denen eine Fricardnummer verifiziert werden könnte.

### 7.3.2 Problem

Zum IPO existiert eine automatische Registrierung der Nutzer, die administrative Eingriffe im Rahmen der Registrierung vermeiden soll. Da dieser Registrierungsvorgang tatsächlich ohne Probleme auch hohe Belastungen verarbeiten konnte, ist also vor diesem Hintergrund eine Anbindung an den Studentenpool nicht erforderlich und würde nur einen geringen Nutzen bewirken. Allerdings ist mit dem derzeitigen Registrierungsvorgang das Problem verbunden, dass Studierende und Angehörige der Universität nicht sicher identifiziert werden können, da gültige Matrikelnummern mangels Prüfverfahren sehr leicht erraten werden können. Vor diesem Hintergrund könnte durch eine Anbindung an den Studentenpool die Identifikation der Studierenden verbessert werden. Allerdings ist auch hier zu beachten, dass nicht alle Studierenden über einen Zugang zum Studentenpool verfügen und dass die Einrichtung eines solchen Zugangs für Studierende, die die Vorlesung nur im Nebenfach belegen, vergleichsweise aufwendig ist.

### 7.3.3 Erinnerung: IPO-Nutzerverwaltung

Das IPO verwendet ein eigenes Registrierungsmodul, über das sich Nutzer eigenständig registrieren können. Dabei werden Vorname, Nachname, E-Mail-Adresse und, sofern vorhanden, eine Fricardnummer abgefragt. Diese Daten durchlaufen eine kurze Plausibilitätsprüfung und die E-Mail-Adresse wird durch eine Test-E-Mail überprüft. Anschließend wird der Nutzerzugang aktiviert und dem Nutzer werden, abhängig davon, ob eine Fricardnummer angegeben wurde, Rollen im IPO zugewiesen. Ein Login wird dem Nutzer automatisch zugewiesen und durch einen Synchronisationsprozess zwischen der LDAP-Nutzerverwaltung und der parallel betriebenen BSCW-Nutzerverwaltung abgeglichen.

### 7.3.4 Realisierungsmöglichkeiten der Nutzerverwaltung

Da weder der Studentenpool noch die Nutzerverwaltung der Fakultät für Informatik in ihrer aktuellen Form als alleinige Quellen für Nutzerdaten ausreichend sind, ist entweder eine Anpassung dieser Nutzerverwaltungen oder eine zusätzliche IPO-eigene Nutzerverwaltung erforderlich. Ziel ist die Verwaltung externer Nutzer, die keinen Bezug zur Fakultät für Informatik haben.

Eine Möglichkeit wäre die Erweiterung der geplanten LDAP-basierten Nutzerverwaltung der Fakultät für Informatik um einen speziellen Bereich für unprivilegierte Nutzer, deren Registrierung nicht verifiziert wurde. Dies ist bisher jedoch nicht vorgesehen und die dazu erforderliche Suche über mehrere LDAP-Bereiche wird gegenwärtig zwar von BSCW, nicht jedoch von Jetspeed unterstützt (siehe 4.4.3). Damit scheidet diese Realisierung vorerst aus.

Auch abseits von LDAP wird eine parallele Nutzerverwaltung in mehreren Datenquellen zwar beispielsweise von BSCW unterstützt, doch trifft das ebenfalls nicht auf Jetspeed zu. Daher wäre entweder eine Anpassung von Jetspeed an die neuen Erfordernisse nötig, oder es müsste ein Weg gefunden werden, gegenüber Jetspeed lediglich eine Nutzerverwaltung zu verwenden.

Diese Nutzerverwaltung könnte über einen regelmäßigen Synchronisationsmechanismus mit den Daten der Fakultätsverwaltung bzw. des Studentenpools abgeglichen werden und zusätzliche IPO-Nutzer verwalten.

Im einfachsten Fall könnte ein Studierender, analog zur Diskussion über Webinscribe, über eine Webseite die Einwilligung zur Übertragung von Daten des Studentenpools an das IPO erteilen. In der Folge könnte ein IPO-Zugang mit den aus dem Studentenpool bekannten Informationen angelegt werden, der auch über die nötigen, gemeinsamen Schlüsselinformationen zur regelmäßigen Synchronisation verfügt. Regelmäßig könnten dann aus dem Studentenpool die IPO-Informationen, wie beispielsweise das aktuelle Passwort, aktualisiert werden. Nutzer, die sich direkt beim IPO registrieren, würden bei einer solchen Synchronisation ignoriert, sodass eine Registrierung auch für jene Nutzer möglich ist, die keinen Bezug zum Studentenpool haben.

Wie auch beim Beispiel Webinscribe können jedoch auch hier Konflikte entstehen, wenn zwei Nutzerverwaltungen unsynchronisiert parallel betrieben werden. So wäre es theoretisch möglich, dass die Logins, die in den Nutzerverwaltungen als Schlüssel dienen, kollidieren.

Beispielsweise könnte ein Nutzer versuchen, einen IPO-Zugang mit einem Login anzulegen, der im Studentenpool bereits vergeben ist. Auch wenn ein IPO-Zugang mit einem freien Login angelegt wird, kann später eine Kollision entstehen, wenn ein Nutzer mit identischem Login im Studentenpool angelegt wird.

Daher würden alle Vorgänge, über die ein Nutzer in einer der beteiligten Verwaltungen angelegt werden kann, Synchronisationsmechanismen benötigen, um Kollisionen erkennen und ggf. auflösen zu können. Folglich müssten mehrere Prozesse überprüft und an die neuen Anforderungen angepasst werden.

Derartige Probleme können im Webinscribe-Szenario vergleichsweise einfach und zuverlässig behandelt werden, insbesondere da über die Matrikelnummer ein gemeinsamer Schlüssel zwischen Webinscribe und Studentenpool existiert. Dies wäre bei einer Verbindung von IPO und Studentenpool jedoch nicht der Fall, da das IPO eine Fricardnummer und der Studentenpool die Matrikelnummer als Schlüssel verwendet, die sich nur mithilfe der Universitätsverwaltung aufeinander abbilden lassen.

Damit ist die technische Realisierung einer Koppelung von IPO und Studentenpool nach derzeitigem Stand wesentlich aufwendiger, als im Fall von Webinscribe und Studentenpool.

### 7.3.5 Juristische Einschränkungen

Die bei Webinscribe durchgeführte Analyse bzgl. Datenschutz und Einwilligung der Studierenden (siehe 7.2.5) gilt auch bei diesem Beispiel und soll daher nicht wiederholt werden. Es sei nur daran erinnert, dass eine Zustimmung der Studierenden zur Nutzung der persönlichen Daten, die zweckgebunden zur Verwendung im Rahmen des Studentenpools erfasst wurden, im Kontext des IPO eingeholt werde muss und dass diese Zustimmung vergleichsweise einfach über ein Webformular erfolgen kann.
Allerdings ändert sich im Fall des IPO die Betrachtung bzgl. des Koppelungsverbots. Sollte durch die Nutzung des Studentenpools oder der Nutzerverwaltung der Fakultät für Informatik die Rollenvergabe innerhalb des IPO derart verändert werden, dass nur noch über diesen Weg eine sichere Anmeldung als Studierender zum IPO möglich ist, so würde möglicherweise das Koppelungsverbot verletzt, da eine privilegierte Nutzung des IPO einen Zugang zum Studentenpool voraussetzen würde. Rechtlich müsste ein alternativer Zugangsweg verfügbar sein, über den Studierende einen gleichwertigen Zugang einrichten können. Damit wäre aber der Vorteil einer Koppelung an den Studentenpool nicht mehr gegeben.

### 7.3.6 Fazit

Da die Anmeldung der Nutzer zum IPO bereits jetzt automatisch und ohne Eingriff der betreuenden Mitarbeiter erfolgt, kann das Verfahren durch Anbindung an den Studentenpool oder die Nutzerverwaltung der Fakultät nur unwesentlich verbessert werden. Eine verbesserte Autorisierung und Identifikation der Studierenden wäre zwar möglich, aufgrund des Koppelungsverbots jedoch rechtlich bedenklich.
Daher bleibt leider festzuhalten, dass der Nutzen, der dem vergleichsweise hohen Aufwand (Anpassung von Jetspeed oder Entwicklung von Synchronisationsprozessen) gegenübersteht, vergleichsweise gering ist, sodass sich eine Koppelung vermutlich nicht lohnen würde.

## 7.4 Automatische Generierung von Nutzerdaten für den Studentenpool

Bisher haben wir drei Systeme identifiziert, bei denen sich die Studierenden an der Fakultät für Informatik zeitnah zu Beginn des Studiums registrieren können bzw. müssen. Durch eine Anbindung von Webinscribe an den Studentenpool kann der damit verbundene Aufwand reduziert werden, im Fall des IPO wurde die Anmeldung schon jetzt weitgehend automatisiert. Lediglich für die Anmeldung der Studierenden zur Nutzung des Studentenpools existiert noch keine einfache Lösung, mit der der große Aufwand, der für Studierende und Mitarbeiter anfällt, reduziert werden könnte. Deshalb soll dieses Problem im Folgenden näher untersucht werden.
Um das Problem der Anmeldung großer Nutzerzahlen zu bewältigen, haben wir für das IPO eine automatische Selbstregistrierung der Nutzer gewählt. Basierend auf den damit verbundenen Erfahrungen werden wir zuerst untersuchen, ob eine Selbstregistrierung der Nutzer für den Studentenpool eine wirksame Alternative darstellen kann.
Für das IPO wurde eine Webanwendung zur Registrierung entwickelt, mit der Daten von den Nutzern abgefragt werden und anschließend die Email-Adresse durch eine Test-Email verifiziert wird (siehe [Ha04]). Die übrigen Informationen werden lediglich einer einfachen Prüfung unterzogen, um grobe Abweichungen erkennen zu könne. Da nach einer Registrierung zur IPO-Nutzung allenfalls Rechte zum Lesen und Schreiben in einem Diskussionsforum sowie zum lesenden Zugriff auf eine Materialsammlung eingeräumt werden, sind die Auswirkungen eines möglichen Missbrauchs sehr gering

und daher die schwache Datenprüfung und der Verzicht auf eine Form von elektronischer Signatur in diesem Umfeld auch akzeptabel.
Im Umfeld des Studentenpools sind jedoch höhere Ansprüche an der Qualität der Registrierung zu stellen. Insbesondere sind mit der Nutzung erhebliche Missbrauchsmöglichkeiten verbunden, die auch unbeteiligte Dritte betreffen können, weshalb eine gültige Zustimmung zur Benutzerordnung der Fakultät für Informatik eine unverzichtbare Nutzungsvoraussetzung darstellt. Um eine gültige Zustimmung zu erhalten müssen die erfassten Daten verifiziert und eine Signatur durchgeführt werden können. Eine Registrierung, die diesen Ansprüchen nicht genügt, kann im Rahmen des Studentenpools nicht eingesetzt werden, sodass eine Adaption der IPO-Lösung hier ausscheidet.
Es muss also eine Lösung gefunden werden, mit der man die Daten eines Studierenden verifizieren und eine elektronische Signatur durchführen kann. Zur automatischen Überprüfung der Daten müsste auf einen bereits existierenden Datenbestand zugegriffen werden, der bereits die Studierendendaten in einer sorgfältig überprüften Fassung enthält. Über einen gemeinsamen Schlüssel, beispielsweise die Matrikelnummer oder die Fricardnummer könnte man die von den Studierenden bei der Registrierung angegebenen Daten mit den bereits erfassten Daten vergleichen. Leider existiert an der Fakultät für Informatik kein geeigneter Datenbestand, auf den zu diesem Zweck zugegriffen werden könnte und die Schaffung eines solchen Datenbestandes würde die erforderliche Datenerfassung, die insgesamt erforderlich ist, zwar für die Registrierung zur Nutzung des Studentenpools vereinfachen, das Problem und den Arbeitsaufwand jedoch nur auf einen anderen Datenbestand verlagern.
Zwar existiert aufseiten der Fakultät für Informatik kein geeigneter Datenbestand, doch werden alle Studierenden bei der Immatrikulation von der Universitätsverwaltung erfasst und der Datenbestand wird kontinuierlich während des Studiums weiter gepflegt. Deshalb soll die Studierendenverwaltung der Universität an dieser Stelle näher betrachtet werden.

### 7.4.1 Studierendenverwaltung der Universität

Die zentrale Verwaltung der Universität erfasst die Daten aller Studierenden bei deren Immatrikulation und verwaltet sie in einem zentralen Werkzeug „HIS-SOS“ (Student Operating System der HIS GmbH). Dabei handelt es sich um Daten wie Vorname, Nachname, Anschrift, Studienrichtung, Zahl der Fachsemester, etc.
Die Studierenden sind verpflichtet, der Verwaltung Änderungen, die sich im Lauf des Studiums ergeben, mitzuteilen und die Daten zu aktualisieren. Dazu kann seit kurzer Zeit auch ein spezielles Portal mit „Selbstbedienungsfunktionen für die Studierende- und Prüfungsverwaltung“, kurz Selbstbedienungs-Portal (SB-Portal), für die Studierenden verwendet werden.
Über die Webseite https://sb.zvw.uni-karlsruhe.de können die Studierenden im Rahmen einer Selbstverwaltung verschiedene Aktionen durchführen und beispielsweise die Änderung der Postanschrift oder den Ausdruck von Studienbescheinigungen veranlassen.
Der Zugang zu diesem SB-Portal wird über die Matrikelnummer als Login und ein Passwort, das den Studierenden von der Universitätsverwaltung zugewiesen wurde, authentifiziert. Einige Aktionen innerhalb des SB-Portals wurden außerdem über Transaktionsnummern (TAN) geschützt, die die Studierenden ebenfalls von der Universitätsverwaltung erhalten.
Die erforderlichen Zugangsdaten (Matrikelnummer, Passwort und eine TAN-Liste) werden den Studierenden kurzfristig nach der Immatrikulation schriftlich mitgeteilt.

Damit verfügen die Studierenden schon frühzeitig über Zugang zu einem verifizierten Account an der Universität Karlsruhe (TH).
Während wir bereits den Fall analysiert haben, dass die im Studentenpool vorhandenen Daten auch von anderen Diensten wie Webinscribe genutzt werden können, haben wir nun den Fall, dass Daten, die in einem anderen Dienst erfasst sind, vom Studentenpool genutzt werden könnten Ob eine Verbindung dieser Datenbestände im Rahmen eines *Federated Identity Management* ermöglicht werden kann, wird daher im Folgenden untersucht.

### 7.4.2 Verbindung der Nutzerverwaltungen von Fakultät und Universität

Wie auch schon im Beispiel der Anbindung von Webinscribe an den Studentenpool ist es zuerst erforderlich, von den betroffenen Studierenden eine Zustimmung zur Weitergabe von Nutzerdaten einzuholen. Diese Zustimmung muss von den Studierenden gegenüber der Datenquelle, also in diesem Fall gegenüber der Universitätsverwaltung, ausgesprochen werden. Um eine elektronische Zustimmung erfassen zu können, ist es daher erforderlich, dass seitens der Universitätsverwaltung eine geeignete Webanwendung entwickelt wird, bei der sich die Studierenden anmelden und über ein Formular einer Datenübertragung von der Universitätsverwaltung an den Betreiber des Studentenpools zuzustimmen. Ein solches Formular muss eindeutig gestaltet sein, die Art und der Zweck der Übertragung müssen für den Studierenden ersichtlich sein.
Da ohnehin eine Anmeldung der Studierenden am SB-Portal der Universitätsverwaltung erforderlich ist, können hier gleichzeitig weiter Informationen erhoben werden, beispielsweise eine Kontakt-Emailadresse oder ein gewünschtes Passwort. Insbesondere kann, deutlich im Formular gekennzeichnet, auch die Zustimmung zur Benutzerordnung der Fakultät für Informatik eingeholt werden. Diese Zustimmung wird gegenwärtig im Rahmen der Benutzeranmeldung durch eine handschriftliche, rechtsgültige Unterschrift erteilt. In dem „Gesetz über Rahmenbedingungen für elektronische Signaturen“, dem so genannten Signaturgesetz (SigG) werden die „einfache“, die „fortgeschrittene“ und die „qualifizierte“ elektronische Signatur definiert, von denen die qualifizierte elektronische Signatur geeignet ist, im Rahmen elektronischer Kommunikation die handschriftliche Unterschrift zu ersetzen.. Aufgrund der vorliegenden Rahmenbedingungen kommt an diese Stelle jedoch nur eine „einfache elektronische Signatur“ in Frage, da die für eine fortgeschrittene oder gar qualifizierte Signatur erforderliche Zertifizierung der Studierenden nicht vorausgesetzt werden kann. Eine einfache Signatur kann bereits durch ein Textfeld realisiert werden, in das ein Nutzer Vornamen und Nachnamen einträgt, und hat daher alleine praktisch keinerlei Aussagekraft und Wirksamkeit.
Angesichts der Anforderungen der Fakultät für Informatik an die Zustimmung zur Nutzerordnung ist eine einfache elektronische Signatur daher unzureichend. Die fehlende Aussagekraft der einfachen elektronischen Signatur kann aber zum Teil kompensiert werden, da die angegebenen Daten, und damit die einfache elektronische Signatur, nur dann im SB-Portal erfasst werden können, wenn zuvor eine Anmeldung mit Matrikelnummer und zugehörigem Passwort durchgeführt wurde und die Zustimmung zusätzlich mit einer Transaktionsnummer bestätigt wurde. Dieser Schutzmechanismus bewirkt ein wesentlich höheres Gewicht der elektronischen Signatur.
Werden nun also die Nutzerdaten (Vorname, Nachname, Matrikelnummer, etc.), ein gewünschtes Passwort, die Zustimmung zur Benutzerordnung der Fakultät für Informatik und eine elektronische Signatur erfasst und, mit einer TAN bestätigt, an die Fakultät für Informatik übermittelt, so kann vollautomatisch ein neuer Zugang zum Studentenpool für den jeweiligen Studierenden angelegt werden.

### 7.4.3 Koppelungsverbot

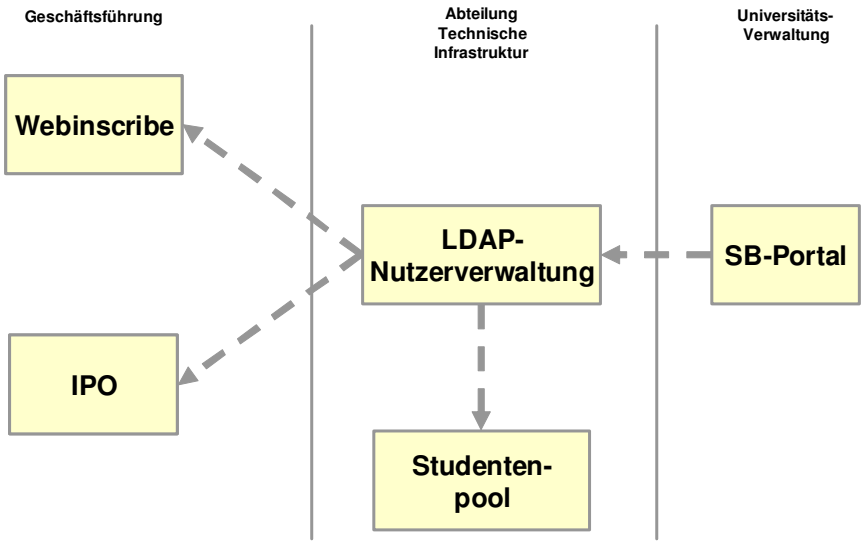
Auch in diesem Fall ist das Koppelungsverbot zu berücksichtigen und auch hier liegt möglicherweise durch die Verbindung der Zustimmung zur Datenübermittelung gegenüber der Universitätsverwaltung und der Zustimmung zur Benutzerordnung der Fakultät für Informatik eine unzulässige Koppelung vor. Aber wie bereits im Fall von Webinscribe diskutiert (siehe 7.2.5), kommt das Koppelungsverbot nicht zur Anwendung, wenn eine zumutbare Alternative existiert. Da die Studierenden auch ohne elektronische Datenübermittelung einen Zugang zum Studentenpool erhalten können, indem sie wie bisher persönlich bei den betreuenden Mitarbeitern erscheinen und einen entsprechenden Antrag ausfüllen, ist eine solche Alternative gegeben und das Koppelungsverbot findet keine Anwendung. Allerdings müssen die Studierenden deutlich im SB-Portal der Universitätsverwaltung auf diese Alternative hingewiesen werden.

### 7.4.4 Ergebnis

Es wurde eine Lösung erarbeitet, mit der sich die Studierenden automatisch als Nutzer für den von der Fakultät für Informatik bereit gestellten Studentenpool anmelden können. Als Voraussetzung müssen diese Studierenden zuvor über einen Zugang zum von der Universitätsverwaltung betriebenen SB-Portal verfügen, was aber für alle ordentlich immatrikulierten Studierenden vorausgesetzt werden kann.
Wir konnten auch die rechtlichen Rahmenbedingungen bzgl. Datenschutz und elektronischer Signatur untersuchen und einen Lösungsansatz finden, unter dem eine Datenübertragung zulässig ist und gleichzeitig eine Zustimmung zur Nutzerordnung der Fakultät für Informatik erteilt werden kann. Dieser Ansatz basiert auf den im SB-Portal verwendeten Passwörtern und Transaktionsnummern, die eine hohe Sicherheit gewährleisten können.
Geklärt wurde außerdem, welche Daten aufseiten der Universitätsverwaltung vorliegen und an die Fakultät für Informatik übertragen werden können. Weitere Daten, wie z. B. ein gewünschtes Passwort, können außerdem im Rahmen der Registrierung abgefragt und ebenfalls übertragen werden.
Nicht untersucht wurden an dieser Stelle die technischen Aspekte einer Datenübertragung, die durch die Studierenden selbst über das SB-Portal initiiert wird. Offen sind insbesondere die genaue Integration einer Registrierungsfunktion in den Portalkontext, sowie die Art und Weise der Datenübertragung. Denkbar wäre etwa eine periodische Datenübertragung in einem definierten Format oder die unmittelbare Anbindung über Schnittstellen wie z. B. Webservices. In beiden Fällen steht eine Spezifikation der zu übertragenden Datensätze und Datenformate noch aus.
Um den Aufwand an der Fakultät zu reduzieren, kann über eine weitere Automatisierung der Verwaltungsprozesse im Rahmen der Nutzerverwaltung nachgedacht werden. Da der größte personelle Aufwand bei der Erfassung und Überprüfung von Nutzerdaten anfällt, ist zu überlegen, ob auch in diesem Bereich Anknüpfungspunkte zur Universitätsverwaltung existieren.

## 7.5 Fazit

Wir haben die verschiedenen Datenbestände im Umfeld der Fakultät für Informatik untersucht, in denen ein Studierender erfasst werden kann und drei verschiedene Fälle analysiert, in denen ein Federated Identity Management zum Einsatz kommen kann.
Ausgehend von Webinscribe haben wir Fragestellungen bearbeitet, die die Betreibersicht, die technische Umsetzbarkeit und juristische Aspekte wie den Datenschutz umfassen.




**Information 41: FIM an der Fakultät für Informatik**

Webinscribe stellt einen vergleichsweise einfachen Fall dar, bei dem mit geringem Aufwand ein großer Nutzen erzielt werden kann. Zwar treten schon hier erste Bedenken auf, da der parallele Einsatz zweier Nutzerdatenverwaltungen erforderlich ist und Konflikte berücksichtigt werden müssen, doch sind die damit verbundenen Erfordernisse einfach zu erfüllen. Das IPO hingegen erscheint zwar ähnlich, stellt jedoch sowohl technisch aufgrund der komplexeren Nutzerverwaltung als auch rechtlich ein wesentlich größeres Problem dar. Zum einen ist es augrund fehlender gemeinsamer Schlüssel nicht möglich, Konflikte der beteiligten Nutzerverwaltungen zu erkennen und automatisch zu behandeln, zum anderen entstehen datenschutzrechtliche Probleme, falls der Studentenpool zur Identifikation der Studierenden eingesetzt wird. Verzichtet man allerdings auf diese Anwendung, so entsteht durch die Anbindung an den Studentenpool kein Vorteil, da schon jetzt durch die automatische Registrierung eine effektive Nutzerverwaltung gegeben ist.
In beiden Fällen wurde die Existenz von Studierendendatensätzen im Studentenpool stillschweigend vorausgesetzt, wodurch schließlich die Frage aufgeworfen wurde, wie dieser Datenbestand effektiv gepflegt werden kann. In unserer Untersuchung haben wir festgestellt, dass große Vorteile entstehen, wenn im Rahmen eines *Federated Identity Management* die Universitätsverwaltung einbezogen und das SB-Portal für die Studierenden genutzt werden kann.
Insgesamt haben wir so einen Pfad aufgezeigt, über den die Nutzerdaten von der Universitätsverwaltung an die Fakultät für Informatik übertragen werden und dort weiteren Diensten zur Verfügung gestellt werden können. Dabei überschreiten wir gleich zweimal die Grenzen von Geschäftsbereichen und haben daher nicht nur eine, sondern gleich mehrere Anwendungen von *Federated Identity Management*.

# 8 AUSWIRKUNGEN AUF BETRIEBSABLÄUFE

Ein Ziel dieser Arbeit war die Untersuchung, welche Optimierungsmöglichkeiten durch einen effektiven Einsatz von *Identity Management* an der Fakultät für Informatik bestehen und wie sich auf diese Weise die existierenden Betriebsabläufe optimieren lassen. Die einzelnen Bereiche, in denen diese Arbeit spürbare Auswirkungen auf die existierenden oder geplanten Betriebsabläufe haben kann, sollen im Folgenden zusammengefasst werden.

## 8.1 Verwaltung IPO

Während die Vorlesung Kommunikation und Datenhaltung (K&D) noch mit einer manuellen Registrierung der Nutzer durchgeführt wurde, war ein solcher Betrieb im Rahmen der INFORMATIK-I-Vorlesung nicht mehr möglich, da die Studierendenzahlen von inzwischen über 650 registrierten Teilnehmern einen zu hohen personellen Aufwand erfordert hätten. Deshalb wurde in dieser Arbeit ein Konzept zur Verwaltung der Teilnehmer entwickelt, das Standardfälle, wie das Anlegen und Verifizieren von Nutzern oder Hilfestellungen bei vergessenen Passwörtern, automatisch durchführen kann. Verglichen mit den bei K&D erforderlichen manuellen Arbeiten konnten über diese Automatismen erhebliche Effizienzsteigerungen realisiert werden, die überdies durch weitere Funktionen wie beispielsweise eine automatische Protokollierung der durchgeführten Aktionen weitere Verbesserungen der Servicequalität bewirken konnten.

## 8.2 Zentrale Nutzerverwaltung innerhalb der Fakultät

Zur zentralen Nutzerverwaltung innerhalb der Fakultät für Informatik wurden bisher verschiedene Werkzeuge verwendet, die über parallele, unsynchronisierte Nutzerverwaltungen gesteuert wurden. Aufgrund der fehlenden Synchronisation waren die Verwaltungsvorgänge weder effizient, da Nutzer manuell in mehreren Verwaltungswerkzeugen angelegt werden mussten, noch ausreichend konsistent, da das Löschen eines Nutzereintrags im Useradm nicht automatisch die Deaktivierung sämtlicher abhängiger Dienste, wie beispielsweise VPN oder Dial-In, bewirken konnte.
Durch die Zusammenführung der bisher getrennten Datenbestände in einem zentralen Verwaltungswerkzeug werden nicht nur Inkonsistenzen vermieden und so die Aktualität und die Sicherheit der verwalteten Nutzerdaten wesentlich verbessert. Darüber hinaus werden die Prozesse zur Dienstverwaltung auch auf das Wesentliche reduziert und dadurch vereinfacht.
Die Zusammenführung in einem Datenbestand erleichtert schließlich auch die Zusammenführung der bisher getrennten Verwaltungsoberflächen, sodass einheitliche Oberflächen und einheitliche Verwaltungsabläufe ermöglicht werden.

## 8.3 Webinscribe

Die bisherige Passwort-Zuteilung für Nutzer von Webinscribe kann durch die aufgezeigte Anbindung an die Studierendendaten der Fakultät für einen Großteil der Studierenden vermieden werden, sodass der personelle Aufwand zur Durchführung der Tutorienvergabe erheblich reduziert werden kann. Auch die Studierenden profitieren von den vereinfachten und automatisierten Abläufen, sofern sie zuvor Ihre Einwilligung erteilt haben. Allerdings kann das Problem nicht vollständig gelöst werden, da nur ein Teil der betroffenen Studierenden an der Fakultät für Informatik eingeschrieben ist, sodass ein alternativer Mechanismus weiterhin betrieben werden muss.

## 8.4 Studentenpool

Der bisherige Ablauf einer Anmeldung zur Nutzung des Studentenpools umfasste das persönliche Erscheinen des Studierenden zu den Sprechzeiten der betreuenden Mitarbeiter, die Überprüfung des Antragsformulars anhand des Studierendenausweises sowie die manuelle Datenerfassung in der Nutzerverwaltung des Studentenpools.
Dieser Ablauf kann nun durch die Koppelung der Anmeldung an das SB-Portal der Universitätsverwaltung erheblich vereinfacht werden, indem der Studierende dort einer Datenübermittelung zustimmt und vollautomatisch einen Zugang zum Studentenpool erhält. Von diesem verbesserten Betriebsprozess profitieren nicht nur die Mitarbeiter, die von Routinetätigkeiten entlastet werden, sondern auch die Studierenden in Form einer schnelleren Bearbeitung und insgesamt höherer Dienstqualität, da beispielsweise Tippfehler im Rahmen der manuellen Datenerfassung vermieden werden können.

# 9 ZUSAMMENFASSUNG UND AUSBLICK

Diese Arbeit hat das Ziel verfolgt, Fragestellungen des *Identity Management* an der Fakultät für Informatik zu untersuchen. Dabei wurde zuerst mit der Untersuchung des eng umrissenen IPO-Szenarios eine Basis geschaffen, auf der die Analyse des *Identity Management* an der Fakultät aufbauen konnte. Im Bereich des IPO konnten die entwickelten Konzepte bereits erfolgreich in den Wirkbetrieb überführt werden und wird auch für künftige Erweiterungen eine solide Basis bilden.

An der Fakultät für Informatik konnten schon wesentliche Komponenten, wie die Synchronisation des aktuellen Useradm mit einem LDAP-Verzeichnis und erste Verwaltungswerkzeuge, implementiert werden. Damit wird schon jetzt die schrittweise Migration auf ein neues System ermöglicht, indem die einzelnen Dienste sukzessive auf die Nutzung des zentralen LDAP-Verzeichnisses umgestellt werden. Diese Umstellung ist bereits geplant und wird bald begonnen werden.

Im Bereich des *Federated Identity Management* konnte noch nicht mit einer Umsetzung begonnen werden, da diese Schritte die künftige Nutzerverwaltung der Fakultät für Informatik voraussetzen. Auch sind wichtige Fragestellungen noch ungeklärt und müssen näher untersucht werden. Beispielsweise wurde die technische Umsetzung einer Koppelung von Universitätsverwaltung und Nutzerverwaltung der Fakultät noch nicht näher betrachtet. Die nötige Untersuchung wird einerseits die Anforderungen der Schnittstellen des SB-Portals untersuchen müssen, andererseits auch einen Mechanismus zur sicheren Datenübertragung an die Fakultät für Informatik erarbeiten müssen. Dazu müssen mögliche Übertragungsprotokolle untersucht und Datenformate spezifiziert werden. Denkbar wären sowohl periodische, gesammelte Übertragung der erfassten Daten als auch direkte Zugriffe, beispielsweise direkt über LDAP oder über einen zwischengeschalteten Webservice, der weitere Funktionen wie eine sichere Protokollierung implementieren könnte. Der Umfang dieser Arbeiten rechtfertigt sicherlich die Ausschreibung einer eigenen Studienarbeit.

Leider wurden im Rahmen dieser Arbeit auch Probleme aufgedeckt, die derzeit noch ungelöst sind. Beispielsweise existieren Anwendungen, die dauerhaft parallel zu einer LDAP-Anbindung einen eigenen Nutzerbestand pflegen müssen. BSCW etwa könnte ohne eigenen Nutzerbestand keinen Einladungsmechanismus realisieren, der an der Fakultät für Informatik als unverzichtbar angesehen wird. Zwar erlaubt BSCW keine Logins in der eigenen Nutzerverwaltung, die bereits in einem LDAP-Datenbestand vergeben sind, doch führt die LDAP-Nutzerverwaltung keine entsprechenden Prüfungen durch und es kann zu Konflikten durch doppelt vergebene Logins kommen.

Insbesondere die Diskussion des *Federated Identity Management* steht noch am Anfang. Wir haben jedoch eine Basis für die weitere Analyse geschaffen, auf der sich Themen wie die *Security Assertion Markup Language* (SAML) oder die *Web Service Federation Language* (WSFL) intensiv untersuchen lassen.

# VERZEICHNISSE

## Abkürzungen und Glossar

| Abkürzung oder Begriff | Langbezeichnung und/oder Begriffserklärung |
|---|---|
| 802 | Gruppe von Standards der IEEE, die sich mit Computer-Netzwerken auf Schicht 1 und Schicht 2 befassen.. |
| 802.1x | Standard zur Authentifizierung in Computernetzwerken nach IEEE-802-Standard. |
| Active Directory | Zentrales Verzeichnis mit Verwaltungsfunktionen für Mircosoft Windows 2000 und Windows XP. |
| ATIS | Abteilung Technische Infrastruktur<br>Einrichtung der Fakultät für Informatik, die für den Betrieb der technischen Infrastruktur (beispielsweise Netzwerk, Mailsystem, etc.) verantwortlich ist. |
| Authentifikation | Eindeutige Identifikation eines Nutzers, beispielsweise durch Angabe eines Logins und eines zugehörigen Passworts. |
| Autorisation | Basierend auf einer erfolgreichen Authentifikation werden einem Nutzer Rechte zugewiesen. |
| BSCW | Basic Support for Cooperative Work<br>System zur Bereitstellung einer Kollaborationsplattform, wird im Rahmen des IPO zur Verwaltung von Materialien und Diskussionsforen verwendet. |
| C&M | Cooperation & Management<br>Name der an der Universität Karlsruhe (TH) angesiedelten Forschungsgruppe. |
| Dial-In | Aufbau einer Einwahlverbindung, um auf ein Computernetzwerk zuzugreifen. In der Regel wird als Protokoll PPP verwendet. |
| Distinguished Name | Global eindeutiger Bezeichner für einen Eintrag in einem LDAP-Verzeichnis. |
| DN | Siehe *Distinguished Name* |
| EAP | *Extensible Authentication Protocol*<br>Erweiterung der Authentifikationsfunktionen des PPP-Protokolls, kann beispielsweise von 802.1x eingesetzt werden. Spezifiziert in RFC 2284. |

| | |
|---|---|
| Einwege-Hash-Funktion | Kryptographische Funktion, um aus einer Eingabe beliebiger Länge eine Zeichenkette definierter Länge zu generieren. Dabei ist die Anwendung der Hash-Funktion effizient möglich, die Umkehrung jedoch nicht.<br>Statt ein Passwort im Klartext zu speichern, kann auch das Ergebnis der Anwendung einer Hash-Funktion gespeichert werden, sodass ein Angreifer keinen Zugriff auf die Original-Passwörter erhalten kann. Zur Passwort-Prüfung ist dann jeweils die Anwendung der Hash-Funktion auf die Eingabe erforderlich. |
| FAQ | *Frequently Asked Questions*<br>*Supportdokument, das die Antworten zu häufig gestellten Fragen beinhaltet.* |
| Fricard | Bezeichnung für den an der Universität Karlsruhe (TH) eingesetzten Studierendenausweis. |
| Fricardnummer | Eindeutige Identifikationsnummer einer Fricard. Da ein Studierender, etwa bei Verlust, eine neue Fricard erhalten kann, ist diese Nummer variabel und daher zur dauerhaften Identifikation nicht geeignet. |
| HIS-SOS | Student Operating System der HIS GmbH<br>Wird an der Universität Karlsruhe (TH) zur Verwaltung der Studierendendaten eingesetzt. |
| HTML | *HyperText Markup Language*<br>Vom W3C standarisierte Auszeichnungssprache, die die Struktur und die Darstellung eines Dokuments beschreibt. |
| HTTP | *HyperText Transfer Protocol*<br>Internet-Anwendungsprotokoll, das innerhalb des World Wide Web benutzt wird. |
| Identity Management | Der Begriff Identitiy Management bezeichnet sowohl die Aufgabe, verteilte Nutzerdaten effizient und konsistent zu pflegen, als auch spezielle Systeme, die die Erfüllung dieser Aufgabe unterstützen sollen. |
| IEEE | *Institute of Electrical and Electronic Engineers*<br>Standarisierungsorganisation |
| IETF | *Internet Engineering Task Force*<br>Organisation, die maßgeblich am Standardisierungsprozess der Internet-Standards beteiligt ist. |
| INFORMATIK-I-Portal | INFORMATIK-I-Portal<br>Ein Webportal, das die INFORMATIK-I-Vorlesung, zusammen mit Übung und Tutorien, unterstützen soll. Es umfasst insbesondere Vorlesungsmaterialien und Diskussionsforen. |

| | |
|---|---|
| IP | *Internet Protocol*<br>Verbindungsloses Protokoll der Vermittlungsschicht. |
| IPO | Siehe INFORMATIK-I-Portal |
| IT-Dienst | Ein IT-Dienst ist ein immaterielles Produkt eines IT-Dienstleisters, das entweder auf IT basiert oder im Zusammenhang mit IT steht und dem Zweck dient, den Zustand des IT-Dienstnehmers unmittelbar zu verbessern. Der Zustand des Dienstnehmers bezieht sich dabei auf seinen Informationsstand, sein Wissen, seine wirtschaftliche Situation usw. Ein IT-Dienst zeichnet sich durch seine Funktionalität bzw. Leistung sowie durch eine Menge von Dienstmerkmalen aus.<br><br>A described set of facilities, IT and non-IT, supported by the IT Service Provider that fulfils one or more needs of the customer and that is perceived by the customer as a coherent whole. [ITIL-Glossar, 16.11.2004, www.serview.de/content/itsm/glossar]<br><br>Synonym: IT-Service, IT-Dienstleistung<br>Englisch: *IT Service* |
| ITU | *Internation Telecommunications Union*<br>Standarisierungsorganisation, die Festlegungen im Telekommunikationsbereich trifft (frühere CCITT). |
| Java | Objektorientierte Programmiersprache, entwickelt von Sun Microsystems. |
| JDBC | *Java Database Connectivity*<br>API, die in Java-Programmen zur Unterstützung des Zugangs zu Datenbanken genutzt wird. |
| Jetspeed | Bezeichnung für einen von der Apache Software Foundation entwickelten Portlet-Container. |
| JSR-168 | *Java Secification Request 168*<br>Beinhaltet die *Java Portlet Specification*, die als Standard von vielen aktuellen Portlet-Umgebungen unterstützt wird. |
| K&D | Kommunikation & Datenhaltung<br>Bezeichnung eines der insgesamt 8 Wahlpflichtfächer, die im Hauptstudium an der Fakultät für Informatik angeboten weren. |
| Kerberos | Am Massachusetts Institute of Technology (MIT) entwickeltes Protokoll zur sicheren Authentifikation in Computer-Netzwerken. Wird beispielsweise von MS Windows 2000 und Windows XP eingesetzt. |

| | |
|---|---|
| Komponente | Eine ausführbare und austauschbare Softwareeinheit mit definierten Schnittstellen und eigener Identität. |
| LDAP | *Lightweight Directory Access Protocol*<br>Ursprünglich Protokoll zum Zugriff über IP-Netzwerke auf Verzeichnisdienste nach X.500-Standard. Später wurden Eingenständige Verzeichnisdienste entwickelt, die den direkten Zugriff über LDAP ermöglichen. |
| LDoc | Living Document<br>Ein in Standard-HTML-Format vorliegendes Textdokument (Document) mit eingebundenen Grafiken, die durch Anklicken um (i. d. R. multimediales) Material ergänzt werden können und damit zum Leben (Living) erweckt werden. |
| Managementkonsole<br>Mcon | Eine webbasierte Plattform für das vereinheitlichte, zentrale Management von verteilten Anwendungen. Die Konsole lässt sich über Managementmodule funktional erweitern und auf betreiberspezifische Anforderungen anpassen. |
| Matrikelnummer | Eindeutige Identifikationsnummer, die einem Studierenden bei der Einschreibung zugewiesen wird und sich während des Studiums nicht verändert. |
| MS | Microsoft |
| NIS | *Network Information System*<br>Früher Standard von Sun Microsystems zur zentralisierten Nutzerverwaltung auf Unix-Systemen. |
| Nutzerverwaltung | Umfasst sowohl die Nutzerdatenbestände, als auch die zugehörigen Verwaltungswerkzeuge und –prozesse. |
| Pdok | Produktdokument<br>Ein gemäß einer vorgegebenen Struktur aufgebautes Dokument, das im Rahmen der bei C&M eingeführten strukturierten Softwareentwicklung erstellt wird und zur Dokumentation eines C&M-Softwareprodukts dient. |
| Portal | Rahmenwerk zur Realisierung eines Browser-basierten Frontends mit Personalisierungskonzept, rollenbasierter Zugriffskontrolle, einmaliger Benutzerauthentifizierung. Ermöglicht zudem die Integration auf Präsentationsebene. |
| Portlet | Komponente zum Einsatz in einem Portlet-Container. Realisiert eine Funktion für den Nutzer, beispielsweise ein Adressbuch oder den Zugriff auf einen Gruppenkalender. |
| Portlet-Container | Portal-Umgebung, die den Einsatz unterschiedlicher Portal-Komponenten (*Portlets*) ermöglicht, sofern diese einer gemeinsamen Spezifikation (z. B. JSR-168) entsprechen. Der |

| | |
|---|---|
| | Portlet-Container stellt Funktionen wie z. B. Authentifikation und *Single Sign On* zur Verfügung und ermöglicht dem Anwender den Zugriff auf *Portlets*. |
| PPP | *Point-to-Point Protocol*<br>Protokoll zum Verbindungsaufbau zu Computernetzen, wird insbesondere zur Nutzung von Wählleitungen verwendet. Spezifiziert in RFC 1661. |
| PPT | Powerpoint |
| Proxy-User | Nutzer, der ein System repräsentiert und stellvertretend für dieses System Zugriffsrechte in einem Verzeichnisdienst erhält. Das jeweilige System tritt über den Proxy-User gegenüber den Verzeichnisdienst auf. Synonyme: System DN und Service DN. |
| Python | Objektorientierte Script-Sprache |
| Radius | *Remote Authentication Dial-In User Service*<br>Protokoll zur Authentifizierung von Nutzern bei Einwahlverbindungen, wird beispielsweise im Zusammenhang mit PPP und VPN verwendet. |
| RDN | Siehe Relative Distinguished Name |
| *Relative Distinguished Name* | Eindeutiger Bezeichner für einen Eintrag in einem LDAP-Verzeichnis, bezogen auf den übergeordneten Vaterknoten. Einträge mit unterschiedlichem Vaterknoten können einen identischen RDN haben, die Konkatenation des RDN mit dem DN des Vaterknotens bildet den eigenen DN. |
| RFC | *Request for Comments*<br>Bezeichnung für Dokumente, die zum Standardisierungsverfahren der IETF eingereicht worden sind. |
| RPC | *Remote Procedure Call*<br>Client/Server-Protokoll, durch das der Client eine Prozedur auf einem entfernten Server aufrufen kann und das Ergebnis der Prozedurausführung zugestellt bekommt. |
| SASL | *Simple Authentication and Security Layer*<br>Rahmenwerk zur Authentifikation und Autorisierung in Internet-Protokollen. Ermöglicht insbesondere die verschlüsselte Datenübertragung und wird in RFC 2222 spezifiziert. |
| SB-Portal | Selbstbedienungfunktionen für Studierende<br>Portal der Universitätsverwaltung für die Studierenden, mit dem diese selbständig Verwaltungsvorgänge wie z. B. den Ausdruck von Studienbescheinigungen veranlassen können. |

| | |
|---|---|
| SNMP | *Simple Network Management Protocol*<br>Internet-Anwendungsprotokoll, das im Internet-Management zur Kommunikation von Managementinformation genutzt wird. |
| SOAP | *Simple Object Access Protocol*<br>Auf XML basierendes, vielseitig verwendbares Anwendungsprotokoll. |
| SQL | *Structured Query Language*<br>Deklarative Sprache zur Durchführung von Datenbankoperationen. Herstellerübergreifender Standard (z. B. ISO). |
| SQL-Datenbank | Datenbank, die Anfragen nach SQL-Standard verarbeitet. |
| SSL | *Secure Sockets Layer*<br>Mechanismus zur Authentifikation und Verschlüsselung von Netzwerkverbindungen nach TCP/IP-Standard. Vorgänger von TLS. |
| System-DN | Siehe Proxy-User. |
| TAN | Transaktionsnummer<br>Einmalig zu verwendende Nummer zur Autorisierung eine Transaktion. Die TAN muss dem Nutzer zuvor auf einem sicheren Kommunikationsweg mitgeteilt worden sein und wird häufig eingesetzt, wenn ein einfacher Passwort-Schutz keine ausreichende Sicherheit bietet. |
| TCP | *Transmission Control Protocol*<br>Ein im Internet verwendetes Schicht4-Protokoll, das einen verbindungsorientierten Transportdienst bereitstellt. |
| TLS | *Transport Layer Security*<br>Mechanismus zur Authentifikation und Verschlüsselung von Netzwerkverbindungen nach TCP/IP-Standard. Nachfolger von SSL, spezifiziert in RFC 2246. Anders als SSL ein Standard der IETF. |
| Virtual Private Network | Ein entfernter Rechner erhält über einen (i. d. R. verschlüsselten) Tunnel Zugriff in ein internes bzw. privates Netzwerk und wird mit den gleichen Rechten ausgestattet, wie ein lokaler Rechner. |
| VLAN | Über eine Erweiterung auf Schicht 2 werden aktive Komponenten wie *Switches* oder *Router* in die Lage versetzt, in einem physikalischen Netz mehrere logische Netze zu unterscheiden. Die Kommunikation zwischen diesen Netzen ist nur mit dem Einsatz von *Routern* möglich, häufig werden *Firewalls* zwischen unterschiedlichen logischen Netzen eingesetzt. |

| | |
|---|---|
| VPN | Siehe *Virtual Private Network* |
| W3C | *World Wide Web Consortium*<br>Organisation, durch die mit dem Web verbundene Technologien und Konzepte standardisiert werden. |
| WebDAV | *Web-based Distributed Authoring and Versioning*<br>Ein Protokoll der IETF, dass es den Nutzern erlaubt, Dateien auf entfernten Web-Servern kollaborativ zu editieren und zu verwalten. |
| Webinscribe | Eine Anwendung, die von der Geschäftsführung der Fakultät für Informatik betrieben wird, um die Zuteilung der Studierenden auf die verfügbaren, vorlesungsbegleitenden Tutorien durchzuführen. |
| Webservice | "A Web service is a software system designated to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjuction with other Web-related standards. A Web service is an abstract notion that must be implemented by a concrete agent. The agent is the concrete piece of software or hardware that sends and receives messages, while the (Web-)service is the resource characterized by the abstract set of functionality that is provided."<br>[W3C Working Group Note 11.02.2004, http://www.w3.org/TR/ws-arch/] |
| WWW | *World Wide Web*<br>Wichtigste Anwendung des Internet. |
| X.500 | Standard der ITU für Verzeichnisdienste. Fand aufgrund hoher Komplexität keine weite Verbreitung. |
| X.509 | Standard der ITU für Public-Key-Infrastrukturen. Spezifiziert insbesondere Standardformate für die Speicherung von digitalen Zertifikaten. Spezifiziert in RFC 3280. |
| XML | *eXtensible Markup Language*<br>Vom W3C standardisierte Auszeichnungssprache, durch die sich die Struktur eines Dokuments beschreiben lässt. |
| XML-RPC | Standard zur Durchführung von *Remote Procedure Calls*. Verwendet http zur Datenübertragung, Anfragen und Antworten werden gemäß XML kodiert. Erweiterte Funktionen wie Transaktions-Mechanismen, Authentifikation etc. werden davon nicht erfasst. |

| | |
|---|---|
| YP | Siehe Yellow Pages |
| Yellow Pages | Ursprüngliche Bezeichnung für NIS |

## Index

## Informationen



## Tabellen

## Literatur


[Ba02] Tom Barton: „Practices in Directory Groups“, NSF Middleware Initiative, http://middleware.internet2.edu/dir/groups/internet2-mace-dir-groups-best-practices-200210.htm

[Ca03] Gerald Carter: „LDAP System Administration“, O'Reilly & Associates, ISBN 1565924916

[Ha04] Patrick von der Hagen: „Pdok IPO-Registrierung“

[HB+04] David Holder, Simon Biles, TimO'Brien, Alistair Matthes: „Solution Guide for Windows Security and Directory Services for Unix v0.9“, veröffentlicht von Microsoft Corporation.

[HK01] Patrick von der Hagen, Christian Knierim: “Neugestaltung der Benutzerverwaltung der Fakultät für Informatik”, Studienarbeit am Institut für Programmstrukturen und Datenorganisation (IPD)

[HR04] Wolfgang Hommel und Helmut Reiser: “Federated Identity Management: Die Notwendigkeit zentraler Koordinationsdienste”

[HS+03] Timothy A. Howes, Mark C. Smith, Gordon S. Good: “Understanding and Deploying LDAP Directory Services”, Addison-Wesley, ISBN 0672323168

[KL03] Dieter Klünter und Jochen Laser: „LDAP verstehen, OpenLDAP einsetzen“, Dpunkt Verlag, ISBN 3898642178

[Le03] Spencer C. Lee: “An Introduction to Identity Management”, SANS Institute 2003

[Me02] META Group Inc.: “The Value of Identity Management: How securing identity management provides value to the enterprise”

[Po04] Thomas Poisl: „Kollaborationplattform BSCW“, Studienarbeit bei C&M

[PW04] Birgit Pfitzmann und Michael Waidner: “Federated Identity-Management Protocols –Where User Authentication Protocols May Go–“, IBM Zurich Research Lab

[Sc02] Peter Schaar: „Datenschutz im Internet“, Verlag C. H. Beck, ISBN 3406486584

[Sc04] Niko Schmid: „Ein Portal zur Unterstützung ausgewählter Geschäftsanwendungsfälle in der Aus- und Weiterbildung“, Diplomarbeit bei C&M

[SE+01] Hal Stern, Mike Eisler, Ricardo Labiaga: “Managing NFS and NIS”, O'Reilly, ISBN 1565925106

[We04] Sven Weih, „Zusammenführung von LDAP-Servern /ATIS Dial-In“, Praktikum bei C&M

# ANHANG

## LDAP-Schema für ATIS-Dial-In und ATIS-VPN

Diese Schema-Definitionen werden von der ATIS zur Bereitstellung des Dial-In-Dienstes und des VPN-Dienstes verwendet. Diese Syntax wird von OpenLDAP verwendet. Es werden ein benötigtes Attribut sowie zwei neue Objektklassen definiert.

```
attributetype
   ( 1.3.6.1.4.1.87.4.2.1.3
      NAME 'atisVpnIp'
      DESC 'used to store a special IP for VPN-systems'
      EQUALITY caseIgnoreIA5Match
      SUBSTR caseIgnoreIA5SubstringsMatch
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
      SINGLE-VALUE
   )

objectclass
   ( 1.3.6.1.4.1.87.4.2.2.2
      NAME 'atisDialinProfile'
      SUP top AUXILIARY
      DESC 'ATIS RadiusDialin Profile als AUXILIARY Klasse'
      MUST ( uid $ userPassword $ mail )
      MAY ( radiusAuthType $ radiusServiceType $
            radiusFramedIPAddress $ radiusFramedIPNetmask $
            radiusFramedProtocol $ radiusFramedRoute $
            radiusFramedRouting $ radiusReplyMessage $
            radiusReplyItem $ radiusCheckItem $
            radiusAscendIdleLimit $ radiusAscendCallback $
            radiusAscendDialNumber $ radiusAscendDataSvc $
            radiusAscendCBCPEnable $ radiusAscendCBCPMode $
            radiusAscendSendAuth $ radiusGroupName $
            radiusAscendBaseChannelCount $
            radiusAscendIncChannelCount $
            radiusAscendDecChannelCount $
            radiusAscendAppletalkPeerMode $
            radiusAscendBridge $
            )
   )

objectclass
   ( 1.3.6.1.4.1.87.4.2.2.3
      NAME 'atisVpnProfile'
      SUP top AUXILIARY
      DESC 'ATIS RadiusVpn Profile als AUXILIARY Klasse'
      MUST ( uid $ userPassword $ mail $ atisVpnIp)
      MAY ( userCertificate )
   )
```

## Beispielnutzer im LDAP-Verzeichnis

Ein Beispielnutzer nach Synchronisation mit dem existierenden Useradm und ohne weitere Dienste.

```
dn: uid=hagen,ou=people,ou=ATIS,dc=ira,dc=uka,dc=de
uid: hagen
sn: von-der-Hagen
givenName: Patrick
cn: Patrick von-der-Hagen
displayName: Patrick von-der-Hagen
loginShell: /bin/bash
gecos: Patrick von-der-Hagen,,,
uidNumber: 707
gidNumber: 70
mail: hagen@ira.uka.de
userPassword: XXXXXXXX
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: qmailUser
telephoneNumber: 072186951106
employeeType: Hiwi
employeeNumber: nichtZutreffend
title: Hochwohlgeboren
o: Abteilung Technische Infrastruktur
ou: ATIS
mailHost: irams1.ira.uni-karlsruhe.de
mailForwardingAddress: hagen@irams1.ira.uni-karlsruhe.de
mailAlternateAddress: patrick.vdhagen@ira.uka.de
mailAlternateAddress: patrick.von-der-hagen@ira.uka.de
homeDirectory: /home/atis/hagen
```

## Beispielnutzer im LDAP-Verzeichnis mit VPN-Dienst

Ein Beispielnutzer nach Synchronisation mit dem existierenden Useradm und mit Daten für den VPN-Dienst.

```
dn: uid=hagen,ou=people,ou=ATIS,dc=ira,dc=uka,dc=de
uid: hagen
sn: von-der-Hagen
givenName: Patrick
cn: Patrick von-der-Hagen
displayName: Patrick von-der-Hagen
loginShell: /bin/bash
gecos: Patrick von-der-Hagen,,,
uidNumber: 707
gidNumber: 70
mail: hagen@ira.uka.de
```

```
userPassword: XXXXXXXX
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: qmailUser
objectClass: atisVpnProfile
telephoneNumber: 072186951106
employeeType: Hiwi
employeeNumber: nichtZutreffend
title: Hochwohlgeboren
o: Abteilung Technische Infrastruktur
ou: ATIS
mailHost: irams1.ira.uni-karlsruhe.de
mailForwardingAddress: hagen@irams1.ira.uni-karlsruhe.de
mailAlternateAddress: patrick.vdhagen@ira.uka.de
mailAlternateAddress: patrick.von-der-hagen@ira.uka.de
homeDirectory: /home/atis/hagen
atisVpnIp: 141.3.6.2
```