Architektur eines Identitätsmanagementsystems an einer Hochschule

Diplomarbeit

von

Steffen Hofmann

zur Erlangung des Grades eines Diplom-Informatikers

angefertigt an der Fakultät für Mathematik und Informatik,
Lehrgebiet Informationssysteme und Datenbanken,
Prof. Dr. Gunter Schlageter,
der FernUniversität in Hagen

eingereicht am 08. Juni 2007

E-Mail:
steffen.hofmann@fu-berlin.de

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Hochschulen stellen komplexe Organisationen dar, zu deren Verwaltung eine Vielzahl von Diensten benötigt wird. Zum Beispiel ist die Studierendenverwaltung verantwortlich für die Einschreibungen und Rückmeldungen für die einzelnen Studiengänge, Prüfungsämter erfassen Noten und die Personalstelle erstellt die Gehaltsabrechungen für Mitarbeiter. Dies erfolgt inzwischen fast immer durch die Unterstützung von IT-Systemen. Dabei lassen sich diverse Anwendungen identifizieren, in denen Datensätze mit Personeninformationen gespeichert werden. Die Systeme sind untereinander jedoch selten gekoppelt. Das folgende Szenario soll einen kleinen Einblick darüber geben, welche Probleme dadurch auftreten können:

Eine Studierende, Korinna Zuse, wird von der Studierendenverwaltung mit ihrer Matrikelnummer, ihrem Studiengang und der privaten Anschrift erfasst. Frau Zuse arbeitet auch an der Hochschule als studentische Hilfskraft für die Fakultät Informatik. Die Personalabteilung, die sie als Mitarbeiterin führt, erfasst ebenfalls einige Daten zu ihrer Person. Für den Zugriff auf die Webanwendung zur Pflege der Lehrveranstaltungen erhält sie eine persönliche Zugangskennung. Inzwischen existieren drei Datensätze zu Korinna Zuse. Die Studentin und Mitarbeiterin heiratet einen Monat nach ihrer Anstellung und nimmt den Namen ihres Mannes an. Mit der Rückmeldung zum nächsten Semester teilt sie die Namensänderung der Studierendenverwaltung mit. Frau von Neumann, so lautet ihr neuer Nachname, vergisst jedoch, dies der Personalstelle mitzuteilen. Die Studentin ändert an der Fakultät Informatik ihr Aufgabengebiet. Die Erfassung der Lehrveranstaltung erwies sich nämlich schnell als mühsame und langweilige Tätigkeit, da zu jeder Lehrveranstaltung immer wieder neu die Dozenten vollständig mit allen Angaben wie E-Mail-Adresse und Telefonnummer eingetragen werden mussten. Die neue Aufgabe, die Erstellung von Videos für das E-Learningsystem, bereitet ihr große Freude. Um die einzelnen Filme in die Learning-Management-Plattform einspielen zu können, wird für Korinna von Neumann ein neues Benutzerkonto eingerichtet. Der Zugang zu der Webanwendung für die Pflege der Lehrveranstaltungen bleibt versehentlich aktiv, obwohl sie diese Aufgabe gar nicht mehr wahrnimmt.

Das Szenario zeigt einige Probleme auf, die durch den Einsatz ungekoppelter IT-Systeme an einer Hochschule entstehen können. An mehreren Stellen werden manuell Daten zu einer Person erfasst. Durch ausbleibende Aktualisierungen und fehlerhafte Eingaben können schnell widersprüchliche Daten entstehen. Die Auflösung dieser Inkonsistenzen über alle Systeme erfordert meist einen immensen administrativen Aufwand. Es entstehen also hohe Personalkosten für die Datenerfassung und -bereinigung. Die Hochschulangehörigen müssen sich zudem eine Vielzahl von Zugangskennungen merken und ständig neu eingeben. Regelmäßig vergessen Benutzer ihre Passwörter oder verwechseln diese. Verwaiste Benutzerkonten stellen ein großes Sicherheitsrisiko dar. So können Hacker unbemerkt diese Zugangsdaten ausnutzen, um einen Zugriff auf bestimmte Systeme zu erhalten. Werden aus dem oben genannten Szenario in dem System für die Pflege der Lehrveranstaltungen die Ankündigungstexte und Zeitangaben geändert, so könnte ein großer Imageschaden entstehen. Welche Person möchte einer Hochschule Daten anvertrauen, die offensichtlich unsichere IT-Systeme betreibt? Es sollten also Regeln existieren, die beispielsweise festlegen, innerhalb welcher Zeit Personen ihre Zugangsrechte verlieren. Diese Regeln müssen für alle Applikationen der Hochschule wirksam werden. Häufig wird zudem anhand einer Gruppenzugehörigkeit festgelegt, welche Funktionen einem Benutzer innerhalb einer An-

wendung zur Verfügung stehen. Studierende dürfen zum Beispiel ihre Noten einsehen, bestimmte Mitarbeiter sind berechtigt, Noten zur erfassen. In diesem Beispiel muss man sogar vermeiden, dass eine Person beide Funktionen wahrnehmen kann. Sonst könnte der Studierende seine eigenen Noten eintragen. Es muss also erkannt werden, ob anhand der gespeicherten Attribute zu einem Mitarbeiter auch ein Datensatz eines Studierenden existiert, bei dem es sich um dieselbe natürliche Person handelt. Gegebenenfalls erfordert dies die Entfernung eines oder sogar mehrerer Duplikate.

Bei der Einführung neuer oder bei dem Austausch alter Systeme stellt sich jedes Mal die Frage, wie die neuen Systeme effizient mit Daten gefüllt werden können. Personendaten, Benutzerkonten und bestimmte Rechte müssen für die neue Anwendung in irgendeiner Form erreichbar sein. Kann ein Mitarbeiter das System nicht nutzen, da seine Daten noch nicht oder unvollständig eingetragen wurden, so kommt es zu einem Produktivitätsverlust. Der Ärger der frustrierten Hochschulangehörigen, die lange Zeit keinen Zugriff erhalten, entlädt sich meist bei den Administratoren. Unter Zeitdruck wird das Problem irgendwie gelöst. Die Konsequenzen sind in der Regel viel zu umfangreiche Rechte und ein weiteres undokumentiertes Benutzerkonto, von dem nach einigen Jahren niemand mehr weiß, wie es zustande gekommen ist.

Ein Identitätsmanagementsystem, mit dem sich die autonomen Systeme einer Hochschule koppeln lassen, könnte eine Vielzahl der genannten Probleme lösen.

Aktuell wird das Thema Identitätsmanagement an vielen Hochschulen behandelt. Die am 19. Juni 1999 von 29 europäischen Staaten unterzeichnete Bologna-Erklärung „soll die Grundlage für die Verwirklichung eines europäischen Hochschulraums bis zum Jahr 2010 bilden.“[1] Diese Hochschulreform beinhaltet die Einführung einer Struktur mit Bachelor- und Masterstudiengängen, die eine Vergleichbarkeit der Abschlüsse unter den europäischen Hochschulen ermöglichen soll. Diese Studienabschlüsse beinhalten jedoch eine neue Prüfungsstruktur, die an den Hochschulen abgebildet werden muss. An einigen deutschen Hochschulen wurden und werden deshalb neue Anwendungen zur Verwaltung von Studiengängen und insbesondere von Prüfungsleistungen eingeführt. Die neuen Anwendungen müssen dafür in die alte historisch gewachsene IT-Landschaft integriert werden. Dies kann jedoch nicht in kurzer Zeit stattfinden. Studierenden- und Prüfungsdaten müssen beispielsweise in ein neues System stufenweise migriert werden oder es findet abhängig vom Studiengang eine parallele Verwaltung entweder in einem alten oder dem neuen System statt. Diese sanfte Integration neuer IT-Systeme kann durch die Einführung eines Identitätsmanagementsystems unterstützt werden.

1.2 Ziele und Adressatenkreis

Ziel der Diplomarbeit ist die Skizzierung einer Architektur für ein Identitätsmanagementsystem. Diese Architektur soll auf der historisch gewachsenen, heterogenen IT-Infrastruktur einer Hochschule basieren und insbesondere folgende Anforderungen umsetzen können:

- Zur Arbeitsersparnis und der daraus im Idealfall resultierenden Kostenersparnis soll eine manuelle Erfassung und Aktualisierung von personenbezogenen und weiteren Daten, die von mehreren Anwendungen der Hochschule benötigt werden, nur ein-

[1] [BMBF05, S. 12]

malig erfolgen. Dies erfordert eine Integration von Datenquellen und die Übermittlung von ausgewählten Informationen an andere Systeme.

- Es muss auf eine einfache Art und Weise möglich sein, bestehende und neue IT-Systeme einbinden und später ersetzen zu können. Die geforderte Flexibilität führt dazu, dass die Architektur nicht nur eine einmalige Integration zu einem festgelegten Zeitpunkt ermöglicht, sondern zukunftsorientiert einen dynamischen Anpassungsprozess erlaubt.

- Die Zahl der verschiedenen Benutzerkonten für eine Person an der Hochschule soll reduziert werden. Mehrfachanmeldungen mit der gleichen Zugangskennung gilt es ebenfalls zu vermindern. Es gilt also die Benutzerfreundlichkeit im Umgang mit den IT-Systemen an einer Hochschule zu erhöhen.

Kooperationen im Bereich der Forschung und Lehre erfordern auch einen Datenaustausch unter den Hochschulen. Dieser Aspekt soll in der Architektur zusätzlich berücksichtigt werden. Das folgende Beispiel macht die Vorteile einer hochschulübergreifenden Kopplung deutlich.

Beispiel:
Der wissenschaftliche Mitarbeiter der FernUniversität in Hagen reist zu seinen Forschungspartnern an die Freie Universität Berlin und möchte vor Ort die Möglichkeit erhalten, mit seinem Notebook E-Mails und Forschungsdaten in Hagen abzurufen. Er benötigt hierfür einen Zugang zum Netzwerk, auf das man erst zugreifen kann, wenn man einen gültigen Benutzernamen mit dem dazugehörigen Passwort eingegeben hat. Die Freie Universität Berlin könnte diese Daten zur Prüfung an die FernUniversität in Hagen weiterreichen. Die Freie Universität Berlin muss also nichts weiter über den mobilen Wissenschaftler wissen und auch keine Daten zu der Person erfassen.

Adressaten dieser Arbeit sind vor allem die Mitarbeiterinnen und Mitarbeiter an den Hochschulen, die sich mit diesen Integrationsprozessen beschäftigen. Viele diskutierte Themen und Architekturmerkmale des Identitätsmanagementsystems sind jedoch nicht nur hochschulspezifisch. So sind auch (IT-)Mitarbeiter anderer Unternehmen angesprochen.

1.3 Gliederung und Aufbau der Arbeit

In Abschnitt 2 („Grundlagen und Stand der Forschung“) wird zunächst geklärt, was ein Identitätsmanagementsystem überhaupt darstellt und welche Anforderungen an ein solches System bestehen. Verbunden mit den Anforderungen erfolgt eine detaillierte Betrachtung des Themas Datenintegration. Weitere Schwerpunkte bilden Autorisierungsmodelle und Standards, die im Bereich föderierter Identitätsmanagementsysteme zum Einsatz kommen.

In Abschnitt 3 („Rahmenbedingungen“) werden die organisatorischen, technischen, rechtlichen und hochschulpolitischen Aspekte untersucht, die bei einem Identitätsmanagementsystem, das an einer deutschen Hochschule zum Einsatz kommt, berücksichtigt werden müssen. Ein besonderer Fokus liegt dabei auf den Datenschutzgesetzen.

In Abschnitt 4 („Architektur“) wird eine Architektur für ein Identitätsmanagementsystem an einer Hochschule unter verschiedenen Gesichtspunkten entwickelt. Den Schwerpunkt bilden dabei die Komponenten, aus denen eine Informationssystem-Architektur entsteht.

2 Grundlagen und Stand der Forschung

2.1 Definition Architektur

Architektur: Eine Architektur besteht aus Komponenten, die bestimmte Verantwortlichkeiten übernehmen und miteinander in Beziehung stehen. Sie soll die gestellten Anforderungen (funktionale und nichtfunktionale) möglichst vollständig berücksichtigen. Die Beschreibung einer Architektur stellt die Verantwortlichkeiten, Beziehungen und Schnittstellen von Komponenten möglichst abstrakt dar und soll die Details einer Realisierung ausblenden.

2.2 Identitätsmanagementsysteme

2.2.1 Definition

Den Begriff Identitätsmanagement (IDM) definiert Spencer C. Lee folgendermaßen:

"Identity management refers to the process of employing emerging technologies to manage information about the identity of users and control access to company resources."[2]

Identitätsmanagement wird demnach nicht als ein starres Gebilde aufgefasst, sondern als ein Prozess, bei dem mehrere (neu aufkommende) Technologien zum Einsatz kommen. Ein wesentlicher Aspekt des Prozesses ist die Verwaltung von Identitäten. Spencer C. Lee schränkt die Identitäten auf Benutzeridentitäten ein. Andere Autoren sprechen nur allgemein von Identitäten. Darüber hinaus beinhaltet Identitätsmanagement die Steuerung des Zugriffs auf Ressourcen. Dabei kann es sich um eine simple Anmeldung an einem PC mit Benutzernamen und Passwort handeln, jedoch sind auch komplexere Zugangsverfahren möglich. Üblich ist auch die Verwendung des Begriffs Identity and Access Management (IAM).

Identitätsmanagementsystem (IDMS): Ein System, das den Prozess Identitätsmanagement unterstützt, wird als Identitätsmanagementsystem bezeichnet.

Der Begriff System kann sehr allgemein aufgefasst werden.

System: Ein System stellt eine nach außen abgrenzbare zweckgebundene Einheit dar, die sich aus dem Verbund von mehreren Elementen ergibt, die untereinander eine Wechselwirkung besitzen.

Im Kontext des Identitätsmanagements werden in aller Regel unter System die beteiligten Software- sowie Hardwarekomponenten verstanden. Bei dieser begrifflichen Auslegung existiert an einer Hochschule mehr als nur ein Identitätsmanagementsystem. Anwendungen zur Personal- und Studierendenverwaltung sind jeweils eigene IDMS. Es ist jedoch kein Widerspruch, wenn ein IDMS andere Identitätsmanagementsysteme integriert. Es entsteht wieder eine zweckgebundene Einheit.

In einem kontinuierlichen Prozess über viele Phasen verändert sich ein IDMS. Es erfolgen immer wieder Anpassungen an neue Applikationen, oder Änderungen der organisatorischen Strukturen erzeugen neue Richtlinien. Durch die unterschiedlichen Strukturen in den

[2] [Lee03], S. 3

Organisationen und den dynamischen Prozessen gibt es kein fertiges Produkt. Meist werden diverse Softwarekomponenten verschiedener Anbieter benötigt.

2.2.2 Identitäten

Identitätsmanagement beschäftigt sich nach der Definition in 2.2.1 also mit der Verwaltung von Informationen zu Identitäten. Eine Abgrenzung des Bergriffes Identität steht jedoch noch aus.

Der mehrdeutige Begriff Identität (engl. identity) stammt von dem lateinischen Wort identitas ab und bedeutet Wesenseinheit bzw. Wesensgleichheit. Bezogen auf Identitätsmanagementsysteme definiert L. Jean Camp den Begriff Identität wie folgt:

„In an identity management system identity is that set of permanent or long-lived temporal attributes associated with an entity."[3]

Eine Identität setzt sich also aus einem Satz mehrerer Attribute zusammen und gehört zu einer bestimmten Person, einem Auto oder anderen Dingen. Es werden nur solche Attribute herangezogen, die dauerhaft ohne Änderung des Wertes zur Verfügung stehen oder zumindest über einen langen Zeitraum beständig sind. Das Geburtsdatum einer Person ändert sich nie, der Nachname hingegen kann sich im Laufe der Zeit zum Beispiel durch Heirat ändern. Beide Attribute wären nach der genannten Definition für die Bildung einer Identität geeignet.

Eine weitere mögliche Definition des Begriffes Identität stammt von Phillip J. Windley:

„A digital identity contains data that uniquely describes a person or thing […] but also contains information about the subject's relationships to other entities."[4]

In dieser Definition wird von einer digitalen Identität gesprochen. In der Literatur zu Identitätsmanagementsystemen werden die Begriffe Identität und digitale Identität häufig synonym verwandt. Das Wort digital gibt einen Hinweis darauf, dass die Daten in einem IT-System gespeichert sind.

Die Definition legt weiterhin fest, dass die gespeicherten Daten die Person oder etwas anderes eindeutig bestimmen. Diese Forderung erscheint jedem sofort einsichtig und selbstverständlich. Aber welche Attribute stellen eine Eindeutigkeit her? Ein Ansatz wäre, all die Merkmale zu verwenden, die sich nicht oder nur sehr selten ändern. Die erfassten Attribute können jedoch nicht ausreichen, um in einer Datenbank einen Schlüssel zu bilden. Künstliche Schlüsselwerte oder eindeutige Namen in einem Namensraum, einer Domäne, wie es bei E-Mail-Adressen der Fall ist, bilden in der Regel den Identifikator (engl. identifier). Auch Beziehungen stellen weitere Informationen dar, um eine Identität zu bilden. Es kommt sogar vor, dass nur mit Hilfe dieser Beziehungen eine Identität beispielsweise einer Person zugeordnet werden kann, da ansonsten alle Attribute zu mehreren Einträgen gleich sind.

Beispiel:
In einer Datenbank zur Dozentenverwaltung sind zwei Jens Müller enthalten. Neben dem vollständigen Namen wurden leider keine weiteren Attribute erfasst. Woher weiß man jetzt,

[3] [Camp04], S. 36
[4] [Wind05], S. 8

ob es sich um dieselbe oder um eine andere natürliche Person handelt? Beide Datensätze sind mit Lehrveranstaltungen verbunden. Die Betrachtung der Lehrveranstaltungen zeigt, dass der eine Eintrag mit Jens Müller nur Kurse aus dem Bereich Datenbanksysteme beinhaltet. Der andere Jens Müller hingegen scheint in vorderasiatischer Geschichte zu referieren. Wahrscheinlich hat man es also mit zwei verschiedenen natürlichen Personen zu tun. Die eindeutige Identität zu dem jeweiligen Dozenten bilden der vollständige Name und die Beziehungen zu den Lehrveranstaltungen. Nur ein Anruf bei einem Jens Müller wird die anzunehmende Eindeutigkeit letztendlich bestätigen.

Beide Definitionen zur Identität schließen die Existenz mehrerer Identitäten zu einer Person nicht aus. Tatsächlich besitzt ein Studierender viele Identitäten an einer Hochschule, wie das folgende Bespiel zeigt:

Beispiel:
In der Studierendenverwaltung existiert zu einem Studierenden eine Identität, die unter anderem die Attribute Name, Geburtsdatum, Anschrift, Matrikelnummer und Studiengang enthält. Im Prüfungsbüro werden zusätzlich Noten erfasst. Die Hochschulbibliothek hingegen speichert eine Identität bestehend aus Bibliotheksausweisnummer, Namen und Anschrift.

Die Identitäten enthalten einen Teilsatz an Attributen, die eindeutig den zugehörigen Studierenden bestimmen, sich aber nur in wenigen Attributen überschneiden können. Die Identitäten lassen sich auch als Ausweis zu der Person oder einer anderen Sache auffassen. Werden Attribute einer Identität aus dem einen System zur Bildung einer neuen Identität in einem anderen System transportiert, so spricht man von einer abgeleiteten Identität.

Zur Vereinfachung wird im nachfolgenden Text ausschließlich der Begriff Identität verwendet. Er beinhaltet auch die Speicherung der Daten in einem IT-System (digitale Identität). Des Weiteren werden in den Definitionen und Beispielen nur Personen und deren Identitäten betrachtet. Die Identitäten könnten jedoch auch zu einem Programm, einem Auto oder etwas anderem gehören. Bei Pseudonymen, die fingierte Namen darstellen, handelt es sich auch um Identitäten, deren Bezug zu einer Person jedoch nicht offensichtlich ist.

Der Lebenszyklus einer Identität

In der Literatur zu dem Thema Identitätsmanagement werden diverse Modelle dargestellt, die den Lebenszyklus einer Identität beschreiben. Abbildung 1 zeigt in Anlehnung an Phillip J. Windley ([Wind05]) die abstrakte Form eines Lebenszyklus.

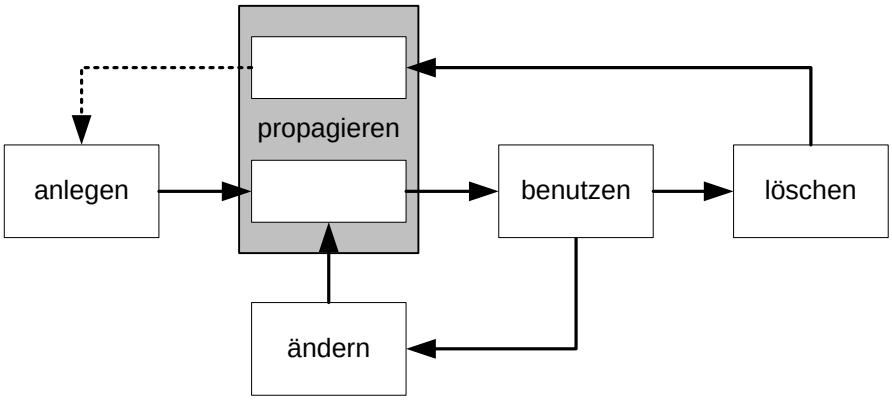

Abbildung 1: Lebenszyklus einer Identität

Die Phasen lassen sich wie folgt beschreiben:

- **Anlegen**
  In dieser Phase wird eine Identität mit allen notwendigen Attributen angelegt. Dies kann durch eine manuelle Eingabe erfolgen. Es ist aber auch möglich, dass ein automatisierter Prozess den Datensatz erzeugt. Windley benennt diese Phase mit dem englischen Begriff provision und definiert den Begriff provisioning aus zwei Perspektiven.

  "Provisioning is the process of preparing an IT system to provide service to a client, customer, or other user. From the perspective of digital identity, provisioning is the creation of the identity record and its population with the correct attributes."[5]

  Die Attribute, die zu einer Identität gespeichert werden, lassen sich unterteilen in Standardattribute und systemspezifische Attribute. Standardattribute besitzen die Eigenschaft, dass sie unabhängig von einem System der Identität angehören. Zusätzlich werden innerhalb eines Systems noch Attribute angelegt, die nur dort eine Bedeutung besitzen.

  Beispiel:
  Der Vorname und Nachname einer Person sind Standardattribute. Ein künstlicher Datenbankschlüssel, der zu der Identität erzeugt wird, ist in der Regel nur für das Datenbanksystem relevant.

- **Propagieren**
  Eine Identität wird in der Regel von mehreren Systemen verwendet. Aus diesem Grund muss die Identität oder eine von ihr abgeleitete Identität eventuell an andere Systeme propagiert werden. Der Verteilungsprozess sollte möglichst zeitnah erfolgen. Unter anderem können technische Gründe zu Aktualisierungsintervallen von mehreren Tagen führen. In diesem Fall müssen die Änderungsanfragen beispielsweise in Warteschlangen zwischengespeichert werden. Klassische Transaktionsmechanismen von Datenbankmanagementsystemen, die dem ACID-Prinzip (atomicity, consistency, isolation, durability) folgen, sind für das Gesamtsystem nicht anwendbar. Der Zustand der temporären Inkonsistenz wird bei Identitätsmanagementsystemen meist in Kauf genommen.

- **Benutzen**
  Die Tatsache, dass eine Identität in Systemen für einen bestimmten Zweck benutzt werden soll, liefert den Grund dafür, dass eine Identität überhaupt gebildet und propagiert wird.

- **Ändern**
  Identitäten unterliegen im Laufe der Zeit immer wieder Änderungen. Abhängig von der Eigenschaft des Attributs können Änderungen innerhalb weniger Sekunden oder in sehr großen Zeitabständen von mehreren Jahren erfolgen. Windley bezeichnet diese Phase mit dem englischen Begriff maintain und weist darauf hin, dass auch geänderte Werte wieder an alle Systeme propagiert werden müssen.

[5] [Wind05], S.30

- **Löschen**
  Löschungen (engl. deprovision nach Windley) müssen allen Systemen bekannt gemacht werden, an die die Identität propagiert wurde. Dieser letzte Schritt wird des Öfteren vergessen und so existieren in einigen Systemen verwaiste Identitäten. Im Falle von Benutzerkonten können so sicherheitskritische Zugänge entstehen.

Windley stellt in seinem Modell nicht explizit dar, dass Löschungen propagiert werden müssen. Eine Wiederbelebung von Identitäten ist nach seiner Vorstellung ebenfalls nicht möglich. Wird jedoch eine Identität vollständig aus einem System gelöscht und nach einiger Zeit wieder mit den identischen Daten angelegt, so kann man auch den Fall der Wiederbelebung (gestrichelter Pfeil in Abbildung 1) in das Modell mit aufnehmen.

In einigen Modellen zu Lebenszyklen einer Identität werden Sperrungen und Freigaben gerne als eigene Phase aufgenommen. Dies ist nach meiner Auffassung eine zu eingeschränkte Sichtweise, die Identitäten immer mit Benutzerkonten gleichsetzt.

Der Begriff Provisioning wird in der Literatur nicht eindeutig definiert. Entgegen der oben genannten Beschreibung von Windley, der unter Provisioning lediglich das Erzeugen und Verteilen einer Identität versteht, verwende ich in dieser Arbeit folgende Definition:

Provisioning: Provisioning beschreibt den Vorgang des Anlegens, Propagierens, Änderns, Löschens und Wiederbelebens von Identitäten.

Jede Phase des Lebenszyklus muss in der Architektur berücksichtigt werden. Nach einer abstrakten Betrachtung sollten alle am IDMS teilnehmenden Systeme untersucht werden, um die Realisierung der einzelnen Phasen zu ermöglichen.

2.2.3 Anforderungen

Die Anforderungen, die an ein IDMS gestellt werden, lassen sich wie auch für Software üblich in funktionale und nicht funktionale Anforderungen unterteilen. Die folgenden Anforderungen gelten allgemein für Identitätsmanagementsysteme und ergeben sich nicht aus dem speziellen Kontext der Hochschule.

Funktionale Anforderungen

Funktionale Anforderungen, die sich auf eine konkrete Organisation beziehen und beispielsweise Geschäftsregeln, spezifische Abfolgen von Operationen oder besondere Schnittstellen beinhalten, können an dieser Stelle nicht genannt werden. Ein Vergleich diverser Whitepapers von Anbietern für Identitätsmanagementsysteme und der geringen Zahl wissenschaftlicher Publikationen zu diesem Thema führt zu der folgenden Klassifizierung von Funktionen, die für ein IDMS typisch sind. Insbesondere wurden [Lee03], [Diam06], [Orac06], [Sun06], [IBM05] und [IBM06] betrachtet.

Ein Identitätsmanagementsystem kann die nachstehenden Funktionen übernehmen:

a) Informationsspeicher

Eine der wesentlichen Aufgaben eines Identitätsmanagementsystems ist die Speicherung aller benötigten Informationen. Abhängig von der Architektur des Systems können sämtliche Identitäten einer Hochschule in einer zentralen Datenbank gespeichert werden (zentra-

ler Architekturansatz) oder nur Referenzen auf Identitäten in anderen Datenquellen und deren Beziehungen untereinander (föderierter Architekturansatz). Bei den gespeicherten Identitäten kann es sich beispielsweise um Personen, Anwendungen, Geräte, Gebäude sowie Räume handeln. Die Art und der Umfang der Attribute zu einer Identität sind abhängig von dem Kontext, in dem die Identität verwendet wird. Bei Identitätsmanagementsystemen kommen häufig Verzeichnisdienste zum Einsatz, in denen die Identitäten in der Regel hierarchisch angeordnet sind.

b) Datenintegration

Die Datenquellen, die durch das Identitätsmanagementsystem in irgendeiner Form integriert werden müssen, weisen Heterogenitäten auf mehreren Ebenen auf. Hierzu gehören beispielsweise Datenbankmodellunterschiede und technische Besonderheiten der einzelnen Systeme. Viele Datenbanken dienen genau einer Anwendung und sind nicht für eine Kopplung oder Zusammenführung mit anderen vorbereitet. Hinzu kommt eine stark divergierende Qualität der Daten.

c) Authentifizierung

Authentifizierung: Unter Authentifizierung wird der Vorgang verstanden, bei dem eine Person oder eine zu ihr gehörende Identität anhand bestimmter Authentifizierungsmerkmale (engl. credentials) überprüft wird.

Authentisierung: Das Erbringen des Nachweises der eigenen Person oder der zugehörigen Identität wird mit dem Begriff Authentisierung wiedergegeben.

Beispiel:
Eine Person möchte den Zugang zu einem Forschungslabor erhalten. Sie authentisiert sich durch Scannen eines Fingerabdrucks gegenüber einem Zugangssystem. Innerhalb des Systems werden die erfassten Daten zu dem Fingerabdruck überprüft. Es erfolgt also eine Authentifizierung.

Im Englischen gibt es keine Unterscheidung zwischen den Begriffen Authentifizierung und Authentisierung. Man spricht nur von authentication.

Eine sehr benutzerunfreundliche Situation entsteht dann, wenn eine Person für jede Anwendung an einer Organisation andere Identitäten besitzt und somit auch verschiedene Authentifizierungsmerkmale erbringen muss. Ein Identitätsmanagementsystem übernimmt die Rolle einer zentralen Authentifizierungsinstanz innerhalb einer Organisation und ermöglicht eine Reduzierung der notwendigen Authentifizierungsmerkmale für alle Systeme. Des Weiteren soll die Anzahl der Eingaben von Authentifizierungsmerkmalen durch eine Person reduziert werden. Hierbei gelten mehrere Stufen der Erleichterung der Authentisierung.

Reduced-Sign-On: Unter Reduced-Sign-On versteht man die Reduzierung der notwendigen Authentisierungen, die eine Person für die Anmeldung an den Systemen einer Organisation durchführen muss. Die Person kann hierzu ein oder mehrere unterschiedliche Sätze von Authentifizierungsmerkmalen verwenden.

Simple-Sign-On: Wird an einer Organisation für die Anmeldung an den Systemen nur ein bestimmter Satz von Authentifizierungsmerkmalen verwendet, der aber bei jeder Anmeldung an ein System teilweise oder vollständig eingegeben werden muss, so spricht man von Simple-Sign-On.

Single-Sign-On (SSO): Wie beim Simple-Sign-On wird nur noch ein bestimmter Satz von Authentifizierungsmerkmalen in der Organisation verwendet, jedoch authentisiert sich die Person nur noch einmal innerhalb eines festgelegten Zeitraumes (z.B. pro Tag). Alle weiteren Authentifizierungen in den Systemen, die die Person benutzen möchte, werden durch Mechanismen realisiert, die keine erneute Eingabe von Authentifizierungsmerkmalen durch die Person erfordern.

Das Verständnis über den Begriff Single-Sign-On ist sehr vielschichtig. Die Definitionen in den Publikationen bewegen sich jedoch in aller Regel innerhalb eines Kontinuums, das durch die hier definierten Sachverhalte Simple-Sign-On und Single-Sign-On begrenzt wird.

Ein sehr populärer Authentifizierungsdienst, der ein Single-Sign-On ermöglicht, ist Kerberos. Die aktuelle Version 5 von Kerberos wird in [Neum05] spezifiziert. Eine kostenlose Implementierung für die Betriebsysteme Unix und Linux findet man beispielsweise unter [MIT07]. Kerberos besitz den Nachteil, dass für Webbrowser und E-Mail-Clients meist Plug-ins geladen werden müssen. Dies ist jedoch nicht immer möglich (Beispiel: *Internet-cafe*).

Eine Person, die sich an mehreren Systemen angemeldet hat, muss sich von diesen auch wieder abmelden. Dabei wäre ein Single-Log-Out wünschenswert.

Single-Log-Out: Unter Single-Log-Out wird eine einzelne Aktion verstanden, die eine Person ausführen muss, um sich von allen Systemen, an denen diese angemeldet ist, abzumelden.

Übernimmt ein IDMS die Rolle der zentralen Authentifizierungsinstanz, so können die erfolgreichen und abgewiesenen Authentisierungsversuche einfach an zentraler Stelle protokolliert werden. Eine regelmäßige Analyse der Authentifizierungsdaten erlaubt es, Missbrauchsfälle aufzuspüren:

Beispiel:
Innerhalb kurzer Zeit erfolgt eine Authentisierung von den zwei geographisch weit entfernten Orten Hagen und Tokio. Wahrscheinlich sind also die Authentifizierungsmerkmale in fremden Besitz gelangt.

Mit den Vorteilen, die sich aus einer zentralen Authentifizierungsinstanz ergeben, sind auch gleichzeitig Nachteile verbunden. Es ist nämlich einfacher möglich, das Verhalten von Personen innerhalb einer Organisation zu kontrollieren. Um dies zu vermeiden, sind Maßnahmen zur Pseudonymisierung von Protokolldaten zu ergreifen. Auch diese Funktion übernimmt ein IDMS. Ein weiterer Nachteil entsteht dadurch, dass die Ausspähung eines einzigen Passworts dazu führen kann, dass der Angreifer für den zugehörigen Benutzer Zugriff auf alle Systeme erhält. Bei besonders kritischen Anwendungen reicht deshalb eine Authentifizierung allein mit einem Benutzernamen und Passwort nicht aus. Zusätzlich müssen noch weitere Authentifizierungsmerkmale herangezogen werden. Dies führt zur Bildung von Authentifizierungskontextklassen, die jeweils für ein festgelegtes Sicherheits-

niveau stehen und eine Kombination mehrere Authentifizierungsarten beinhalten können. Die Steuerung der Authentifizierung abhängig von einem bestimmten Sicherheitsniveau zu einer Anwendung und den erbrachten Authentifizierungsmerkmalen ist eine weitere wichtige Funktionalität, die ein IDMS bereitstellt.

d) Autorisierung

Nach erfolgreicher Authentifizierung für eine Ressource steht fest, welche Identität sich gegenüber dem System ausgewiesen hat. In einem zweiten Schritt muss geklärt werden, welche Zugriffsrechte die Identität auf die Ressource besitzt. Es wird eine Autorisierung durchgeführt.

Autorisierung: Wird einer Person der Zugriff auf eine Ressource innerhalb einer Organisation gewährt, so spricht man von Autorisierung. Autorisierung umfasst dabei die Zuweisung und die Überprüfung von Zugriffsrechten. Vor einer Autorisierung muss immer eine Authentifizierung durchgeführt werden.

Beispiel:
Über ein Web-Portal können Studierende nach erfolgreicher Authentifizierung ihre Prüfungsleistungen abrufen. Die Mitarbeiter des für den Studierenden zuständigen Prüfungsbüros können die erbrachten Leistungen des Studierenden eintragen. Abhängig von den Funktionen, die eine Person an der Hochschule wahrnimmt, werden unterschiedliche Zugriffsrechte für die Systeme zugewiesen und bei jedem Zugriff überprüft. Es findet also eine Autorisierung statt.

Die Funktionen Authentifizierung und Autorisierung werden in einigen Identitätsmanagementsystemen auch zusammengefasst und als Zugriffskontrollkomponente beschrieben.

e) Provisioning

Provisioning wurde bei der Beschreibung des Lebenszyklus einer Identität in Abschnitt 2.2.2 mit den Vorgängen zum Anlegen, Propagieren, Ändern, Löschen und Wiederbeleben einer Identität definiert. Die Abbildung dieser Funktionalität wird auch als Life-Cycle-Management bezeichnet. Zu jeder Personengruppe existiert in der Regel mindestens eine Datenquelle, in der durch manuelle Erfassung wichtige Attribute zu einer Person gespeichert sind. Mit Hilfe von Regelwerken, die passend zu diesen Daten erstellt wurden, wird der Lebenszyklus der Identitäten in den Zielsystemen abgebildet. Es findet eine automatisierte Provisionierung statt. Nicht alle Identitäten, die ein Zielsystem benötigt, können jedoch über diesen Weg in die einzelnen Systeme eingefügt, aktualisiert oder gelöscht werden.

Beispiel:
Eine Firma soll für Wartungsarbeiten den Zugriff auf ein System erhalten. Die hierfür notwendigen Einträge werden dann von einem Administrator vorgenommen. Wichtig ist jedoch, dass diese Einträge speziell markiert sind. Ein fester Endtermin kann dafür sorgen, dass der Zugang mit Abschluss des Wartungsvertrages gesperrt wird.

Die Provisioning-Komponente des IDMS prüft in regelmäßigen Abständen (z.B. täglich) sämtliche Einträge auf Konsistenz. Zu jeder Person in einem Zielsystem wird geprüft, ob die zugehörigen Attribute noch dem Regelwerk entsprechen. Spezielle Einträge, wie die manuell eingepflegte Wartungsfirma, werden gesondert aufgelistet. Einen hohen Sicher-

heitsgewinn und eine Reduzierung von Kosten erreicht man vor allem, wenn Sperrungen und Löschungen für Benutzer sehr schnell und zuverlässig durch die Provisioning-Komponente ausgeführt werden. Hackern wird die Zugriffsmöglichkeit über verwaiste Benutzerkonten genommen und Ressourcen, bei denen hohe Lizenzgebühren anfallen, können nicht mehr genutzt werden.

f) Passwort-Management

Die Verwaltung der Passwörter kann als eigener Funktionsbaustein betrachtet werden. Es wird ein Dienst zur Verfügung gestellt, über den Benutzer unter bestimmten Voraussetzungen Passwörter selbst ändern können. Eine mögliche Voraussetzung kann das Wissen über das alte bestehende Passwort sein. Nach erfolgreicher Änderung des Passwortes muss dieses gegebenenfalls in mehrere Systeme über einen gesicherten Weg transportiert werden. Die Synchronisation der Passwörter sollte möglichst schnell, also innerhalb weniger Sekunden erfolgen. Ansonsten kann schnell die Situation eintreten, dass sich Benutzer zum Teil mit dem alten, zum Teil mit dem neuen Passwort authentisieren müssen. Regelmäßig vergessen Anwender ihre Passwörter und richten sich dann an einen Kundendienst der Hochschule (Hotline), der das Passwort neu setzen kann. Den autorisierten Mitarbeitern muss ein Dienst hierfür zur Verfügung gestellt werden. Alternativ sind auch Verfahren denkbar, mit denen ein Benutzer selbst in die Lage versetzt wird, sein vergessenes Passwort neu zu setzen. Bekannt sind beispielsweise Fragen nach dem Mädchennamen der Mutter, dem Lieblingshaustier oder Ähnlichem. Der Benutzer muss mit der Einrichtung seiner Zugangskennung Antworten zu diesen Fragen formulieren. Um ein an die Bedürfnisse angepasstes Sicherheitsniveau zu erreichen, müssen Richtlinien für die Passwörter gesetzt werden können. Passwortlänge, erlaubte Zeichen, Ähnlichkeiten zu existierenden Wörtern, Gültigkeitsdauer etc. sind Gegenstand solcher Passwortrichtlinien (engl. password policies). Die Definition dieser Richtlinien ist Bestandteil des Passwortmanagements. Eine Analyse der an dem IDMS partizipierenden Systeme kann dazu führen, dass mehrere Sicherheitsniveaus notwendig werden. Systeme mit sehr sensiblen Daten erfordern ein höheres Sicherheitsniveau und könnten längere Passwörter gekoppelt mit biometrische Daten oder Karten verlangen. Man spricht in diesem Zusammenhang auch von einem erweiterten Passwort- und Richtlinien-Management (engl. advanced password (and) policy management).

g) User Self-Service

Identitätsmanagementsysteme sollen zum einen zu einer Entlastung des IT-Personals führen und zum anderen benutzerfreundlich sein. Ermöglicht man es den Benutzern, auf einfache Weise einige persönliche Attribute wie Anschrift oder E-Mail-Einstellungen in ihrem Profil in Selbstbedienung zu ändern (engl. profile management), so kommt man diesem Ziel etwas näher. Man spricht auch vom User Self-Service. Innerhalb einer Organisation sind diverse Antragsverfahren vorstellbar, die im Hintergrund weitere Prozesse auslösen.

Beispiel:
Eine Mitarbeiterin der Personalstelle beantragt online anhand der Stellenbeschreibung für eine neue Mitarbeiterin bestimmte Rechte für IT-Systeme an der Hochschule. Die Rechte werden erst wirksam, wenn der Leiter der Fakultät, an der die neue Mitarbeiterin arbeitet, via Webinterface die Genehmigung erteilt.

Diese Funktionalität der Verwaltung von Antragsverfahren wird mit dem englischen Begriff request management oder allgemeiner workflow management bezeichnet.

h) Zentrale Administration

Die Zentrale Administration beinhaltet alle Funktionen, die zur Verwaltung des Identitätsmanagementsystems und der darin geführten Daten benötigt werden. Meist handelt es sich um eine große Sammlung von Werkzeugen mit visuellen Modellierungshilfen für die Architektur. Alle beschriebenen Funktionen müssen konfiguriert und überwacht werden. Um die Definition des Regelwerkes für das Provisioning zu erleichtern, können spezialisierte Konfigurationsprogramme verwendet werden.

i) Dezentrale Administration

Die zentralen Administratoren eines Identitätsmanagementsystems, die meist einer IT-Abteilung zugeordnet sind, können in dem komplexen Gebilde ihrer Organisation nicht alle Besonderheiten der einzelnen Bereiche kennen. Deshalb ist es sinnvoll, bestimmte Aufgaben des Identitätsmanagements an ausgewählte Verantwortliche in den einzelnen Abteilungen, so genannte lokale Administratoren, zu delegieren. Mit einer Komponente des IDMS zur dezentralen Administration können diese beispielsweise in die Lage versetzt werden, Identitäten oder Zugriffsrichtlinien für ihren Bereich zu verwalten. Funktionen, die Änderungen an der Architektur des Identitätsmanagementsystems zulassen, sind jedoch nicht Bestandteil dieser dezentralen Werkzeuge.

j) Workflowmanagement

In der Funktion des User Self-Service wurde bereits mit dem Szenario der Urlaubsbeantragung ein möglicher Arbeitsablauf, ein Workflow beschrieben, der an einer zentralen Stelle definiert werden sollte. Das Workflowmanagement muss die Möglichkeit bereitstellen, komplexe Abläufe zu konfigurieren. Dies beinhaltet unter Umständen auch diverse Funktionsaufrufe in anderen Systemen. Neben den Workflows auf der Dienstebene für Benutzer existieren auch Workflows im Bereich des Provisioning im Kern des IDMS. Visuelle Werkzeuge zur Einrichtung der Workflows sollten zur Verfügung gestellt werden.

k) Auditing

Allgemein drückt Auditing die Protokollierung und Analyse aller Ereignisse in einem komplexen System aus. Bei Identitätsmanagementsystemen stehen vor allem die Überprüfung der Authentifizierungen und Autorisierungen im Vordergrund. Zunächst einmal werden Daten über erfolgreiche und fehlgeschlagene Anmeldungen an den Systemen gesammelt. Anhand dieser Daten findet eine Analyse statt. Auffällige Ereignisse, wie systematische Benutzernameneingaben, werden herausgefiltert und gemeldet. Bezüglich der Autorisierung wird geprüft, ob die Zugriffsregeln auch eingehalten werden. Zu jeder Zeit muss die Frage beantwortet werden können, wer zu welcher Zeit Zugang zu welchen Systemen hatte. Dies beinhaltet ebenfalls eine Phase der Datensammlung und eine Phase der Analyse. Das Auditing ermöglicht auch die Nachvollziehbarkeit sämtlicher Schritte im Provisioning. Werden die genannten Funktionen von der Auditing-Komponente realisiert, so spricht man auch von einem revisionsfähigen IDMS. In einigen Wirtschaftszweigen existieren hierbei gesetzliche Auflagen, für deutsche Hochschulen jedoch nicht.

l) Monitoring

Das Monitoring prüft die beteiligte Hard- und Software auf Fehlfunktionen. Auf Hardwareebene sind dies beispielsweise die Server, Netzwerkkomponenten und Klimaanlagen. Auf Softwareebene werden die Verfügbarkeit und das Antwortverhalten der einzeln bereitgestellten Dienste getestet. Beim Provisioning können beispielsweise Fehler durch nicht erreichbare Systeme entstehen. Dies sollte sofort angezeigt werden, um je nach Situation entsprechende Maßnahmen ergreifen zu können. Die Statusüberwachung einer jeden Provisioning-Anfrage stellt eine zentrale Funktion des Monitorings dar. Auditing und Monitoring werden manchmal auch als eine Funktion angesehen.

m) Reporting

Das Reporting ist verantwortlich für die Erstellung von Statistiken über die gespeicherten Identitäten, die Zugriffe auf einzelne Dienste und die Auslastung der beteiligten Hardware. Hierfür müssen die Daten aus diversen Datenquellen zusammengetragen und aggregiert werden. Die Daten dienen vor allem den Administratoren sowie der Leitung der Organisation, die auf eventuelle Engpässe oder andere Missverhältnisse reagieren müssen.

Nicht funktionale Anforderungen

In den folgenden nicht funktionalen Anforderungen sind sowohl technische Anforderungen als auch Qualitätsanforderungen enthalten.

a) Sicherheit

In Identitätsmanagementsystemen werden sensible Daten gespeichert und über das Netzwerk transportiert. Hierzu gehören Personeninformationen, Benutzerkennungen sowie Passwörter. Die Nutzer des IDMS verlassen sich auf die Korrektheit und die vertrauliche Behandlung der Daten. Das Identitätsmanagementsystem stellt einen Verbund sich untereinander vertrauender IT-Systeme dar. Man spricht deshalb auch häufig von einem circle of trust. Wird eine der beteiligten Komponenten kompromittiert, besteht eine Gefahr für die gesamte Organisation. Verfahren zur Verschlüsselung der Kommunikation sowie einzelner Inhalte und der Integritätswahrung sind zwingende Anforderungen. Die Verschlüsselung auf Transportebene betrifft nicht nur die Authentifizierung. Die Replikations-, Synchronisations- sowie Backupprozesse müssen auch geschützt werden.

Die Netzstruktur innerhalb der Organisation beeinflusst ebenfalls die Sicherheit des IDMS. Eine genaue Planung über die Systeme die weltweit oder nur eingeschränkt erreichbar sein sollen, muss erfolgen. Die Analyse, Dokumentation und Strukturierung der genannten Prozesse stellt bereits einen ersten Sicherheitsgewinn dar.

b) Zuverlässigkeit

Ein Identitätsmanagementsystem übernimmt eine zentrale Rolle innerhalb der IT-Infrastruktur eines Unternehmens. Je mehr Systeme daran partizipieren, umso wichtiger ist es, dass das IDMS jederzeit seine Funktionen wahrnehmen kann. Ist das System nicht verfügbar, so können viele Prozesse zum Erliegen kommen. Ein IDMS sollte demnach eine gewisse Reife aufweisen, sollte also über die gesamte Zeit stabil und mit wenigen Fehlern laufen. Die Forderung nach Stabilität impliziert, dass das System eine gewisse Fehlertoleranz aufweist. Die beteiligten Komponenten zur Bildung eines Identitätsmanagementsys-

tems müssen redundant ausgelegt sein. Hierzu gehören insbesondere die beteiligte Hardware (Server, Netzwerkkomponenten), die Anwendungen und qualifiziertes Personal. Eine schnelle Wiederherstellbarkeit im Falle eines Ausfalls zeichnet ein gutes IDMS aus. Viele unberechenbare Ereignisse müssen im Rahmen einer Risikoanalyse betrachtet werden. Es gilt diese Risiken zu identifizieren, zu klassifizieren und je nach Gefährdungsgrad entsprechende Maßnahmen zu ergreifen. Eine Ermittlung der Kosten, die durch entsprechende Sicherung der Systeme entstehen, darf auch nicht außer Acht gelassen werden. Häufig führen die stark begrenzten Budgets an den Hochschulen dazu, dass auf bestimmte Maßnahmen verzichtet wird, obwohl ein erhöhtes Risiko besteht. Es gilt immer, den politischen Schaden sowie die Gefährdung der Daten gegen die Kosten abzuwägen. Die Verfügbarkeit der Systeme kann auch durch eine zu hohe Systemlast eingeschränkt sein. Verfahren zur Lastverteilung (load balancing) sollten bei der Planung berücksichtigt werden. Bei denial-of-service (DOS) Angriffen, die meist aus fremden Netzen kommen, handelt es sich um unzählige und zum Teil nicht standardkonforme Anfragen. Auch diese Attacken können die Verfügbarkeit eines Systems stark einschränken.

c) Benutzerfreundlichkeit

Die Akzeptanz eines Identitätsmanagementsystems ist von den Komponenten abhängig, die der Anwender benutzt, um eine bestimmte Funktionalität zu erhalten. Neben der Sicherheit und Zuverlässigkeit müssen die Schnittstellen zu den Anwendern leicht verständlich und einfach erlernbar sein. Zu den Anwendern gehören die allgemeinen Nutzer, Personen, an die einige dezentrale Aufgaben delegiert wurden, Administratoren sowie Entwickler. All diese Gruppen müssen mit wenig Aufwand, also möglichst effizient, ihre Arbeit an dem IDMS durchführen können.

d) Flexibilität

Die Einführung zu dieser Arbeit hat bereits verdeutlicht, welche heterogenen Strukturen an einer Hochschule vorliegen. Jede Hochschule besitzt ihre Besonderheiten. Bei anderen Organisationen gilt Vergleichbares. Ein Identitätsmanagementsystem sollte sich leicht in eine Vielzahl heterogener IT-Infrastrukturen einbinden lassen. Vorhandene Systeme müssen leicht integrierbar sein. Dies gilt zum Beispiel für die Datenquellen, die einen wichtigen Beitrag für die Bildung der Identitäten liefern. Organisationen sind keine starren Konstrukte. Änderungen in der Organisationsstruktur sowie Einführung neuer und Ablösung alter Anwendungen führen zu einer Dynamik, die eine einfache Anpassbarkeit des IDMS erfordern. Eine modulare Architektur sorgt für eine „… evolution without revolution …“[6].

e) Verwendung von Standards

Eng verbunden mit der Forderung nach Flexibilität ist die Verwendung von Standards. Standardisierte Protokolle vereinfachen die Integration und den Austausch einzelner Komponenten. Eine meist große Gemeinschaft von Entwicklern liefert wichtige Erfahrungen und Hilfestellungen zu verfügbaren Softwareprodukten und Programmierbibliotheken. Standardisierte Prozesse und Architekturen helfen dabei, dass sich neue Entwickler, Administratoren und andere Beteiligte schnell in das System hineinfinden können und dass einzelne Entwicklungen auch für andere Organisationen zur Verfügung gestellt werden können. Eine Reihe von Standards für Protokolle und Architekturen im Bereich des Identitätsmanagements entwickelten in den letzten Jahren die Organization for the Advancement

[6] [Orac06], S. 10

of Structured Information Standards (OASIS, http://www.oasis-open.org/) und das Liberty Alliance Project (LAP, http://www.projectliberty.org), an dem sich über 150 Firmen beteiligen.

2.2.4 Erfolgsfaktoren

Die Einführung eines IDMS wird nicht nur durch technische Probleme bestimmt. Im Vorfeld müssen eine Reihe von Prozessen analysiert werden. Eine Betrachtungsebene stellen die Datenflüsse dar. Neben der Frage, wo welche Daten erfasst werden, muss auch geklärt werden, wer welche Daten benötigt. Die einzelnen Mitgliedsgruppen benötigen bestimmte IT-Dienste für ihre tägliche Arbeit. Des Weiteren existieren für die einzelnen Benutzerklassen unterschiedliche Rechte, Kompetenzen und Verantwortlichkeiten. In Zusammenarbeit mit der Universitätsleitung und den organisatorisch, inhaltlich und technisch verantwortlichen Mitarbeitern einzelner Bereiche müssen diese Prozesse definiert und umgesetzt werden. Die allgemeine Bereitschaft aller Mitarbeiter hierfür ist ein wesentlicher Erfolgsfaktor für ein Identitätsmanagementsystem. Es muss vielen Mitarbeitern bewusst sein, dass Änderungen von Daten und Prozesse über die Grenze der aktuellen Anwendung hinaus wirksam werden können.

2.3 Datenmodelle für Datenbanken

E. F. Codd beschreibt ein Datenmodell für Datenbanken als eine Kombination aus einer Sammlung von „data structure types“[7], „operators or inferencing rules“[8] und „general integrity rules“ [9].

Im Folgenden werden die bekannten Datenmodelle für Datenbanken kurz aufgeführt:

- Relationales Datenmodell
  Im relationalen Datenmodell wird die Realwelt mit Hilfe von Relationen modelliert. Dabei können die Relationen als Tabellen aufgefasst werden, die Entitäten und Beziehungen enthalten.

- Hierarchisches Datenmodell
  Im hierarchischen Datenmodell wird die Realwelt mit Hilfe von Baumstrukturen abgebildet. Jeder Datensatz entspricht einem Knoten innerhalb der hierarchischen Struktur. Die Knoten besitzen dabei maximal einen Vorgänger.

- Netzwerkdatenmodell
  Mit dem hierarchischen Datenmodell lassen sich bestimmte Aspekte der Realwelt nur schwierig oder gar nicht abbilden (Beispiel: *Ein Mitarbeiter arbeitet in mehreren Projekten mit.).* Netzwerkdatenmodelle erweitern das hierarchische Datenmodell, in dem sie mehrere Vorgänger für einem Knoten zulassen. So entsteht letztlich ein Graph.

[7] [Codd80], S.112
[8] [Codd80], S.112
[9] [Codd80], S.112

- Objektorientiertes Datenmodell
  Im objektorientierten Datenmodell bilden Objekte abgeschlossene Einheiten. Objekte enthalten dabei eine Sammlung von zusammenhängenden Attributen. Die Ausprägung dieser Attribute bestimmt den Zustand eines Objektes. Der Zugriff auf Objekte erfolgt über festgelegte Methoden. Durch Verweise auf andere Objekte entsteht ein Netz von Objekten.

- Objektrelationales Datenmodell
  Das objektrelationale Datenmodell stellt ein Bindeglied zwischen dem relationalen Datenmodell und dem objektorientierten Datenmodell dar.

Eine detaillierte Behandlung des Themas Datenbankmodelle findet man beispielsweise in [Voss00] oder [Kris92].

2.4 Verzeichnisdienste und LDAP

Sehr verbreitet für die Speicherung von Identitäten, die sich auf Personen oder Benutzerkonten beziehen, sind Verzeichnisdienste (engl. directory services). In Verzeichnisdiensten sind die Daten hierarchisch organisiert. Dabei stellt ein Knoten in der Regel ein Objekt dar. Es liegt also meist ein objektorientiertes Datenmodell zugrunde, bei dem die Objekte durch ihre Beziehungen eine Baumstruktur bilden.

Verzeichnisdienste sind häufig für den lesenden Zugriff optimiert und werden in Bereichen eingesetzt, in denen eine sehr hohe Zahl von Anfragen pro Tag erwartet wird.

Verzeichnisdienste besitzen gegenüber den sehr verbreiteten relationalen Datenbanken meist einige Schwächen. So unterstützen zum Beispiel viele Verzeichnisdienste keine Transaktionen und können referenzielle Integrität nicht sicherstellen.

Im Zusammenhang mit Verzeichnisdiensten werden oft die Begriffe Shared Directory, Metadirectory und Virtual Directory genannt:

- Shared Directory
  Ein Shared Directory ist ein Verzeichnis(-dienst), in dem Informationen enthalten sind, die von mehreren unterschiedlichen Systemen genutzt werden.

- Metadirectory
  Ein Metadirectory stellt eine Ansammlung von Informationen dar, die aus mehreren Verzeichnissen zusammengetragen werden. Ziel ist die Realisierung einer integrierten Sicht auf die gesamten Informationen. Metadirectories replizieren in der Regel die Daten aus den beteiligten Verzeichnissen. Man spricht von materialisierter Integration.

- Virtual Directory
  Ein Virtual Directory stellt wie ein Metadirectory eine integrierte Sicht auf Informationen mehrerer Verzeichnisse her. Jedoch werden die Daten in Echtzeit aus den Verzeichnissen abgerufen. Eine permanente Speicherung von Informationen findet also im Gegensatz zu Metadirectories nicht statt.

Die zwei bedeutendsten Standards im Bereich Verzeichnisdienste stellen die Standards X.500 und LDAP dar. X.500 ist eine von der ITU[10] spezifizierte sehr umfangreiche Familie von Standards, die auf dem ISO/OSI-Modell[11,12] aufsetzen. Dabei wird unter anderem das Directory Access Protocol (DAP) festgelegt. Das Lightweight Directory Access Protocol (LDAP) hingegen ist ein auf dem Client/Server-Paradigma basierendes Anwendungsprotokoll des TCP/IP-Stacks und setzt nur eine Auswahl der Funktionen um, die in DAP spezifiziert wurden. LDAP ist als Zugriffsprotokoll für Verzeichnisdienste sehr verbreitet und wird im Folgenden etwas detaillierter dargestellt.

LDAP

Das Lightweight Directory Access Protocol (LDAP) wurde 1993 in [Yeon93] erstmalig spezifiziert. Es folgten 1995 die Version 2 (vgl. [Yeon95]) und 1997 die aktuelle Version 3 (vgl. [Wahl97]).

Ein Objekt repräsentiert in LDAP einen Verzeichnisdiensteintrag (engl. directory entry). Dabei stellt das Objekt eine Instanz einer oder mehrerer Objektklassen dar. Die Objektklassen definieren wiederum einen der Realwelt entsprechenden zusammengehörigen Satz von obligatorischen und optionalen Attributen. Für die Attribute wird festgelegt, wie die Werte verglichen werden sollen und wann eine Übereinstimmung besteht. Objektklassen, Attribute und die Syntax für Objektklassen und Attribute erhalten einen weltweit eindeutigen Identifier. Man spricht auch vom Object Identifier (OID). In einem Schema werden die Definitionen der Attribute und Objektklassen festgehalten.

Jedes Objekt im Baum des Verzeichnisdienstes, im so genannten Directory Information Tree (DIT), erhält einen eindeutigen Namen. Dieser Distinguished Name (DN) gibt den Pfad innerhalb des DIT von der Wurzel bis zu dem entsprechenden Objekt an.

Beispiel für einen Distinguished Name:
***ou**=Informatik, **ou**=Fakultät für Mathematik und Informatik, **o**=fernuni-hagen, **c**=de*

Das Beispiel zeigt, dass der Distinguished Name vom Blatt zur Wurzel gelesen wird. Der DN setzt sich aus den Knoten zusammen, die durch Kommata voneinander getrennt werden. Jeder Knoten erweiterten dabei den Pfad seines Vorgängers um eine weitere Komponente, die man Relative Distinguished Name (RDN) nennt. Der RDN besteht aus einem Attribut des Objektes mit zugehörigem Wert. In dem genannten Beispiel ist ***ou**=Informatik* ein Relative Distinguished Name.

LDAP stellt unter anderem die elementaren Operationen wie add, modify, delete und search bereit. Darüber hinaus werden noch Operationen zum Verbinden, Operationen für verschiedene Authentifizierungsverfahren, Operationen zum Vergleichen sowie Operationen zur Veränderung der Baumstruktur angeboten. Eine detaillierte Beschreibung der genannten und weiterer Operationen können unter anderem in [LiUn06] und [KlLa03] nachgelesen werden. In der angegeben Literatur sind auch praktische Beispiele enthalten.

[10] ITU = International Telecommunication Union, URL: http://www.itu.int
[11] ISO = International Organization for Standardization, URL: http://www.iso.org
[12] OSI = Open Systems Interconnection

2.5 Grundlagen zur Datenintegration

Ein Identitätsmanagementsystem, das an einer Hochschule eingeführt wird, setzt in der Regel auf bereits existierenden Anwendungen zur Verwaltung von Identitäten auf. Diese verteilten autonomen Anwendungen erfassen eigenverantwortlich Identitätsdaten. Mit der Einbindung in die Struktur des Identitätsmanagementsystems müssen sie jedoch einen Teil ihrer Autonomie aufgeben und übernehmen die Funktion einer autoritativen Datenquelle. Es liegt der Fall einer heterogenen Postintegration von Daten vor.

Autoritative Datenquelle: Ein System, in dem bestimmte Arten von Identitäten in einer hohen Qualität manuell erfasst werden und das die eingegebenen Daten teilweise oder vollständig anderen Komponenten in einem Identitätsmanagementsystem zur Verfügung stellt, wird als autoritative Datenquelle bezeichnet. Ausschließlich dieses System ist verantwortlich für die Erfassung dieser Identitätsarten.

Fehler bei der Datenerfassung in den autoritativen Quellen können Auswirkungen auf andere Systeme in der Hochschule haben und dort neue Fehler produzieren. Änderungen in den Datenquellen auf beispielsweise technischer, struktureller oder semantischer Ebene müssen im Vorfeld kommuniziert werden. Gegebenenfalls ist eine Anpassung von Komponenten des IDMS notwendig.

2.5.1 Datenintegration versus Prozessintegration

Bei Identitätsmanagementsystemen kommen Formen sowohl der Datenintegration als auch der Prozessintegration zum Einsatz.

Datenintegration: Die Datenintegration basiert auf einem bereits existierenden Datenbestand und versucht diesen entweder zu integrieren oder eine integrierte Sicht darauf zu schaffen.

Prozessintegration: „In der Prozessintegration steht ... die Verknüpfung von Prozessen zur Laufzeit im Vordergrund.“[13]

Beispiel Datenintegration:
Eine Anwendung für die Hotline benötigt zu dem Benutzerkonto eines Mitarbeiters einige Informationen. Die Personalnummer, die E-Mailadresse und die Zugangskartennummer werden hierfür aus mehreren Datenbanken gelesen und dem Hotliner präsentiert.

Beispiel Prozessintegration:
Ein Mitarbeiter wird an der Hochschule neu eingestellt. Die Sachbearbeiterin trägt die Person in ein System zur Mitarbeiterverwaltung ein. Dies löst entweder sofort (synchron) oder zu einem späteren Zeitpunkt (asynchron) mehrere Prozesse im IDMS aus. Für den Mitarbeiter wird eine Benutzerkennung erzeugt, im E-Mail-System wird ein Postfach angelegt und an einer anderen Stelle wird eine Zugriffskarte für bestimmte Gebäude und Räume präpariert.

Das Beispiel zur Prozessintegration zeigt, dass zur Abbildung eines Prozesses Aktionen in mehreren IT-Systemen ausgelöst werden müssen. Der Aufruf der hierfür notwendigen Funktionen in den jeweiligen Anwendungen kann mit Nachrichten, die zwischen den IT-Systemen ausgetauscht werden, oder so genannten Remote Procedure Calls (RPC) reali-

[13] [LeNa07], S. 401

siert werden. Man spricht allgemein von Enterprise Application Integration (EAI), wenn Softwareprodukte zur Integration von Prozessen eingesetzt werden.

Prozessintegration beinhaltet immer Aspekte der Datenintegration. Damit wie in dem oben genannten Beispiel ein Benutzerkonto angelegt werden kann, müssen mit einer Nachricht auch Identitätsdaten transportiert werden. Dies erfordert wiederum eine Extraktion von Daten aus einer Datenbank und deren Transformation in eine für das empfangende System verständliche Form.

2.5.2 Problemfelder der Datenintegration

Leser und Naumann diskutieren in [LeNa07] die Grundprobleme Verteilung, Autonomie und Heterogenität, die bei der Integration von Daten bestehen. Diese Problemfelder können bei der Integration unabhängig voneinander auftreten und lassen sich daher als zu einander orthogonale Dimensionen darstellen.

Verteilung

Die Verteilung der Daten lässt sich nach zwei Arten klassifizieren, die physische und die logische Verteilung.

Physische Verteilung: „Daten sind physisch verteilt, wenn sie auf physisch getrennten und damit meist auch geographisch entfernten Systemen verwaltet werden.“[14] Physische Verteilung wird auch Allokation genannt.

Aus der physischen Verteilung ergeben sich mehrere Probleme. Die Systeme, in denen die Daten vorliegen, müssen zunächst identifiziert und zu jeder Zeit lokalisierbar sein. Die Schemata, die für die Speicherung der Daten in den jeweiligen Systemen verwendet werden, unterscheiden sich in aller Regel. Jeder Zugriff über ein Netzwerk erzeugt Datenverkehr. Eine physische Verteilung erfordert deshalb auch eine Optimierung der Zugriffe über das Netzwerk.

Logische Verteilung: „Daten sind logisch verteilt, wenn es für ein Datum mehrere mögliche Orte zu seiner Speicherung gibt.“[15]

Logische Verteilung lässt sich gut anhand von zwei Tabellen in einer Datenbank verdeutlichen:
Zwei Tabellen nehmen Personendaten auf und besitzen die gleichen Felder Name, Vorname, Titel und Geschlecht. In einer Tabelle werden alle Mitarbeiter, in der anderen alle Dozenten einer Hochschule gespeichert. Dozenten können Mitarbeiter sein oder durch Honorarverträge einen Lehrauftrag erhalten. D. Hellmann, ein Mitarbeiter der Fakultät Informatik, bietet unter anderem Vorlesungen im Bereich der Kryptographie an. Sein Datensatz ist also in beiden Tabellen enthalten. Es liegt somit eine logische Verteilung vor.

Charakteristisch bei der logischen Verteilung ist also eine Überlappung von Einträgen. Ein Datum, das an mehreren Orten gespeichert wird, muss auch an all diesen Orten aktualisiert werden. Ansonsten entstehen Inkonsistenzen.

[14] [LeNa07], S. 51
[15] [LeNa07], S. 51

Bei Replikationen zur Lastverteilung und Ausfallsicherheit liegen beide Formen der Verteilung vor. Auf unterschiedlicher Hardware an geographisch getrennten Orten (physische Verteilung) liegen die identischen Datenbanktabellen mit den gleichen Einträgen vor (logische Verteilung).

Autonomie

Bei der Einführung eines Identitätsmanagements existieren innerhalb einer Organisation bereits viele Systeme, die integriert werden sollen. Diese Systeme werden in der Regel eigenverantwortlich durch die einzelnen Abteilungen betrieben. Die dadurch entstehenden Autonomien lassen sich folgendermaßen klassifizieren:

- Designautonomie
- Schnittstellenautonomie
- Zugriffsautonomie
- Juristische Autonomie

Designautonomie: Die Designautonomie beinhaltet die eigenverantwortliche Festlegung von Datenmodellen, Schemata, Datentypen, Datenformaten, Begriffsdefinitionen und Semantik innerhalb eines Datenbanksystems.

Schnittstellenautonomie: Schnittstellenautonomie liegt dann vor, wenn ein System die Schnittstellen zum Datenaustausch selbst vorgibt.

Mögliche Schnittstellen können beispielsweise auf SOAP[16] oder ODBC[17] basieren. Im ungünstigen Fall stehen nur Import- und Exportmechanismen auf Basis von Textdateien zur Verfügung.

Zugriffsautonomie: Bei der Zugriffsautonomie wird vom System vorgegeben, welche Arten der Authentifizierung und Autorisierung verwendet werden, um zugreifen zu können.

Juristische Autonomie: Besitzen die Betreiber eines Systems das Recht, die Integration ihrer Daten zu verbieten, so spricht man von juristischer Autonomie.

Beispiel:
Die Leiterin der Studierendenverwaltung könnte der Auffassung sein, dass die dort erfassten Daten auf keinen Fall das System verlassen dürfen. Sie kann datenschutzrechtliche Bedenken äußern und so einen langwierigen juristischen Entscheidungsprozess initiieren.

Es wird an diesem kleinen Bespiel zur juristischen Autonomie schnell klar, dass die Einführung eines Identitätsmanagementsystems auch langwierige Überzeugungsarbeit bedeuten kann.

Heterogenität

Heterogenitäten zwischen den Datenquellen sowie zwischen Datenquelle und den Integrationskomponenten des Identitätsmanagementsystems existieren auf vielen Ebenen. Diese lassen sich folgendermaßen klassifizieren:

[16] Siehe Abschnitt 2.11.1 „SOAP, WSDL und UDDI“
[17] ODBC = Open Database Connectivity

a) Technische Heterogenität

Beim Zugriff auf eine Datenquelle ist eine Vielzahl von technischen Heterogenitäten zu überwinden. Auf die Datenquelle kann man zum einen über das Netzwerk zugreifen. Zum anderen ist auch ein lokaler Zugriff, beispielsweise auf eine Textdatei, möglich. In der nachfolgenden Beschreibung der technischen Heterogenität wird davon ausgegangen, dass die Kommunikation mit der Datenquelle über ein Netzwerk erfolgt. Nachdem man eine Verbindung zu der Datenquelle hergestellt hat, erfolgt der Nachrichtenaustausch über verschiedene Kommunikationsprotokolle. Protokolle können beispielsweise JDBC[18], LDAP oder SOAP sein. Die Informationen, die ausgetauscht werden, liegen in diversen Textformaten wie XML oder binär vor. Eingebettet in das Austauschformat sind die Anfragen an die Datenquelle und die dazugehörigen Ergebnisse. Ein Beispiel für eine Anfragesprache ist SQL. Die genannten Heterogenitäten bezeichnet man auch mit Schnittstellenheterogenität. Die Authentifizierungs- und Autorisierungsvarianten bilden eine weitere Unterart der technischen Heterogenität. Man spricht auch von Zugriffsheterogenität.

b) Syntaktische Heterogenität

Die syntaktische Heterogenität umfasst die Darstellungsvarianten gleicher Inhalte. Hierzu gehören unterschiedliche Zeichenkodierungen, die bei Identitätsmanagementsystemen oft große Probleme bereiten. Kodierungen mit einem kleinen Satz an Zeichen, wie es bei ASCII der Fall ist, müssen nach sehr umfangreichen Zeichensätzen wie Unicode transformiert werden. Diese Richtung ist noch einfach, der Rückweg ist in der Regel mit Verlusten behaftet. Unterschiedliche binäre Zahlenformate (z.B. big endian und little endian) sowie unterschiedliche Trennzeichen (z.B. Tabulatoren und Semikolons in Textdateien) sind auch Merkmale der syntaktischen Heterogenität. Sie lassen sich jedoch meist einfach ineinander umwandeln.

c) Datenmodellheterogenität

Datenmodellheterogenitäten existieren auf der Ebene der Datenquelle, des Datenaustausches und des Integrationssystems.

Beispiel:
Die Datenquelle basiert auf einem relationalen Datenmodell. Für den Datenaustausch wird eine XML-Struktur verwendet, die ein hierarchisches Datenmodell darstellt. Das Integrationssystem beinhaltet ein objektorientiertes Datenmodell.

Auch semantische Aspekte gehen mit der Heterogenität auf Datenmodellebene einher. In objektorientierten Datenmodellen können Beziehungen in Form von Spezialisierungen und Generalisierungen hergestellt werden. Relationale Datenmodelle beinhalten diese Konzepte hingegen nicht und so lassen sich Vererbungsprinzipien schwer oder gar nicht abbilden. Es geht eventuell ein Teil der Semantik verloren, denn die Beziehungen sind Bestandteil der Semantik.

[18] JDBC = Java Database Connectivity

d) Strukturelle Heterogenität

Verwenden zwei Datenquellen das gleiche Datenmodell, aber verschiedene Schemata, um einen bestimmten Aspekt aus der realen Welt darzustellen, so spricht man von struktureller Heterogenität. Eine der Hauptursachen dafür liegt in der Designautonomie der Entwickler. Bestimmte Sachverhalte lassen sich in mehreren Varianten abbilden. Teilweise entscheidet der Geschmack der Entwickler darüber, welches Schema verwendet wird. Andererseits kann ein normalisiertes Schema bewusst denormalisiert werden, um eine Anfrageoptimierung zu erhalten. Ein beispielsweise mehrere Millionen Male am Tag aufgerufener JOIN-Operator für zwei große Tabellen einer relationalen Datenbank kann durch das Zusammenführen der Tabellen vermieden werden. Auch die strukturelle Heterogenität kann wie bei der Datenmodellheterogenität zu Unterschieden in der Semantik führen. Als Spezialfall der strukturellen Heterogenität gilt die schematische Heterogenität. Hierunter versteht man, dass gleiche Sachverhalte durch Beziehungen, Attribute oder Werte ausgedrückt werden können und so jeweils ein anderes Schema existiert.

Beispiel:
Eine Person kann auf verschiedene Weise zu einem Mitglied der Hochschule werden. Es werden die Mitgliedsgruppen Studierende, Mitarbeiter und Alumni (Ehemalige) betrachtet. Die Personen werden in einer Tabelle einer relationalen Datenbank mit den wichtigsten persönlichen Attributen gespeichert. Die Mitgliedschaft wird nun in drei verschiedenen Formen modelliert. Im ersten Fall existiert eine zweite Tabelle, in der die Mitgliedsgruppen in Listenform enthalten sind. Über eine n:m-Relation werden nun die Personen den einzelnen Gruppen zugeordnet (Modellierung als Beziehungen). Die zweite Variante sieht in der Personentabelle pro Mitgliedsgruppe eine eigene Spalte mit dem Datentyp boolean vor. Die Mitgliedzugehörigkeit wird mit true oder false ausgedrückt (Modellierung als Attribute). Die letzte Möglichkeit enthält nur eine Spalte in der Personentabelle. Die Mitgliedgruppen werden einfach der Reihe nach durch Semikolon getrennt in die Spalte geschrieben. Denkbar ist auch eine Bitmaske, bei der jedes Bit für eine Mitgliedsgruppe steht (Modellierung als Werte).

Die Abfragen und Aktualisierungen bestimmter Informationen, wie in dem Beispiel die Mitgliedszugehörigkeit pro Person, sehen je nach Modellierungsform sehr unterschiedlich aus. Auch die Erweiterungen eines Sachverhaltes, wie eine neue Mitgliedsgruppe, gestalten sich sehr mannigfaltig. In dem ersten Fall des Beispiels wird einfach ein neuer Eintrag in die Tabelle der Mitgliedsgruppen eingefügt. Bei der zweiten Modellierungsvariante muss hingegen eine neue Spalte in die Personentabelle eingefügt werden.

e) Semantische Heterogenität

„Daten werden für einen menschlichen Benutzer erst durch Interpretation zu Information, und diese Interpretation erfordert sowohl Wissen über die konkrete Anwendung als auch Weltwissen“.[19]

Ein passendes Beispiel zu diesem Zitat ist die Darstellung von Noten in einem System:
Die Zahl 10 als Note besitzt alleine wenig Aussagekraft. Es kann sich hierbei um Leistungspunkte zu einem Bachelorkurs handeln, es könnte aber auch eine ganzzahlige Darstellung der Note 1,0 in einer Diplomprüfung sein.

[19] [LeNa07], S. 73

Die häufigsten Konflikte, die sich durch semantische Heterogenitäten ergeben, sind Synonyme und Homonyme. Die Definitionen der beiden Begriffe sind aus [Lehn03], S. 126 entnommen:

Synonyme: Zwei Beschreibungen werden als Synonyme bezeichnet, wenn sie den gleichen Sachverhalt in der realen Welt beschreiben. Mit den Wörtern männlich/weiblich und maskulin/feminin wird beispielsweise jeweils das entsprechende Geschlecht der Person ausgedrückt.

Homonyme: Ein Begriff wird Homonym bezeichnet, wenn er mindestens zwei unterschiedliche Sachverhalte in der realen Welt beschreibt. Das Wort null steht entweder für den Zahlenwert oder für die Tatsache, dass kein Wert existiert.

Auch Hyperonyme spielen eine Rolle. Sie werden als Oberbegriffe definiert. Zum Beispiel ist Obst ein Hyperonym zu dem Begriff Apfel.

Die Auflösung semantischer Heterogenitäten erfordert eine Reihe von Informationen. Die Schemata liefern einen ersten Beitrag zu dem Kontext der Daten und liegen meist in lesbarer Form vor. Weitere Hilfen stellen die Anwendung zu diesem Schema, die Entwickler und Benutzer dar.

Transparenz

Mit der Lösung der einzeln genannten Probleme wird unter anderem versucht, Transparenz herzustellen. Schafft man es, die physische Verteilung durch das integrierte Informationssystem zu verstecken, so spricht man von Ortstransparenz. Nicht immer ist die Herstellung von Transparenz gewünscht. So möchte man beispielsweise mit einem bestimmten Protokoll direkt auf eine Schnittstelle zugreifen, um dadurch eine höhere Datentransferrate zu erzielen.

2.5.3 Metadaten

Metadaten: „Unter dem Begriff Metadaten versteht man gemeinhin jede Art von Informationen, die für den Entwurf, die Konstruktion und die Benutzung eines Informationssystems benötigt wird.“[20]

Identitätsmanagementsysteme, die auch als Informationssysteme aufgefasst werden können, benötigen Metadaten in vielen Bereichen:

Beispiel:
Das IDMS enthält als Komponente eine Datenbank, in der alle Identitäten der Hochschule gespeichert werden. Es existieren Metadaten, in denen festgelegt wird, welche Computer sich mit der Datenbank über das Netzwerk verbinden dürfen. Die Beschreibung des Datenbankschemas stellt ebenfalls eine Form von Metadaten dar.

Eine weitere Behandlung des Themas Metadaten ist im Rahmen dieser Arbeit nicht möglich. Eine ausführliche Darstellung erfolgt beispielsweise in [BaGü04] und [LeNa07].

[20] [BaGü04], S. 328

2.5.4 Architekturen zur Datenintegration

Für die Integration von Daten existieren viele Architekturen. Sie sind im Wesentlichen durch den Heterogenitätsgrad der Datenquellen und den Zeitpunkt der Datenintegration geprägt. Findet eine Integration von Datenquellen statt, nachdem diese eingeführt wurden, so liegt der Fall der Postintegration vor. Im Gegensatz dazu steht die Preintegration.

Den speziellen Architekturen liegen bestimmte Grundprinzipien für die Datenintegration zu Grunde:

- Virtuelle versus materialisierte Datenintegration
  Im virtuellen Fall wird, beispielsweise ausgelöst durch eine Anfrage, nur ein Ausschnitt von Daten zu dem integrierten Informationssystem transportiert, um nach kurzer Zeit wieder verworfen zu werden. Es werden die Quellsysteme also direkt genutzt. Bei der materialisierten Integration werden die kompletten Daten von dem Quellsystem in ein Zielsystem kopiert und dort dauerhaft gespeichert.

- Passive Schnittstelle/Pull-Modus versus aktive Schnittstelle/Push-Modus
  Löst die Integrationskomponente eine Anfrage aus, so spricht man von einer passiven Schnittstelle oder Pull-Modus. Bei Datenveränderungen kann auch die Datenquelle den Datentransport initiieren. Es handelt sich dann um eine aktive Schnittstelle oder um einen Push-Modus.

- Synchrone versus asynchrone Datenverarbeitung
  Werden die Daten sofort verarbeitet und erfolgt eine direkte Rückmeldung über das Ergebnis der Verarbeitung, so spricht man von einem synchronen Vorgang. Im asynchronen Fall können die Verarbeitung der Daten und die Ergebnisübermittlung zeitlich weit auseinander liegen. Die beteiligten Systeme sind darauf vorbereitet und können sich zwischenzeitlich anderen Aufgaben widmen.

Da Identitätsmanagementsysteme an einer Hochschule eine Integration bereits bestehender sehr heterogener Systeme beinhalten (heterogene Postintegration), wird nur auf die dazu passenden Architekturen eingegangen. Ausführlich werden Architekturen zur Datenintegration in [BaGü04], [LeNa07] und [ERS99] vorgestellt.

Föderierte Datenbanksysteme

Mit föderierten Datenbanksystemen (FDBS) werden Datenquellen integriert, die bereits vor dem Integrationsprozess existieren und untereinander Heterogenitäten aufweisen können. Dabei ist den Datenquellen bekannt, dass sie Bestandteil eines föderierten Systems sind und somit einen Teil ihrer Autonomie aufgeben müssen.

Isolierte Datenbanksysteme weisen in der Regel eine Drei-Schichten-Architektur auf [vgl. ANSI 75]. Diese besteht im Wesentlichen aus den Komponenten internes Schema, konzeptionelles Schema und mehrere externe Schemata. Abbildung 2 verdeutlicht den Aufbau der Drei-Schichten-Architektur.

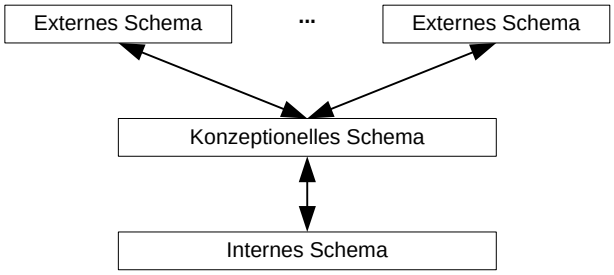

Abbildung 2: Drei-Schichten-Architektur eines autonomen Datenbanksystems

Mit dem konzeptionellen Schema werden auf logischer Ebene die gesamten Daten beschrieben, die innerhalb des Datenbanksystems verwaltet werden sollen. Es entsteht durch Abstraktion der Aspekte, die es aus der Realwelt zu modellieren gilt. Das konzeptionelle Schema ändert sich in der Regel nach der Erstellung gar nicht mehr oder nur im geringen Umfang. Es stellt einen relativ stabilen Bezugspunkt für die externen Schemata dar. Die externen Schemata sorgen dafür, dass der Benutzer einer Datenbank nur einen Ausschnitt der Daten sieht. Dabei muss die Struktur der Daten nicht dem konzeptionellen Schema entsprechen. Mit dem internen Schema legt man fest, wie die logische Struktur der Daten aus dem konzeptionellen Schema auf physischer Ebene gespeichert werden sollen.

In föderierten Datenbanksystemen wird ein globales konzeptionelles Schema verwendet. Wie das konzeptionelle Schema in der Drei-Schichten-Architektur stellt das globale konzeptionelle Schema auch einen Bezugspunkt dar. Es ist jedoch Bestandteil einer erweiterten Schichten-Architektur, die in Abbildung 3 dargestellt wird.

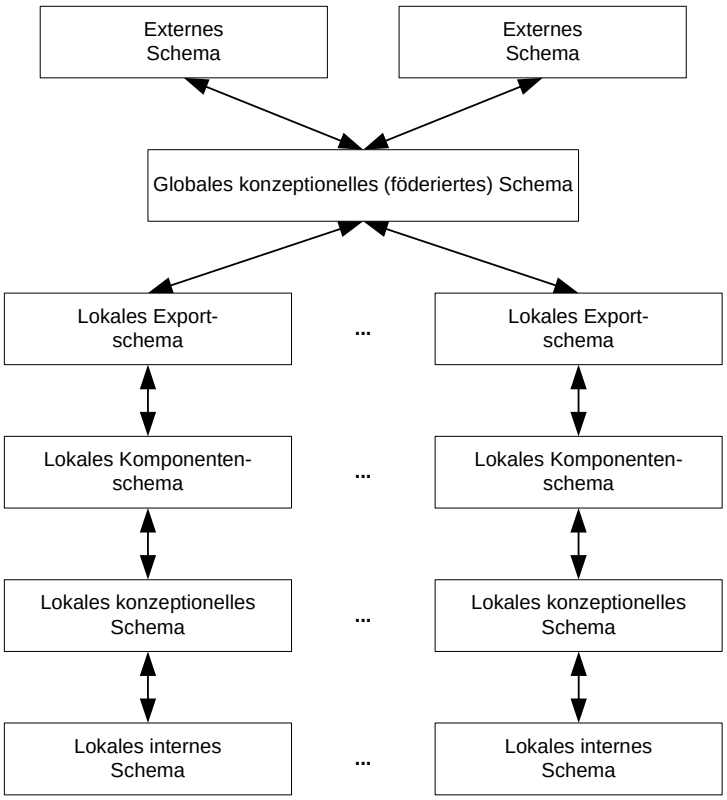

Abbildung 3: Schichten-Architektur im föderierten Datenbanksystem[21]

[21] Abbildung nach [LeNa07], S. 95

Das lokale interne Schema und das lokale konzeptionelle Schema entsprechen dem jeweiligen Schema aus der Drei-Schichten-Architektur. Das lokale Komponentenschema entsteht aus der Überführung des lokalen konzeptionellen Schemas in das Datenmodell, das für das globale konzeptionelle Schema verwendet wird. Man spricht in diesem Zusammenhang auch vom kanonischen Datenmodell. Das lokale Exportschema sorgt dafür, dass nur eine Teilmenge des lokalen Komponentenschemas zu sehen ist.

Das globale konzeptionelle Schema stellt basierend auf den lokalen Exportschemata eine integrierte Sicht auf die Datenquellen her. Es kann auf zwei Arten entstehen. Bei dem Bottom-up-Entwurf wird ein Schemaintegrationsprozess durchgeführt. Dabei erfolgt eine Analyse und anschließende Zusammenfassung der Schemata all derjenigen Datenquellen, die integriert werden sollen. Bei dem Top-down-Entwurf wird unabhängig von den Datenquellen ein globales konzeptionelles Schema entworfen. Im Anschluss ist eine Abbildung des globalen Schemas auf die lokalen Exportschemata durchzuführen. Man spricht auch vom Schema Mapping.

Anwendungen nutzen externe Schemata, um auf das globale konzeptionelle Schema zuzugreifen. Dabei stellen die externen Schemata einen Ausschnitt aus dem konzeptionellen Schema dar. In Abbildung 3 werden nur zwei externe Schemata dargestellt. Die Anzahl der externen Schemata kann jedoch beliebig sein. Insbesondere muss sie nicht mit der Anzahl der integrierten Datenquellen korrespondieren.

Der Exkurs zu den föderierten Datenbanksystemen zeigt, dass die zu integrierenden Datenquellen nicht ihr lokales konzeptionelles Schema ändern müssen. Das lokale Komponentenschema und das lokale Exportschema können im föderativen Verbund von der Datenquelle selbst realisiert werden. In diesem Fall gibt die Datenquelle jedoch einen Teil ihrer Autonomie auf, da Änderungen des globalen Schemas auch Anpassungen des lokalen Exportschemas und eventuell sogar des lokalen Komponentenschemas bedingen. Kann oder soll die Datenquelle das lokale Exportschema und das lokale Komponentenschema nicht umsetzen, so muss dies von einer anderen Komponente im föderierten Datenbanksystem übernommen werden. Die Zuweisung dieser Verantwortlichkeit kann von Datenquelle zu Datenquelle differieren und erzeugt somit unerwünschte Heterogenitäten in der Architektur föderierter Datenbanksysteme.

Mediator-Wrapper-Architektur

Mediator-Wrapper-Architekturen können als ein Spezialfall föderierter Datenbanksysteme aufgefasst werden. Bei dieser Architektur müssen die zu integrierenden Datenquellen nur einen sehr geringen Teil ihrer Autonomie aufgeben. Das lokale Komponentenschema und das lokale Exportschema werden immer von einer separaten Komponente realisiert.

Die nachfolgende Beschreibung basiert auf den Literaturquellen [Panc99], [Wied92] und [LeNa07].

Mediator-Wrapper-Architekturen werden bei so genannten mediatorbasierten Informationssystemen verwendet. Abbildung 4 stellt die grundlegende Architektur dar.

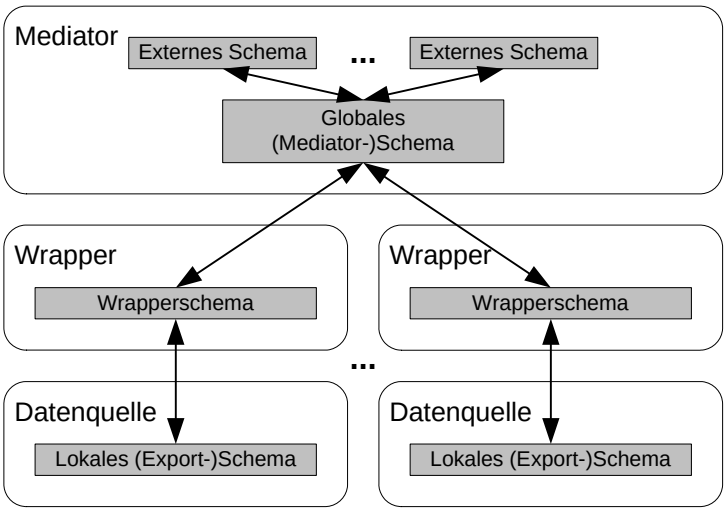

Abbildung 4: Mediator-Wrapper-Architektur[22]

Die Wrapper dienen einer einzelnen Datenquelle und greifen entweder auf ein lokales Exportschema oder auf das lokale konzeptionelle Schema der Datenquelle zu. Dabei entsprechen das Exportschema und das konzeptionelle Schema dem jeweiligen Schema aus der Drei-Schichten-Architektur, die im vorangegangenen Abschnitt vorgestellt wurde.

Die Wrapper besitzen eine einheitliche Schnittstelle, über die Anfragen in einer von den Datenbankquellen unabhängigen Form gestellt werden können. Der Wrapper überführt die Anfrage in eine quellenspezifische Anfrage. Das quellenspezifische Ergebnis wird in eine für alle Wrapper gleiche Form übersetzt.

Das vom Wrapper verwendete Schema stellt einen Teilausschnitt des globalen Mediator-Schemas dar. Es kann als Wrapper-Exportschema aufgefasst werden. Das Exportschema verwendet somit das kanonische Datenmodell des globalen Mediator-Schemas. Der Wrapper muss also gegebenenfalls Heterogenitäten auf Datenmodellebene überwinden.

Insgesamt sorgt der Wrapper für die Überwindung technischer, syntaktischer, Datenmodell- und struktureller Heterogenitäten. Dabei greifen diese auf Datenquellen zu, die (maximal) ein lokales Exportschema zur Verfügung stellen müssen. Somit bleiben die Datenquellen fast vollständig autonom.

Mediatoren greifen auf mehrere Wrapper zu, um mit Hilfe des globalen Mediator-Schemas eine integrierte Sicht auf mehrere Datenquellen zu schaffen. Darüber hinaus können die Mediatoren Schematransformationen, Datentransformationen und –aggregationen durchführen. So entstehen mehrere externe Schemata. Die Mediatoren schaffen also einen Mehrwert bei der Integration.

Mediatoren können auch für einen einzelnen Wrapper eingesetzt werden. Außerdem ist es möglich, dass ein Mediator auf mehrere andere Mediatoren zugreift. So entsteht eine hierarchische Struktur von Mediatoren. Dabei kann jeder einzelne Mediator einen anderen Mehrwert schaffen (z.B. Normalisierung einzelner Werte, Umbenennung von Attributen).

[22] Abbildung nach [LeNa07], S. 97

Die Mediator-Wrapper-Architektur zeichnet sich durch eine hohe Flexibilität aus. Außerdem können Wrapper gegebenenfalls mehrfach verwendet werden. So ist beispielsweise ein Wrapper denkbar, der allgemein für SQL-Datenbanken verwendet werden kann. Mehrere Anbieter von Identity Management Systemen verwenden das Prinzip des Wrappers und der Mediatoren. Bezeichnungen wie Adapter und Preconfigured Connectors stehen dabei für spezielle Wrapper.

2.5.5 Datenfehler und Datenfehlerbereinigung

Die Daten in einer einzelnen Datenquelle weisen unter Umständen Fehler auf. Die Ursachen hierfür können falsche Dateneingaben und -erfassungen, ausbleibende Aktualisierungen und somit Veralterung der Daten sowie Transformationsfehler auf Schema- und Datenebene sein. Durch die Integration von Daten aus mehreren Datenquellen entstehen meist weitere Fehler wie beispielsweise Duplikate. Datenfehler bezeichnet man oft auch als Anomalien.

In der Literatur werden verschiedene Klassifikationen von Anomalien vorgenommen. Die folgende Klassifikation wurde adaptiert aus [RaDo00] und [MüFr05]. Die Abbildung 5 stellt die einzelnen Fehlerklassen dar. Zunächst wird zwischen einer Datenquelle und der Integration mehrerer Datenquellen unterschieden. In beiden Fällen können Fehler auf Schema- und Datenebene entstehen.

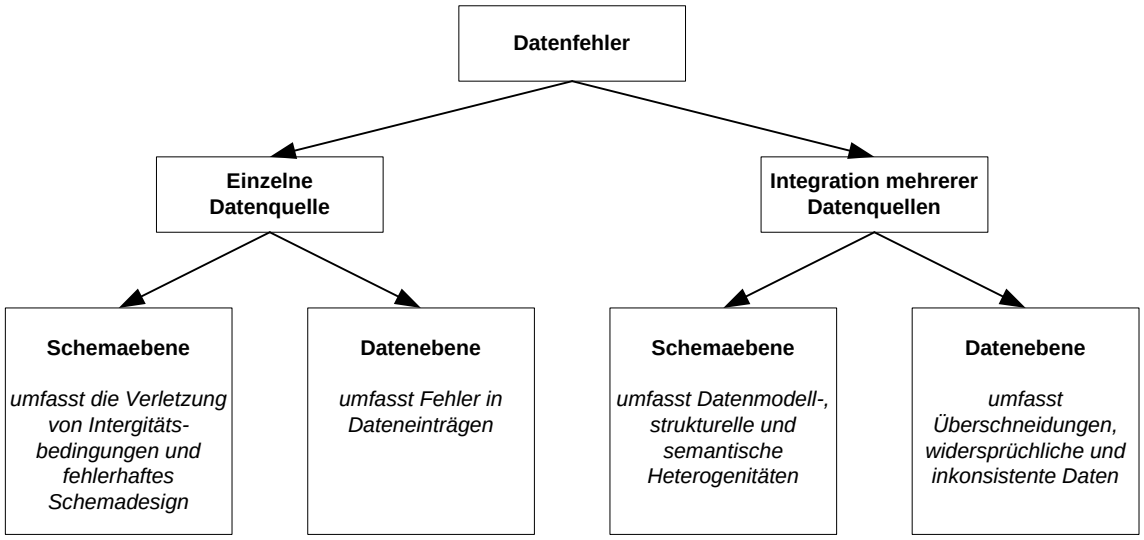

Abbildung 5: Klassifikation von Datenfehlern[23]

Nachfolgend werden für die vier Fehlerklassen, die in der Abbildung 5 aufgeführt sind, typische Fehlerarten angegeben.

Schemafehler innerhalb einer einzelnen Datenquelle

- Unzulässiger Wert (betrifft einzelne Attribute)
  Hierbei handelt es sich um unzulässige Datenwerte.
  →Beispiel: *Datum=32.15.2007*

[23] In Anlehnung an [RaDo00], S. 3

- (Funktionale) Attributabhängigkeit verletzt (betrifft einen einzelnen Datensatz)
  Zwei voneinander abhängige Attribute enthalten widersprüchliche Informationen.
  →Beispiel: *Alter= 97 und Geburtsdatum=01.01.2007*

- Eindeutigkeit verletzt (betrifft mehrere Datensätze eines Typs)
  Im Schema wird für die Werte eines oder mehrerer Attribute angegeben, dass diese nur für einen Datensatz vorkommen dürfen. Diese Eindeutigkeit wird nicht eingehalten, wenn mehrere Datensätze den gleichen Wert erhalten.

- Referenzielle Integrität verletzt (betrifft die gesamte Datenquelle)
  Ein Fremdschlüssel verweist auf einen nicht existenten Eintrag.

Datenfehler innerhalb einer einzelnen Datenquelle

- Fehlende Werte (betrifft einzelne Attribute)
  Aufgrund fehlender Informationen wird ein verpflichtendes Attribut mit einem Dummywert angegeben.
  → Beispiel: Geburtsdatum=31.12.9999

- Schreibfehler (betrifft einzelne Attribute)
  →Beispiel: *Ort=Bierlin*

- Kryptische Werte (betrifft einzelne Attribute)
  Ein Wert kann nicht mehr aus angegebenen Abkürzungen und Kodierungen abgeleitet werden.

- Eingebettete Werte (betrifft einzelne Attribute)
  Wenn Attribute in einem Schema fehlen, werden oft Werte in anderen Attributen zusammengefasst.
  →Beispiel: *Nachname=Graf von Hagen* anstelle von *Titel=Graf von, Nachname=Hagen*

- Falsch zugeordnete Werte (betrifft einzelne Attribute)
  →Beispiel: *Ort=58089, PLZ=Hagen*

- Domänenspezifische Formatierungsfehler (betrifft einzelne Attribute)
  Die Formatierungskonvention für ein Attribut wird nicht eingehalten.
  →Beispiel: *Telefon=+49-30-83856031* anstelle von *Telefon:+49/30/838-56031*

- Falsche Werte (betrifft einzelne Attribute)
  Die Zuordnung des Wertes zum Attribut ist zwar richtig, der Wert entspricht aber nicht den Gegebenheiten der Realwelt. Diese Art von Fehlern ist sehr schwierig zu erkennen.
  → Bespiel: *Name=Johann Wolfgang von Goethe, Geburtsort=Frankfurt (Oder)*

- Widersprüchliche Werte (betrifft einen einzelnen Datensatz)
  →Beispiel: *Ort=Berlin, PLZ=58089*

- Duplikate (betrifft mehrere Datensätze eines Typs)
  Bei zwei Datensätzen handelt es sich um die Repräsentation desselben Objektes in der Realwelt.

- Widersprüchliche Angaben in Duplikaten (betrifft mehrere Datensätze eines Typs)
  → Beispiel: 1. *Vorname=Karl, Nachname=Neumann, Geburtsdatum: 19.12.1975*
  2. *Vorname=Karl, Nachname=Neumann, Geburtsdatum: 19.04.1975*

- Transpositionen von Wörtern (betrifft mehrere Datensätze eines Typs)
  Innerhalb eines Attributes werden die Wörter, aus denen sich der Wert zusammensetzt, in verschiedenen Reihenfolgen angegeben.
  → Beispiel: 1. *Vorname=Hugo Egon Karl*
  2. *Vorname=Egon Karl Hugo*

Schemafehler durch die Integration mehrerer Datenquellen

- Namenskonflikte
  → siehe Abschnitte 2.5.2e) „Semantische Heterogenität“

- Strukturelle Konflikte
  → siehe Abschnitt 2.5.2d) „Strukturelle Heterogenität“

Datenfehler durch die Integration mehrerer Datenquellen

- Weitere Duplikate (mit widersprüchlichen Werten)
  Bei der Integration mehrerer Datenquellen können weitere Duplikate entstehen. Wie bei einzelnen Datenquellen, können die Duplikate untereinander widersprüchliche Werte aufweisen.

- Unterschiedliche Repräsentation von Daten
  → Beispiel: Datenquelle 1: *Werte für das Geschlecht: M / W*
  Datenquelle 2: *Werte für das Geschlecht: male / female*
  (Das Beispiel gilt dann, wenn im Schema keine Bedingungen für die zulässigen Werte angegeben werden können. Ansonsten läge ein Fehler auf Schemaebene vor.)

- Unterschiedliche Genauigkeit
  → Beispiel: Datenquelle 1: *Änderungszeitpunkt=2007-01-01*
  Datenquelle 2: *Änderungszeitpunkt=2007-01-01 03:12:05*

Den Prozess, bei dem Datenfehler identifiziert und korrigiert werden, bezeichnet man als Data Cleaning oder Data Cleansing. Findet eine Datenbereinigung nur auf einem Datensatz statt, so spricht man von Data Scrubbing. Den Vorgang der Duplikaterkennung nennt man Record Matching, Record Linkage, Object Identification oder Entity Resolution. Werden Duplikate zu einem Datensatz zusammengeführt, so bezeichnet man dies als Record Merging.

Für die Gestaltung der Prozesse zur Fehlererkennung und zur Fehlerbereinigung existieren verschiedene Ansätze. In der Regel wird für die Fehlererkennung zunächst der Datenbestand auf typische Charakteristika hin untersucht (Profiling). Danach werden für die Attribute Bedingungen angegeben (Assessment), die dann ständig überwacht werden (Monitoring). Erkannte Fehler müssen je nach Fehlerart unterschiedlich behandelt werden.

Im Folgenden werden ausschließlich Verfahren zur Duplikaterkennung beschrieben. Auch wenn die einzelnen Datenquellen über qualitativ hochwertige Daten verfügen, ist bei der Datenintegration, die in Identitätsmanagementsystemen erfolgt, eine Duplikaterkennung und –bereinigung erforderlich. Eine ausführliche Diskussion weiterer Verfahren zur Fehlererkennung und Fehlerbereinigung findet man beispielsweise in [MüFr05] und [KiCa04].

Duplikaterkennung

Bei Duplikaten handelt es sich um zwei oder mehr Datensätze, die dasselbe Objekt der Realwelt repräsentieren. Nun kann es jedoch vorkommen, dass die zwei Datensätze nicht vollständig syntaktisch und lexikalisch übereinstimmen. Deshalb ist es zur Behebung des Fehlers notwendig, auch Einträge zu finden, die sich nur ähnlich sind.

Ähnlichkeitmaß: Ein Ähnlichkeitsmaß gibt einen Grad für die syntaktische und lexikalische Übereinstimmung zweier Datensätze an.

Im Folgenden werden die einzelnen Klassen von Ähnlichkeitsmaßen vorgestellt:

a) Editierabstände

Sehr verbreitete Ähnlichkeitsmaße sind Editierabstände. Dabei wird die minimale Anzahl von Editieroperationen bestimmt, die notwendig sind, um eine Zeichenkette in eine andere Zeichenkette zu überführen. Dabei können die folgenden bekannten Verfahren zum Einsatz kommen:

- Hamming-Distanz
  Die Hamming-Distanz ist nur anwendbar auf Zeichenketten gleicher Länge. Zur Bestimmung der Distanz wird die minimale Anzahl der Ersetzungsoperationen bestimmt, die notwendig sind, um eine Zeichenkette in eine andere zu überführen.

- Levenshtein-Distanz
  Die Levenshtein-Distanz gibt an, wie viele Einfüge-, Ersetzungs- und Löschoperationen von Zeichen minimal notwendig sind, um eine Zeichenkette in eine andere zu überführen. Die Levenshtein-Distanz ist zwar für die Ähnlichkeitsbestimmung von zwei Zeichenketten anwendbar, die sich in der Länge unterscheiden können, jedoch zeigt dieses Ähnlichkeitsmaß deutliche Schwächen, wenn Abkürzungen vorkommen (Beispiel: *Univ-Prof. wird mit Universitätsprofessor verglichen.*)

b) Tokenbasierte Ähnlichkeitsmaße

Die Bestimmung von Editierabständen besitzt den Nachteil, dass unterschiedliche Reihenfolgen von Wörtern innerhalb einer Zeichenkette nicht berücksichtigt werden. Dies kann dazu führen, dass Duplikate als solche nicht erkannt werden, da der Editierabstand zu groß ist.

Beispiel:

- *Personendatensatz 1: Hugo Egon Karl Schmidt*
- *Personendatensatz 2: Karl Hugo Egon Schmidt*

Die beiden Datensätze werden sowohl nach der Hamming- als auch nach der Levenshtein-Distanz nicht als Duplikate erkannt. Tokenbasierte Verfahren berücksichtigen Reihenfolgevertauschungen. Dabei stellen Token die Wörter in einer Zeichenkette dar. Man erhält die einzelnen Wörter, indem man eine vorliegende Zeichenkette anhand bestimmter Trennzeichen wie Leerzeichen, Bindestriche, Zeilen- und Seitenumbrüche sowie Satzzeichen zerlegt. Zu den tokenbasierten Ähnlichkeitsmaßen gehören die Jaccard-Ähnlichkeit und die Term-Frequency/Inverse-Document-Freqeuncy (TFIDF):

- Jaccard-Ähnlichkeit
  Bei der Jaccard-Ähnlichkeit wird für zwei Zeichenketten das Verhältnis gebildet aus der Anzahl von Token, die jeweils in beiden Zeichenketten vorkommen, zu der Anzahl aller Token aus beiden Zeichenketten. Das Verfahren wird beispielsweise in [Ferb03] näher beschrieben.

- Term-Frequency/Inverse-Document-Frequency (TFIDF)
  Angenommen, es wird ein bestimmtes Token bzw. ein bestimmter Term i in einem Dokument j gesucht. Bei TFIDF erfolgt zunächst die Berechung der Term-Frequency (TF):

  $$\mathrm{TF}(i, j) = \frac{\text{Anzahl des Terms i in Dokument j}}{\text{Anzahl aller Terme in Dokument j}}$$

  Als nächstes wird die Inverse-Document-Frequency bestimmt:

  $$\mathrm{IDF}(i) = \log \frac{\text{Anzahl aller betrachteten Dokumente}}{\text{Anzahl der Dokumente, in denen der Term i vorkommt}}$$

  Die TFDIF wird als das Produkt von TF und IDF definiert:

  $$\mathrm{TFDIF}(i, j) = \mathrm{TF}(i, j) * \mathrm{IDF}(i)$$

  Der Wert von TFDIF ist verhältnismäßig hoch, wenn ein Term sehr häufig ein einem einzelnen Dokument vorkommt, wenn also TF einen hohen Wert besitzt. In je mehr Dokumenten ein Term vorkommt, umso kleiner wird IDF und somit fällt der Wert von TFDIF dann geringer aus.

  Weitere Informationen zu TFDIF können beispielsweise [SaGi84] entnommen werden.

c) Phonetische Ähnlichkeitsmaße

Die bisher vorgestellten Ähnlichkeitsmaße orientierten sich ausschließlich an der Syntaktik von Attributwerten. Phonetische Ähnlichkeitsmaße versuchen eine Ähnlichkeit aufgrund der Aussprache von zwei Begriffen herzustellen.

Beispiel:
Die Wörter DAX und Dachs besitzen einen relativ hohen Editierabstand, klingen aber in der Aussprache gleich.

Phonetische Ähnlichkeitsmaße sind stark abhängig von einer Sprache und der natürlichen Aussprache innerhalb eines Landes oder sogar einer Region. Sehr populär für die englische Sprache ist der Soundex-Algorithmus (vgl. [Newc67]). Für die deutsche Sprache existiert beispielsweise die Kölner Phonetik (vgl. [Post69]).

Die Duplikaterkennung kann bei einer hohen Anzahl von Datensätzen sehr aufwendig sein. Dies führt dazu, dass nicht immer ein Vergleich aller n Datensätze mit allen anderen n-1 Datensätzen erfolgt. Die Gesamtmenge wird in einzelne Teilmengen, so genannten Partitionen zerlegt. Die Duplikaterkennung wird dann nur pro Partition durchgeführt. Die Partitionen können mit Hilfe verschiedener Strategien erstellt werden. Bekannte Verfahren sind die Sorted-Neighbourhood-Method (SNM), die Multi-Pass-Sorted-Neighbourhood-Method (MP-SNM) und die inkrementelle Sorted-Neighbourhood-Method (I-SNM). Die genannten Verfahren werden beispielsweise in [LeNa07] vorgestellt.

Erkannte Duplikate sollen in der Regel zusammengeführt werden. Dabei können die folgenden Fälle vorliegen:

- Identität: Alle Attribute der beiden Datensätze stimmen exakt überein.
- Komplementarität: Die Duplikate enthalten nicht gemeinsame Attribute.
- Konflikt: In mindestens einem Attribut liegen unterschiedliche Werte vor.

Je nach vorliegendem Fall müssen unterschiedliche Methoden bei der Zusammenführung der Duplikate angewandt werden.

Mit der Datenfehlererkennung und Datenfehlerbereinigung wird das Ziel verfolgt, die Datenqualität insgesamt zu erhöhen. Mögliche Bewertungskriterien sind Genauigkeit, Vollständigkeit, Konsistenz und Aktualität. Eine ausführliche Diskussion zum Thema Datenqualität findet man beispielsweise in [Lehn03].

2.6 Sicherheit von IT-Systemen

Von IT-Systemen verlangt man, dass diese sicher sind. Hierbei muss zunächst ein gemeinsames Verständnis für den Begriff Sicherheit hergestellt werden. Der Mensch empfindet eine subjektive Sicherheit, wenn er glaubt, dass das IT-System vor potentiellen Gefahren geschützt ist. Dem gegenüber steht eine objektive Sicherheit, die bedeutet, dass tatsächlich keine Gefahren für ein System vorhanden sind.

IT-Systeme können durch verschiedene Angriffe in ihrer Sicherheit verletzt werden. Die möglichen Angriffziele lassen sich folgendermaßen klassifizieren:

- Computer
  Können mehrere Personen an einem Computer arbeiten, so ist es möglich, dass Programme durch einen nicht berechtigten Benutzer aufgerufen oder verändert werden. Neben den Softwarekomponenten kann auch die Hardware manipuliert oder sogar zerstört werden.

- Anwendungen
  Mehrere Nutzer können eine bestimmte Anwendung benutzen. Dabei bietet die Anwendung verschiedene Operationen auf Objekten an. Es ist möglich, dass ein Benutzer mehr Rechte erlangt, als für ihn eigentlich vorgesehen sind.

- Daten
  Daten, die in Datenbanken enthalten sind oder in anderer Form auf Datenträgern vorliegen, können von Unbefugten gelesen, manipuliert oder sogar gelöscht werden.

- Kommunikation(swege)
  Zwischen IT-Systemen werden in der Regel Daten über verschiedene Kommunikationswege ausgetauscht. Es ist möglich, dass die Kommunikation mitgelesen oder manipuliert wird.

Unabhängig von den Komponenten, aus denen ein IT-System besteht, lassen sich folgende Eigenschaften definieren, die es zu schützen gilt. Man spricht auch von Schutzzielen:

- Vertraulichkeit (engl. confidentiality)
  Bestimmte Daten sollen nur von berechtigten Personen gelesen werden können. Um dies bei der Kommunikation sicher zu stellen, werden die Daten verschlüsselt. Dabei unterscheidet man grundsätzlich zwischen symmetrischen Verschlüsselungen (zwei Kommunikationspartner verwenden den gleichen Schlüssel) und asymmetrischen Verschlüsselungen, die häufig unter dem Begriff Public-Key-Verfahren in der Literatur aufgeführt werden. Gängige Verfahren für Netzwerkprotokolle, bei denen sowohl symmetrische wie asymmetrische Verschlüsselung zum Einsatz kommt, sind die Secure Sockets Layer (SSL) oder Transport Layer Security (TLS). Vertraulichkeit muss nicht nur für die Daten gewahrt bleiben, die über Kommunikationswege fließen. Auch Daten, die auf Datenträgern vorliegen, können vertraulich sein und müssen durch eine Verschlüsselung geschützt werden.

- Integrität (engl. integrity)
  Unabhängig davon, ob Daten übermittelt oder auf Datenträgern gespeichert werden, muss sichergestellt werden, dass keine unbefugte Person diese verändert. Gleiches gilt für Programme, für die eine Manipulation und somit ein fehlerhaftes Verhalten vermieden werden muss. Die Integrität wird in der Regel mit digitalen Signaturen überprüft. Dazu werden verschiedene Hash-Algorithmen angewandt, mit denen aus den eigentlichen Daten eine Zeichenkette zur Überprüfung erzeugt wird. Werden die Daten manipuliert, so verändert sich der Hash-Wert.

- Authentizität (engl. authenticity)
  Findet eine Kommunikation zwischen zwei Teilnehmern statt, so muss sichergestellt werden, dass die Daten, die übermittelt werden, auch tatsächlich von dem angegebenen Erzeuger stammen. Dabei kommen verschiedene Verfahren der Authentifizierung zum Einsatz.

- Verfügbarkeit (engl. availability)
  Computer, Anwendungen, Daten sowie Kommunikationswege müssen für befugte Personen immer nutzbar, also verfügbar sein. Auf Hardwareebene bedeutet dies, dass defekte Komponenten schnell ausgetauscht werden können oder redundant vorhanden sind. Für die Anwendungen gilt insbesondere, dass diese fehlerfrei und effizient ablaufen. Der Zugriff auf Daten muss ebenfalls zu jeder Zeit für Befugte möglich sein. Die Kommunikationswege dürfen nicht blockiert werden und sollten auch nicht überlastet sein.

Eine detaillierte Betrachtung des Themas Sicherheit von IT-Systemen findet beispielsweise in [Bish03] statt.

2.7 Authentifizierungsverfahren

Die Authentifizierungsverfahren werden nach den folgenden Eigenschaften gruppiert:

- Wissen
  Eine Authentifizierung anhand von Wissen verlangt, dass die Person, die sich authentisieren möchte, mehrere Werte weiß, die sie dann angeben kann. Das wohl bekannteste und am weitesten verbreitete Authentifizierungsverfahren durch Wissen ist die Verwendung eines Passwortes. Das Passwort wird in der Regel zusammen mir einem Benutzernamen eingegeben und geprüft. Dabei hängt die Sicherheit des Passwortes von Bedingungen wie beispielsweise der minimalen Länge und den verwendeten Zeichen ab. Diese Bedingungen oder Richtlinien für Passwörter werden oft als Password Policies bezeichnet.

- Besitz
  Bei Authentifizierungsverfahren durch Besitz muss eine Person ein materielles Merkmal vorweisen. Hierzu gehören Magnetkarten, Chipkarten, Smardcards u.a. Meist wird auf den Karten ein Zertifikat und ein privater Schlüssel hinterlegt.

- Biometrie
  Bei biometrischen Authentifizierungsverfahren wird eine Person anhand unverwechselbarer und individueller Körpermerkmale identifiziert. Beispiele für solche Körpermerkmale sind Fingerabdrücke, die Iris oder die Stimme.

Jedes einzelne Authentifizierungsverfahren bringt ein gewisses Maß an Sicherheit mit sich. Dabei ist es beispielsweise einfacher, ein Passwort auszuspähen, als Fingerabdrücke oder die Iris nachzuahmen. Um die Sicherheit bei der Authentifizierung insgesamt zu erhöhen, werden die einzelnen Verfahren miteinander kombiniert.

Beispiel:
An Geldautomaten erfolgt anhand einer Bankkarte eine Authentifizierung durch Besitz. Zusätzlich muss noch ein PIN eingegeben werden. Es findet also zusätzlich eine Authentifizierung durch Wissen statt.

Würde in dem Beispiel nur die Bankkarte ohne PIN genügen, um Geld abzuheben, so stellte der Verlust der Bankkarte ein viel höheres Risiko dar.

2.8 Autorisierungsmodelle

Autorisierungsmodell: Ein Autorisierungsmodell beschreibt das Verfahren, nach dem Zugriffe auf Objekte autorisiert werden.

Im Kontext von Autorisierungsmodellen werden die Begriffe Subjekt und Objekt genannt, die zunächst definiert werden:

Subjekt (engl. subject): „A subject is a computer system entity that can initiate requests to perform an operation or series of operations on objects.“[24]

Objekt (engl. object): „An object is a system entity on which an operation can be performed.“[25]

Des Weiteren gelten für die Entwicklung von Autorisierungsmodellen Grundsätze, von denen einige kurz vorgestellt werden:

- Prinzip der geringsten Privilegien (engl. Principle of Least Privilege)
  Ein Subjekt erhält danach den minimalen Satz an Privilegien bzw. Rechten, die notwendig sind, um eine Anwendung oder Funktion auszuführen.

- Prinzip der Separation von Privilegien (engl. Principle of Separation of Privilege)
  Bei diesem Prinzip wird verlangt, dass ein Zugriff auf beispielsweise ein System nicht basierend auf einer einzigen Bedingung bzw. einem einzelnen Recht erfolgen darf.

- Prinzip der Separation von Aufgaben (engl. Separation of Duty (SoD))
  Dieses Prinzip ist ähnlich dem Prinzip der Separation von Privilegien. Dabei wird verlangt, dass auf Personen oder Subjekte die Aufgaben in der Form aufgeteilt sind, dass für die Ausführung kritischer Vorgänge, Aktionen oder Operationen mehr als eine Person oder mehr als ein Subjekt erforderlich ist.

  Beispiel:
  An einer Hochschule existiert eine Stelle in der Studierendenverwaltung, die Bareinzahlungen von Semestergebühren entgegen nimmt. Diese Einzahlungen werden von einer Person in einer speziellen Anwendung gebucht. Es wird jedoch verlangt, dass eine Person, die Einzahlungen bucht, keinen weiteren Zugriff auf die Datenbank erhält, die für die Anwendung verwendet wird. Ansonsten könnte nämlich die Person unbemerkt Geld hinterziehen, in dem sie nachträglich Buchungen in der Datenbank verändert.

Sehr verbreitet sind die Autorisierungsmodelle Discretionary Access Control (DAC), Mandatory Access Control (MAC) sowie Role-Based Access Control (RBAC). Im Nachfolgenden werden die Autorisierungsmodelle vorgestellt und auch auf ihre Nachteile verwiesen. Ein besonderes Interesse gilt dabei dem RBAC-Modell, das in verschiedenen Abwandlungen in aktueller Identitätsmanagementsoftware verwendet wird.

Die dargestellten Autorisierungsmodelle sind nicht nur isoliert zu betrachten. Auch eine Kombination der Modelle lässt sich in vielen Systemen wiederfinden.

[24] [FKC03], S. 30
[25] [FKC03], S. 30

2.8.1 Discretionary Access Control (DAC)

1983 veröffentlichte das U.S. Department of Defense (DoD) die Trusted Computer System Evaluation Criteria (TCSEC), in denen die zwei Modelle Discretionary Access Control und Mandatory Access Control zur Zugriffskontrolle vorgestellt werden (vgl. [DoD85]).

Bei der Discretionary Access Control werden den Objekten Subjekte zugeordnet. Die Subjekte, bei denen es sich beispielsweise um Personen oder Gruppen von Personen handeln kann, stellen die Eigentümer der Objekte dar. Zugriffe auf Prozesse, die ein Objekt repräsentieren, werden nur für zugewiesene Subjekte gestattet. Der Eigentümer kann im eigenen Ermessen (engl. discretionary) anderen Subjekten alle Rechte für seine Objekte übertragen. Durch diese Zuweisung werden weitere Subjekte Eigentümer von Objekten und können wiederum die vollen Rechte für Objekte auf andere Subjekte übertragen.

Eine sehr bekannte Form der Implementierung von DAC sind Access Control Lists (ACLs). Wie der Name bereits zum Ausdruck bringt, handelt es sich um Listen, in denen zu jedem Objekt die Subjekte oder Gruppen von Subjekten und ihre Zugriffsrechte zugeordnet sind.

Die Vergabe von Rechten nach DAC ist für eine Organisation nicht kontrollierbar. Durch das Übertragen der Eigentümerrechte auf andere Subjekte ist es sogar möglich, dass der ursprüngliche Eigentümer seine Zuweisung zu einem Objekt verlieren kann, da ein neuer Eigentümer ihm diese entzieht.

2.8.2 Mandatory Access Control (MAC)

Die Vergabe von Zugriffsrechten erfolgt bei der Mandatory Access Control im Gegensatz zu DAC nicht durch eine auf Vertrauen basierende Selbstbestimmung der Mitarbeiter. Stattdessen werden den Objekten anhand ihrer Vertraulichkeit Sicherheitsstufen zugeordnet, die sich an einem von der Organisation aufgestellten Regelwerk orientieren. Den Subjekten werden ebenfalls Sicherheitsklassen zugeordnet. Der Zugriff auf ein Objekt durch ein Subjekt ist nur dann möglich, wenn beide der entsprechenden Sicherheitsklasse genügen. Es existieren viele Varianten zu diesem Modell, die unterschiedliche Ziele verfolgen. Das Bell-LaPadula Model konzentriert sich auf den Aspekt der Vertraulichkeit von Daten, während das Biba Integrity Model die Integrität der Daten als besonders wichtig ansieht.

Die Mandatory Access Control entzieht den Benutzern jegliche Möglichkeit, den Zugriffschutz zu verändern. Dies ist nur mit Hilfe von Administratoren möglich. Dadurch wird zwar die organisationsweite Kontrolle über die Objekte erhöht, jedoch erweist sich das gesamte Verfahren als recht unflexibel.

2.8.3 Role-Based Access Control (RBAC)

RBAC besteht nach [FKC03] im Ausgangsmodell aus Benutzern, Rollen und Rechten. Rechte werden definiert als Operationen auf Objekten. Mehrere Rechte können mehreren Rollen zugeordnet werden und mehrere Benutzer können Mitglieder von mehreren Rollen sein. Abbildung 6 zeigt die Relationen zwischen Benutzern, Rollen, Operationen und Objekten im Entity-Relationship-Diagramm (ERD).

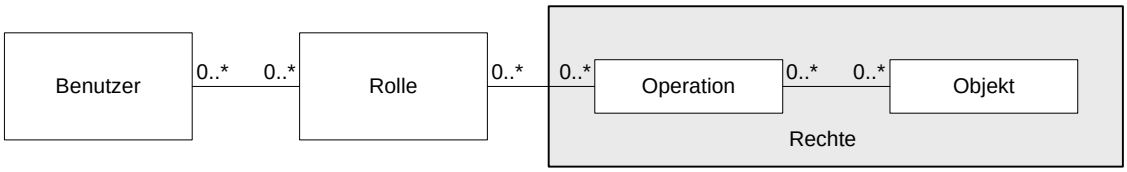

Abbildung 6: RBAC Basiselemente

Bei der Diskussion um RBAC wird teilweise der Begriff der Gruppe als Synonym für eine Rolle verwendet. In dieser Arbeit wird jedoch unter den Begriffen Rolle und Gruppe etwas Unterschiedliches verstanden:

Rolle: Eine Rolle repräsentiert eine Zusammenfassung von Rechten. Der Name einer Rolle und deren mögliche Beschreibung geben die Funktion wieder, die ein Benutzer innerhalb des Anwendungskontextes wahrnimmt.

Gruppe: Eine Gruppe ist ausschließlich eine Strukturierungshilfe. Sie dient der Gruppierung von Benutzern, Rollen oder Rechten und stellt keine Beziehung zwischen diesen drei Elementen her.

Eine weitere mehrdeutige Verwendung tritt in der Literatur bei den englischen Begriffen permissions und privileges auf. In dieser Arbeit werden beide Begriffe synonym für den deutschen Begriff Rechte verwendet.

Die Zwischenschaltung von Rollen zwischen Benutzern und ihren Rechten bringt mehrere Vorteile mit sich. Rollen fassen Rechte anhand der Funktionen zusammen, die Benutzer innerhalb einer Organisation wahrnehmen können. Sicherheitsregeln einer Organisation lassen sich durch diese Zuweisung einfach abbilden und nachvollziehen. Administratoren können mit einem relativ geringen Aufwand die Rollen entsprechend der Kompetenzen und Verantwortlichkeiten den Benutzern zuordnen. Übernimmt ein neuer Benutzer die gleichen Aufgaben wie ein bereits existierender, so erhält der neue Benutzer die gleichen Rollen. Hierdurch müsse die Rechte nicht mühsam dupliziert werden. Auch Änderungen der Rechte durch neue Aufgabengebiete eines Benutzers oder durch Beendigungen des Beschäftigungsverhältnisses lassen sich durch entsprechende Änderungen der Rollenzuordnungen einfach abbilden. Es ist möglich, die Definition von Rollen und Rechten einerseits sowie die Zuweisung von Benutzern, Rollen und Rechten andererseits in die Zuständigkeitsbereiche zweier verschiedener Personen zu übertragen (Prinzip der Separation of Duty). Für die Umsetzung bestimmter Sicherheitskonzepte sind dadurch immer mehrere Personen erforderlich, was die Sicherheit im Allgemeinen erhöht. Das RBAC-Modell ist zum einen abstrakt, da es die Details einer Implementierung ausblendet, zum anderen ist es aber auch generell formuliert, d.h. es lässt sich in einer Vielzahl von Anwendungen verwenden.

Die bisher dargestellte Form von RBAC entspricht einer ausschließlich statischen Sichtweise des Modells. Das Modell wird noch um eine dynamische Komponente ergänzt, die das bekannte Konzept von Subjekten und Objekten enthält.

Zu jedem Benutzer können stellvertretend mehrere aktive Subjekte zur gleichen Zeit zugeordnet sein. Beispielsweise existieren in einem Betriebssystem zwei laufende Prozesse (Subjekte) zu einem Benutzer. Jedes Subjekt eines Benutzers kann dabei eine unechte Teilmenge der Rollen verwenden, die dem Benutzer zugeordnet sind. Die Rollen, die ein

aktives Objekt verwendet, werden als aktive Rollen bezeichnet und erzeugen die Dynamik in dem Modell. Jedes Subjekt erhält dabei nur das Minimum an Rollen, die es aktiv benötigt (Prinzip der Least Privileges).

Das bisher vorgestellte Basismodell von RBAC wird auch mit RBAC0 bzw. RBAC Version 0 bezeichnet.

Die Erweiterung von RBAC0 um Hierarchien von Rollen bezeichnet man als RBAC1. Hierarchien erlauben eine bessere Darstellung der Verteilung von Funktionen, die in einer Organisation wahrgenommen werden. Einige Funktionen überlappen sich in den Aufgabengebieten der Benutzer.

Beispiel: *Abbildung 7 zeigt zwei typische Rollen an einer Hochschule. Studierende und Mitarbeiter sind Mitglieder der Hochschule. Mitglieder der Hochschule dürfen beispielsweise Wireless LAN benutzen, egal, welche spezielle Funktion sie übernehmen. Studierende und Mitarbeiter erben die Rechte, die mit der Rolle Mitglied der Hochschule verbunden sind und erweitern diese um spezifische Rechte der jeweiligen Unterrolle.*

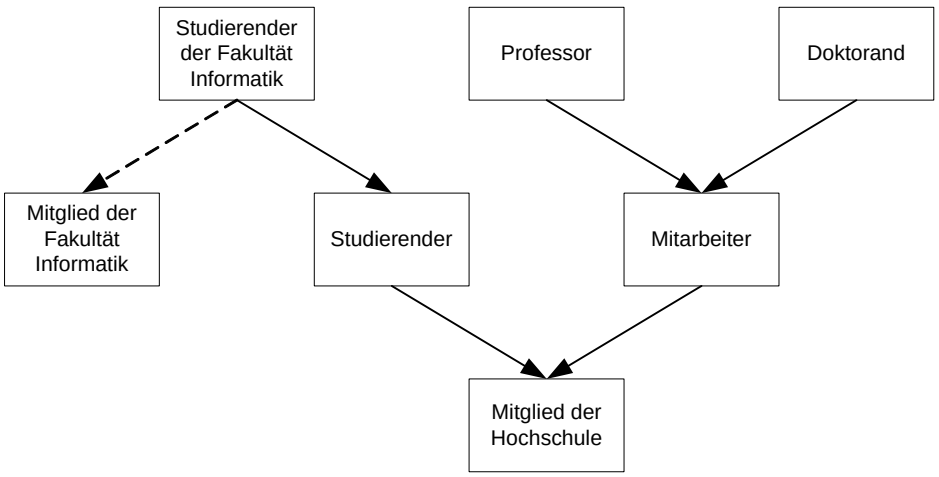

Abbildung 7: Beispiel einer Rollenhierarchie

In einer limitierten Rollenhierarchie existiert zu jeder Rolle nur ein direkter Nachfolger. Bei der Darstellungsvariante wie in Abbildung 7 darf es pro Rolle also nur eine ausgehende Kante geben. Nach der Graphentheorie entspricht diese Darstellung einem Baum mit zur Wurzel gerichteten Kanten (In-Tree). Generelle Hierarchien erlauben mehrere direkte Nachfolger. Dies wird in Abbildung 7 durch den gestrichelten Pfeil symbolisiert.

Beispiel:
Der Studierende der Fakultät Informatik erbt alle Rechte, die aus seiner Mitgliedschaft zur Fakultät Informatik und aus der Rolle des Studierenden und Mitglieds der Hochschule resultieren.

Eine weitere Erweiterung von RBAC0 stellt die Einführung von Bedingungen (engl. constraints) dar. Insbesondere soll durch die Einführung von Bedingungen eine Aufteilung von Zuständigkeiten erfolgen. Dies erreicht man durch so genannte SoD (Separation of Duty) constraints, die sich in statische und dynamische gliedern lassen. Mit statischen SoD constraints stellt man generell sicher, dass ein Benutzer nicht für eine Rolle A und zusätzlich

für eine Rolle B autorisiert sein darf. Sind die Rollen hierarchisch aufgebaut und schließen sich die Rollen A und B gegenseitig aus, so gilt dies auch für die Unterrollen von A und B.

Beispiel:
Nach Abbildung 7 könnte für die Rollen Studierender und Mitarbeiter eine solche Ausschlussbedingung existieren. Damit sind auch die folgenden anderen Kombinationen von Rollen nicht erlaubt:

- *Studierender und Professor*
- *Studierender und Doktorand*
- *Studierender der Fakultät Informatik und Mitarbeiter*
- *Studierender der Fakultät Informatik und Professor*
- *Studierender der Fakultät Informatik und Doktorand*

Dynamische SoD constraints erlauben es den Benutzern, zwar Mitglied einer Rolle A und einer Rolle B zu sein, aber die Rollen dürfen nicht gleichzeitig aktiv benutzt werden. Dies bedeutet, dass die aktiven Subjekte, die zu einem Benutzer gehören, bestimmte Rollen nicht gleichzeitig aktiv verwenden können.

Beispiel:
Eine Person kann sowohl Studierender als auch Mitarbeiter der Hochschule sein. Deshalb kann man keinen generellen Ausschluss für beide Rollen festlegen. In bestimmten Anwendungen darf der Benutzer hingegen nur mit einer dieser beiden Rollen agieren, da sonst nicht klar ist, welche Funktionen bereitgestellt werden sollen.

Sicherheitskritische Funktionen in einer Anwendung lassen sich absichern, in dem man festlegt, dass mindestens zwei Rollen notwendig sind, um diese auszuführen. In Verbindung mit einer statischen oder dynamischen Ausschlussbedingung für die Rollen erhält man die Klasse von operational SoD constraints.

Die Festlegung, dass ein Benutzer nur eine bestimmte Anzahl kritischer Operationen in einer festgelegten Abfolge auf einem Objekt ausführen darf, führt zu dem Konzept der history and object-based SoD constraints. Für jedes Objekt wird dabei eine Historie angelegt, in der vermerkt wird, welcher Benutzer welche Operationen auf diesem ausgeführt hat. Wurde eine kritische Operation von dem Benutzer auf einem Objekt ausgeführt, so können weitere nicht mehr möglich sein. Die Speicherung aller Informationen zu den ausgeführten Operationen und die Prüfung der Ausschlussbedingungen lassen die Komplexität stark ansteigen, insbesondere, wenn dies auch systemübergreifend ermöglicht werden soll.

Die letzte Klasse von speziellen Bedingungen für Rollen stellen die temporal constraints dar. Es handelt sich dabei um zeitliche Restriktionen. So können für Rollen ein Gültigkeitszeitraum oder eine Gültigkeitsdauer definiert werden. Die zeitliche Einschränkung kann sich auch auf die Beziehungen zwischen Benutzer und Rollen beziehen. Gleiches gilt für die Zuweisungen von Rechten zu Rollen.

Beispiel:
Eine Urlaubsvertretung für einen Mitarbeiter soll bestimmte Aufgaben während der Abwesenheit übernehmen. Es werden für die Zeit des Urlaubs bestimmte Rollen zugewiesen. Die Zuweisung trägt ein Aktivierungs- und ein Deaktivierungsdatum. Zum Ende des Urlaubs sind die Rollen für die Vertretung nicht mehr wirksam. Niemand muss mehr daran denken, diese Zuweisungen wieder aufzuheben. Bei einer weiteren Urlaubsvertretung können wieder neue Daten eingestellt werden.

Die Erweiterung von RBAC0 um die genannten Bedingungen für Rollen und deren Zuweisung zu Benutzern oder Subjekten wird als RBAC2 bezeichnet.

RBAC3 stellt eine Vereinigung von RBAC1 (Rollenhierarchien) und RBAC2 (Bedingungen) dar.

Rollen lassen sich für verschiedene Ebenen einer Organisation verwenden. Sehr nahe liegend ist die Darstellung der allgemeinen Organisationsstruktur mit Hilfe hierarchischer Rollen. Man spricht auch von enterprise view. Für die Anwendungssysteme, die eine Autorisierung für Benutzer benötigen, müssen die Rollen und Rechte, die abstrakt formuliert wurden, in ein implementiertes Autorisierungsmodell übernommen werden. Dies erfordert Umsetzungsregeln, die für jedes System individuell formuliert werden müssen. Im einfachsten Fall werden die abstrakten Rollen und Rechte in gleichartige systemspezifische Rollen und Rechte übersetzt. Wesentlich komplexer und nicht immer vollständig möglich ist die Umsetzung von RBAC in andere Autorisierungsmodelle (z.B. MAC, DAC) und darauf aufbauende Systeme. Vorschläge hierfür findet man ebenfalls in [FKC03].

2.9 Architekturansätze für Identitätsmanagementsysteme

Für Identitätsmanagementsysteme kommt eine Vielzahl unterschiedlicher Architekturen zum Einsatz. Eine Klasse von Architekturen, aus denen sich eine Referenzarchitektur ableiten lässt, ist aktuell nicht erkennbar. Einer der Gründe mögen die sehr organisationsspezifischen (IT-)Strukturen sein. Grundsätzlich lassen sich jedoch zwei Ansätze von Architekturen für Identitätsmanagementsysteme erkennen:

- Zentralisierter Architekturansatz
  Bei zentralisierten Architekturen für Identitätsmanagementsysteme wird versucht, sämtliche Identitätsdaten an einer zentralen Stelle zusammenzuführen und zu speichern. Dabei steht das Ziel einer integrierten Sicht auf alle Identitätsdaten im Vordergrund. Viele der damit verbundenen Probleme wurden bereits im Abschnitt 2.5 („Grundlagen zur Datenintegration“) aufgezeigt. Auch die anderen Aufgaben wie beispielsweise Authentifizierung und Autorisierung werden bei zentralisierten Architekturen von einer zentralen Komponente realisiert. Zentralisierte Architekturen besitzen den Nachteil, dass sie schlecht skalieren. Jeder neue Sachverhalt aus der Realwelt, der zentral gespeichert werden soll, erfordert unter anderem eine Anpassung des zentralen Identitätsspeichers.

- Föderierter Architekturansatz
  Föderierte Architekturen für Identitätsmanagementsysteme verfolgen den Grundsatz, dass die Identitäten in den ursprünglichen IT-Systemen verbleiben. Über bestimmte Mechanismen wird dafür gesorgt, dass Identitäten miteinander verknüpft werden. Bei Bedarf erfolgt ein direkter Austausch einiger Identitätsattribute zwischen zwei IT-Systemen. Neben der Verwaltung der Identitätsdaten werden auch die anderen Aufgaben eines Identitätsmanagementsystems verteilt gelöst.

  Beispiel:
  Ein Studierender ist aufgrund eines Kooperationsstudiengangs an zwei Hochschulen registriert. An beiden Hochschulen erhält der Studierende ein Benutzerkonto. Authentifiziert sich der Studierende an der Hochschule A, so sorgt ein spezieller Mechanismus dafür, dass er auch auf Dienste der Hochschule B zugreifen kann.

Dabei wird im Hintergrund eine Verknüpfung zwischen beiden Benutzerkonten des Studierenden hergestellt.

Föderierte Architekturen verfolgen nicht das Ziel einer integrierten Sicht auf alle Identitätsdaten. Vielmehr stehen die Flexibilität des Gesamtsystems und der Erhalt der Autonomie der einzelnen Systeme im Vordergrund. Föderierte Identitätsmanagementsysteme verlangen einen hohen Aufwand zur Abstimmungen aller Teilnehmer.

Während zentralisierte Architekturen eher innerhalb einzelner Organisationseinheiten oder Organisationen verwendet werden, finden föderierte Architekturen für organisationsübergreifende Identitätsmanagementsysteme eine Verwendung. Auch beliebige Kombinationen aus beiden Architekturansätzen kommen zum Einsatz.

In den nachfolgenden Kapiteln werden Architekturen und Standards vorgestellt, die für föderierte Identitätsmanagementsysteme zum Einsatz kommen können. Außerdem werden die wichtigsten Initiativen in diesem Bereich aufgezeigt.

2.10 Serviceorientierte Architekturen

In diversen Publikationen werden serviceorientierte Architekturen (SOA) teils widersprüchlich zueinander dargestellt. Die Organization for the Advancement of Structured Information Standards (OASIS) spezifiziert in [OASIS06a] ein Referenzmodell für SOAs, durch das ein gemeinsames Verständnis für diese Klasse von Architekturen hergestellt werden soll. Das Referenzmodell bildet die Grundlage für die nachfolgende Beschreibung serviceorientierter Architekturen.

Serviceorientierte Architektur: "Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains."[26]

Nach der Definition stellen serviceorientierte Architekturen ein Paradigma dar, mit dem sich verteilte Ressourcen bzw. Fähigkeiten organisieren und nutzbar machen lassen. Dabei können die Ressourcen unter der Kontrolle verschiedener Bereiche sein.

Die Ressourcen dienen der Befriedigung von Bedürfnissen. Die Bedürfnisse können sich auf eine oder mehrere Ressourcen richten und eine Ressource kann mehrere Bedürfnisse befriedigen. Es besteht also eine n:m-Relation.

Beispiel:
Ein Betreiber eines E-Learningsystems an einer Hochschule benötigt Daten von Studierenden und Mitarbeitern. Der Betreiber besitzt also das Bedürfnis, bestimmte Daten zu erhalten. Die Ressourcen, die er zu Befriedigung seiner Bedürfnisse benötigt, sind die Daten der Studierendenverwaltung und der Personalstelle. Die Studierendendaten werden zusätzlich von den Prüfungsbüros benötigt, die ebenfalls das Bedürfnis nach diesen Daten aufweisen.

[26] [OASIS06a], S. 8

Der Mechanismus, der die Bedürfnisse und die Ressourcen zusammen führt, wird als Service bezeichnet. Der Zugriff auf den Service erfolgt über eine festgelegte und beständige Schnittstelle, das Service Interface.

Die Komponente, die einen Service bereitstellt, um eine oder mehrere Ressourcen zugänglich zu machen, wird als Service Provider bezeichnet. Die Komponente, die den Service über das Service Interface in Anspruch nimmt, nennt man Service Consumer. Service Provider und Service Consumer sind undifferenziert betrachtet Service Participants.

Dadurch, dass Services von einer Komponente angeboten und durch eine andere genutzt werden, entsteht eine Dynamik zwischen den Teilnehmern. Wichtige Konzepte für die Ermöglichung, Erzeugung und Beeinflussung dieser Dynamik sind Sichtbarkeit (engl. visibility), Interaktion (engl. interaction) und Auswirkung in der realen Welt (engl. real world effect).

2.10.1 Sichtbarkeit

Ein Service Consumer und ein Service Provider müssen sich in irgendeiner Form „sehen“ können, um zu interagieren. Dabei gelten die folgenden Bedingungen:

- Bewusstheit (engl. awareness)
  Es muss den Serviceteilnehmern bewusst sein, dass jeweils weitere existieren. Dies erreicht man mit Mechanismen, die ein Auffinden von Serviceteilnehmern ermöglichen. Anhand einer Servicebeschreibung, die ein Service Provider zur Verfügung stellt, kann ein Service Consumer entscheiden, ob der angebotene Service seinen Bedürfnissen teilweise oder vollständig gerecht wird.

- Bereitschaft (engl. willingness)
  Die Bereitschaft dazu, dass ein Service Provider einem Service Consumer die Nutzung eines Service gestattet, wird im Wesentlichen durch Richtlinien (engl. policies) bestimmt. Die Richtlinien, die der Service Provider verwendet, können dem Service Consumer mitgeteilt werden.

- Erreichbarkeit (engl. reachability)
  Zwischen den Serviceteilnehmern muss in irgendeiner Form eine Kommunikation möglich sein.

2.10.2 Interaktion

Die Interaktion zwischen den Teilnehmern kann durch den Austausch von Nachrichten, die Veränderung einer gemeinsam genutzten Entität oder durch eine Kombination von beidem erfolgen. Die Interaktion enthält die zwei folgenden Modelle:

- Informationsmodell (engl. information model)
  Mit dem Informationsmodell wird festgelegt, welche Daten ausgetauscht werden können. Die Definitionen von Formaten, Bezeichnungen und Beziehungen von Entitäten bestimmen die Struktur der Daten. Zusätzlich muss die Semantik der Daten für alle Serviceteilnehmer gleich sein. Uneindeutigkeiten bezüglich der Bedeutung der Dateninhalte sind also zu vermeiden.

- Verhaltensmodell (engl. behaviour model)
  Das Verhaltensmodell enthält ein Aktionsmodell (engl. action model) und ein Prozessmodell (engl. process model). Dabei wird in dem Aktionsmodell festgelegt, welche Aktionen gegen einen Service möglich sind und welche Reaktionen erfolgen können. Die Antwort auf eine Aktion stellt letztlich nur eine weitere Aktion dar. Mit dem Prozessmodell werden die zeitlichen Abhängigkeiten und Eigenschaften der Aktionen definiert. Danach sind bestimmte Aktionen nur in einer bestimmten Abfolge durchführbar.

2.10.3 Auswirkung in der realen Welt

Mit der Benutzung eines Service soll ein bestimmter Effekt erzeugt werden. Dabei geht es nicht hauptsächlich um die Veränderung eines bestimmten Wertes, sondern um die tatsächliche Bedeutung der jeweiligen Wertänderung in der realen Welt.

Beispiel:
Bei einer Studierendenidentität wird über einen Service der Hörerstatus von irgendeinem Wert auf ‚E' geändert. ‚E' steht für exmatrikuliert. Die zu der Identität gehörige Person ist also in der realen Welt dadurch nicht mehr Studierender dieser Hochschule.

2.10.4 Vorteile

Serviceorientierte Architekturen eignen sich aufgrund ihrer hohen Flexibilität und Skalierbarkeit gut zur Kopplung verteilter heterogener IT-Systeme, wie man sie häufig innerhalb einer Organisation oder eines Verbundes von Organisationen vorfindet. Ausgewählte Ressourcen eines IT-Systems werden über einen Service-Provider zugänglich gemacht. Service Consumer nutzen diesen Service. Für diese bleibt jedoch die tatsächliche Realisierung des Services transparent. Ein bestehendes IT-System kann durch ein neues einfach ersetzt werden, sofern das neue IT-System das gleiche Service Interface anbietet. Neue Services können nach der Bekanntgabe sofort genutzt und mit bestehenden kombiniert werden, um wiederum neue Services zu bilden. Im letzten Fall spricht man auch von Kompositionen (engl. composites) von Services.

2.11 Standards für föderierte Identitätsmanagementsysteme

2.11.1 SOAP, WSDL und UDDI

Serviceorientierte Architekturen werden sehr häufig mit Hilfe von SOAP[27] realisiert. SOAP ist ein vom World Wide Web Consortium (W3C) spezifiziertes zustandloses Protokoll (siehe [W3C03]), das dem Austausch strukturierter und typisierter Informationen in verteilten Umgebungen dient. Ein Sender (SOAP sender) schickt eine XML-basierte Nachricht (SOAP message) an einen Empfänger (SOAP receiver). Beim Empfänger werden durch die in der Nachricht enthaltenen Operationen Aktionen ausgelöst. In diesem Zusammenhang wird auch häufig der Begriff der Remote Procedure Calls (RPC) genannt. Nach der Ausführung der Aktion kann dem Sender das Ergebnis wieder in Form einer XML-Nachricht mitgeteilt werden. Die Nachricht kann entweder eine Erfolgsinformation mit zusätzlichen Daten enthalten oder es wird ein Fehler angezeigt. SOAP-Nachrichten

[27] SOAP stand ursprünglich für Simple Object Access Protocol, wird aber seit der Version 1.2 als Eigenname verwendet.

können auch über mehrere Zwischenstationen (SOAP intermediaries) zu ihrem Ziel gelangen. Der eigentliche Transport von SOAP-Nachrichten erfolgt mit Hilfe eines anderen Internetprotokolls. Man spricht auch vom Binding.

Binding / Reverse Binding: Der Transport spezifizierter Nachrichten auf Basis eines zugrunde liegenden Protokolls wird als Binding bezeichnet. Unter Reverse Binding versteht man die Extraktion der Nachricht aus dem zugrunde liegenden Protokoll.

In der Regel werden SOAP-Nachrichten mit Hilfe von HTTP transportiert. Jedoch sind auch andere Anwendungsprotokolle aus dem TCP/IP-Protokollstapel wie beispielsweise SMTP und POP3 möglich.

SOAP ermöglicht letztlich die Interaktion innerhalb einer serviceorientierten Architektur. Da meist HTTP für den Transport von SOAP-Nachrichten verwendet wird, um auf Services zuzugreifen, spricht man auch häufig von Web Services.

SOAP ist als Standard anerkannt und plattformunabhängig. SOAP-Nachrichten besitzen aufgrund der XML-Strukturen und diverser Protokollinformationen einen hohen Overhead. Der Anteil der Nutzinformationen ist also im Verhältnis zu den Gesamtinformationen gering.

Mit der WSDL (Web Services Description Language) erfolgt die Beschreibung von Funktionen, die über eine Web Service Schnittstelle angeboten werden. Im XML-Format liegen Angaben zu den Datentypen, den Funktionsnamen, den Funktionsparametern sowie den Ergebnistypen vor. Außerdem sind Informationen zu der Adresse, unter der ein Web Service zu erreichen ist und zu dem Binding (z.B. SOAP über HTTP) enthalten.

Mit UDDI (Universal Description, Discovery and Integration) wird ein Verzeichnisdienst über SOAP zur Verfügung gestellt, der unter anderem das Auffinden von Web Services ermöglicht. Man spricht von Green Pages, wenn in dem Verzeichnis Informationen über Web Services abrufbar sind. Mit UDDI sind auch die bekannten Branchenverzeichnisse (Yellow Pages) und allgemeine Telefonbücher (White Pages) umsetzbar.

WSDL und UDDI realisieren die Sichtbarkeit von Services innerhalb einer serviceorientierten Architektur.

2.11.2 SAML

Die Grundlage für die nachfolgende Beschreibung der Security Assertion Markup Languague (SAML) bilden die Spezifikationen, die von der OASIS für die Versionen 1.0, 1.1 und 2.0 veröffentlicht wurden (vgl. [OASIS03a], [OASIS03b], [OASIS05a], [OASIS05b], [OASIS05c], [OASIS05d], [OASIS05e]).

Mit SAML wird ein XML-basiertes Rahmenwerk für den standardisierten Austausch von Informationen für die Authentifizierung und Autorisierung bereitgestellt. Außerdem können auch beliebige weitere Attribute zu Personen oder anderen Dingen übermittelt werden. Abbildung 8 stellt den Aufbau des Rahmenwerks dar.

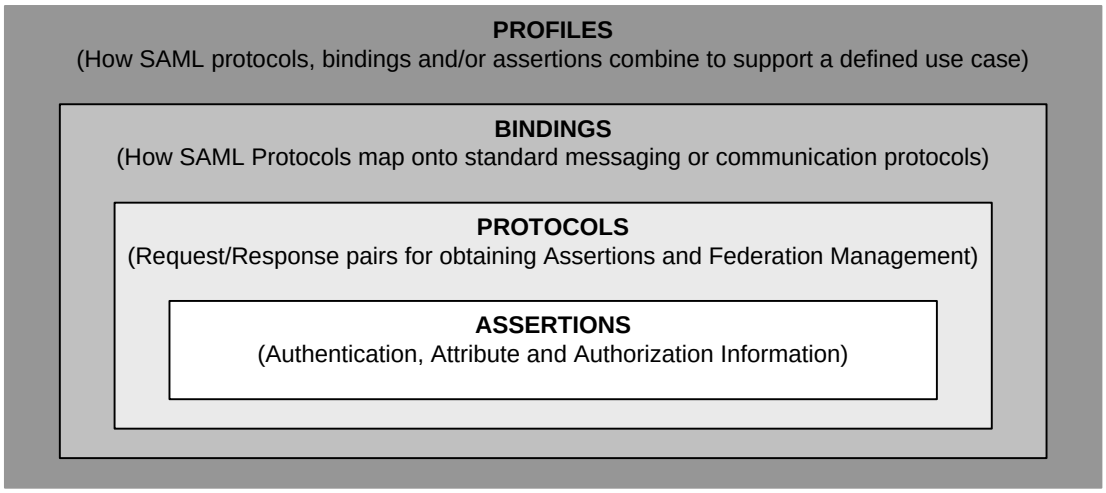

Abbildung 8: SAML Rahmenwerk[28]

Den Kern des Rahmenswerks bilden Assertions, die zwischen sich gegenseitig vertrauenden Teilnehmern ausgetauscht werden (circle of trust). Eine Assertion stellt dabei eine Behauptung dar, die von einem behauptenden Teilnehmer (engl. asserting party) an einen der Behauptung trauenden Teilnehmer (engl. relying party) übermittelt wird. Jede Assertion bezieht sich auf ein Subjekt (im Sinne von Autorisierungen, vgl. Abschnitt 2.8) oder allgemeiner auf eine Identität. Die OASIS definiert drei Arten von Assertions:

- Authentication Assertion
  Der Teilnehmer, gegenüber dem sich ein Subjekt erfolgreich authentisiert hat, stellt eine Authentication Assertion für den Teilnehmer aus, dessen Ressourcen das Subjekt in Anspruch nehmen möchte.

- Attribute Assertion
  Attribute Assertions enthalten spezifische Informationen zu einer Identität. Dies kann beispielsweise der Name zu einer Person oder Statusinformationen eines Benutzerkontos sein.

- Authorization Decision Assertion
  Mit einer Authorization Decision Assertion wird festgelegt, ob ein Subjekt Zugriff auf eine angefragte Ressource erhält. Das bedeutet, dass die Autorisierungsentscheidung nicht von der Ressource selbst, sondern von einer vertrauenswürdigen Instanz getroffen wird.

Jede Assertion enthält Informationen über die ausstellende Instanz und das Subjekt, auf die sich die Assertion bezieht. Darüber hinaus kann eine Vielzahl unterschiedlicher Bedingungen angegeben werden. So ist beispielsweise die Angabe einer Gültigkeitsdauer der Behauptung möglich. Mit der SAML-Version 2.0 kann zu Authentication Assertions ein Authentication Context mit angegeben werden. Der Authentication Context gibt an, welche Prozesse, Prozeduren, Mechanismen etc. seitens der Authentifizierungsinstanz zum Einsatz kommen, um ein Subjekt zu authentifizieren. Dabei geht es nicht nur um technische Aspekte, sondern auch um Aspekte, die die organisatorischen Maßnahmen und rechtlichen

[28] Abbildung nach [OASIS05e], S. 9

Vorschriften innerhalb einer Organisation betreffen. Das folgende Beispiel soll dies verdeutlichen.

Beispiel:
Die Studierenden einer Hochschule erhalten nur gegen Vorlage ihres Personalausweises eine Zugangskennung (organisatorischer Prozess). Die Zugangskennung kann ausschließlich für Anwendungen verwendet werden, die mit der zentralen Authentifizierungsinstanz über eine mit SSL 3.0 oder TLS 1.0 verschlüsselte Verbindung kommunizieren (organisatorischer Prozess und Verschlüsselungsmechanismen).

Mit Protocols werden die Interaktionsmöglichkeiten zwischen den Teilnehmern festgelegt. Die Protokolle folgen dem Request-Response-Paradigma. Es werden also Nachrichten (engl. messages) ausgetauscht. Der SAML Requester sendet eine Anfrage (SAML Request) an einen SAML Responder. Die Anfrage enthält eine oder mehrere Assertions. Der Empfänger der Anfrage verarbeitet die enthaltenen Assertions und gibt in einer Antwort (SAML Response) den Erfolg der Verarbeitung an. Der empfangende Teilnehmer übernimmt die Rolle des SAML Responder.

Mit Bindings wird festgelegt, über welche Standardprotokolle die SAML-Nachrichten abgebildet werden. Zu den Standardprotokollen gehören SOAP und HTTP. Je nach SAML-Version sind verschiedene Binding-Varianten pro Standardprotokoll möglich. So werden beispielsweise in der SAML-Version 2.0 ein SOAP Binding und ein Reverse SOAP (PAOS) Binding spezifiziert.

In Profiles wird für ausgewählte Anwendungsfälle beschrieben, wie SAML Assertions und SAML Protocols einzusetzen sind. Zu den Profiles gehört beispielsweise das Web Browser SSO Profile, das angibt, wie Authentication Assertions zwischen einem Identity Provider und mehreren Service Providern ausgetauscht werden müssen, um ein Single-Sign-On über mehrere Webanwendungen zu realisieren.

2.11.3 SPML

Die Service Provisioning Markup Languague (SPML) wurde von der OASIS in den Versionen 1.0 und 2.0 spezifiziert (vgl. [OASIS03c] und [OASIS06b]). Mit SPML wird ein XML-basiertes Protokoll definiert, das das Anlegen, Ändern sowie Löschen von Identitäten ermöglicht. Damit werden Operationen bereitgestellt, die man für ein Provisioning benötigt. Darüber hinaus können mit SPML einzelne Identitäten abgefragt werden.

In der Version 2.0 von SPML wird ein Domänenmodell bestehend aus den folgenden vier Elementen angegeben:

- Requesting Authority (RA)
  Eine Requesting Authority ist eine Softwarekomponente, die SPML-Anfragen an einen Provisioning Service Provider stellt. Im Sinne serviceorientierter Architekturen handelt es sich bei der RA um einen Service Consumer, der einen Service in Anspruch nimmt, um Identitäten beispielsweise anzulegen oder zu verändern.

- Provisioning Service Provider (PSP)
  Der Provisioning Service Provider ist eine Softwarekomponente, die SPML-Anfragen entgegennimmt und verarbeitet. Das Ergebnis einer jeden SPML-Anfrage übermittelt der PSP der Requesting Authority.

- Provisioning Service Target (PST)
  Ein Provisioning Service Target ist eine Komponente, die an einen Provisioning Service Provider angebunden ist und die das tatsächliche Ziel einer Provisionierungsanfrage darstellt. Gemäß serviceorientierter Architekturen handelt es sich um eine Ressource, für die mit Hilfe eines Providers ein Service bereitgestellt wird. Bei dem PST kann es sich beispielsweise um einen Verzeichnisdienst handeln, in dem Personendaten enthalten sind.

  Es wird verlangt, dass mindestens ein Provisioning Service Target pro Provisioning Service Provider existiert. Dabei müssen die PST eindeutig identifizierbar sein. Die Schnittstellen zwischen den Provisioning Service Providern und den Provisioning Service Targets kann beliebig sein. So erfolgt der Zugriff auf ein PST über Protokolle wie beispielsweise LDAP oder JDBC. Ein Provisioning Service Target agiert niemals als Provider.

- Provisioning Service Object (PSO)
  Ein Provisioning Service Target enthält eine Sammlung von Provisioning Service Objects. Bei den Objekten handelt es sich letztlich um die Identitäten, die von einem PST verwaltet werden. Jedes PSO ist eindeutig identifizierbar und wird nur von einem PST verwaltet.

Die Abbildung 9 zeigt ein Beispiel für die genannten Komponenten und deren Beziehungen zueinander.

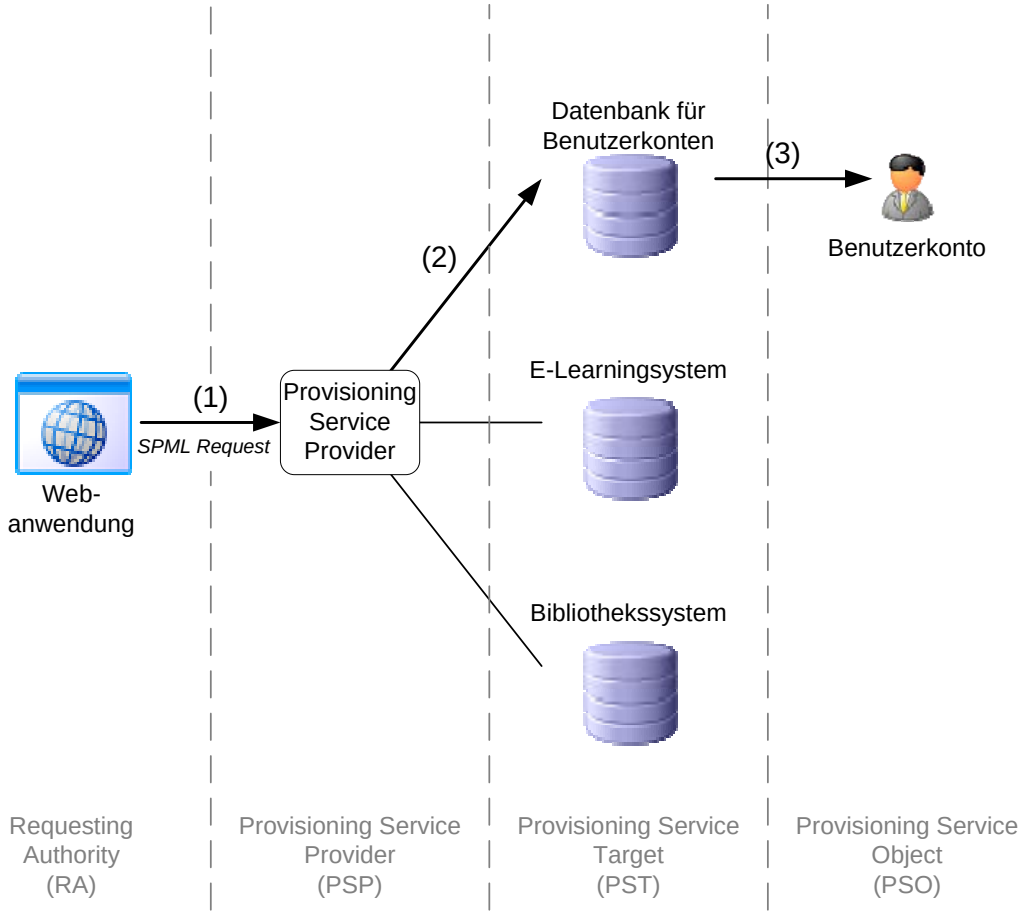

Abbildung 9: Komponenten von SPML

Die Webanwendung stellt eine Requesting Authority (RA) dar, die eine SPML-Anfrage zum Einfügen eines Benutzerkontos (1) an einen Provisioning Service Provider (PSP) richtet. An dem PSP sind mehrere Provisioning Service Targets (PST) angebunden: Eine Datenbank für Benutzerkonten, ein E-Learningsystem und ein Bibliothekssystem. Die SPML-Anfrage wird vom Provisioning Service Provider in einen Datenbankaufruf für Benutzerkontendatenbank umgesetzt (2). Dies führt wiederum dazu, dass ein neuer Eintrag, ein neues Provisioning Service Object (PSO), erzeugt wird (3). Nachdem das Benutzerkonto angelegt wurde, meldet der Provisioning Service Provider als Ergebnis den Erfolg der Operation an die Requesting Authority.

Das SPML-Protokoll stellt neben den genannten Operationen zum Einfügen (add), Ändern (modify), Löschen (delete) und Abfragen (lookup) einzelner Identitäten mit der Version 2.0 eine weitere Operation bereit, die die Provisioning Service Targets eines Provisioning Service Provider auflistet (listTargets). Zusätzlich zu diesen unter dem Begriff Core Operations zusammengefassten Operationen enthält das Protokoll noch Erweiterungen, so genante Capabilities, die die Serviceteilnehmer optional unterstützen können. Die Standard Capabilities der SPML-Version 2.0 können der Tabelle 1 entnommen werden. Darüber hinaus erlaubt SPML die Definition benutzerspezifischer Zusätze (Custom Capabilities).

Capability	**Beschreibung**
Async Capability	Ermöglicht die asynchrone Bearbeitung von SPML-Anfragen.
Batch Capability	Ermöglicht die Abarbeitung einer Serie von gleichen oder verschiedenen Operationen, die sich auf beliebige Provisioning Service Targets sowie Provisioning Service Objects beziehen können.
Bulk Capability	Ermöglicht die Änderung oder Löschung mehrerer Provisioning Service Objects eines Provisioning Service Targets mit einer einzigen SPML-Anfrage.
Password Capability	Ermöglicht das Setzen von Passwörtern sowie das Prüfen von Passwörtern auf ihre Gültigkeit für ein Provisioning Service Object. Des Weiteren können Passwörter gesperrt und zurückgesetzt werden.
Reference Capability	Ermöglicht, dass ein Provisioning Service Object Referenzen auf andere Objekte enthält.
Search Capability	Ermöglicht die Suche nach Provisioning Service Objects innerhalb eines Provisioning Service Targets unter Angabe verschiedener Suchkriterien. Ein Iterieren über die im Ergebnis enthaltenen Provisioning Service Objects ist ebenfalls möglich.
Suspend Capability	Ermöglicht das Aktivieren und Deaktivieren von Provisioning Service Objects. Dabei handelt es sich in der Regel um Benutzerkonten.
Updates Capability	Ermöglicht wie bei der Bulk Capability die Änderung von Attributen mehrerer Provisioning Service Objects. Dabei können zusätzlich Zeitkriterien mit angegeben und über die Anfragen und Antworten iteriert werden.

Tabelle 1: Standard Capabilities der SPML-Version 2.0

SPML unterstützt keine Transaktionen. OASIS begründet dies in [OASIS06b] auf Seite 39 folgendermaßen:

„Provisioning operations are notoriously difficult to undo and redo. For security reasons, many systems and applications will not allow certain identity management operations to be fully reversed or repeated."

OASIS schlägt vor, Custom Capabilities zu verwenden, um beispielsweise Atomarität zu ermöglichen.

Als Binding für das SPML-Protokoll werden seit der Version 1.0 (vgl. [OASIS03d]) nur ein Binding über SOAP und über einfache Textdateien spezifiziert.

Die Identitätsdaten, die in einer SPML-Nachricht enthalten sein können, unterliegen keinem vorgeschriebenen Datenmodell. Aus diesem Grund spezifiziert die OASIS optionale Profile, mit denen das Datenmodell festgelegt wird (vgl. [OASIS06c] und [OASIS06d]).

2.11.4 XML Encryption, XML Signature

Zur Wahrung der Vertraulichkeit der Daten kann bei den vorgestellten Standards wie beispielsweise SOAP, SAML und SPML die gesamte Kommunikation mittels SSL/TLS verschlüsselt werden. Neben dieser Verschlüsselung auf Verbindungsebene besteht auch die Möglichkeit, einzelne XML-Nachrichten oder Teile von Nachrichten mit Hilfe von XML Encryption (vgl. [W3C02a]) zu verschlüsseln. Der Vorteil der Verschlüsselung auf Nachrichtenebene liegt in der Reduzierung der zu transportierenden Datenpakete. Nur vertrauliche Inhalte werden dabei mit XML Encryption verschlüsselt. Dabei können auch die Inhalte ineinander geschachtelt verschlüsselt werden. So kann beispielsweise der Empfänger einer Nachricht nur Teile der Information entschlüsseln und die noch verschlüsselten Informationen an weitere Empfänger senden.

Mit XML Signature (vgl. [W3C02b]) kann eine Nachricht oder Teile der Nachricht digital signiert werden. Die Signierung der Daten dient der Wahrung der Integrität.

2.11.5 XACML

Bei der eXtensible Access Control Markup Language (XACML) handelt es sich um einen weiteren OASIS-Standard, der in den Versionen 1.0, 1.1 und 2.0 vorliegt. Zum Zeitpunkt der Erstellung dieser Arbeit wurden die ersten Entwürfe für die Version 3.0 veröffentlicht.

Mit XACML können standardisierte Richtlinien für den Zugriff von Subjekten auf Objekte erstellt werden. In XACML werden drei Grundelemente für die Erstellung von Richtlinien verwendet:

- Rule
  Eine Rule enthält eine Regel, die für sich ausgewertet werden kann und als Ergebnis wahr oder falsch liefert. Eine Rule alleine reicht jedoch noch nicht aus, um eine Autorisierungsentscheidung zu treffen.

- Policy
  Eine Policy enthält einen Satz von Attributen. In der Policy wird angegeben, wie die einzelnen Regeln zu kombinieren und auszuwerten sind.

- PolicySet
  Ein PolicySet enthält mehrere Policy- und PolicySet-Elemente. Auch hierbei wird angegeben, wie die einzelnen Elemente kombiniert und ausgewertet werden.

Weitere Informationen zu XACML erhält man unter [OASIS07].

2.12 Initiativen für föderierte Identitätsmanagementsysteme

2.12.1 WS-* (WS-Star)

Unter WS-* werden mehrere Standards im Bereich des föderierten Identitätsmanagement zusammengefasst, an deren Entwicklung im Wesentlichen Microsoft und IBM beteiligt sind. WS-* beinhaltet unter anderem die Standards WS-Security, WS-Trust, WS-Policy sowie WS-Federation. Informationen zu den Standards findet man unter [OIO07] und [VSV06].

2.12.2 Liberty Alliance

Die Liberty Alliance, auch bekannt unter dem Namen Project Liberty, ist ein Konsortium von über 160 Mitgliedsorganisationen, das die Entwicklung von Spezifikationen für föderiertes Identitätsmanagement vorantreibt. Dabei wird auf existierende Standards wie SAML zurückgegriffen. Zu den wichtigsten Spezifikationen gehören die folgenden:

- Liberty Identity Federation Framework (ID-FF)
  In diesem Rahmenwerk wird ein Single-Sign-On (SSO) basierend auf SAML für diverse Szenarien beschrieben. Darüber hinaus spezifiziert es die Kopplung verschiedener Benutzerkonten einer Person.

- Liberty Identity Web Services Framework (ID-WSF)
  In ID-WSF wird festgelegt, welche Voraussetzungen Identitätsdienste erfüllen müssen, um zueinander kompatibel zu sein. Dabei werden Verfahren zur Beschreibung und zum Auffinden (engl. discovery) von Identitätsservices spezifiziert. ID-WSF ermöglicht, dass Benutzer die Übermittlung ihrer Attribute steuern können.

- Liberty Identity Services Interface Specifications (ID-SIS)
  ID-SIS stellt einen Aufsatz von ID-WSF dar und ermöglicht spezielle kompatible Identitätsdienste wie calendar service oder contacts service.

Weitere Informationen zu dieser Initiative können [Liberty07] entnommen werden.

2.12.3 Shibboleth und DFN-AAI

Eine besonders wichtige Initiative für deutsche Hochschulen stellt aktuell die Authentifizierungs- und Autorisierungs-Infrastruktur des Deutsches Forschungsnetz e.V. (DFN-AAI) dar. Aus diesem Grund wird auf das Thema etwas näher eingegangen.

Mit Hilfe von DFN-AAI wird für die wissenschaftlichen Einrichtungen in Deutschland der Zugang zu geschützten Ressourcen ermöglicht. Bei den Ressourcen handelt es sich beispielsweise um Online-Zeitschriften, Online-Recherchedatenbanken, E-Learningsysteme sowie Download-Server für Software von Firmen wie Microsoft und Autocad.

Die DFN-AAI verwendet das Produkt Shibboleth vom Middleware Architecture Committee for Education (MACE) des Internet2-Konsortiums[29]. Shibboleth selbst basiert in der aktuellen Version 1.3 auf dem OASIS-Standard SAML 1.1 und beinhaltet die folgenden drei Komponenten:

- Identity Provider (IdP)
  Ein Identity Provider verwaltet Berechtigungsnachweise und Attribute zu Personen. Der IdP stellt Authentication Assertions und Attribute Assertions (vgl. Abschnitt 2.11.2 „SAML“) für Service Provider aus. Dabei werden Attribute Assertions auch zweckentfremdet, um Autorisierungsinformationen zu übertragen.

- Service Provider (SP)
  Der Service Provider verwaltet geschützte Ressourcen. Anhand der Authentication Assertions und Attribute Assertions entscheidet der Service Provider darüber, ob der Zugriff auf eine geschützte Ressource gewährt wird.

- WAYF Service
  Die Abkürzung WAYF steht für „Where are you from?“ Diese optionale Komponente kann von einem Service Provider genutzt werden, um herauszufinden, welcher Identity Provider für den Benutzer zuständig ist. Die Zuordnung zu einem IdP kann entweder automatisch oder mit Interaktion des Benutzers erfolgen.

Zwischen den Identity Providern, den Service Providern und den WAYF Services muss eine Vertrauensbeziehung hergestellt werden. Dazu werden Informationen wie Zertifikate und digitale Signaturen in Metadaten gespeichert, mit denen insbesondere die Schutzziele Authentizität, Integrität und Vertraulichkeit verfolgt werden.

Der DFN verwaltet für die wissenschaftlichen Einrichtungen in Deutschland die Metadaten, die jeder Teilnehmer übernehmen muss. Des Weiteren betreibt der DFN einen WAYF Service.

Das folgende Beispiel skizziert ein mögliches Szenario im föderativen Verbund der DFN-Mitglieder.

Beispiel:
Ein Mitarbeiter der FernUniversität in Hagen möchte auf eine Online-Datenbank mit Zeitschriften zugreifen, die an der Freien Universität Berlin angeboten wird. Bei der Online-Datenbank handelt es sich um ein Produkt eines Verlags. In der Lizenz wurde vereinbart, dass nur Mitarbeiter deutscher Hochschulen dieses Produkt nutzen dürfen. Die Online-Datenbank wird durch einen Shibboleth Service Provider geschützt. Der Mitarbeiter der FernUniversität, der nun die Ressource aufruft, muss zunächst authentifiziert werden. Durch den Aufruf eines Links auf der Anmeldeseite teilt er mit, dass er nicht Mitglied der Freien Universität Berlin ist. Der Mitarbeiter gelangt zum WAYF Service des DFN. Dort

[29] siehe [MACE07]

gibt er an, dass die FernUniversität in Hagen seine Heimathochschule ist. Daraufhin wird der Mitarbeiter zum Identity Provider der FernUniversität weitervermittelt. Nach erfolgreicher Authentifizierung mit Benutzernamen und Passwort kehrt der Mitarbeiter mit einer Authentication Assertion zum Service Provider für die Online-Datenbank zurück. Der Service Provider prüft die Gültigkeit der Authentication Assertion gegen den Identity Provider der FernUniversität in Hagen. Nach der Bestätigung der Authentifizierung fragt der Service Provider den Identity Provider nach weiteren Attributen zu dieser Person. Insbesondere wird gefragt, ob die Person Mitarbeiter ist. Nach der Bestätigung des Mitarbeiterstatus erhält der Mitarbeiter der FernUniversität in Hagen Zugriff auf die Online-Datenbank.

Das Beispiel zeigt vereinfacht eine mögliche Interaktion zwischen den Komponenten in einer Shibboleth-Infrastruktur. Die Festlegung, in welcher Reihenfolge bestimmte Nachrichten ausgetauscht werden, um beispielsweise ein Single-Sign-On zu realisieren, bezeichnet man wie bei SAML auch bei Shibboleth als Profile. Eine detaillierte Beschreibung der Profile von Shibboleth können in [ScCa05] und [Cant05] nachgelesen werden.

Shibboleth unterstützt auch Single-Sign-On. Das Single-Sign-On beschränkt sich jedoch im Wesentlichen nur auf Webbrowseranwendungen. Service Provider für beispielsweise den Anmeldeprozess von Betriebssystemen oder E-Mail-Clients befinden sich noch in der Entwicklung.

Weitere wichtige Funktionen stellt Shibboleth mit der Attribute Release Policy (ARP) des Identity Providers sowie der Attribute Acceptance Policy (AAP) des Service Providers bereit. Mit der ARP wird für den Identity Provider festgelegt, welche Attribute an welchen Service Provider übermittelt werden dürfen. Mit der AAP regelt wiederum der Service Provider, welche Attribute und Attributwerte dieser von einem Identity Provider akzeptiert.

Informationen zur DFN-AAI erhält man in [DFN07]. Detaillierte Beschreibungen zu dem Produkt Shibboleth können unter [Internet2_07a] gefunden werden.

2.13 Workflowmanagement

Zunächst einmal wird der bereits in Abschnitt 2.2.3j) verwendete Begriff Workflow präzisiert.

Workflow: „The computerised facilitation or automation of a business process, in whole or part.“[30]

Die Erstellung von Dokumenten, die Verteilung von Informationen sowie komplexere Aufgaben müssen innerhalb einer Organisation zwischen mehreren Personen koordiniert werden. Mit Workflows versucht man, diese Prozesse zu unterstützen oder zu automatisieren. Dabei werden die Workflows mit Hilfe von Computern verwaltet. Man spricht von Workflowmanagementsystemen.

Workflowmanagementsystem: „A system that completely defines, manages and executes ‘workflows’ through the execution of software whose order of execution is driven by a computer representation of the workflow logic.“[31]

[30] [WFMC95], S. 6
[31] [WFMC95], S. 6

Für die Beschreibung von Workflows existiert eine Reihe von Standards. Hierzu gehören insbesondere die XML Process Definition Language (XPDL), die Business Process Execution Language (BPEL) sowie die Business Process Modelling Language (BPML).

Die Workflow Extensible Markup Languague (Wf-XML), die auf dem Asynchronous Service Access Protocol (ASAP) basiert, dient der einheitlichen Kommunikationen zwischen den einzelnen Komponenten eines oder mehrerer Workflowmanagementsysteme.

Weitere Informationen zu dem Thema Workflowmanagement können beispielsweise [WFMC07] entnommen werden.

3 Rahmenbedingungen

3.1 Organisatorische Strukturen

Die Organisationsstruktur der deutschen Hochschulen folgt einem gewissen Grundmuster. An oberster Stelle der Hierarchie steht die Hochschulleitung. Auf der zweiten Ebene befinden sich in der Regel die Fakultäten (bzw. Fachbereiche), die Bibliotheken sowie weitere Einrichtungen, die einen zentralen Charakter besitzen. Zu den zentralen Einrichtungen gehören unter anderem die Studierenden- und Personalverwaltung. Diese Auflistung der Organisationseinheiten ist sehr vereinfacht und bei weitem nicht vollständig. Sie hilft jedoch, in etwa die Verantwortlichkeiten für ein Identitätsmanagement zuzuordnen.

Die Hochschulleitung setzt einige Vorgaben, die im Zusammenhang mit dem Identitätsmanagementsystem bestehen. In den folgenden Abschnitten werden einige solcher Vorgaben genannt. Die Fakultäten sind in der Regel Konsumenten von Identitätsdaten. Ein besonderes Interesse besteht für die Fakultäten in der Frage, welche Personen welche Aufgaben innerhalb der Fakultät zu erfüllen haben. Einige zentrale Einrichtungen besitzen die Verantwortung, bestimmte Personengruppen an einer Hochschule IT-gestützt zu erfassen und zu pflegen.

Bei der Betrachtung der Bereiche ist nicht nur die Bestimmung der Verantwortlichkeiten von Interesse, sondern auch die Identifizierung verfügbaren Wissens des Personals, das für den Aufbau und Betrieb des Identitätsmanagementsystems benötigt wird.

3.2 Technische Strukturen

An Hochschulen existieren diverse IT-Systeme, die meist miteinander vernetzt sind. Zunächst einmal müssen all die Systeme identifiziert werden, die an dem Identitätsmanagementsystem teilnehmen sollen. Für jedes dieser Altsysteme muss dann die Frage gestellt werden, ob es in kurzer Zeit abgelöst werden oder weiterhin mit möglichst wenigen Änderungen fortbestehen soll.

Zu den Systemen, in denen Identitätsdaten an einer Hochschule verwaltet werden, gehören in jedem Fall die Systeme der Studierendenverwaltung, der Personalverwaltung sowie der Bibliotheken. Hinzu kommen noch Anwendungen im Bereich E-Learning und Dienste wie E-Mail, die von ausgewählten Organisationseinheiten einer Hochschule angeboten werden. Für jedes der Systeme muss letztlich geklärt werden, ob es in das Identitätsmanagementsystem zu integrieren ist und welche Aufgabe es übernehmen soll. Die Qualität, Aktualität und Verfügbarkeit der Daten entscheiden darüber, ob ein System zu einer autoritativen Datenquelle werden kann oder ob eher Identitätsdaten aus anderen Systemen übernommen werden sollen.

Neben der Betrachtung der zur Verfügung stehenden Identitätsdaten müssen auch die Schnittstellen der einzelnen Systeme sowie die Leistungsfähigkeit und Ausfallsicherheit der beteiligten Hardware analysiert werden.

3.3 Rechtliche Anforderungen

In diesem Abschnitt werden die Gesetze, Verordnungen und Richtlinien untersucht, die für den Integrationsprozess von Identitätsdaten relevant sind. Die Anwendbarkeit bestimmter

Rechtsvorschriften hängt davon ab, ob und in welchen Umfang die Identitätsdaten nur innerhalb einer Hochschule oder sogar bedingt durch eine Kooperation von Universitäten über Landesgrenzen hinweg verwendet werden. Je nach Integrationsszenario gelten Regelungen auf Hochschul-, Landes-, Bundes-, europäischer oder internationaler Ebene. Eine besondere Betrachtung erfolgt für die gesetzlichen Vorschriften zum Datenschutz, da die Identitäten häufig personenbezogene Daten enthalten oder mit solchen verknüpft sind.

3.3.1 Internationale und europäische Richtlinien

Mit internationalen und europäischen Richtlinien werden unter anderem Grundsätze verabschiedet, die in nationalen Gesetzgebungen berücksichtigt werden sollen. Man kann diese als Mindestanforderungen auffassen, die immer gelten müssen. Die wesentlichen internationalen und europäischen Richtlinien sind dem Verzeichnis der Gesetze, Verordnungen, Richtlinien und Chartas (Abschnitt 8) zu entnehmen. Sie beziehen sich auf personenbezogene Daten, auf die elektronische Kommunikation sowie elektronische Signaturen. Als Beispiel sei die Charta der Grundrechte der Europäischen Union genannt. Artikel 8 regelt den Schutz personenbezogener Daten. Danach dürfen Daten „... nur nach Treu und Glauben für festgelegte Zwecke und mit Einwilligung der betroffenen Person oder auf einer sonstigen gesetzlich geregelten legitimen Grundlage verarbeitet werden.“[32] Die Zweckbindung der Daten und die Vorgabe der Einwilligung der betroffenen Person stellen Grundsätze vieler Rechtsvorschriften zum Datenschutz dar.

3.3.2 Bundesgesetze

Bundesdatenschutzgesetz (BDSG)

Das Bundesdatenschutzgesetz enthält Regelungen, die den Umgang mit personenbezogenen Daten festlegen. Zweck des BDSG ist es, „… den einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird.“[33] Es gilt unter anderem nach §1 Abs. 2 Nr. 1 für öffentliche Stellen des Bundes und nach §1 Abs. 2 Nr. 2 für „… öffentliche Stellen der Länder, soweit der Datenschutz nicht durch Landesgesetz geregelt ist …“ Da die Hochschulen öffentliche Einrichtungen der Länder darstellen, müssen also insbesondere die jeweiligen Datenschutzgesetze der Bundesländer für den Integrationsprozess betrachtet werden. Diese orientieren sich jedoch alle an den Grundsätzen des Bundesdatenschutzgesetzes. Deshalb erfolgt an dieser Stelle eine ausführlichere Betrachtung des BDSG.
Personenbezogene Daten werden definiert als „… Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener).“[34] Im Kontext der Hochschule können diese zum Beispiel Matrikel- und Personalnummern oder auch E-Mail-Adressen sein. Unter einer automatisierten Verarbeitung wird die Erhebung, Verarbeitung und Nutzung der Daten durch EDV-Systeme verstanden.[35] Die Verarbeitung beinhaltet „… das Speichern, Verändern, Übermitteln, Sperren und Löschen personenbezogener Daten.“[36] Das Bundesdatenschutzgesetz fordert die Datenvermeidung und Datensparsamkeit.[37] Es sollen also so wenige personenbezogene Daten wie möglich erhoben, verarbeitet und genutzt werden. Des Weiteren sollen Anonymisierung und Pseu-

[32] Artikel 8 Abs. 2 Satz 1 Charta der Grundrechte der Europäischen Union
[33] §1 Abs. 1 BDSG
[34] §3 Abs. 1 BDSG
[35] Vgl. §3 Abs. 2 BDSG
[36] §3 Abs. 4 BDSG
[37] Vgl. § 3a Satz 1 BDSG

donymisierung verwendet werden.[38] Die automatisierte Verarbeitung der Daten ist nur dann zulässig, wenn die betroffene Person nach freier Entscheidung einwilligt oder eine Legitimierung durch andere Rechtsvorschriften besteht.[39] Die Einwilligung sollte schriftlich erfolgen, wenn nicht andere Umstände eine andere Form erfordern.[40] Betrachtet man den Aufwand, der entsteht, um von allen Studierenden eine schriftliche Einwilligung für eine zweckgebundene Verarbeitung ihrer Daten zu erhalten, so bestehen wahrscheinlich besondere Umstände, die zum Beispiel ein Online-Verfahren rechtfertigen. Hierzu müssen die Regelungen der Bundesländer herangezogen werden. Nach §4b Abs. 5 des Bundesdatenschutzgesetzes trägt „[d]ie Verantwortung für die Zulässigkeit der Übermittlung ... die übermittelnde Stelle." Der Daten empfangenden Stelle muss der Verwendungszweck mitgeteilt werden.[41] Der Betroffene kann eine Auskunft darüber verlangen, welche personenbezogenen Daten von ihm gespeichert wurden, welche Stelle diese erhoben hat und welche Empfänger diese für welchen Zweck speichern.[42] Regelungen zur Korrektur, Sperrung und Löschung von Datensätzen, die der Betroffene veranlassen darf, finden sich in den §20 und §35 des Bundesdatenschutzgesetzes.

Hochschulrahmengesetz (HRG)

Im Hochschulrahmengesetz werden Grundsätze für Hochschulen aufgestellt, die nach Landesrecht als staatliche Hochschulen aufgefasst werden. Nach §13 Abs. 1 des HRG sollen „[b]ei der Reform von Studium und Lehre und bei der Bereitstellung des Lehrangebots … die Möglichkeiten eines Fernstudiums sowie der Informations- und Kommunikationstechnik genutzt werden". Ein weiterer Aspekt für den Integrationsprozess wird im dritten Kapitel des Hochschulrahmengesetzes, Mitglieder der Hochschule, aufgezeigt. An einer Hochschule existieren mehrere Mitgliedsgruppen, die unterschiedliche Merkmale und Rechte besitzen können. Nach §36 Abs. 1 des Hochschulrahmengesetzes sind „Mitglieder der Hochschule … die an der Hochschule nicht nur vorübergehend oder gastweise hauptberuflich Tätigen und die eingeschriebenen Studierenden." Professoren stehen durch §36 Abs. 2 auch die „…nach dem Eintritt in den Ruhestand … mit der Lehrbefugnis verbundenen Rechte zur Abhaltung von Lehrveranstaltungen und zur Beteiligung an Prüfungsverfahren zu." Neben den nach dem Hochschulrahmengesetz klar zu identifizierenden Mitgliedsgruppen einer Hochschule wird nach §36 Abs. 1 durch Landesrecht die Stellung weiterer Personengruppen geregelt.

Signaturgesetz (SigG)

Nach den Abschnitten 2.6 und 2.7 können für bestimmte Bereiche elektronische Signaturen verwendet werden. Das Signaturgesetz (SigG) muss in diesem Fall beachtet werden. So wird beispielsweise in §7 des SigG der Inhalt von qualifizierten Zertifikaten, also Zertifikaten für natürliche Personen, festgesetzt. Nach §7 Abs.1 Nr. 1 des SigG kann anstelle des Namens des Signaturschlüsselinhabers ein unverwechselbares Pseudonym verwendet werden Dieser Fall ist auch für Personen mit einem Künstlernamen interessant. Auch kommt es vor, dass in Publikationen ein anderer Name verwendet wird, als auf dem Ausweis oder Reisepass. Die Ursache hierfür sind häufig falsche oder abweichende Transkriptionen ausländischer Namen bei den Behörden.

[38] Vgl. § 3a Satz 2 BDSG
[39] Vgl. § 4 Abs. 1 BDSG
[40] Vgl. § 4a Abs. 1 Satz 3 BDSG
[41] Vgl. §4b Abs. 6 BDSG
[42] Vgl. §19 Abs. 1, §34 Abs. 1 BDSG

Signaturverordnung (SigV)

Aufgrund des Signaturgesetzes und europäischer Richtlinien erließ die Bundesregierung die Signaturverordnung (SigV), in der unter anderem die Inhalte eines nach §4 Abs. 2 Satz 4 des SigG geforderten Sicherheitskonzeptes festgelegt werden. Dieses enthält unter anderem Regelungen darüber, wie eine Identitätsprüfung und ein Attributnachweis erfolgen müssen. So müssen der Personalausweis oder Dokumente mit einer vergleichbaren Sicherheit zur Prüfung vorgelegt werden.[43] Die hierfür nötigen organisatorischen Maßnahmen können, wenn überhaupt, nur von wenigen Stellen der Hochschule geleistet werden. Zudem müssen die geprüften Daten als solche markiert werden und besitzen im Vergleich zu ungeprüften Daten eine höhere Priorität.

Gesetze zur Telekommunikation und zu Telediensten

Wegen der Netzinfrastruktur an der Hochschule und der über das Internet angebotenen Dienste gelten das Telemediengesetz (TMG) und Telekommunikationsgesetz (TKG). Je nachdem welche Dienste, wie zum Beispiel WLAN (Wireless Local Area Network), angeboten werden, ist die Hochschule verpflichtet, Verbindungs- und Teilnehmerdaten zu speichern. Eine weitere Betrachtung dieser Gesetze ist im Rahmen dieser Arbeit nicht möglich.

3.3.3 Gesetze der Bundesländer

Als Beispiele für Gesetze auf Landesebene werden die Regelungen in den Bundesländern Nordrhein-Westfalen oder Berlin betrachtet. Diese Entscheidung richtet sich nicht nach irgendwelchen Besonderheiten in der Rechtsprechung in diesen Bundesländern, sondern wird durch den Sitz der FernUniversität in Hagen und meinen Wohnort bestimmt.

Datenschutzgesetze der Bundesländer

Eine Betrachtung der einzelnen Datenschutzgesetze aller Bundesländer würde an dieser Stelle zu weit führen. Es erfolgt deshalb nur eine kurze exemplarische Betrachtung des Berliner Datenschutzgesetzes (BlnDSG). In diesem finden sich, wie in den anderen Datenschutzgesetzen auf Landesebene, die Grundsätze aus dem Bundesdatenschutzgesetz wieder. Zusätzlich werden noch weitere Forderungen für die automatisierte Verarbeitung personenbezogener Daten aufgestellt. So müssen nach §5 Abs. 2 Nr. 4 des Berliner Datenschutzgesetzes Maßnahmen getroffen werden, die die „...Vertraulichkeit[,] … Integrität[,] … Verfügbarkeit[,] … Authentizität[,] … Revisionsfähigkeit[,] … [und] …Transparenz …“ des Datentransportes gewährleisten. Nach §16 Abs. 1 Nr. 3 muss dem Betroffenen auf Antrag eine gebührenfreie Auskunft über „… die Herkunft der Daten und die Empfänger von Übermittlungen innerhalb der letzten zwei Jahre…“ erteilt werden. Es wird also neben dem bekannten Auskunftsrecht aus dem Bundesdatenschutzgesetz eine Regelung zur Aufbewahrungsdauer der Daten festgeschrieben. Dieser Aspekt ist wichtig, um zum Beispiel eine Abschätzung des benötigten Speicherplatzes für die Datenbank- und Backupsysteme vornehmen zu können.

Hochschulgesetze der Bundesländer

Zur exemplarischen Betrachtung der Hochschulgesetze der Bundesländer wird das Beispiel Nordrhein-Westfalen (NRW) herangezogen. Nach §7 Abs.1 des Hochschulgesetzes (HG)

[43] Vgl. §3 Abs. 1 SigV

für NRW sollen „… Möglichkeiten des Fern- und Verbundstudiums sowie der Informations- und Kommunikationstechnik …“ gefördert werden. In den Hochschulgesetzen der Bundesländer werden die Stellungen der sonstigen Mitgliedsgruppen einer Hochschule definiert (vgl. Abschnitt 3.3.2). Diese Festlegung erfolgt durch den §11 des Hochschulgesetzes für Nordrhein-Westfalen. Benannt werden darin Rektor/in, Kanzler/in, Professoren bzw. Professorinnen sowie Mitarbeiter/innen und Studierende.[44] §11 Abs. 4 des HG für NRW erweitert die Gruppe der Personen, die eine Zugehörigkeit zur Hochschule besitzen um zum Beispiel Privatdozentinnen und Privatdozenten, wissenschaftliche Hilfskräfte oder Gasthörerinnen und Gasthörer. Diesen zusätzlichen Gruppen möchte und muss man einen Zugriff auf die IT-Systeme der Hochschule erlauben, jedoch mit eingeschränkten Rechten.
Der zwölfte Abschnitt des Hochschulgesetzes für Nordrhein-Westfalen trifft Regelungen für das Zusammenwirken von Hochschulen. Die Kooperation kann sich sowohl auf Lehre und Studium beziehen als auch auf gemeinsame Einrichtungen mehrerer Hochschulen. Ein Studiengang, der von mehreren Hochschulen gemeinsam angeboten wird, verlangt unter anderem einen Austausch von Studierendendaten. Der Studierende muss allein schon aufgrund der Chancengleichheit die Möglichkeit erhalten, das entsprechende Leistungsangebot aller beteiligten Hochschulen nutzen zu können. Dies könnte zum Beispiel eine Recherche-Datenbank sein. Um notwendige Daten ohne Konflikte austauschen zu können, wird eine Zuordnung der Daten zu der jeweiligen Organisation, also der Hochschule, benötigt. Es reicht nicht aus, nur die Matrikelnummer zu speichern. Vielmehr muss die die Matrikelnummer in Kombination mit der Hochschule gespeichert werden.

3.3.4 Ordnungen, Satzungen und Vereinbarungen der Hochschule

Die gesamte Hochschule und einzelne Einrichtungen können Ordnungen, Satzungen und Vereinbarungen erlassen, die organisatorische Regelungen zu den IT-Systemen und der Datenerfassung beinhalten. Auf technischer Ebene können zum Beispiel Sicherheitsrichtlinien für die zu verwendenden Kommunikationsprotokolle und Verschlüsselungsverfahren manifestiert werden. Auch Regelungen zur Art und Weise der Dokumentation können Bestandteil einer Dienstvereinbarung oder Ähnlichem sein. Es ist also im Kontext des Integrationsprozesses wichtig, sich über die entsprechenden Regelungen innerhalb der Hochschule zu informieren.

3.4 Hochschulpolitische Vorgaben und Rahmenbedingungen

Die Hochschulleitung muss entscheiden, welchen Anforderungen das Identitätsmanagementsystem genügen soll. Einige der Anforderungen ergeben sich aus der Zusammenarbeit zwischen der Hochschule und anderen Organisationen, denen dann u. a. IT-Ressourcen zur Verfügung gestellt werden sollen. Beispiele hierfür sind Vereinbarungen über Kooperationsstudiengänge, die mit anderen Hochschulen getroffen werden, und die Verlagerung bestimmter Aufgaben der Hochschule in hochschuleigene bzw. hochschulnahe Kapitalgesellschaften.

Verbunden mit der an deutschen Hochschulen immer wichtiger werdenden Diskussion über Budget und Personal muss auch eine Entscheidung darüber getroffen werden, wie viel Personal für den Aufbau und Betrieb eines Identitätsmanagementsystems zum Einsatz kommen soll. Außerdem müssen Gelder für die notwendigen Softwarelizenzen und die Hardware bereitgestellt werden und eine Entscheidung darüber getroffen werden, in wie weit eine Refinanzierung zu erfolgen hat. Dabei ist zu beachten, dass eine Refinanzierung

[44] Vgl. §11 Abs. 1 und Abs. 1 HG für NRW

dazu führen kann, dass bestimmte Bereiche an dem IDMS nicht teilnehmen wollen oder können und somit von vornherein eine Eintrittsbarriere geschaffen wird. Das reduziert allerdings den Nutzen des Identitätsmanagementsystems erheblich.

Mögliche weitere Vorgaben können aus der Abstimmung der beteiligten Bereiche mit dem Personalrat und dem Datenschutzbeauftragten entstehen.

Die vorangegangenen Abschnitte haben gezeigt, dass sich aus bereits existierenden IT-Systemen und anderen Rahmenbedingungen Anforderungen an das Identitätsmanagementsystem ergeben können. Jedoch können auch Entscheidungen über zukünftig einzusetzende Softwareprodukte dadurch beeinflusst werden, ob eine Integration in ein geplantes oder bestehendes Identitätsmanagementsystem möglich ist. Somit sorgt ein IDMS selbst für neue Rahmenbedingungen.

4 Architektur

In diesem Abschnitt wird nun eine Architektur eines Identitätsmanagementsystems für eine Hochschule skizziert.

Die Architektur setzt sich aus vier Hauptkomponenten zusammen, die jeweils eine eigene Klasse von Architekturen darstellen. Abbildung 10 zeigt diese vier Architekturen, die über das Identitätsmanagementsystem-Rahmenwerk (IDMS-Rahmenwerk) zusammengefasst werden.

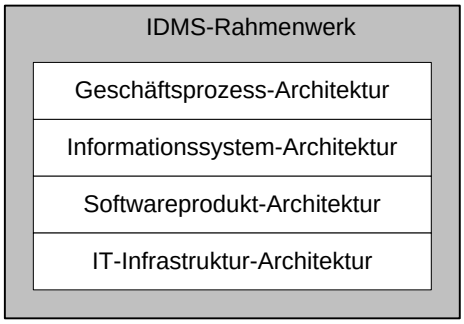

Abbildung 10: Rahmenwerk für ein Identitätsmanagementsystem

Die oberste Schicht des IDMS-Rahmenwerks stellt die Geschäftsprozess-Architektur dar, die sich aus den Geschäftsprozessen an einer Hochschule zusammensetzt. Sie wird folgendermaßen definiert:

Geschäftsprozess-Architektur: Eine Geschäftsprozess-Architektur setzt sich aus den einzelnen Schritten von Geschäftsvorgängen, deren Abhängigkeiten und Verknüpfungspunkten zusammen.

Ein möglicher Geschäftsprozess könnte der folgende sein:

Beispiel:
Eine Person soll erstmalig einen Arbeitsvertrag an einer Hochschule erhalten. Zu diesem Zweck erfasst ein Mitarbeiter der Personalstelle die Daten in einem System zur Mitarbeiterverwaltung. Wird der Arbeitsvertrag durch alle notwendigen Unterschriften gültig, so wird ein hochschulweites Benutzerkonto für den neuen Mitarbeiter in der Benutzerdatenbank des Hochschulrechenzentrums angelegt.

Der in dem Beispiel genannte Geschäftsprozess lässt sich sicherlich noch weiter fortführen. Der Ausschnitt reicht aber aus, um zu zeigen, dass mehrere Personen, Bereiche und auch IT-Systeme involviert sind.

Die nächste Schicht im IDMS-Rahmenwerk beinhaltet eine Informationssystem-Architektur, die wie folgt definiert wird:

Informationssystem-Architektur: Eine Informationssystemarchitektur basiert auf einer Daten-Architektur, die um Komponenten zur Erfassung, Extraktion, Transformation, Qualitätssicherung, Verteilung und Analyse ergänzt wird. Eine Daten-Architektur beinhaltet die Art und Weise und den Ort der Datenspeicherung sowie den Datentransport.

Für den Aufbau der Informationssystem-Architektur werden Ergebnisse aus der Geschäftsprozess-Architektur genutzt. Das vorangegangene Beispiel zur Geschäftsprozess-Architektur zeigte exemplarisch zwei IT-Systeme, nämlich die Benutzer- und Mitarbeiterdatenbank, zwischen denen Daten ausgetauscht werden müssen.

Für die Realisierung der Komponenten, die in der Informationssystem-Architektur abstrakt beschrieben werden, erfolgt in der Softwareprodukt-Architektur eine konkrete Zuweisung von ein oder mehreren Produkten. Die Softwareprodukt-Architektur wird dabei folgendermaßen definiert:

Softwareprodukt-Architektur: Eine Softwareprodukt-Architektur beinhaltet als Komponenten die Softwareprodukte, die zur Realisierung eines komplexen IT-Systems benötigt werden. Zusätzlich werden die Abhängigkeiten der Produkte untereinander und deren Schnittstellen zueinander beschrieben.

Die Softwareprodukt-Architektur generiert Anforderungen für die unterste Schicht im IDMS-Rahmenwerk, die so genannte IT-Infrastruktur-Architektur. Die IT-Infrastruktur-Architektur wird dabei wie folgt definiert:

IT-Infrastruktur-Architektur: Eine IT-Infrastruktur-Architektur beinhaltet sämtliche Hardware wie Server und Netzwerke, die zur Realisierung eines komplexen IT-Systems benötigt wird. Darüber werden in dieser Architektur die Gebäude sowie gebäudetechnische Komponenten erfasst, die zum Betrieb der Hardware notwendig sind.

Die eingesetzten Softwareprodukte sowie der Umfang der zu verarbeitenden Daten beeinflussen maßgeblich den Umfang und die Leistung von Servern und Netzwerken. Des Weiteren müssen Sicherheitsanforderungen wie beispielsweise Vertraulichkeit und Verfügbarkeit berücksichtigt werden.

In den nachfolgenden Kapiteln wird zunächst beschrieben, wie Geschäftsprozesse an einer Hochschule erfasst werden können (siehe Abschnitt 4.1). Die Angabe einer konkreten Geschäftsprozess-Architektur erfolgt im Rahmen dieser Arbeit nicht, da jede Hochschule ihre ganz individuellen Geschäftsprozesse besitzt, die sich schwer oder gar nicht in eine allgemeingültige Struktur überführen lassen.

Im Abschnitt 4.2 wird eine Informationssystem-Architektur für ein Identitätsmanagementsystem entwickelt, die sich allgemein in Hochschulen und anderen Organisationen umsetzen lässt. Diese Informationssystem-Architektur soll in dieser Arbeit den Schwerpunkt der Architekturen des IDMS-Rahmenwerks bilden und wird deshalb ausführlich dargestellt.

Abschnitt 4.3 stellt in Ausschnitten Softwareprodukte vor, die für die Realisierung der Informationssystem-Architektur eingesetzt werden können. Dabei werden nur solche Produkte angegeben, die im Rahmen dieser Arbeit getestet wurden oder teilweise zur Realisierung der Informationssystem-Architektur an einer deutschen Hochschule produktiv eingesetzt werden.

Auf die IT-Infrastruktur-Architektur wird nicht weiter eingegangen. Auch diese ist wie die Geschäftsprozess-Architektur sehr stark abhängig von den jeweiligen Gegebenheiten an einer Hochschule.

4.1 Erfassung der Geschäftsprozess-Architektur

4.1.1 Vorgehensweise

Grundsätzlich werden mit der Geschäftsprozess-Architektur die aktuellen aber auch die zukünftigen Geschäftsprozesse an einer Hochschule beschrieben. Zunächst erfolgt eine Bestandsaufnahme. Die Prozesse werden dann im Hinblick auf ihre Effektivität und weitere Kriterien bewertet. Für nicht effektive Prozesse werden verbesserte Prozesse beschrieben. Hierbei kann es auch zu einer Zusammenlegung oder Trennung von einzelnen Vorgängen oder ganzen Geschäftsprozessen kommen.

Als Ausgangspunkt für die Analyse vorhandener Prozesse können die Personengruppen verwendet werden, die an der Hochschule in irgendeiner Form tätig sind. Hierzu gehören Mitarbeiter, Studierende, Alumni und Gäste. Diese Gruppen können auch noch weiter unterteilt werden.

Für jeden Prozess sollten dabei folgende Aspekte untersucht werden:

- Beteiligte Organisationseinheiten
  Bei jedem Prozess sind mehrere Organisationseinheiten der Hochschule beteiligt, die bestimmte Verantwortlichkeiten besitzen.

- Beteiligte Personen
  Innerhalb einer Organisationseinheit können mehrere Personen an dem Geschäftsprozess beteiligt sein. Die Personen besitzen hierbei Leitungs-, inhaltliche oder technische Kompetenzen.

- Beteiligte IT-Systeme
  In bestimmten Fällen sind IT-Systeme zur Unterstützung oder auch Realisierung bestimmter Geschäftsprozesse beteiligt. Ist dies nicht der Fall, so sollte die Frage gestellt werden, ob der Einsatz eines IT-Systems eine Effektivitätssteigerung darstellen kann.

- Sensibilität der Informationen
  Bei jedem Geschäftsprozess werden Informationen transportiert und generiert. Die Sensibilität dieser Informationen muss abgeschätzt werden, da sich daraus entsprechende Sicherheitsmaßnahmen ergeben.

- Authentifizierung
  Abhängig von der Sensibilität der Daten müssen an das Sicherheitsniveau angepasste Authentifizierungsverfahren angewandt werden. Auch bei diesem Aspekt kann man zu der Einschätzung gelangen, dass der aktuelle Prozess die Sicherheitsbedürfnisse nicht vollständig erfüllt.

- Autorisierung
  Für die am Geschäftsprozess beteiligten Personen oder Mitgliedsgruppen sollte untersucht werden, wer wann welche Berechtigung benötigt. Zu viele Berechtigungen stellen ein Sicherheitsrisiko, zu wenige eine Behinderung bei der Aufgabenerfüllung dar.

- Notwendigkeit zur Erfüllung der Aufgaben
  Hochschulmitglieder müssen an einer Hochschule gewisse Aufgaben wahrnehmen. Mit diesen Aufgaben sind bestimmte Geschäftsprozesse verbunden. Hierbei kann man unterscheiden zwischen notwendigen Prozessen, die für die Erfüllung der Aufgaben unverzichtbar sind, und Prozessen, die lediglich eine Erleichterung darstellen.

- Weitere Bewertungskriterien
  Die Einschätzung des Schutzbedarfs der Informationen zeigt ein erstes Bewertungskriterium auf. So können Geschäftsprozesse, bei denen hochsensible Informationen verwendet werden, vorrangig betrachtet werden. Weitere Bewertungskriterien sind die mit dem Prozess verbundenen Kosten, Zeitabhängigkeiten sowie die politische Bedeutung (z.B. Außenwirkung der Hochschule).

Die Geschäftsprozesse können mit Hilfe der Business Process Modelling Language (BPML), die im Zusammenhang mit Workflowmanagement bereits erwähnt wurde (siehe Abschnitt 2.13), beschrieben werden.

4.1.2 Nutzung der Ergebnisse

Die Erfassung und Bewertung der einzelnen Geschäftsprozesse kann die folgenden Konsequenzen haben:

- Neugestaltung des Prozesses
  Geschäftsprozesse, die beispielsweise bestimmten Zeitvorgaben nicht entsprechen oder zu teuer sind, müssen in der Regel neu gestaltet werden.

- Trennung in obligatorische und optionale Systeme
  Alle IT-Systeme, die für eine Personengruppe als notwendig zur Erfüllung der Aufgabe angesehen werden, gelten als obligatorisch. In den obligatorischen Systemen werden unter anderem personenbezogene Daten verarbeitet und gegebenenfalls an weitere Systeme übermittelt. Eine freiwillige Zustimmung der Betroffenen kann jedoch bei den obligatorischen Systemen nicht erfolgen. Für die benötigten Daten muss dann eine Datenschutzrichtlinie an der Hochschule erlassen werden.

  Beispiel:
  Ein Mitarbeiter, dessen Hauptaufgabe darin besteht, Prüfungsdaten in einer dafür vorgesehenen Datenbank zu erfassen, sollte nicht darüber entscheiden können, ob für ihn ein Benutzerkonto für die Authentifizierung angelegt werden darf oder nicht. Gestattet er dies nämlich nicht, so kann er auch seiner Tätigkeit nicht nachkommen.

  Bei optionalen Systemen muss hingegen ein Benutzer die Möglichkeit erhalten zu entscheiden, ob Daten zu seiner Person von dem System verarbeitet werden dürfen oder nicht. Gestattet er nicht die Verarbeitung bestimmter oder aller Daten, so kann

es dazu führen, dass er das entsprechende System nicht nutzen kann. Trotzdem kann er seinen Aufgaben an der Hochschule nachkommen, vielleicht nur etwas unbequem.

Beispiel:
Das Hochschulrechenzentrum bietet einen kostenpflichtigen Druckservice für Studierende an. Möchte ein Studierender diesen Service nutzen, so muss er bestimmte Angaben wie Name, E-Mail-Adresse und Bankverbindung auf einer Webseite eingeben. Möchte er dies nicht, so kann er sich auch persönlich an den Druckservice wenden und gegen Vorkasse und Vorlage des Studierendenausweises seine Ausdrucke anfertigen lassen.

- Festlegung der Sicherheitsanforderungen für IT-Systeme
  Werden bei einem Geschäftsprozess sensible Daten verwendet, so müssen hohe Sicherheitsanforderungen auch für die beteiligten IT-Systeme gelten. Die Sicherheitsanforderungen werden in Sicherheitsrichtlinien festgelegt.

- Festlegung der involvierten Personen für ein Identitätsmanagementsystem
  Die Geschäftsprozesse geben Auskünfte darüber, welche Personen in welchen Organisationseinheiten sich an dem Aufbau und Betrieb des Identitätsmanagementsystems beteiligen müssen. Außerdem können die Personengruppen erfasst werden, die das IDMS nutzen.

- Festlegung von Anforderungen an die Informationssystem-Architektur
  Die Analyse der Geschäftsprozesse bringt weitere Anforderungen an die Informationssystem-Architektur hervor. Hierzu gehört die Ermittlung von IT-Systemen, bei denen eine initiale Erfassung von Identitätsdaten erfolgt. Anhand der verwalteten Personengruppen und der Bewertung der Datenqualität in diesen Systemen werden die autoritativen Datenquellen festgelegt. Außerdem werden Informationen über die IT-Systeme gewonnen, die bestimmte Identitätsdaten benötigen, um eine weitere manuelle Erfassung und Pflege zu vermeiden.

Es lassen sich sicherlich noch weitere Konsequenzen aus der Analyse der Geschäftsprozess-Architektur ermitteln. Die wichtigsten sind jedoch genannt worden.

4.2 Informationssystem-Architektur

4.2.1 Architektur zur System-, Daten- und Prozessintegration

Die Architektur zur System-, Daten- und Prozessintegration wird für das bessere Verständnis anhand eines Beispiels entwickelt.

Ausgangssituation

Abbildung 11 zeigt exemplarisch fünf IT-Systeme einer Hochschule, in denen Identitätsdaten gespeichert werden.

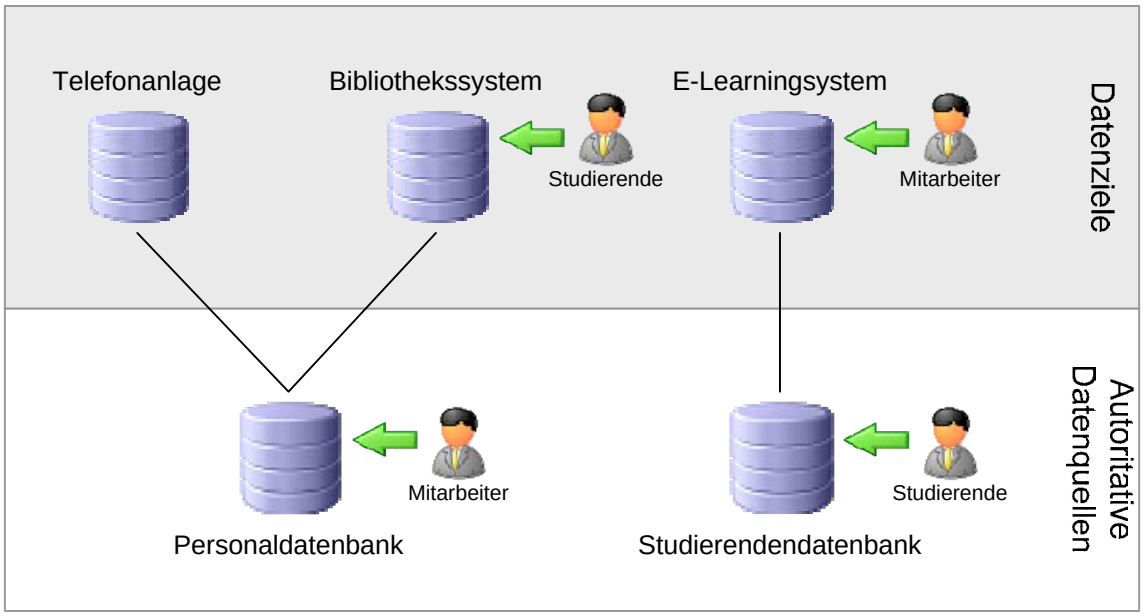

Abbildung 11: Architekturbeispiel - Ausgangssituation

Mitarbeiterdaten gelangen durch manuelle Eingabe in die Personaldatenbank der Hochschule. Die Personaldatenbank dient als autoritative Datenquelle für die Telefonanlage und das Bibliothekssystem. Durch eine Übertragung der Mitarbeiteridentitäten an die Telfonanlage und das Bibliothekssystem wird eine erneute manuelle Eingabe der gleichen Daten vermieden. Eine weitere autoritative Datenquelle stellt die Studierendendatenbank dar. Aus der Studierendendatenbank werden Identitätsdaten an das E-Learningsystem übermittelt, um die erneute manuelle Eingabe von Studierendendaten einzusparen.

Die Telefonanlage, das Bibliothekssystem sowie das E-Learningsystem stellen Ziele für Identitätsdaten dar, die in anderen Systemen erstmalig manuell erfasst wurden. Man spricht in diesen Fällen von Datenzielen:

Datenziele: Die Systeme, die die Identitätsdaten aus einer autoritativen Datenquelle erhalten, werden als Datenziele (engl. data targets) bezeichnet. Die Begriffe Datensenken und Identitätsdatenkonsumenten können dabei als Synonym verwendet werden.

Die einfachen Linien in Abbildung 11 symbolisieren die Übermittlung der Identitätsdaten mit Hilfe von proprietären Protokollen zwischen den genannten Systemen.

Bei dem Bibliotheksystem und dem E-Learningsystem erfolgt trotz der Übermittlung von Identitätsdaten noch eine manuelle Erfassung und Pflege von Personengruppen, die nicht in der Datenlieferung enthalten sind. Bei dem Bibliothekssystem sind dies die Studierenden und beim E-Learningsystem die Mitarbeiter. Eine Kopplung der Datenziele mit den autoritativen Datenquellen, in denen die benötigten Identitätsdaten vorliegen, besteht jedoch nicht.

Schritt 1: Vermeidung der mehrfachen manuellen Erfassung / Einführung von Service Providern

Ein wichtiges Ziel der Architektur ist ja die Reduzierung oder sogar die vollständige Elimination der mehrfachen manuellen Erfassung von Identitäten. Es wird daher festgelegt, dass innerhalb einer Hochschule jede Art von Identität nur an jeweils einem IT-System manuell erfasst wird. Alle anderen IT-Systeme erhalten die Identitätsdaten aus den so festgelegten autoritativen Datenquellen.

Die autoritative Datenquellen und die Identitätsdatenkonsumenten müssen zum Zweck des Identitätsdatenaustauschs über einen Mechanismus verbunden werden. Diese Integrationsaufgabe soll mit einer serviceorientierten Architektur gelöst werden. Jedes IT-System, das die Rolle einer autoritativen Datenquelle oder einer Datensenke übernimmt, stellt mit Hilfe einer neuen Komponente einen einheitlichen Service zur Verfügung. Wie in Abschnitt 2.10 („Serviceorientierte Architekturen“) bereits dargestellt wurde, erfolgt der Zugriff auf einen Service über ein Service Interface. Die Komponente, die den Service bereitstellt, heißt Service Provider.

Im Falle einer autoritativen Datenquelle stellt der Service Provider einen Service für die Abfrage von Identitäten bereit. Der Service wird deshalb auch als Identity Service bezeichnet.

Identity Service Provider: Ein Identity Service ist ein Mechanismus für die Abfrage von einer oder mehrerer Arten von Identitäten aus einer Datenquelle. Die Identitätsdaten können gelesen, nicht aber verändert werden. Die Komponente, die den Identity Service zur Verfügung stellt, wird Identity Service Provider (ISP) genannt. Der Zugriff auf einen Identity Service Provider erfolgt über das Identity Service Interface.

Im Gegensatz zu autoritativen Datenquellen wird für ein Datenziel ein Service über einen Service Provider realisiert, der das Einfügen, Ändern und Löschen von Identitäten ermöglicht. Einfügen, Ändern und Löschen sind die elementaren Operationen für das Provisioning und deshalb wird der Service als Provisioning Service bezeichnet.

Provisioning Service Provider: Ein Provisioning Service ermöglicht das Einfügen, Ändern und Löschen einer oder mehrerer Arten von Identitäten in ein Datenziel. Die Komponente, die den Provisioning Service ermöglicht, wird Provisioning Service Provider (PSP) genannt. Die Schnittstelle für den Provisioning Service Provider wird Provisioning Service Interface genannt.

Abbildung 12 zeigt für das begleitende Beispiel die Erweiterungen um die Service Provider mit deren Schnittstellen.

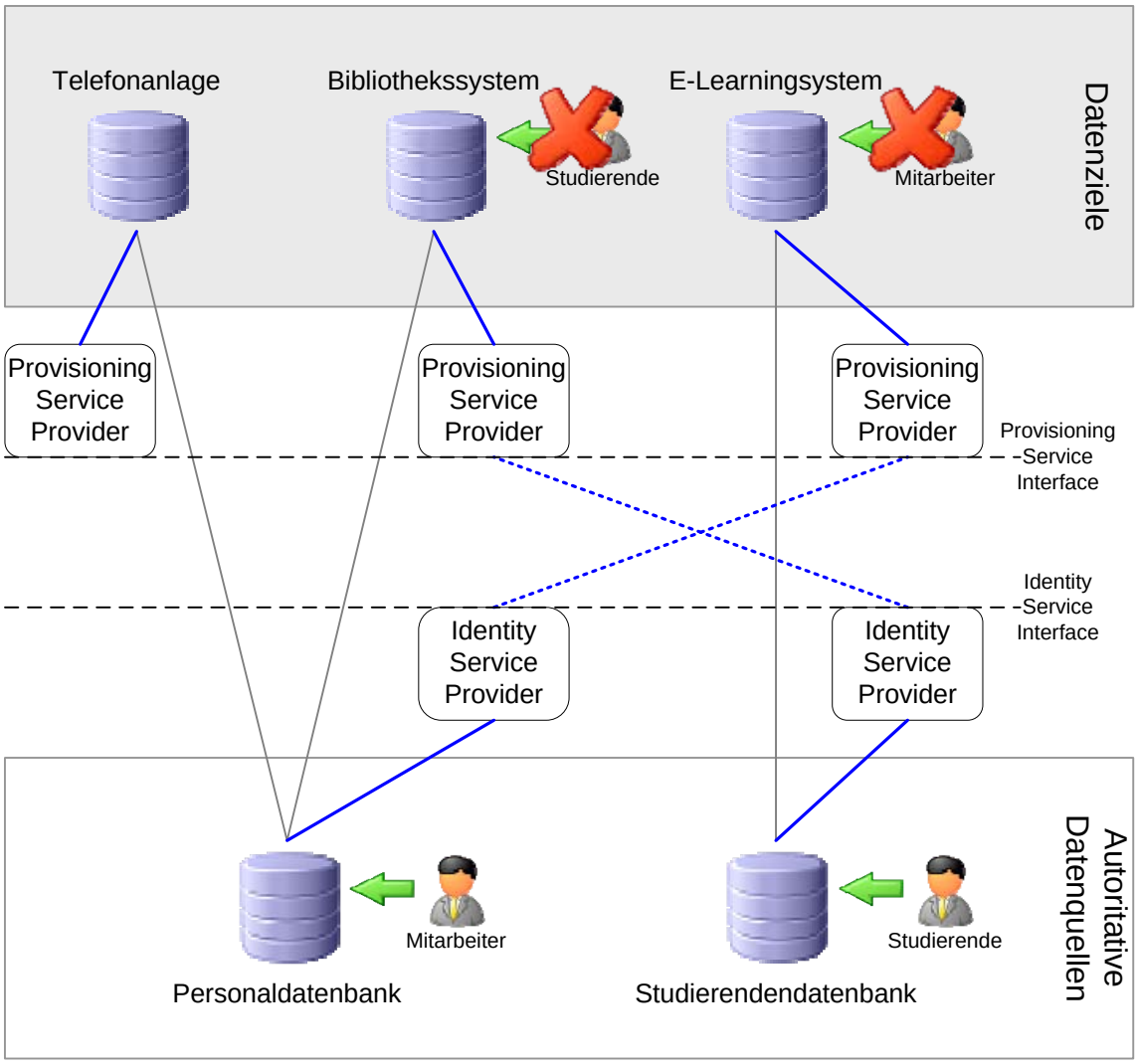

Abbildung 12: Architekturbeispiel - Erweiterung um Service Provider

Die Abfrage von Identitätsdaten aus der Personaldatenbank und aus der Studierendendatenbank wird zusätzlich über einen Identity Service Provider ermöglicht. Die Datenziele Telefonanlage, Bibliothekssystem und E-Learningsystem können nun auch über den Provisioning Service Provider mit Identitäten versorgt werden. Die Anbindung der Datenquellen und Datenziele an die jeweiligen Service Provider werden mit den fetten durchgehenden Linien in Abbildung 12 wiedergeben.

Die manuelle Erfassung von Studierenden für das Bibliothekssystem und die manuelle Erfassung von Mitarbeitern für das E-Learningsystem entfallen im nächsten Schritt. Stattdessen erfolgt der Datenaustausch über die neuen Service Provider (siehe gestrichelte Linien in Abbildung 12).

Ein wesentlicher Vorteil der beschriebenen Vorgehensweise besteht darin, dass die alten etablierten Verfahren zum Austausch der Mitarbeiter- und Studierendendaten zunächst bestehen bleiben können. Lediglich die Identitätsdaten, die mehrfach manuell erfasst wurden, gelangen nun aus den autoritativen Datenquellen über die serviceorientierte Architektur zu den Datenzielen.

Eine Interaktion innerhalb einer serviceorientierten Architektur erfolgt zwischen einem Service Provider und einem Service Consumer. Damit also die Identitäten von den Datenquellen zu den Datenzielen gelangen können, muss entweder der Identity Service Provider oder der Provisioning Service Provider die Rolle eines Service Consumers übernehmen. Die Rollenverteilung hängt vom Modus, der verwendet wird. Wie in Abschnitt 2.5.4 erwähnt wurde, kommen hierfür der Pull- oder der Push-Modus in Frage. Es folgt eine Definition der beiden Modi bezogen auf die hier verwendete serviceorientierte Architektur:

Pull-Modus: Im Pull-Modus werden vom Provisioning Service Provider die Identitätsdaten beim Identity Service Provider abgerufen. Der Provisioning Service Provider übernimmt die Rolle eines Service Consumers, da er durch die Abfrage von Identitäten den Identity Service in Anspruch nimmt. Aktionen im Datenziel initiieren dabei die Übertragung von Identitäten.

Push-Modus: Im Push-Modus werden vom Identity Service Provider neue oder geänderte Identitäten an einen Provisioning Service Provider übermittelt. Der Identity Service Provider, der die Rolle des Service Consumers übernimmt, verwendet den Provisioning Service, um die Identitätsdaten in dem Datenziel einzutragen.

Sicherheitsaspekte, beispielsweise in Form von Firewallregeln, und Datenschutzaspekte können bei der Wahl zwischen Pull- oder Push-Modus eine entscheidende Rolle spielen.

Die bisher eingeführte serviceorientierte Architektur ermöglicht zwei Formen der Integration. Mit Hilfe der Service Provider werden IT-Systeme miteinander verbunden. Es liegt also eine Systemintegration vor. Des Weiteren werden Daten integriert. Bei der Datenintegration lassen sich wiederum zwei Fälle unterscheiden. Werden die Identitäten aus einer Datenquelle in einem Datenziel gespeichert, so liegt der Fall einer materialisierten Datenintegration vor. Fragt hingegen ein Service Consumer, wie beispielsweise eine Webanwendung, Identitätsdaten bei mehreren Identity Services ab, um diese in einer aggregierten Form anzuzeigen, so handelt es sich um eine virtuelle Integration von Daten.

Schritt 2: Einführung eines zentralen Datenspeichers für Identitäten

Das Datenziel erhält nun alle Daten aus autoritativen Datenquellen. Deshalb benötigt es die Information, ob es sich bei zwei vorliegenden Identitäten um dieselbe natürliche Person handelt. Bei der rein manuellen Erfassung wurde diese Aufgabe mehr oder weniger gut von einzelnen Menschen erfüllt. Beim direkten Datenaustausch zwischen den Service Providern benötigt jedoch jedes Datenziel ein Verfahren, um eine integrierte Sicht auf die Identitäten der Hochschule zu realisieren. Es wäre vorteilhaft, wenn nur eine Komponente dieses aufwendige Verfahren durchführt. Mit der Einführung eines zentralen Datenspeichers an einer Hochschule, der Main Identity Store genannt wird, übernimmt eine separate Komponente diese Verantwortung.

Abbildung 13 zeigt für das begleitende Beispiel, wie der Main Identity Store in die gesamte Architektur eingefügt wird.

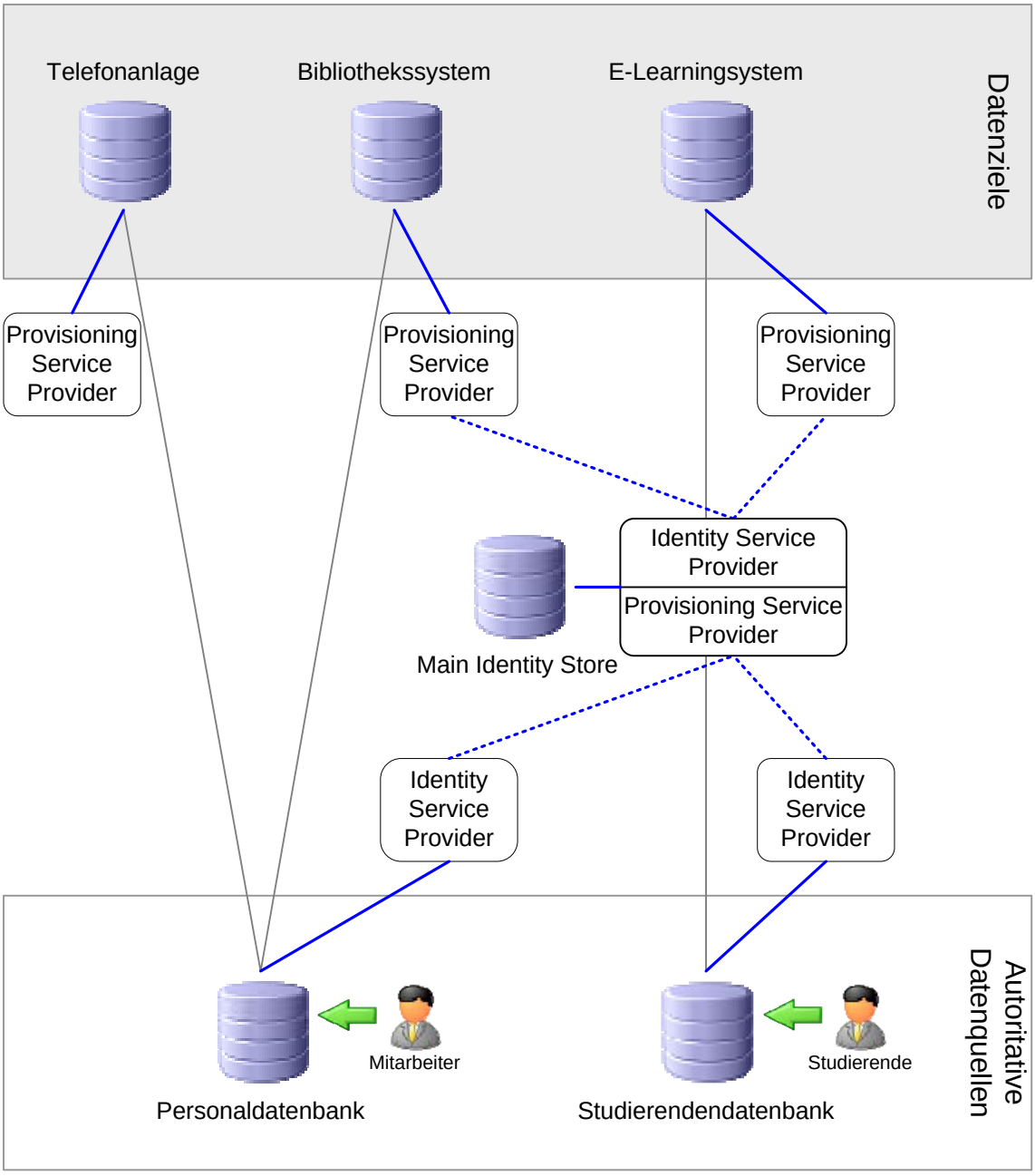

Abbildung 13: Architekturbeispiel – Main Identity Store

Die Identitäten aus der Personal- und der Studierendendatenbank werden in dem Main Identity Store gespeichert. Ist ein Mitarbeiter gleichzeitig Studierender der Hochschule, so werden dort die beiden Identitäten miteinander verknüpft. Der Main Identity Store stellt somit ein Datenziel dar. Für das Bibliothekssystem und das E-Learningsystem übernimmt der Main Identity Store hingegen die Rolle einer Datenquelle, in der die Identitätsdaten in aufbereiteter Form vorliegen. Der Main Identity Store muss demnach sowohl einen Identity Service, als auch einen Provisioning Service bereitstellen. Die Services können mit getrennten Identity und Provisioning Service Providern realisiert werden. In Abbildung 13 werden diese beiden Komponenten zu einer Komponente zusammengefasst.

Werden Identitäten in den Main Identity Store eingefügt, gelöscht oder geändert, so ist eine (sofortige) prozessgesteuerte Erzeugung und Übermittlung von abgeleiteten Identitäten zu

den Datenzielen möglich. Die Architektur leistet dann auch eine Integration von Prozessen. Wie später noch gezeigt wird, können alle in diesem Abschnitt vorgestellten Service Provider solche Prozesse realisieren.

Schritt 3: Aufhebung alter proprietärer Verbindungen / Vollständige Umsetzung der serviceorientierten Architektur

Nach der Etablierung von Identity und Provisioning Services und dem Aufbau des Main Identity Stores können alte proprietäre Verbindungen, die für den Austausch von Identitätsdaten verwendet wurden, durch die neuen Services ersetzt werden. Dieser Übergang kann schrittweise erfolgen. Abbildung 14 zeigt für das begleitende Beispiel die vollständig realisierte serviceorientierte Architektur zum Austausch von Identitäten.

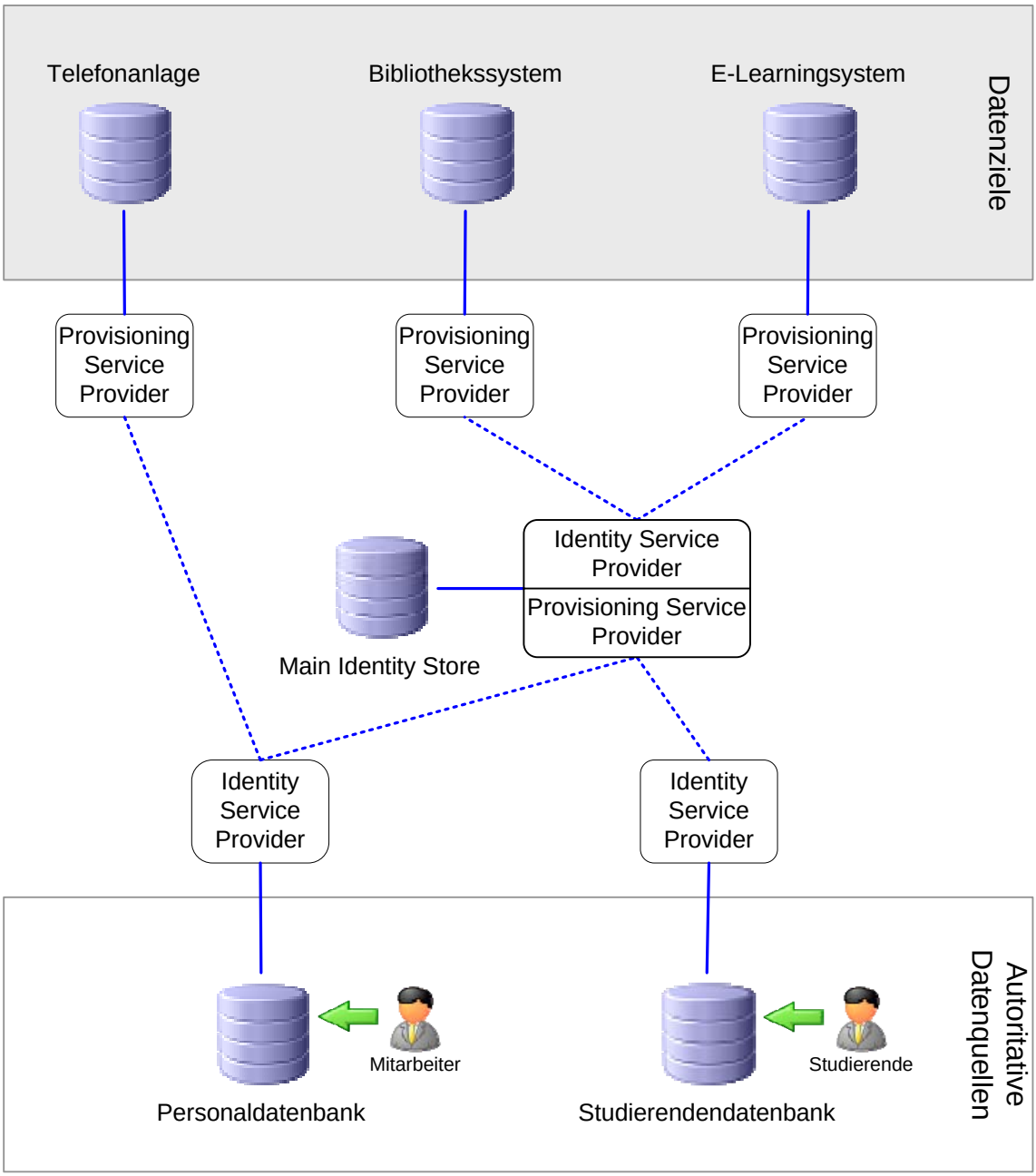

Abbildung 14: Architekturbeispiel - Vollständig serviceorientierte Architektur

Ein Main Identity Store nimmt nicht alle Arten von Identitäten auf. Nur die Identitätsdaten, für die eine mehrfache Verwendung vorliegt, werden gespeichert. Bei speziellen Identitäten, für die nur ein einziger Datenkonsument Bedarf zeigt, erfolgt eine direkte Übermittlung. Aus der Fortführung des begleitenden Beispiels wird dieser Sachverhalt deutlich:

Das Bibliothekssystem und das E-Learningsystem erhalten nun sämtliche Mitarbeiter- und Studierendendaten von dem Main Identity Store. Die neue zentrale Datenbank, die eine integrierte Sicht auf die Identitäten schafft, erhält die benötigten Daten wiederum von der Personaldatenbank und der Studierendendatenbank. Eine Besonderheit stellt die Verbindung zwischen der Personaldatenbank und der Telefonanlage dar. Die Telefonanlage benötigt zu den vollständigen Namen noch eine Kostenstelle. Die Kostenstellen weisen eine spezielle Struktur auf. Da die Kostenstelleninformationen nur noch von der Telefonanlage benötigt werden und kein weiteres IT-System einen Bedarf für diese Identitätsdaten besitzt, wird auf die Speicherung der Kostenstellenattribute in dem Main Identity Store verzichtet. Stattdessen erfolgt eine direkte Übermittlung der speziellen Identitäten von der Personaldatenbank zu der Telefonanlage.

Die vorgestellte Architektur zur System-, Daten- und Prozessintegration stellt demnach einen Kompromiss zwischen rein zentralisierter und vollständig föderierter Architektur dar. Zudem wird eine schrittweise Integration neuer Komponenten in eine bestehende Struktur und somit ein hohes Maß an Flexibilität ermöglicht.

Es sei an dieser Stelle angemerkt, dass die Einführung des Main Identity Stores (Schritt 2) und die Aufhebung der proprietären Verbindungen (Schritt 3) auch ohne weiteres in der Reihenfolge hätten getauscht werden können.

Nachdem die Grundkonzepte der Architektur zur System-, Daten- und Prozessintegration vorgestellt wurden, findet in den nachfolgenden Kapiteln eine Beschreibung der Verantwortlichkeiten für den Main Identity Store sowie für den Identity Service Provider und den Provisioning Service Provider statt. Für jede dieser drei Komponenten wird ein Entwurf mit angegeben.

4.2.2 Main Identity Store

Verantwortlichkeiten

In dem Main Identity Store erfolgt die Speicherung von Identitäten, die mehrfach an der gesamten Hochschule benötigt werden.

Entwurf

Der in diesem Abschnitt ausgearbeitete Entwurf für einen Main Identity Store wird den grundlegenden Anforderungen einer Hochschule gerecht. Etwaige Besonderheiten an den Hochschulen führen dann in der Regel zu einer Erweiterung des Main Identity Stores.

Abbildung 15 zeigt den ersten Ausschnitt aus dem Entity-Relationship-Diagramm (ERD), das zur Visualisierung des Entwurfs verwendet wird.

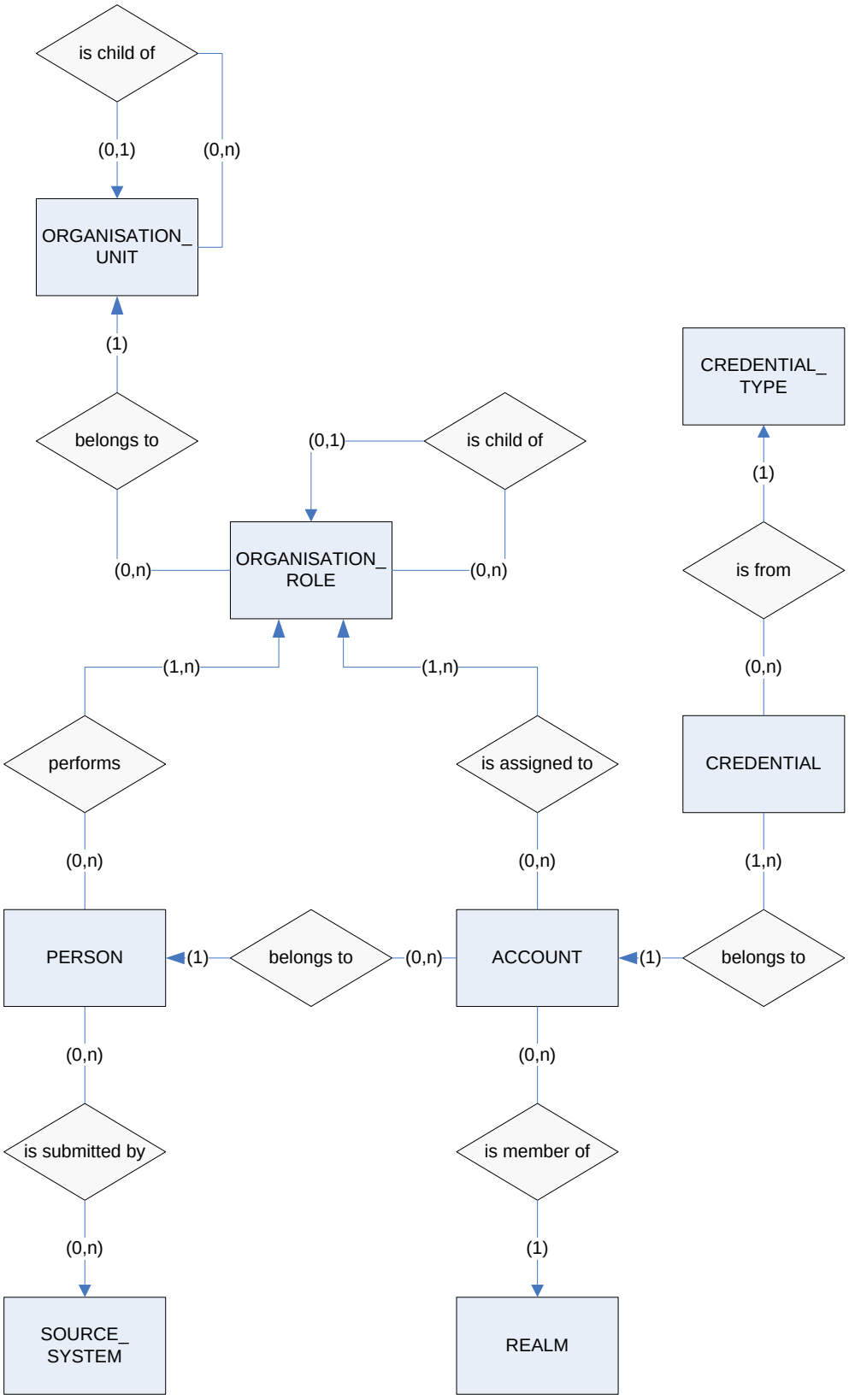

Abbildung 15: Personen, Benutzerkonten, Organisationsrollen und -einheiten

Im Mittelpunkt des Main Identity Store stehen die Identitäten zu den natürlichen Personen. Die Personenidentitäten (PERSON) setzen sich ausschließlich aus Attributen zusammen,

die unabhängig von der Organisation inhärent zu der Person sind und diese eindeutig charakterisieren. Die Daten zu den Personen werden von verschiedenen autoritativen Datenquellen (SOURCE_SYSTEM) an den Main Identity Store übermittelt. In der Relation zwischen den Datenquellen und den Personen erfolgt die Speicherung der Identifier bzw. Schlüssel, die in den jeweiligen autoritativen Datenquellen für die Personen verwendet werden. Diese Information kann genutzt werden, um zu einem späteren Zeitpunkt weitere Daten zu einer Person abzurufen oder um nach Aktualisierungen zu fragen.

Eine Hochschule setzt sich wie andere Organisationen aus mehreren Organisationseinheiten (ORGANISATION_UNIT) zusammen. Dabei bilden die Organisationseinheiten eine limitierte hierarchische Struktur. Limitiert bedeutet, dass jede Organisationseinheit nur maximal einer direkt übergeordneten Organisationseinheit zugeordnet werden kann.

Die Personen der Hochschule können den Organisationseinheiten der Hochschule zugeordnet werden. Diese Zuordnung ist immer mit Aufgaben oder Verantwortlichkeiten verbunden, die die jeweilige Person innerhalb der Organisationseinheit übernimmt. Zu jeder Organisationseinheit existieren deshalb Organisationsrollen (ORGANISATION_ROLE). Dabei gibt die Rolle die Aufgabe und die damit verbundenen Rechte wieder. Ebenso wie die Organisationseinheiten können die Organisationsrollen limitiert hierarchisch angeordnet sein. Generelle Hierarchien von Rollen, wie sie in Abschnitt 2.8.3 „Role-Based Access Control (RBAC)“ beschrieben werden, kommen aufgrund der damit verbundenen Komplexitätssteigerung nicht zum Einsatz. Das folgende Beispiel gibt mögliche Organisationsrollen für eine Organisationseinheit an.

Beispiel:

- *Dekan der Fakultät Mathematik und Informatik*
- *Studierender des Fachbereichs Informatik*
- *Hochschullehrer des Lehrgebiets Informationssysteme und Datenbanken*
- *Leiter des Hochschulrechzentrums*
- *Mitarbeiter des Hochschulrechenzentrums*
- *Sachbearbeiter für Personalangelegenheiten der Drittmittelverwaltung*

Die Fakultät für Mathematik und Informatik, der Fachbereich Informatik und das Lehrgebiet für Informationssysteme und Datenbanken geben eine mögliche hierarchische Abfolge von Organisationseinheiten innerhalb der Hochschule an. Die Organisationsrollen Mitarbeiter und Leiter des Hochschulrechenzentrums bilden eine Hierarchie von Rollen innerhalb einer Organisationseinheit. Der Leiter des Hochschulrechenzentrums ist auch Mitarbeiter des Hochschulrechenzentrums.

Jede Person muss mindestens einer Organisationsrolle zugeordnet werden. Das bedeutet, dass man zu jeder Person immer eine Aussage erhält, welche Aufgabe oder Funktion sie an der Hochschule besitzt. Die notwendigen Informationen, die man für eine Zuordnung zur Organisationsrolle benötigt, liefern hauptsächlich die autoritativen Datenquellen. Das bedeutet, dass dieser Prozess weitestgehend automatisiert werden kann.

Für die Authentifizierung an den diversen IT-Systemen der Hochschule benötigt eine Person ein oder mehrere Benutzerkonten (ACCOUNT). Es existieren nur persönliche Benutzerkonten. Das bedeutet, dass jedes Benutzerkonto immer nur einer Person zugeordnet ist. Nicht jede Person muss jedoch zwangsläufig ein Benutzerkonto besitzen. Die Benutzerkonten sind innerhalb eines bestimmten Bereiches (REALM) eindeutig. Die Unterstützung mehrerer Realms für Benutzerkonten vereinfacht unter anderem die Migration von Benut-

zerkonten aus anderen IT-Systemen. Zu jedem Benutzerkonto gehört mindestens ein Berechtigungsnachweis (CREDENTIAL). Mit CREDENTIAL_TYPE wird angegeben, ob es sich beispielsweise um ein verschlüsseltes Passwort, einen bestimmten Hash für ein Passwort oder ein biometrisches Datum handelt.

Auch Benutzerkonten müssen mindestens einer Organisationsrolle zugeordnet werden. Diese Information wird insbesondere für die zu versorgenden Zielsysteme genutzt. Die Person entscheidet dabei durch die Verwendung unterschiedlicher Benutzernamen darüber, in welcher Funktion oder mit welchen Rechten sie in dem entsprechenden System agieren möchte. Das folgende Beispiel soll den Sachverhalt verdeutlichen.

Beispiel:
Frau Wilhelmine Dickens ist Studierende und Mitarbeiterin am Fachbereich Geisteswissenschaften. Sie besitzt die Benutzerkonten wilhelmine@hochschule.de und wdickens@hochschule.de. Das erste Benutzerkonto ist der Organisationsrolle „Studierende am Fachbereich Geisteswissenschaften“ zugeordnet. Das zweite Benutzerkonto erhält eine Verknüpfung zur Rolle „Mitarbeiter am Fachbereich Geisteswissenschaften“. Die Benutzerkonten- und Rolleninformationen werden dem E-Learningsystem an der Hochschule übermittelt. Dabei erfolgt eine Umsetzung der Organisationsrolle in systemspezifische Rollen oder direkt in Rechte. Meldet sich Frau Dickens mit wilhelmine@hochschule.de an, so kann sie für Studierende typische Funktionen nutzen, wie Online-Übungen oder Abruf von Vorlesungsfolien. Mit dem Benutzerkonto wdickens@hochschule.de hingegen ist es ihr möglich, jegliche Art von Inhalten zu Vorlesungen des Fachbereichs Geisteswissenschaften abzulegen.

Nach dem ERD im Abbildung 15 ist es möglich, dass eine Person über ihr Benutzerkonto anderen Organisationsrollen zugeordnet ist, als sie direkt als Person innehat. Dies kann für Übergangsregelungen oder Vertreterfunktionen sinnvoll sein. Soll dies in jedem Fall verhindert werden, so müssen entsprechende Bedingungen (engl. constraints) innerhalb des Main Identity Store dafür sorgen, dass Benutzerkonten nur zu solchen Organisationsrollen zugewiesen werden können, denen der Benutzerkontoinhaber auch zugeordnet ist.

Die Rollenzuordnung zu den Benutzerkonten erfolgt über einen weitestgehend automatisierten Prozess. Ausgehend von den Zuordnungen der Personen zu den Organisationsrollen werden regelbasiert die Zuordnungen zwischen den Benutzerkonten und den Organisationsrollen erzeugt.

In dem vorangegangenen Beispiel wurde beschrieben, dass ein Zielsystem mit Benutzerkonten und Organisationsrollen aus dem Main Identity Store versorgt wird. Dabei erfolgte eine Umsetzung der Organisationsrollen zu systemspezifischen Rollen oder direkt in Rechte erst in dem Zielsystem. Aus den organisatorischen Rollen können jedoch auch systemspezifische Rollen abgeleitet werden, die direkt im Main Identity Store gespeichert werden.

Abbildung 16 zeigt den Ausschnitt aus dem Entity-Relationship-Diagramm, der die Beziehungen zu den systemspezifischen Rollen (TARGET_SYSTEM_ROLE) wiedergibt.

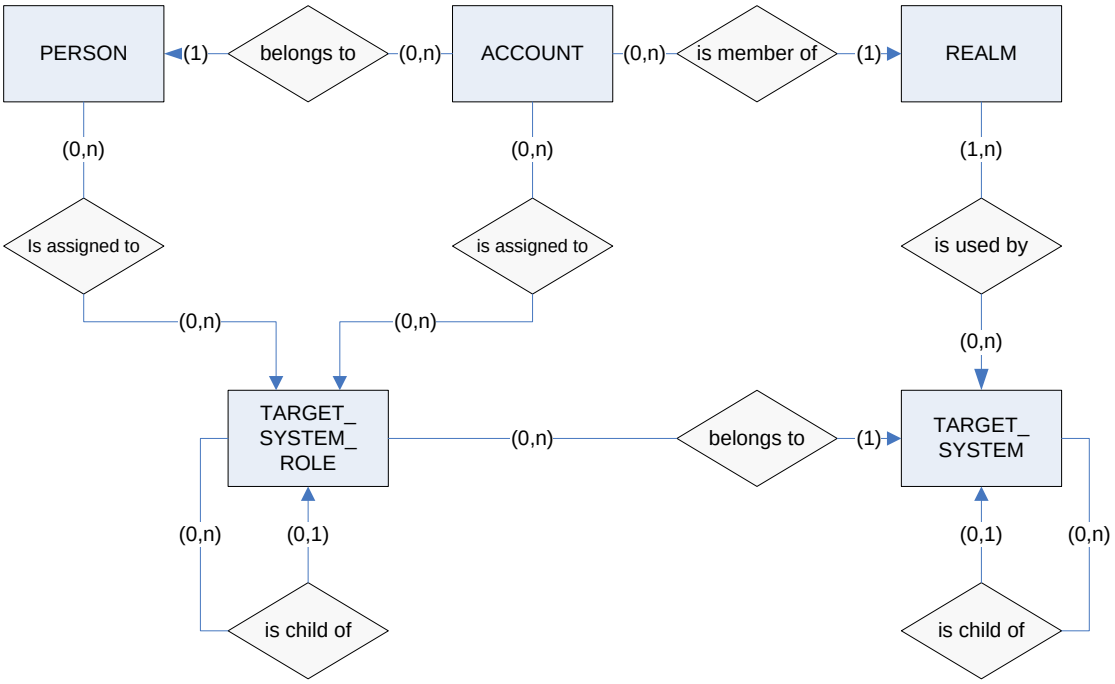

Abbildung 16: Systemspezifische Rollen

Wie die Organisationsrollen können auch die systemspezifischen Rollen in einer limitierten Hierarchie angeordnet sein. Die systemspezifischen Rollen sind dabei immer einem Zielsystem (TARGET_SYSTEM) zugeordnet. Die Zielsysteme, bei denen es sich auch um abstrakte Systeme handeln kann, können wieder hierarchisch angeordnet sein. Diesen Sachverhalt verdeutlicht das folgende Beispiel.

Beispiel:
An der Hochschule wird ein Online-Service der Bibliotheken (abstraktes System) angeboten. Dieser Service beinhaltet Services der Fachbereichsbibliotheken (abstrakte Systeme). Die Services der Fachbereichsbibliotheken setzen sich aus verschieden konkreten Systemen zusammen.

Sowohl Personen als auch Benutzerkonten können den systemspezifischen Rollen zugeordnet sein. Die Zuordnung der systemspezifischen Rollen folgt einem Regelwerk, das ausgehend von den Organisationsrollen abgearbeitet wird.

Die Verwendung der Organisationsrollen und der systemspezifischen Rollen entspricht dem Konzept der Role Based Access Control (RBAC), das in Abschnitt 2.8.3 beschrieben wurde. Personen und Benutzerkonten werden zunächst Rollen zugeordnet. Die Rollen wiederum fassen einzelne Rechte zusammen. Das RBAC-Konzept enthält auch Bedingungen für Rollen. Eine einfache Umsetzung ist für die folgenden zwei Bedingungen möglich.

- Static Separation of Duty (SoD) Constraints
  Zwei Rollen, die einer Person oder einem Benutzerkonto zugewiesen werden sollen, schließen sich gegenseitig aus. Es ist also nur die Zuordnung zu einer der beiden Rollen möglich.

- Temporal Constraints
  Die Zuordnung einer Person oder eines Benutzerkontos zu einer Rolle ist nur für einen bestimmten Zeitraum gültig.

Die in Abbildung 16 dargestellte Beziehung zwischen REALM und TARGET_SYSTEM ermöglicht die Realisierung einer weiteren Bedingung. Zu einem Realm gehören Zielsysteme, die diese Benutzerkonten verwenden. Mit einer zusätzlichen Regel wird dafür gesorgt, dass Benutzerkonten nur zu solchen systemspezifischen Rollen zugeordnet werden können, deren Zielsysteme zu demselben Realm gehören wie das Benutzerkonto. Somit umfasst der Realm letztlich Benutzerkonten, Rollen und Systeme.

Zur Unterstützung der Verwaltung von Personen und Benutzerkonten werden Gruppen eingeführt. Dabei stellen die Gruppen, wie in Abschnitt 2.8.3 „Role-Based Access Control (RBAC)“ definiert, ausschließlich eine Strukturierungshilfe dar. Personenidentitäten und Benutzerkonten können entweder in jeweils separaten Gruppen zusammengefasst werden (siehe Abbildung 17) oder es existiert nur eine Art von Gruppen, in denen gemischt Personen und Benutzerkonten zusammengefasst sind. Die Gruppen können eine limitierte Hierarchie bilden.

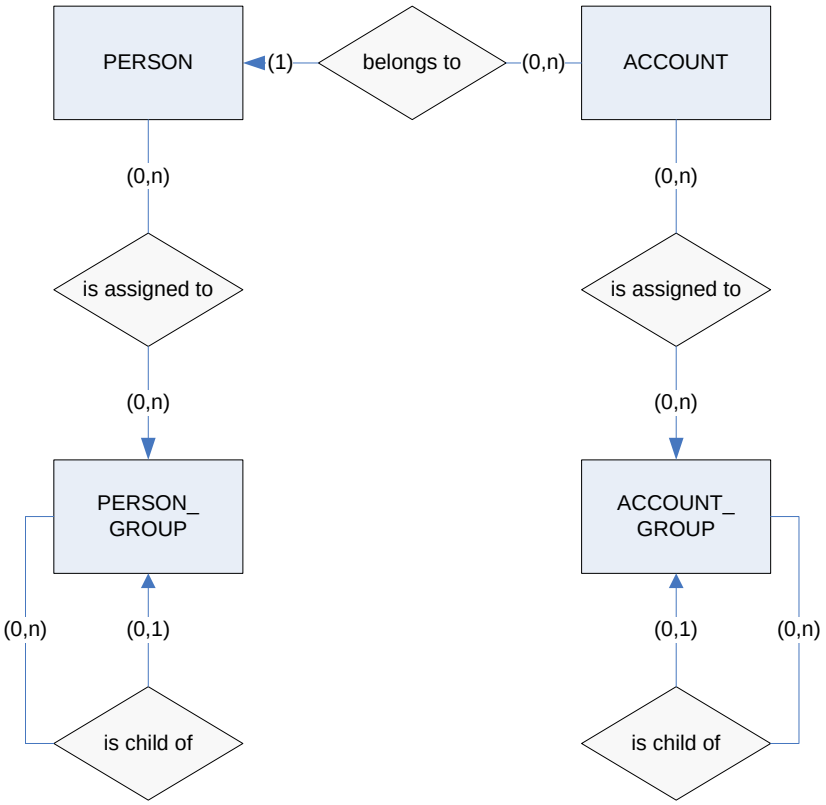

Abbildung 17: Getrennte Gruppen für Personen und Benutzerkonten

Für die Gruppen existieren verschiedene Verwendungszwecke. In Tabelle 2 werden einige Beispiele dafür angegeben.

Verwendungszweck	*Beschreibung*
Exportgruppen	*Die Personen und Benutzerkonten, die zu einem Zielsystem übertragen werden sollen, können in den Exportgruppen zusammengefasst werden. Dabei kann beispielsweise der Betreiber eines Zielsystems eine manuelle Zuordnung vornehmen.*
Ausschlussgruppen für die Rollenzuordnung	*Die Zuordnung der Personen und Benutzerkonten zu den Rollen erfolgt regelbasiert. Möchte man, dass bestimmte Personen oder Benutzerkonten von einer Regel ausgenommen werden, so kann man diese in Gruppen zusammenfassen und die Gruppe mit in die Regel aufnehmen.*
Verteilergruppen	*Soll ein ausgewählter Kreis von Personen regelmäßig bestimmte Informationen erhalten, so können diese in Verteilergruppen aufgenommen werden.*

Tabelle 2: Verwendungszwecke für Gruppen

Der Main Identity Store muss noch eine zusätzliche Rechteverwaltung erhalten, mit der geregelt wird, welche Operationen ein Subjekt ausführen darf. Die Erzeugung von Rollen und Gruppen sowie deren Zuweisung sind Beispiele für solche Operationen auf den jeweiligen Objekten im Main Identity Store. Die Rechteverwaltung stellt letztlich einen einfachen Aufsatz für den Main Identity Store dar.

Die bisher dargestellten Entity-Relationship-Diagramme enthalten keine Attribute zu den Entitäten. Im Rahmen dieser Arbeit kann eine detaillierte Betrachtung der Attribute für die einzelnen Entitäten nicht durchgeführt werden. Lediglich für die Entität PERSON erfolgt eine Auflistung der möglichen Attribute. Diese werden unter anderem für die Diskussion zur Duplikaterkennung im späteren Verlauf der Arbeit verwendet.

Die Tabelle 3 zeigt die Attribute, die die Entität PERSON enthält.

Name des Attributs	**Beschreibung**
Id	Eindeutiger Identifier für jede Person.
givenName	Vorname(n) zu einer Person gemäß den Angaben im Personalausweis, Reisepass oder der Geburtsurkunde. Ein eventuell angegebener Rufname findet keine besondere Berücksichtigung. Beispiele: *Albert, Karl-Heinz, Hugo Egon Paul*
surName	Familienname(n) zu einer Person gemäß den Angaben im Personalausweis, Reisepass oder der Geburtsurkunde. Beispiele: *Einstein, Ipendahl-Modry, Martinez Herrera*
nobilityTitle	Angabe des Adelsprädikats. Bei dem Adelsprädikat kann es sich um einen Adelstitel, einen Namenszusatz oder einer Aneinanderreihung von beidem handeln. Beispiele: *von, de, Gräfin, Sir, Freiherrin von, Lord of*
academicTitle	Angabe der akademischen Grade zu einer Person Beispiele: *Prof., M.A., Dr. rer. nat., B.A.*
auxTitle	Angabe von Titeln, die nicht zu den Kategorien Adelsprädikate und akademische Grade gehören.

	Beispiele: Reg.-R. (Regierungsrat), Staats-R. (Staatsrat)
displayName	Der Anzeigename enthält den Namen zu einer Person, der beispielsweise in Webanwendungen oder als Absender von E-Mails angezeigt werden soll. Mögliche Gründe: - *Die Person möchte, dass von den Vornamen nur ein ausgewählter (z.B. der Rufname) erscheint.* - *Die Person möchte nicht, dass aus dem Namen das Geschlecht ersichtlich ist.* - *Die Person ist eher unter ihrem Künstlernamen bekannt* - *In Publikationen wurde ein etwas anders geschriebener Name verwendet, als er im Personalausweis steht.*
Gender	Angabe des Geschlechts. Die Angabe erfolgt in der Regel nach ISO 5218 (0=not known, 1=male, 2=female, 9=not applicable)
birthDate	Geburtsdatum der Person.
birthName	Geburtsname der Person.
birthPlace	Geburtsort der Person.
birthCountry	Geburtsland der Person. Die Angabe der Länder erfolgt in der Regel mit der Kodierung nach ISO 3166.
citizenship	Staatsbürgerschaft bzw. Nationalität zu einer Person. Die Angabe der Staatsbürgerschaft erfolgt in der Regel mit der Kodierung nach ISO 3166. In Ausnahmefällen kann eine Person eine doppelte Staatsbürgerschaft besitzen. Soll die zweite Staatsbürgerschaft im Main Identity Store mit gespeichert werden, so wird PERSON um das Attribut secondaryCitizenship erweitert.

Tabelle 3: Attribute zu Personen

Neben den Attributen aus Tabelle 3 können E-Mail-Adressen, Telefonnummern und Postanschriften zu einer Person gespeichert werden. Die drei Arten von Kontaktinformationen können mehrfach zu einer Person existieren und stellen somit eigene Entitäten dar. Abbildung 18 stellt diesen Sachverhalt im Entity-Relationship-Diagramm dar.

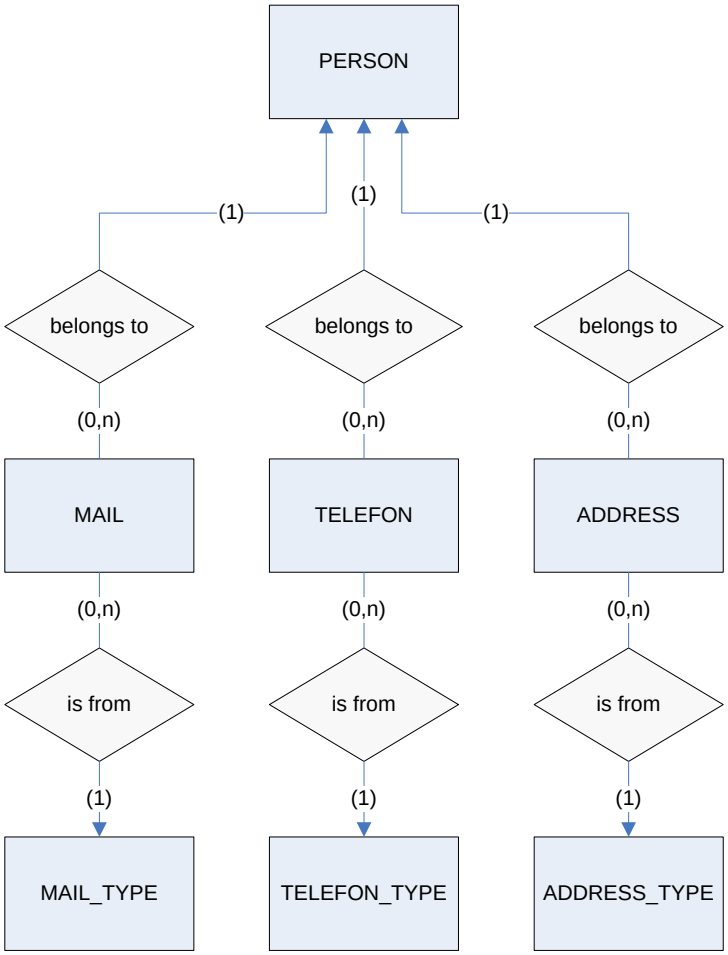

Abbildung 18: Kontaktinformationen zu Personen

E-Mail-Adressen, Telefonnummern und Postanschriften sind immer von einem bestimmten Typ.

Beispiel:

- *Private E-Mail-Adresse, geschäftliche E-Mail-Adresse*
- *Mobiltelefon, Telefax, Diensttelefon*
- *Heimatanschrift, Rechnungsadresse, Semesteranschrift*

4.2.3 Identity Service Provider und Provisioning Service Provider

Verantwortlichkeiten

Der Identity Service Provider bietet eine Operation an, über die die Abfrage einer einzelnen Identität aus einer Datenquelle möglich ist. Optional unterstützt ein Identity Service Provider die Suche nach Identitäten unter Angabe verschiedener Suchkriterien. Identitätsdaten können für einen Service Consumer pseudonymisiert werden.

Der Provisioning Service Provider stellt Operationen zum Einfügen, Ändern und Löschen einer Identität für ein Datenziel zur Verfügung. Optional übermittelt der Provisioning Service Provider erhaltene Identitätsdaten in gleicher oder veränderter Form an weitere Provisioning Service Provider. Dabei erhält der übermittelnde Service Provider die Rolle eines

Service Consumers. Zusätzlich können Datenbereinigungen auf Tupelebene, aber auch tupelübergreifend durchgeführt werden.

Neben den Verantwortlichkeiten, die für die beiden genannten Arten von Service Providern spezifisch gelten, lassen sich gemeinsame Verantwortlichkeiten festlegen.

Sowohl der Identity Service Provider als auch der Provisioning Service Provider unterstützen eine synchrone Bearbeitung von Anfragen. Optional wird eine asynchrone Bearbeitung von Anfragen angeboten.

Zur Verringerung der Anzahl auszutauschender Nachrichten können mehrere verschiedene Operationen für unterschiedliche Identitäten in einer Nachricht zusammengefasst werden. Der Identity und der Provisioning Service Provider zerlegen solche Operationsfolgen in Einzeloperationen.

Beide Arten von Providern unterstützen optional Verfahren zur Authentifizierung und Autorisierung von Service Consumern. Mit der Verschlüsselung auf Kommunikations- und Inhaltsebene kann zusätzlich die Vertraulichkeit der Daten bewahrt bleiben. Mit Hilfe von digitalen Signaturen ist eine Überprüfung der Integrität der Daten möglich.

Wird ein Auditing benötigt, so protokollieren die Provider ausgewählte Aktionen. Darüber hinaus können die Provider aufgetretene Fehler protokollieren, die aufgrund von fehlerhaften Daten, fehlerhafter Software oder Hardware entstehen.

Entwurf

Der Main Identity Store zeigt, dass es sich bei einer Datenquelle und bei einem Datenziel um dieselbe Komponente innerhalb der Architektur handeln kann. In dem nachfolgenden Entwurf wird nur noch von Datenbanken gesprochen. Dabei wird der Begriff der Datenbank in einem erweiterten Sinne verstanden und schließt insbesondere auch einfache Textdateien mit ein, in denen Identitätsdaten in strukturierter Form enthalten sind.

Wie aus den Verantwortlichkeiten für den Identity Service Provider und den Provisioning Service Provider hervorgeht, besitzen beide Service Provider viele Gemeinsamkeiten. Für einige Systeme an einer Hochschule gilt sogar, dass diese zugleich Datenziel und Datenquelle sind. In diesem Fall werden dann auch beide Service Provider benötigt. Deshalb werden in dem Entwurf der Identity Service Provider und Provisioning Service Provider zu einem Identity and Provisioning Service Provider (IPSP) zusammengefasst.

Der IPSP ist für eine Vielzahl von Aufgaben verantwortlich. Dabei kann die Verantwortlichkeit auf verschiedene Komponenten innerhalb des Identity and Provisioning Service Providers aufgeteilt werden. Im Folgenden werden diese Komponenten Schritt für Schritt entwickelt.

a) Wrapper

Identity and Provisioning Service Provider greifen auf Datenbanken zu, um Identitätsdaten abzufragen, hinzuzufügen, zu ändern und zu löschen. Dabei weisen die Datenbanken untereinander Heterogenitäten auf verschiedenen Ebenen auf. Der IPSP stellt mit dem angebotenen Service einen einheitlichen Zugriff auf die heterogenen Datenbanken für die Service Consumer zur Verfügung. Für die einzelnen Komponenten, aus denen der Identity and

Provisioning Service Provider aufgebaut ist, wäre eine Vereinheitlichung des Zugriffs auf die Datenbanken auch wünschenswert. Dadurch könnten diese Komponenten für jeden Identity and Provisioning Service Provider wieder verwendet werden.

Es wird das Konzept der Mediator-Wrapper-Architekturen aufgegriffen (vgl. Abschnitt 2.5.4), bei denen Wrapper die Aufgabe besitzen, die technischen, syntaktischen, Datenmodell- und strukturellen Heterogenitäten der unterschiedlichen Datenbanken zu überwinden. Ein IPSP enthält für den Zugriff auf eine Datenbank einen solchen Wrapper. Dabei wird der Zugriff auf alle zum Einsatz kommenden Wrapper durch eine festgelegte Schnittstelle vereinheitlicht. Die weiteren Komponenten des Identity and Provisioning Service Providers, die teilweise den Wrapper direkt über die einheitliche Schnittstelle nutzen, stellen im erweiterten Sinne die Mediatoren dar. Wie die nachfolgenden Abschnitte zeigen werden, wird dabei ein Mehrwert auf mehreren Ebenen erreicht.

Gegen das Wrapper Interface können Anfragen zum Einfügen, Ändern, Löschen und Herauslesen von Identitäten gestellt werden. Das Herauslesen ist nur für einzelne oder alle Identitäten möglich.

Die jeweilige Anfrage wird in eine datenbankspezifische Anfrage übersetzt, ausgeführt und das datenbankspezifische Ergebnis wieder in eine Wrapper einheitliche Ergebnisform überführt. Es kann vorkommen, dass sich bestimmte Operationen für eine Datenbank nicht umsetzen lassen. Dies kann beispielsweise technische Ursachen haben oder der Betreiber der Datenbank lässt bestimmte Operationen einfach nicht zu. In diesen Fällen liefert der Wrapper ein Ergebnis, das zum Ausdruck bringt, dass die jeweilige Operation nicht unterstützt wird.

Jeder Wrapper stellt ein oder mehrere Wrapper-Exportschemata zur Verfügung. Zur Auswahl des Exportschemas, das für den Zugriff verwendet werden soll, stellt das Wrapper Interface entsprechende Operationen bereit. Tauscht ein Identity and Provisioning Service Provider Daten mit dem Main Identity Store aus, so entspricht das Exportschema einen Teilausschnitt eines Schemas, das vollständig kompatibel zum Main Identity Store ist. Im Folgenden wird dieses durch den Main Identity Store vorgegebene Schema als globales Schema des Identitätsmanagementsystem bezeichnet. Nicht jedes Exportschema eines Wrappers muss jedoch diesem globalen Schema entsprechen. Die Einführung der Architektur zur System-, Daten- und Prozessintegration hat ja gezeigt, dass auch Identitäten zwischen Identity and Provisioning Service Providern ausgetauscht werden können, ohne den Weg über den Main Identity Store zu gehen. Es existieren also neben dem globalen Schema auch mehrere „multilaterale“ Schemata, an denen sich ein Exportschema orientieren kann.

Für die globalen und die multilateralen Schemata wird festgelegt, dass sie einem einheitlichen Datenmodell (z.B. relational, hierarchisch) entsprechen. Dies vereinfacht das Wrapper Interface, da die Struktur der Anfragen einem festgelegten Muster folgt. Es kann also eine einheitliche Anfragesprache verwendet werden.

Als kanonisches Datenmodell empfiehlt sich das hierarchische Datenmodell. Dies lässt sich wie folgt begründen:

- Die in vielen Datenquellen verwendeten relationalen Strukturen lassen sich gut in hierarchische überführen.

- Das für den Main Identity Store vorgestellte Schema enthält bereits hierarchische Strukturen. Die ansonsten dargestellten Relationen und Beziehungen im Main Identity Store lassen sich gut in eine vollständige hierarchische Struktur überführen. Auch der Rückweg ist ohne Informationsverluste möglich.
- Protokolle, wie beispielsweise SOAP, die bei serviceorientierten Architekturen zum Einsatz kommen, verwenden ebenfalls das hierarchische Datenmodell.
- Verzeichnisdienste, die in Kombination mit LDAP sehr häufig als Provisionierungsziel verwendet werden, basieren ebenfalls auf dem hierarchischen Datenmodell. Die Objekte in den Verzeichnisdiensten verwenden keine Vererbungsmechanismen, die nicht in einfache relationale Strukturen überführt werden können.

Zur Überwindung der syntaktischen Heterogenität müssen im Wrapper gegebenenfalls Zeichen in eine andere Kodierung überführt werden. Wenn die Zielkodierung über einen geringeren Zeichenvorrat verfügt, kann es vorkommen, dass einzelne Zeichen nicht überführt werden können. Es gibt zwei Möglichkeiten, diesem Problem zu begegnen. Entweder die Verarbeitung der Identitäten wird abgelehnt oder Zeichen werden durch andere ersetzt. So kann beispielsweise ein ø in ein o überführt werden. Dies bedeutet aber immer ein Informationsverlust und senkt die Qualität. Auch eine Kombination beider Verfahren ist denkbar. So können für einige Zeichen Ersetzungsregeln existieren. Alle anderen nicht abbildbaren Zeichen führen zu einer Ablehnung der Verarbeitung.

b) Protocol and Security Handler

Der Protocol and Security Handler ermöglicht die Sichtbarkeit und die Interaktion des Identity and Provisioning Service Providers innerhalb der serviceorientierten Architektur. Dabei setzt sich die Sichtbarkeit aus den Eigenschaften Bewusstheit, Bereitschaft sowie Erreichbarkeit zusammen (vgl. Abschnitt 2.10.1).

Die Bewusstheit umfasst, dass ein Service Consumer anhand der Servicebeschreibung, die beim IPSP abgerufen werden kann, Informationen über die angeschlossene Datenbank und über die unterstützten Operationen erhält. Zu den Operationen gehören die vom Wrapper Interface angebotenen Operationen (Einfügen, Ändern, Löschen, Herauslesen von Identitäten, Abruf von Schemainformationen, Auswahl des zu verwendenden Wrapper Exportschemas). Darüber hinaus wird eine Operation für die Suche nach Identitäten angeboten.

Die Bereitschaft bei serviceorientierten Architekturen beinhaltet, dass ein Service Consumer den Service beim Service Provider in Anspruch nehmen darf. Es wird deshalb eine Authentifizierung und anschließende Autorisierung im Protocol and Security Handler durchgeführt. Die Authentifizierung erfolgt dabei gegen eine zentrale Authentifizierungsinstanz, die später noch beschrieben wird (siehe Abschnitt 4.2.4). Nach erfolgreicher Authentifizierung ist die Glaubwürdigkeit (Authentizität) der anfragenden Identität hergestellt. Die anschließende Autorisierung wird ebenfalls mit Hilfe einer zentralen Komponente realisiert (siehe Abschnitt 4.2.5). Um eine Vertraulichkeit der Identitätsdaten zu erreichen, unterstützt der Protocol and Security Handler eine Verschlüsselung auf Inhalts- und Verbindungsebene. Die Integrität der Daten wird durch die Verwendung digitaler Signaturen überprüft.

Der Protocol and Security Handler ermöglicht die Erreichbarkeit über das Netzwerk. Dabei kommt ein Serviceprotokoll zum Einsatz, mit dem die Interaktionsmöglichkeiten zwischen den Serviceteilnehmern festgelegt werden. Das Serviceprotokoll folgt dem Request-

Response-Paradigma. Es unterstützt sowohl synchrone als auch asynchrone Anfragen. Darüber hinaus können mehrere Operationen auf verschiedenen Identitäten in einer Anfrage enthalten sein. Das Serviceprotokoll stellt entweder selbst ein Anwendungsprotokoll des TCP/IP-Protokollstapels dar oder es erfolgt ein Binding an ein gängiges Anwendungsprotokoll wie beispielsweise HTTP.

Abbildung 19 zeigt die zwei beschriebenen Kernkomponenten, aus denen sich der Identity and Provisioning Service Provider zusammensetzt. Sie werden über den im nächsten Abschnitt beschriebenen Operation Workflow Handler verbunden.

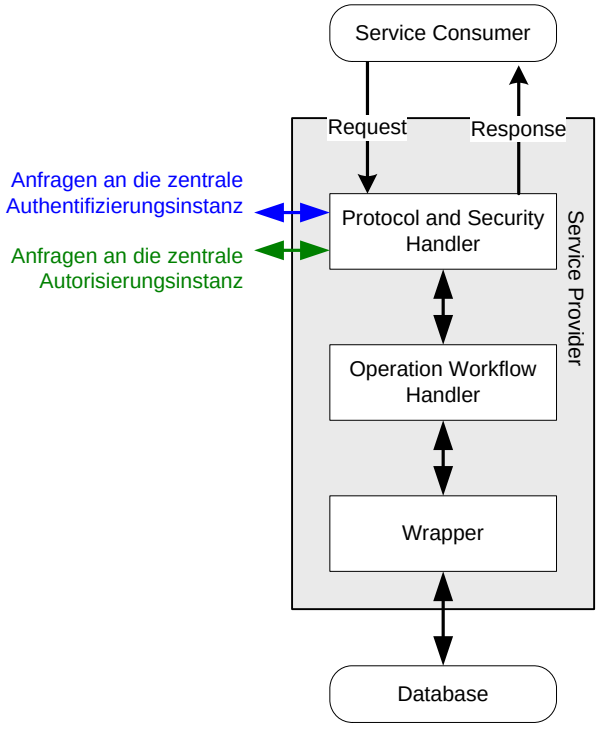

Abbildung 19: Kernkomponenten des Identity and Provisioning Service Providers

c) Operation Workflow Handler

Der Operation Workflow Handler ist dafür verantwortlich, die Operationen innerhalb des Identity and Provisioning Service Providers an die richtigen Komponenten zu delegieren. Gleiches gilt für die Ergebnisse der durchgeführten Operationen. Dabei kann der Operation Workflow Handler auch weitere Operationsaufrufe für andere Komponenten des IPSP generieren. Es entsteht ein Workflow innerhalb des Identity and Provisioning Service Providers.

Der Operation Workflow Handler erzeugt nicht nur neue Operationen für Komponenten innerhalb des Identity and Provisioning Service Providers. Er kann auch nach dem erfolgreichen Einfügen, Ändern oder Löschen einer Identität Anfragen zum Einfügen, Ändern und Löschen an andere IPSP erzeugen. Dabei greift der Operation Workflow Handler auf ein Regelwerk zurück, das er entsprechend der erfolgten Operation für jedes Datenziel abarbeitet. Der Operation Workflow Handler übernimmt in diesem Fall die Rolle eines Service Consumers, der Identitätsdaten an andere Datenziele propagiert.

d) Attribute Release Filter

Um die wesentlichen Datenschutzanforderungen einhalten zu können, wird der Identity and Provisioning Service Provider um den Attribute Release Filter erweitert. Der Attribute Release Filter filtert Identitätsdaten, die dem Protocol and Security Handler von einer Komponente innerhalb des IPSP übergeben werden sollen. Die Art der Filterung richtet sich danach, welches System die Daten erhalten soll. Dabei können die zu übertragenden Identitätsdaten folgendermaßen verändert werden:

- Löschen von einzelnen Attributen
- Löschen von einzelnen Attributen in Abhängigkeit von den Attributwerten
- Pseudonymisierung von Identitäten

Eine Umbenennung von Attributen und eine Veränderung der Repräsentation von Werten werden im Attribute Release Filter nicht vorgenommen. Service Consumer und IPSP einigen sich ja über das zu verwendende Schema. Der Wrapper innerhalb des Service Providers realisiert dieses Schema und führt die entsprechenden Schematransformationen durch.

In einer Attribute Release Policy erfolgt die Definition der Filterregeln. Die Attribute Release Policy kann hochschulweit erstellt werden. Das Ergebnis der Geschäftsprozess-Architektur ist die Trennung in obligatorische und optionale Systeme (vgl. Abschnitt 4.1.2.). Da für die obligatorischen Systeme eine Datenschutzrichtlinie existiert, in der die erlaubten Attribute festgelegt sind, können für diese Systeme auch feste Filterregeln definiert werden. Für die optionalen Systeme müssen dagegen pro Person und System Filterregeln gespeichert werden, die von der betroffenen Person jederzeit änderbar sind.

Die Abbildung 20 stellt die Operations- und Ergebnisabfolge beim erfolgreichen Herauslesen einer Identität dar. Dabei erhält der Attribute Release Filter die herausgelesene Identität vom Operation Workflow Handler (4), filtert diese und übergibt die gefilterte Identität dem Protocol and Security Handler (5).

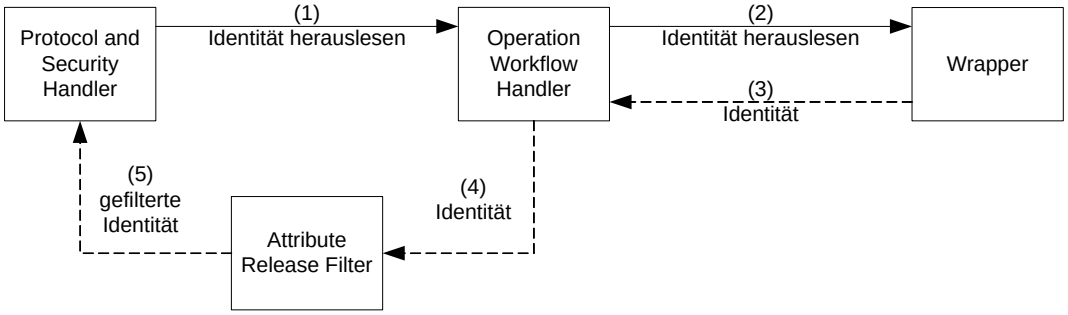

Abbildung 20: Vorgänge im IPSP beim Herauslesen einer Identität[45]

[45] Die Interaktionen mit den Service Consumern sowie die Zugriffe auf die Datenbank werden nicht betrachtet.

e) Harvester

Der Wrapper selbst unterstützt nur das Herauslesen einer einzelnen oder aller Identitäten. Der Harvester ermöglicht hingegen durch den Aufbau von geeigneten Suchindizes die komplexe und unscharfe Suche nach Identitäten. Der Aufbau der Indizes erfolgt entweder asynchron durch Herauslesen aller Identitäten über den Wrapper aus der Datenbank oder synchron bei allen Veränderungen an Identitäten.

Die Abbildung 21 zeigt die Operations- und Ergebnisabfolge bei der erfolgreichen Suche nach Identitäten. Bevor die Identitäten über den Wrapper aus der Datenbank gelesen werden können, erfolgt zunächst eine Suche mit Hilfe des Harvesters.

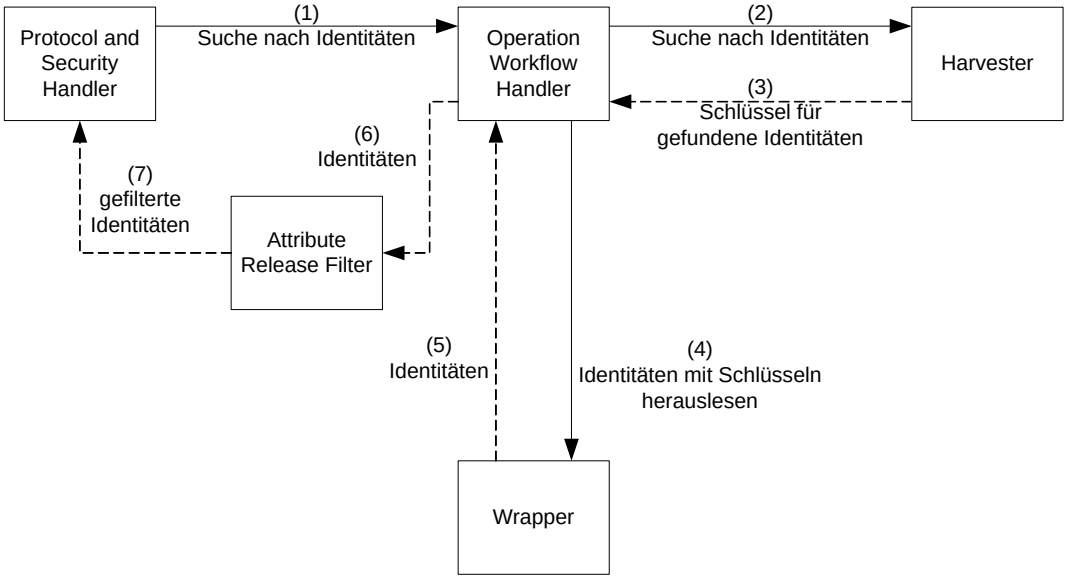

Abbildung 21: Vorgänge im IPSP bei der Suche nach Identitäten[46]

f) Data Scrubber

Der Data Scrubber führt Datenfehlerbereinigungen auf einzelnen Attributen oder einzelnen Identitäten durch. Die möglichen Fehler, die zu eliminieren sind, wurden in Abschnitt 2.5.5 („Datenfehler und Datenfehlerbereinigung“) vorgestellt. Der Data Scrubber kann über eigene Schnittstellen auf unterschiedliche externe Datenbanken zugreifen, um beispielsweise Adressen korrigieren zu können oder Plausibilitätsprüfungen durchzuführen (z.B. Adresse zu Postleitzahl oder Postleitzahl zu Wohnort).

g) Record Matcher

Der Record Matcher übernimmt die Aufgabe, Duplikate zu ermitteln. Dabei kommen verschiede Verfahren zur Duplikaterkennung zum Einsatz. Für die Identitäten von Personen, wie sie im vorgestellten Main Identity Store gespeichert werden, dient das folgende Verfahren zur Erkennung von Duplikaten:

[46] Die Interaktionen mit den Service Consumern sowie die Zugriffe auf die Datenbank werden nicht betrachtet.

Schritt 1: Data Scrubbing
Zunächst werden die einzelnen Identitäten mit Hilfe des Data Scrubbers von Fehlern bereinigt.

Schritt 2: Normalisierung
Die Vornamen (givenName), die Nachnamen (surName) und die Geburtsnamen (birthName) werden in die Zeichenkodierung ASCII überführt und einheitlich klein geschrieben. Für die Umwandlung nach ASCII existieren zu den vorkommenden Zeichenkodierungen entsprechende Ersetzungsregeln. Punkte und Apostrophe in den genannten Attributen werden eliminiert. Für Bindestriche erfolgt eine Ersetzung durch Leerzeichen.

Schritt 3: Überprüfung der exakten Übereinstimmung
Anhand der Attribute givenName, surName und birthDate wird überprüft, ob Identitäten exakt übereinstimmen. Ist dies der Fall, so wird ein weiteres Merkmal herangezogen, um das Vorliegen eines Duplikats zu verifizieren. Dies kann die Anschrift, die E-Mail-Adresse oder der Geburtsname sein. Der Grund dafür, dass der Geburtsname nicht für die erste Überprüfung herangezogen wird, besteht darin, dass er meist nicht angegeben bzw. erfasst wird.

Schritt 4: Bestimmung von Ähnlichkeitsmaßen
Durch eine Kombination von Editierabständen und tokenbasierten Ähnlichkeitsmaßen werden weitere Duplikate ermittelt. Dabei werden ein sichererer und ein unsicherer Bereich für die Ergebnisse der Ähnlichkeitsbestimmung festgelegt. Die sicheren Duplikate können automatisiert weiterverarbeitet werden. Für die unsicheren Kandidaten ist eine manuelle Überprüfung erforderlich. Bei der manuellen Prüfung wird gegebenenfalls mit dem Betreiber einer autoritativen Datenquelle oder direkt mit einer Person Kontakt aufgenommen, um Unstimmigkeiten zu klären. Das Ergebnis der Klärung ist entweder die Korrektur der Daten, die dazu führt, dass die Duplikate bei einer erneuten Überprüfung automatisch erkannt werden, oder eine manuelle Zuweisung, die verhindert, dass eine Duplikatbestimmung erneut durchgeführt werden muss.

Schritt 5:
Mit Hilfe der Komponente Record Merger, die nachfolgend beschrieben wird, erfolgt eine Konsolidierung von Duplikaten.

Es stellt sich die Frage, in wie weit nach den genannten Verfahren Identitäten als Duplikate erkannt werden, die in der Realität gar keine sind. Ein exaktes mathematisches Modell zur Berechnung der Wahrscheinlichkeit von solchen Fehlzuordnung lässt sich nicht aufstellen. Je nach Region lässt sich beispielsweise eine andere Verteilung von Vornamen, Nachnamen und Geburtjahren beobachten. Solche Parameter müssten in ein Modell mit aufgenommen werden. Das folgende Fallbeispiel soll für Schritt 3 zeigen, wie häufig der gleiche Name und der gleiche Namen in Verbindung mit einem gleichen Geburtsdatum an einer deutschen Hochschule auftreten.

Fallbeispiel:
An der Freien Universität Berlin wird unter dem Namen FUDIS (Freie Universität Directory and Identity Service) ein zentrales Identitätsmanagementsystem betrieben, mit dem Identitätsdaten von Mitarbeitern, Studierenden, Alumnis und Gästen zusammengeführt werden. Die Zahl der verwalteten Personenidentitäten liegt bei ca. 40.000. Die maximale Anzahl von Personen, die einen gleichen Vor- und Nachnamen besitzen, beträgt 14. Nimmt

man das Geburtsdatum hinzu, so erhält man keine Duplikate mehr. Es könnte hierbei der Verdacht aufkommen, dass bei der Zusammenführung der Personengruppen bereits Duplikate bereinigt wurden. Betrachtet man allein die größte Personengruppe, nämlich die mit ca. 30.000 Studierenden, so erhält man das gleiche Ergebnis.

h) Record Merger

Der Record Merger erhält Duplikate und besitzt die Aufgabe, diese Duplikate zusammenzuführen. In Abschnitt 2.5.5 wurden die damit verbundenen Probleme erläutert. Enthalten beispielsweise zwei Duplikate für ein Attribut unterschiedliche Informationen, so muss entschieden werden, ob eine oder keine richtig ist. Dies ist nicht immer automatisiert realisierbar. Eine manuelle Korrektur von Daten in den autoritativen Datenquellen ist erforderlich.

Abbildung 22 zeigt, wie der Data Scrubber, der Record Matcher und der Record Merger vom Operation Workflow Handler eingebunden werden, um eine Identität hinzuzufügen, zu der es in der Datenbank bereits ein Duplikat gibt.

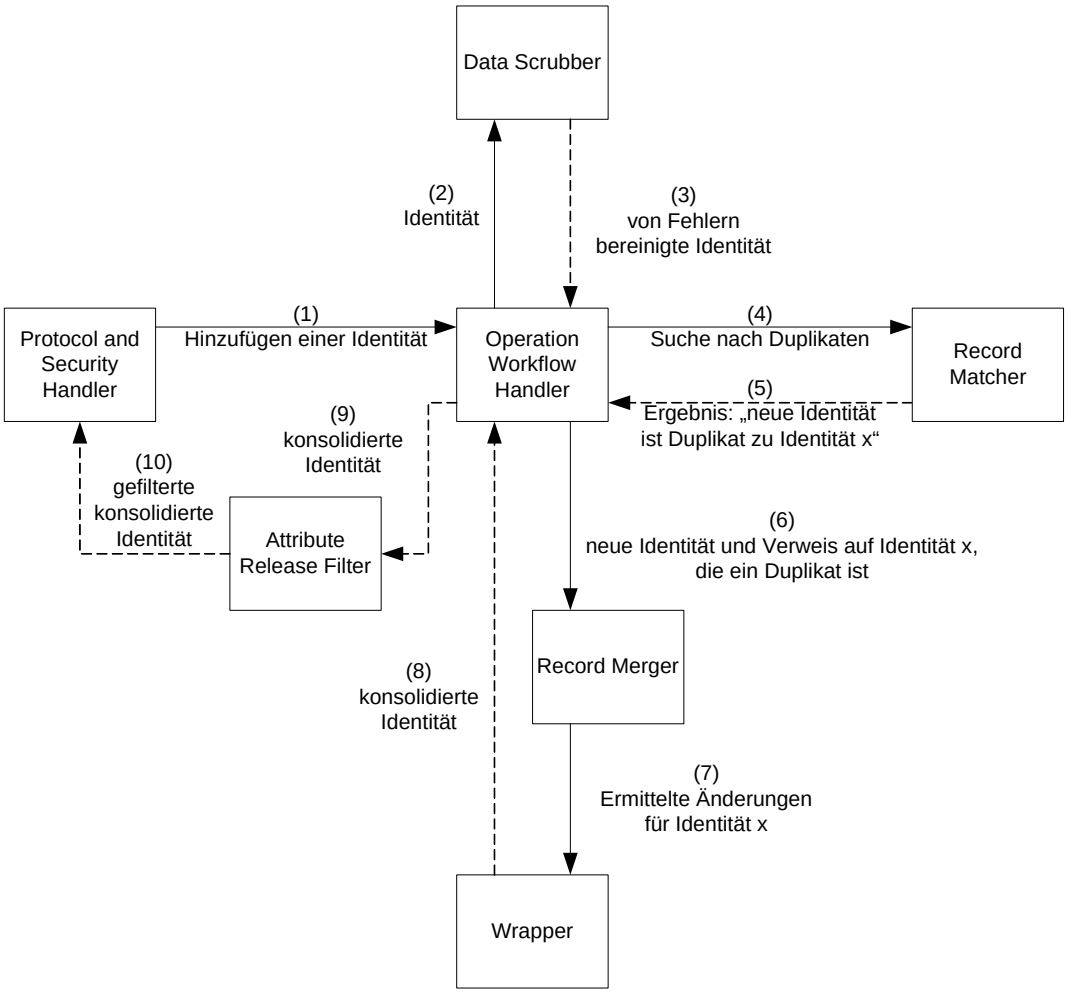

Abbildung 22: Vorgänge im IPSP beim Hinzufügen einer Identität mit Duplikat

i) Auditing

Die Auditing Komponente protokolliert alle Informationen, die für ein Auditing benötigt werden. Insbesondere werden die Operationen zum Einfügen, Ändern und Löschen einer Identität erfasst. Die Protokollinformationen müssen nicht zwangsläufig in dem Identity and Provisioning Service Provider gespeichert werden. Es ist auch möglich, die Daten an eine zentrale Auditinginstanz zu übermitteln.

j) Monitor

Der Monitor überprüft, ob die anderen Komponenten des IPSP verfügbar sind und fehlerfrei funktionieren. Außerdem können die anderen Komponenten auftretende Fehler dem Monitor melden. Der Monitor selbst kann Fehler an eine zentrale Instanz weiterleiten, die das gesamte Identitätsmanagementsystem überwacht.

k) Consistency Checker

Der Consistency Checker besitzt die Verantwortung, den zugreifbaren Datenbestand der an den IPSP angeschlossenen Datenbank in regelmäßigen Abständen (z.B. täglich oder wöchentlich) vollständig daraufhin zu überprüfen, ob die beim Speichern hergestellte Konsistenz noch besteht. Dazu nutzt er den Data Scrubber, um zunächst Fehler in einzelnen Identitäten zu bereinigen. Anschließend führt der Consistency Checker mit Hilfe des Record Matchers eine Duplikaterkennung durch. Für aufgetretene Duplikate wird versucht, diese über den Record Merger zusammenzuführen. Bei Identitäten, die der Identity and Provisioning Service Provider von autoritativen Datenquellen erhält, erfolgt zusätzlich die Überprüfung, ob diese noch nach der Ursprungsdatenbank in der vorliegenden Form gültig sind. Gegebenfalls folgen weitere Korrekturen oder sogar das Löschen einer Identität. Es kann dabei auch vorkommen, dass eine Identität von einer autoritativen Datenquelle fälschlicherweise nicht übertragen wurde. Fehlende Identitäten werden dann entsprechend nachgetragen.

l) Observer

Mit den Operationen zur Suche und zum Herauslesen von Identitäten initiieren Service Consumer eine Datenübermittlung vom Identity and Provisioning Service Provider zum Service Consumer. Der IPSP arbeitet also in dem bereits beschriebenen Pull-Modus. Um seitens des Service Consumers die Daten aktuell zu halten, ist es erforderlich, dass dieser in regelmäßigen Abständen nach Aktualisierungen beim Identity and Provisioning Service Provider fragt. Dies kann in einigen Fällen (z.B. Webanwendungen) das adäquate Mittel sein.

In dem Main Identity Store und in den anderen Datenzielen müssen nach den Anforderungen die Daten möglichst aktuell und konsistent vorliegen. Für den Main Identity Store liegt die Zahl der gespeicherten Identitäten je nach Größe der Hochschule schätzungsweise zwischen 10.000 und 200.000. Ein häufiger Abgleich von allen Identitätsdaten nach dem Pull-Prinzip würde eine permanent hohe Netz- und Systemlast erzeugen.

Mit Hilfe des Observers wird der IPSP in die Lage versetzt, neue, geänderte sowie gelöschte Identitäten an mehrere Datenziele zu übermitteln. Der Observer ermöglicht also den Push-Modus. Dadurch kann auch ein Verbindungsaufbau aus einer demilitarisierten Zone

(DMZ) nach außen erfolgen. Ein Identity and Provisioning Service Provider, der sich in einer von außen nicht mehr zugreifbaren DMZ befindet, kann zwar für außen stehende Systeme keinen Service mehr anbieten, ist aber durch den Push-Modus in der Lage, kontrolliert Identitäten zu übermitteln. Sensible Systeme, die am Identitätsmanagementsystem teilnehmen sollen, bleiben dadurch weiterhin geschützt.

Der Observer ist neben dem Wrapper die zweite Komponente des Identity and Provisioning Service Providers, die direkt auf die angebundene Datenbank zugreift. Dabei nutzt der Observer die Funktionalitäten aus, die die jeweilige Datenbank besitzt, um Änderungen der Daten zu ermitteln. Im günstigsten Fall existiert in der Datenbank ein Mechanismus, der dem Observer alle Änderungen sofort mitteilt. Im schlechtesten Fall muss der Observer in regelmäßigen Abständen den gesamten Datenbestand abrufen und die Änderungen gegenüber dem Ergebnis des vorherigen Abrufs selber ermitteln.

Der Observer muss die gleichen Heterogenitäten wie der Wrapper überwinden. Aus diesem Grund nutzt der Observer einzelne Komponenten des Wrappers.

Die vom Observer ermittelten Änderungen werden dem Operation Workflow Handler übergeben. Dieser löst dann alle weiteren erforderlichen Aktionen aus, um die Identitätsdaten zu den Datenzielen zu übermitteln.

Abbildung 23 zeigt den vollständigen Identity and Provisioning Service Provider mit allen beschriebenen Komponenten.

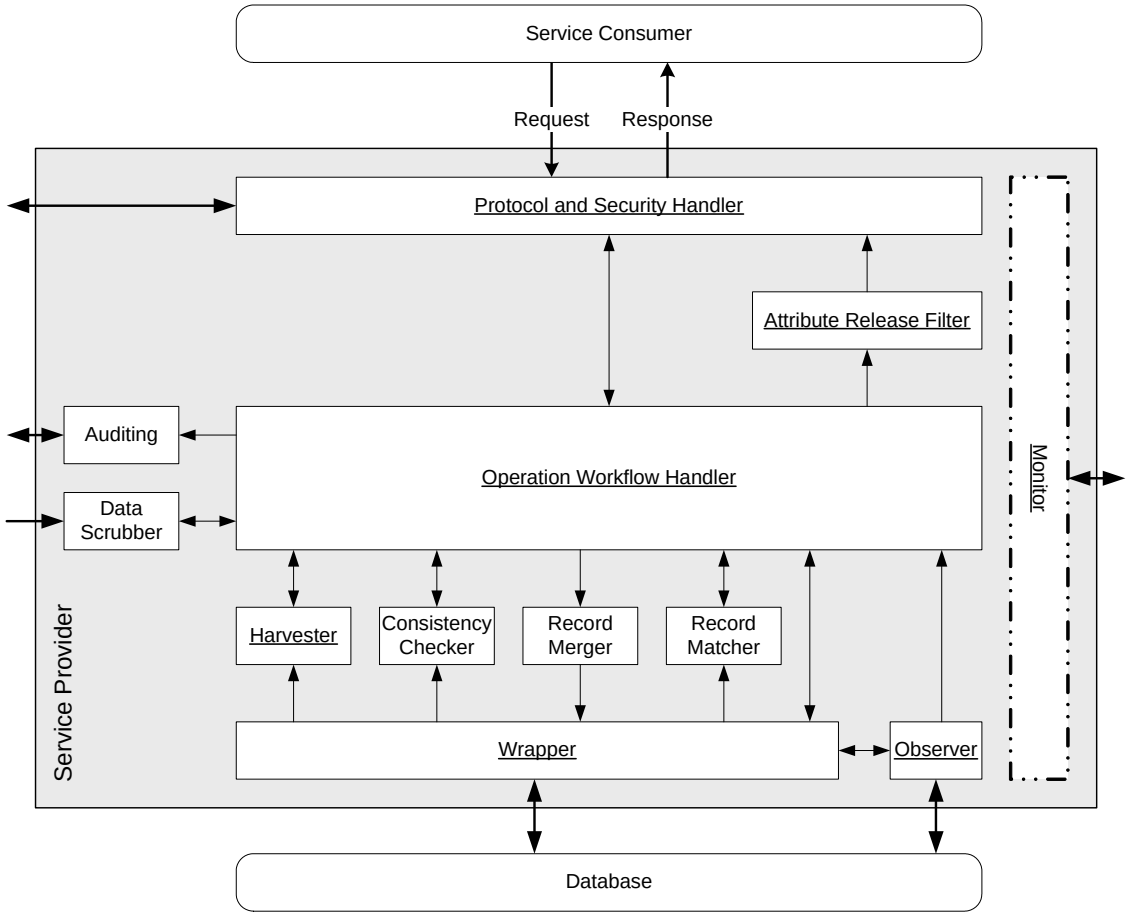

Abbildung 23: Vollständiger Identity and Provisioning Service Provider

Der Monitor ist verbunden mit allen anderen Komponenten im IPSP. Aus Gründen der Übersicht werden die Verbindungen nicht mit dargestellt. Die Komponenten Protocol and Security Handler, Monitor, Data Scrubber sowie Auditing kommunizieren noch wie bereits beschrieben mit (zentralen) Komponenten außerhalb des Identity and Provisioning Service Providers.

Der Identity Service Provider und der Provisioning Service Provider wurden in diesem Abschnitt ja zu einem Identity and Provisioning Service Provider zusammengeführt. Benötigt man beispielsweise nur einen reinen Identity Service Provider, so reichen die Komponenten aus, deren Namen in Abbildung 23 unterstrichen dargestellt sind.

Transaktionen

In der so beschriebenen Form unterstützt der Identity and Provisioning Service Provider keine Transaktionen. Um eine Transaktionsunterstützung für die an einen IPSP jeweils angeschlossene Datenbank zu erhalten, bedarf es einer Erweiterung das Wrappers. Der Wrapper selbst kann dabei die jeweiligen Transaktionsmechanismen der Datenbank nutzen, insofern solche Mechanismen vorhanden sind. Ansonsten muss diese Aufgabe vollständig vom Wrapper übernommen werden. Eine Komponente innerhalb des IPSP oder für die gesamte Informationssystem-Architektur, die Transaktionen über alle Datenbanksysteme, die am Identitätsmanagementsystem teilnehmen, realisiert, fehlt völlig. In den Abschnitten 2.2.2 und 2.11.3 wurde bereits angedeutet, dass es bei Identitätsmanagementsystemen kaum oder gar nicht möglich ist, globale Transaktionen zu realisieren. Man nimmt in Kauf, dass eine Änderung in einer autoritativen Datenquelle sich erst zu einem späteren Zeitpunkt bei den Datenzielen niederschlägt. Es muss nur sichergestellt werden, dass die Zeitpunkte nicht zu weit auseinander liegen (max. ein Tag).

4.2.4 Main Authentication Authority

Die Main Authentication Authority stellt die zentrale Authentifizierungsinstanz an einer Hochschule dar. Sie bietet mehrere Schnittstellen an, über die eine Authentifizierung möglich ist. Die Main Authentication Authority bietet Verfahren für ein Single-Sign-On und ein Single-Logout an. Die Main Authentication Authority erhält die Passwörter vom Main Identity Store.

4.2.5 Main Authorization Authority

Die Main Authorization Authority ist eine zentrale Komponente, die Organisations- und Systemrollen aus dem Main Identity Store erhält, Die Rolleninformationen können über mehrere Schnittstellen von den IT-Systemen der Hochschule abgerufen werden. Dabei wird davon ausgegangen, dass die IT-Systeme die Informationen nicht lokal speichern müssen, sondern die zentrale Instanz nutzen können.

4.2.6 Weitere Komponenten

In dem Abschnitt zum Identity Service Provider und Provisioning Service Provider wurden bereits eine zentrale Komponente für das Überwachen aller Komponenten (Monitor) sowie eine zentrale Komponente für das Auditing aufgeführt.

Um ein Workflowmanagement, einen User Self-Service und weitere Funktionalitäten zu realisieren, bedarf es weiterer Komponenten, die nicht weiter beschrieben werden. Viele Komponenten nutzen die existierenden Services und fügen diese gegebenenfalls zu neuen Services zusammen.

Beispiel:
Bei einem User Self-Service sollen die Mitglieder der Hochschule die Möglichkeit erhalten, ihre private Anschrift zu ändern. Dafür wird eine Webanwendung programmiert, die den Identity and Provisioning Service Provider des Main Identity Stores benutzt.

Für die zentrale Administration des Identitätsmanagementsystems werden diverse Werkzeuge benötigt. Die Werkzeuge nutzen neben den Service Providern auch andere Schnittstellen, um die einzelnen Komponenten zu konfigurieren und um Veränderungen an den Identitätsdaten vorzunehmen. Zum Beispiel müssen in dem Main Identity Store Systeme und Systemrollen angelegt und die Regeln für die Zuweisung der Benutzerkonten zu den Systemen definiert werden.

4.2.7 Abschließende Übersicht

Abbildung 24 zeigt noch einmal vereinfacht die wichtigsten Komponenten der Informationssystem-Architektur, die in den vorangegangen Abschnitten beschrieben wurden. Dabei werden die Systeme in autoritative Datenquellen, zentrale Komponenten und Datenziele eingeteilt.

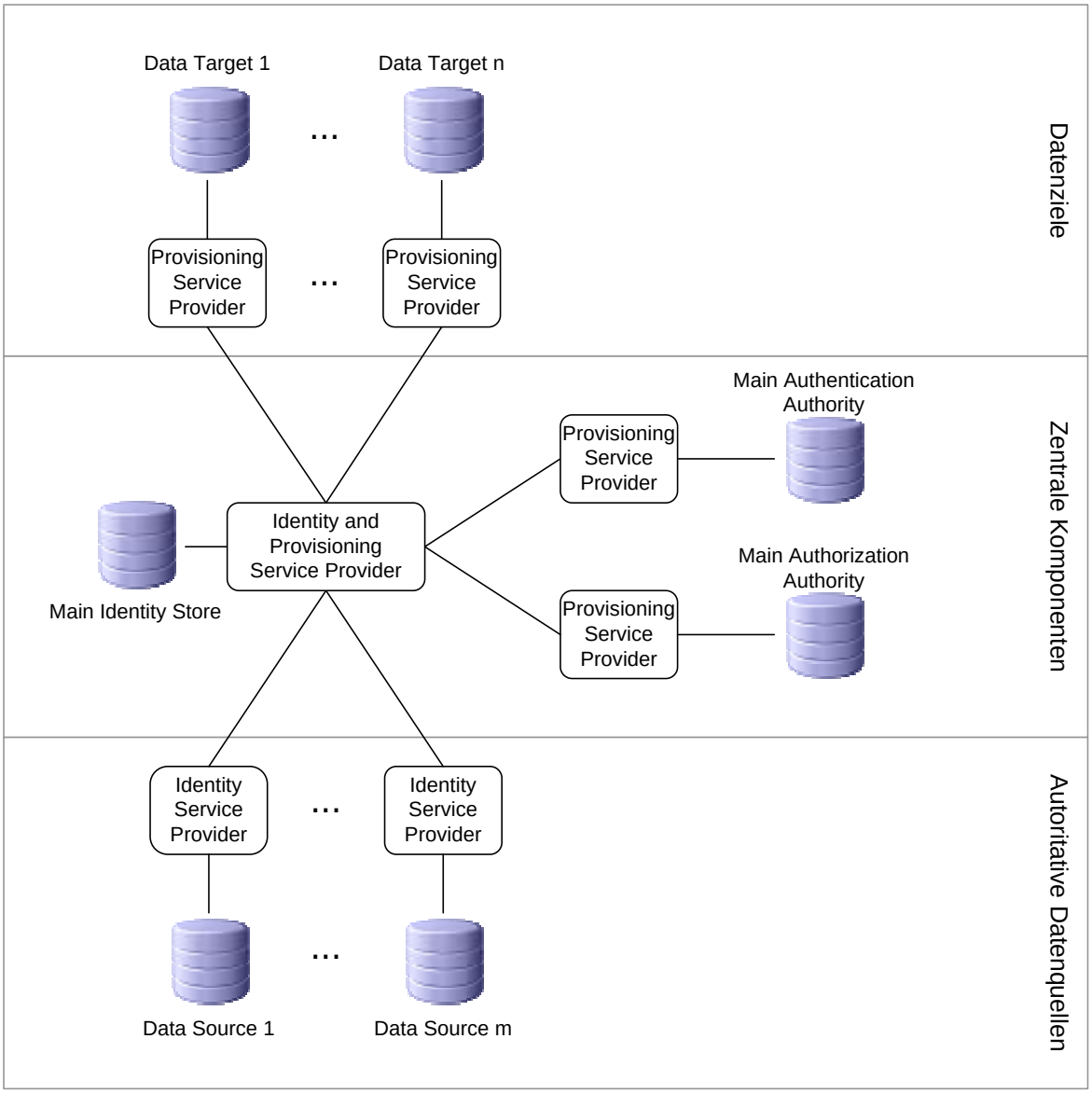

Abbildung 24: Komponenten der Informationssystem-Architektur

4.3 Softwareprodukt-Architektur

In der Softwareprodukt-Architektur werden wie angekündigt Softwareprodukte für die Realisierung einiger Komponenten vorgestellt. Die meisten Produkte basieren auf der Programmiersprache Java; fast alle unterliegen einer Open-Source-Lizenz. Dabei wird nur die Software genannt, die für eine Testimplementierung im Rahmen dieser Arbeit zum Einsatz kamen. Kommerzielle Systeme wurden nur theoretisch betrachtet, sind aber sicherlich auch für die Realisierung vieler Komponenten geeignet. Es wird nochmals auf die Quellen [Orac06], [Sun06], [IBM05] und [IBM06] verwiesen, die mit eine Grundlage für den Abschnitt 2.2.3 darstellten.

a) Softwareprodukte für den Main Identity Store

Der Main Identity Store kann mit der Datenbanksoftware ProstgreSQL [PostgreSQL07] ab der Version 8.0 realisiert werden. PostgreSQL ermöglicht die Erstellung relationaler Datenbanken, die sich über SQL verwalten lassen. Das Produkt verfügt über einen Listen/Notify-Mechanismus, der Anwendungen darüber informieren kann, wenn sich in Tabellen bestimmte Daten geändert haben. Leider existiert keine SQL-Spracherweiterung für die Verwaltung von Hierarchien. PostgreSQL bietet jedoch mit PL/pgSQL eine eigene Programmiersprache an, mit der sich datenbankinterne Funktionen erstellen lassen. Damit können auch Funktionen zur Unterstützung hierarchischer Strukturen erstellt werden.

Neben der Frage, welche Datenbanksoftware für die Speicherung von den genannten Identitäten verwendet werden kann, muss auch geklärt werden, welche Software die genannte automatische Zuweisung zu Organisationsrollen, Systemrollen und Gruppen übernimmt. Die Software Grouper [Internet2_07b] dient zum einen der Verwaltung von Gruppen und Rollen. Zum anderen enthält Grouper einen Mechanismus, über den regelbasiert die Zuordnungen von Identitäten zu Gruppen erfolgt. Dabei werden Informationen darüber genutzt, in welchen Gruppen sich eine Identität bereits befindet.

b) Identity and Provisioning Service Provider

Für den Identity and Provisioning Service Provider, der der Abfrage und Provisionierung von Identitäten dient, werden zu den folgenden Komponenten Softwareprodukte genannt:

- Protocol and Security Handler
  Das vom Protocol and Security Handler verwendete Service Protokoll kann durch eine Kombination von SPML und SAML realisiert werden. Dabei sind SAML-Assertions in eine SPML-Nachricht eingebunden. Erste Versuche einer Spezifikation hierfür findet man in [OASIS06e]. Die Software OpenSAML (siehe [Internet2_07c]) und die Software OpenSPML (siehe [OpenSPML07]) implementieren die jeweiligen Standards. Das aus SAML und SPML bestehende Service Protokoll kann an SOAP gebunden werden. SOAP selbst lässt sich mit Apache Axis 2 (siehe [Apache07a]) umsetzen. Der Apache Tomcat (siehe [Apache07b]) ermöglicht den Transport von SOAP-Nachrichten über HTTP und HTTPS.

- Harvester
  Um komplexe und unscharfe Suchen im Harvester umzusetzen, kann Apache Lucene (siehe [Apache07c]) verwendet werden. Lucene ist eine Volltextsuchmaschine, die unter anderem unscharfe Suchen ermöglicht. Außerdem können verschiedene Suchanfragen miteinander kombiniert werden.

- Record Matcher
  Apache Lucene unterstützt diverse Ähnlichkeitsmaße und ist auch für den Record Matcher eine geeignete Software. Der so genannte FuzzyQuery basiert beispielsweise auf der Levenshtein-Distanz. Auch die Jaccard-Ähnlichkeit und TFDIF sind mit Lucene umsetzbar.

c) Main Authorization Authority

Die Main Authorization Authority enthält Personen und Benutzerkonten mit hierarchisch angeordneten Organisations- und Systemrollen. Diese können in einem Verzeichnis gespeichert werden, das über LDAP schnell abfragbar ist. Als LDAP-Server kann dabei OpenLDAP [OpenLDAP07] verwendet werden.

d) Main Authentication Authority

Zur Realisierung der verschiedenen Schnittstellen, über die eine Authentifizierung möglich sein soll, können mehrere Softwareprodukte kombiniert werden. Mit Microsofts Active Directory in Windows 2003 Server (vgl. [Microsoft07]) erhält man eine Kerberos-Implementierung. Außerdem können Authentifizierungen über LDAP durchgeführt werden. Als Alternative für die LDAP-Schnittstelle von Active Directory bietet sich wieder OpenLDAP an. Um ein Single-Sign-On und Single-Logout im Bereich von Webanwendungen zu realisieren, kann die bereits vorgestellte Software Shibboleth eingesetzt werden. Shibboleth selbst leitet die Authentifizierungsanfrage intern an vorhandene Schnittstellen wie beispielsweise LDAP weiter.

Wie in der Informationssystem-Architektur beschrieben, können vorhandene Services miteinander kombiniert werden, um weitere Funktionalitäten zu realisieren. Damit eine Hochschule an DFN-AAI teilnehmen kann, reicht es nicht aus, dass über Shibboleth nur eine Authentifizierung angeboten wird. Es werden nach einer erfolgreichen Authentifizierung auch noch Identitätsdaten und teilweise sogar Autorisierungsinformationen benötigt. Das bedeutet, dass Shibboleth auch auf den Identity Service des Main Identity Stores zugreift, um Identitätsdaten an Teilnehmer im föderativen Verbund der Hochschulen zu übermitteln.

5 Zusammenfassung und Ausblick

In dieser Arbeit wurde zunächst ein allgemeines Verständnis für Identitätsmanagementsysteme hergestellt. Dafür war es notwendig, den Begriff der Identität näher zu untersuchen. Mit der Darstellung des Lebenszyklus einer Identität konnte auf die ersten Probleme hingewiesen werden, die im Zusammenhang mit dem Erstellen, Ändern, Löschen sowie Verteilen einer Identität entstehen können. Durch die Trennung der Anforderungen an ein Identitätsmanagementsystem in funktionale und nicht funktionale Anforderungen erfolgte eine klare Abgrenzung der für Identitätsmanagementsysteme spezifischen Eigenschaften von den Kriterien, die auf viele IT-Systeme innerhalb einer Organisation anwendbar sind.

Die Datenintegration, die Authentifizierung und die Autorisierung stellen die Kernaufgaben eines Identitätsmanagementsystems dar. Aus diesem Grund erfolgte eine nähere Betrachtung der einzelnen Themenbereiche. Der Begriff der Datenintegration wurde zunächst von der Prozessintegration abgegrenzt, die ebenfalls bei Identitätsmanagementsystemen stattfindet. Die Datenintegration beschäftigt sich mit den Problemen der Verteilung, der Autonomie sowie den Heterogenitäten von Daten und Datenquellen. Diese Problemfelder wurden in der Arbeit näher diskutiert. Außerdem erfolgte mit den föderierten und mediatorbasierten Datenbanksystemen eine Betrachtung von Architekturklassen zur Datenintegration, die eine Relevanz für die bereits existierenden heterogenen Datenquellen einer Hochschule besitzen. Bei der Integration von Identitätsdaten soll eine hohe Datenqualität erzielt werden. Es ist also notwendig, die Datenfehler zu erkennen und zu bereinigen. In der Arbeit wurden die Fehler zunächst klassifiziert. Bei den Verfahren zu Fehlererkennung fand eine besondere Betrachtung der Verfahren zur Duplikaterkennung statt, die anwendbar auf die Identitäten einer Hochschule sind.

Die Behandlung der Themen Authentifizierung und Autorisierung machte deutlich, dass beide in einem Prozess miteinander verbunden sind. Zunächst muss eine Authentifizierung stattfinden, die auf den beschriebenen unterschiedlichen Verfahren beruhen kann. Erst nach der Authentifizierung kann eine Autorisierung erfolgen. Bei der Autorisierung werden verschiedene Modellansätze verfolgt. Mit der Role Based Access Control wurde unter anderem das für Identitätsmanagementsysteme populärste Modell vorgestellt.

Nach der Unterscheidung zwischen zentralisierten und föderierten Architekturansätzen für Identitätsmanagementsysteme, wurden im weiteren Verlauf der Arbeit die Konzepte serviceorientierter Architekturen vorgestellt. Eng verbunden mit dem Thema serviceorientierte Architekturen sind die diversen beschriebenen Standards wie SAML und SPML. Die Initiativen im Bereich Identitätsmanagementsysteme verwenden einige dieser Standards. Eine für deutsche Hochschulen wichtige Initiative stellt dabei die Authentifizierung- und Autorisierungs-Infrastruktur des Deutsches Forschungsnetz e.V. (DFN-AAI) dar, die die auf SAML basierende Software Shibboleth verwendet.

Neben der Betrachtung der ausgewählten Theorien und Technologien im Umfeld von Identitätsmanagementsystemen war es auch wichtig, die technischen, organisatorischen, rechtlichen und hochschulpolitischen Rahmenbedingungen herauszuarbeiten, die für deutsche Hochschulen gelten. Den Schwerpunkt bildeten dabei die rechtlichen Rahmenbedingungen. Hierbei wurden die Datenschutzgesetze näher untersucht. Die Betrachtung der Hochschulgesetze machte zudem deutlich, welche Personengruppen als Mitglieder einer Hochschule eingestuft werden.

Im Lösungsteil der Arbeit wurde dann die Architektur eines Identitätsmanagementsystems für eine Hochschule entwickelt. Dabei entstand ein Rahmenwerk, das mehrere Architekturklassen für ein Identitätsmanagementsystem enthält. Mit der Geschäftsprozess-Architektur erfolgte die Beschreibung eines Verfahrens zur Erfassung von Geschäftsprozessen an einer Hochschule. Dabei wurde deutlich, dass es notwendig ist, die erfassten Prozesse zu bewerten und gegebenenfalls neu zu gestalten.

Den Schwerpunkt der dargestellten Architekturklassen bildete die Informationssystem-Architektur. Nach dem Konzept serviceorientierter Architekturen wurden die an einer Hochschule existierenden Datenbanken für Identitäten mit Hilfe eines einheitlichen Identity Services und eines einheitlichen Provisioning Services zugänglich gemacht. Existierende Schnittstellen und Verbindungen zwischen den Datenbanken blieben dabei zunächst unberührt. Um eine mehrfache manuelle Erfassung von Identitätsdaten zu vermeiden, erfolgte eine Festlegung der Datenquellen, die für die Pflege einer oder mehrer Gruppen von Identitäten verantwortlich sind. Diese so genannten autoritativen Datenquellen wurden über die neuen Services mit den anderen IT-Systemen gekoppelt, in denen dadurch eine erneute manuelle Erfassung von Identitäten entfallen konnte. Mit der Einführung eines zentralen Identitätsspeichers erhielt man dann eine Komponente, mit der eine integrierte Sicht auf die häufig verwendeten Identitätsarten hergestellt wird. Es wurde aber auch gezeigt, das spezielle selten verwendete Identitätstypen nicht in dem so genannten Main Identity Store gespeichert werden müssen, sondern über die serviceorientierte Architektur direkt zwischen zwei IT-Systemen ausgetauscht werden können. So gewinnt man ein hohes Maß an Flexibilität. Nach der Einführung des zentralen Identitätsspeichers konnten alte proprietäre Verbindungen zwischen vorhandenen IT-Systemen nach und nach aufgehoben werden. Sämtliche Systeme waren nach der Überführung in die serviceorientierte Architektur über einheitliche Services erreichbar. Neu hinzukommende Systeme lassen sich ohne Problem in die die Architektur integrieren. Außerdem können Komponenten die vorhandenen Services nutzen, um durch Kompositionen neue Services bereitzustellen.

Für die Komponenten, die für die neue serviceorientierte Architektur benötigt werden, erfolgte im Anschluss eine genaue Angabe der Verantwortlichkeiten sowie ein Entwurf. Für den Main Identity Store wurde ein relationales Schema entwickelt, das im Kern die Identitäten zu Personen und Benutzerkonten enthält. Diese beiden Identitätstypen erhielten mit der Zuweisung zu hierarchisch angeordneten Organisations- und Systemrollen zusätzlich Autorisierungsinformationen. Zur Realisierung der einheitlichen Services innerhalb der serviceorientierten Architektur wurde ein sehr umfangreicher Identity und Provisioning Service Provider entworfen. Dieser greift auf das vorgestellte Konzept mediatorbasierter Datenbanksysteme zurück. So ermöglicht ein Wrapper den einheitlichen Zugriff für alle Komponenten innerhalb des Identity and Provisioning Service auf die angeschlossene Datenbank. Neben den Wrappern wurden verschiedene Mediatoren dargestellt, die unter anderem eine Fehlerbereinigung, Duplikaterkennung und Filterung von Identitäten zur Einhaltung der Datenschutzgesetze durchführen.

Für die beschriebenen Komponenten wurden am Ende der Arbeit innerhalb der Softwareprodukt-Architektur einige Softwareprodukte vorgestellt, mit denen eine Realisierung der einzelnen Komponenten der Informationssystem-Architektur möglich ist.

Eine konkrete Umsetzung der genannten Architekturklassen muss noch erfolgen. Die Geschäftsprozess-Architektur kann an jeder Hochschule sehr individuell ausfallen. Dabei erhält die Hochschule aber die Möglichkeit, eine Übersicht über alle Geschäftsprozesse zu erhalten und kann diese leistungs- und kostenoptimierend anpassen. Dies ist insofern wich-

tig, als die deutschen Hochschulen ständig mit Kürzungen ihrer finanziellen Mittel konfrontiert werden. Die vorhandenen Mittel sollten dann nicht für ineffiziente Verwaltungsprozesse verwendet werden, sondern der eigentlichen Forschung und Lehre zur Verfügung stehen.

Für die Informationssystem-Architektur fehlt noch eine Implementierung der einzelnen Komponenten, wie beispielsweise für den Main Identity Store und den Identity and Provisioning Service Provider. An der Freien Universität Berlin wird im Rahmen des Projektes FUDIS (FU Directory and Identity Service) der Versuch unternommen, die dargestellte Architektur vollständig umzusetzen. Dabei müssen für die Integration der bestehenden IT-Systeme in das Identitätsmanagementsystem nicht nur technische Hürden überwunden werden. Insbesondere müssen einzelne Bereiche davon überzeugt werden, dass eine solche flexible Architektur zu einer Effizienzsteigerung führt und die immer schneller umzusetzenden neuen IT-Systeme dadurch einfach integrierbar sind.

Die bundesweiten Arbeitskreise und Initiativen zeigen, dass ein hoher Bedarf nach Identitätsmanagementsystemen an deutschen Hochschulen existiert und dass bei der Realisierung immer wieder die gleichen oder ähnliche Probleme auftreten. Eine mögliche Lösung für viele dieser Probleme stellt die in dieser Arbeit dargestellte Architektur dar.

6 Abkürzungsverzeichnis

ACL	Access Control List
ARP	Attribute Release Policy
ASAP	Asynchronous Service Access Protocol
BPEL	Business Process Execution Language
BPML	Business Process Modelling Language
DAC	Discretionary Access Control
DAP	Directory Access Protocol
DOS	Denial-Of-Service
ERD	Entity-Relationship-Diagramm
FDBS	Föderiertes Datenbanksystem
IAM	Identity and Access Management
IDM	Identitätsmanagement
IDMS	Identitätsmanagementsystem
IPSP	Identity and Provisioning Service Provider
ISP	Identity Service Provider
JDBC	Java Database Connectivity
LDAP	Lightweight Directory Access Protocol
MAC	Mandatory Access Control
ODBC	Open Database Connectivity
PSP	Provisioning Service Provider
RBAC	Role-Based Access Control
RPC	Remote Procedure Call
SAML	Security Assertion Markup Languague
SOA	Serviceorientierte Architekturen
SoD	Separation of Duty
SP	Service Provider
SPML	Service Provisioning Markup Languague
SSL	Secure Sockets Layer
SSO	Single-Sign-On
TFIDF	Term-Frequency/Inverse-Document-Freqeuncy
TLS	Transport Layer Security
UDDI	Universal Description, Discovery and Integration
Wf-XML	Workflow Extensible Markup Languague
WSDL	Web Services Description Language
XACML	eXtensible Access Control Markup Language
XPDL	XML Process Definition Language

7 Literaturverzeichnis

[ANSI75] ANSI / X3 / SPARC Study Group on Data Base Management Systems (1975):
Interim Report 75-02-08.
FDT (Bulletin of ACMSIGMOD), Jahrgang 7, Heft 2

[Apache07a] The Apache Software Foundation (2007):
Welcome to Apache Axis2/Java.
URL: http://ws.apache.org/axis2/
[Abfrage: 02.04.2007]

[Apache07b] The Apache Software Foundation (2007):
Apache Tomcat.
URL: http://tomcat.apache.org/
[Abfrage: 02.04.2007]

[Apache07c] The Apache Software Foundation (2007):
Welcome to Lucene!
URL: http://lucene.apache.org/
[Abfrage: 02.04.2007]

[BaGü04] Bauer, Andreas / Günzel, Holger (2004):
Data-Warehouse-Systeme – Architektur – Entwicklung – Anwendung.
2., überarbeitete und aktualisierte Auflage, Heidelberg: dpunkt.verlag

[Bish03] Bishop, Matt (2005):
Computer Security- Art and Science.
Boston u. a.: Addison-Wesley

[BMBF05] BMBF – Bundesministerium für Bildung und Forschung (2005):
Stand der Einführung von Bachelor- und Master-Studiengängen in Bologna-Prozess sowie in ausgewählten Ländern Europas im Vergleich zu Deutschland.
URL: http://www.bmbf.de/pub/bachelor_u_master_im_bolognaprozess_in_eu.pdf
[Stand: 28.02.2005, Abfrage: 07.01.2007]

[Camp04] Camp, L. Jean (2004):
Digital Identity.
URL: http://social.cs.uiuc.edu/class/papers/digital_identity.pdf
[Stand: 29.09.2004, Abfrage: 12.12.2006]

[Cant05] Cantor, Scott (2005):
Shibboleth Architecture - Protocols and Profiles.
URL: http://shibboleth.internet2.edu/docs/internet2-mace-shibboleth-arch-protocols-latest.pdf
[Stand: 10.09.2005, Abfrage: 07.01.2007]

[Codd80] Codd, E. F. (1980):
Data models in database management.
International Conference on Management of Data, Proceedings of the 1980 workshop on Data abstraction, databases and conceptual modeling table of contents, Pingree Park, Colorado, United States, Seite 112 – 114

[DFN07] Deutsches Forschungsnetz (2007)
DFN-AAI - Authentifikation Autorisierungs Infrastruktur.
URL: http://www.aai.dfn.de/
[Abfrage: 29.04.2007]

[Diam06] Diamelle Technologies (2006):
Identity and Access Management Architecture - Diamelle White Paper.
URL: https://openiam.dev.java.net/files/documents/5435/37818/Diamelle_IDM_Architecture.pdf
[Stand: 14.07.2006, Abfrage: 12.12.2006]

[DoD85] Department of Defense (1985):
Department of Defense Trusted Computer System Evaluation Criteria - DoD 5200.28-STD.
URL: http://csrc.nist.gov/publications/history/dod85.pdf
[Stand: 26.12.1985, Abfrage: 26.02.2007]

[ERS99] Elmagarmid, Ahmed / Rusinkiewicz, Marek / Sheth, Amit (1992):
Management of Heterogeneous and Autonomous Database Systems.
San Francisco (Ca): Morgan Kaufmann Publishers

[Ferb03] Ferber, Reginald (2003):
Information Retrieval - Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web.
Heidelberg: dpunkt.verlag

[FKC03] Ferraiolo, David F. / Kuhn, D. Richard / Chandramouli, Ramaswamy (2003):
Role-Based Access Control.
Boston u. a.: Artech House

[IBM05] IBM - International Business Machines Corporation (2005):
IBM Tivoli Identity Manager - Automatisches Lebenszyklusmanagement für Benutzeridentitäten.
URL: ftp://ftp.software.ibm.com/software/emea/de/tivoli/Tivoli-Identity-Manager.pdf
[Stand: 11.01.2005, Abfrage: 21.11.2006]

[IBM06] IBM - International Business Machines Corporation (2006):
Extend the value of System z investments through federated identity management solutions from Tivoli software - Security management solutions - White paper.
URL: ftp://ftp.software.ibm.com/software/tivoli/whitepapers/wp-fimz.pdf
[Stand: 21.07.2006, Abfrage: 21.11.2006]

[Internet2_07a] Internet2 (2007):
Shibboleth.
URL: http://shibboleth.internet2.edu/
[Abfrage: 29.04.2007]

[Internet2_07b] Internet2 (2007):
Grouper.
URL: http://middleware.internet2.edu/dir/groups/grouper/
[Abfrage: 02.04.2007]

[Internet2_07c] Internet2 (2007):
OpenSAML - an Open Source SecurityAssertion Markup Language implementation.
URL: http://www.opensaml.org/
[Abfrage: 02.04.2007]

[KiCa04] Kimball, Ralph / Caserta, Joe (2004):
The Data Warehouse ETL Toolkit – Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data.
Indianapolis (IN): Wiley Publishing, Inc.

[KlLa03] Klünter, Dieter / Laser, Jochen (2003):
LDAP verstehen, OpenLDAP einsetzen.
Heidelberg: dpunkt.verlag

[Kris92] Krishna, S. (1992):
Introduction to database and knowledge-base systems.
Singapore u. a.: World Scientific Publishing

[Lee03] Lee, Spencer C. (2003):
An Introduction to Identity Management.
URL: http://www.sans.org/reading_room/whitepapers/authentication/852.php
[Stand: 28.04.2003, Abfrage: 12.12.2006]

[Lehn03] Lehner, Wolfgang (2003):
Datenbanktechnologien für Data-Warehouse-Systeme – Konzepte und Methoden.
Heidelberg: dpunkt.Verlag

[LeNa07] Leser, Ulf / Naumann, Felix (2007):
Informationsintegration – Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen.
Heidelberg: dpunkt.Verlag

[Liberty07] Liberty Alliance Project (2007):
The Liberty Alliance.
URL: http://www.projectliberty.org/
[Abfrage: 02.04.2007]

[LiUn06] Liebel, Oliver / Ungar, John Martin (2006):
OpenLDAP.
Bonn: Galileo Press

[MACE07] Middleware Architecture Committee for Education (2007):
Middleware Architecture Committee for Education.
URL: http://middleware.internet2.edu/MACE/
[Abfrage: 02.04.2007]

[Microsoft07] Microsoft Corporation (2007):
Windows Server 2003 Active Directory.
URL: http://www.microsoft.com/windowsserver2003/technologies/directory/
activedirectory/default.mspx
[Abfrage: 29.04.2007]

[MIT07] Massachusetts Institute of Technology (2007):
Kerberos: The Network Authentication Protocol.
URL: http://web.mit.edu/kerberos/www/
[Abfrage: 03.04.2007]

[MüFr05] Müller, Heiko / Freytag, Johann-Christoph (2005):
Problems, Methods, and Challenges in Comprehensive Data Cleansing.
URL: http://www.dbis.informatik.hu-berlin.de/fileadmin/research/papers/techreports/2003-hub_ib_164-mueller.pdf
[Stand: 02.03.2005, Abfrage: 12.01.2007]

[Neum05] Neuman, C. u.a. (2005):
The Kerberos Network Authentication Service (V5).
Request for Comments (RFC) 4120
URL: http://www.ietf.org/rfc/rfc4120.txt
[Abfrage: 07.01.2007]

[Newc67] Newcombe, Howard B. (1967):
Record Linking: The Design of Efficient Systems for Linking Records into Individual and Family histories.
American Journal of Human Genetics, Jahrgang 19, Heft 3, Seite 335-359

[OASIS03a] OASIS - Organization for the Advancement of Structured Information Standards (2003):
Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1.
URL: http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf
[Stand: 02.07.2003, Abfrage: 11.01.2007]

[OASIS03b] OASIS (2003):
Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1.
URL: http://www.oasis-open.org/committees/download.php/3405/oasis-sstc-saml-bindings-1.1.pdf
[Stand: 02.07.2003, Abfrage: 11.01.2007]

[OASIS03c] OASIS (2003):
Service Provisioning Markup Language (SPML) Version 1.0.
URL: http://www.oasis-open.org/committees/download.php/4137/os-pstc-spml-core-1.0.pdf
[Stand: 05.11.2003, Abfrage: 11.01.2007]

[OASIS03d] OASIS (2003):
Bindings for the Service Provisioning Markup Language (SPML) Version 1.0.
URL: http://www.oasis-open.org/committees/download.php/4136/os-pstc-spml-bindings-1.0.pdf
[Stand: 05.11.2003, Abfrage: 11.01.2007]

[OASIS05a] OASIS (2005):
Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0.
URL: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
[Stand: 15.03.2005, Abfrage: 11.01.2007]

[OASIS05b] OASIS (2005):
Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0.
URL: http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf.
[Stand: 15.03.2005, Abfrage: 11.01.2007]

[OASIS05c] OASIS (2005):
Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0.
URL: http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
[Stand: 15.03.2005, Abfrage: 11.01.2007]

[OASIS05d] OASIS (2005):
Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0.
URL: http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf
[Stand: 15.03.2005, Abfrage: 11.01.2007]

[OASIS05e] OASIS (2005):
Security Assertion Markup Language (SAML) 2.0 Technical Overview - Working Draft 03.
URL: http://www.oasis-open.org/committees/download.php/11511/sstc-saml-tech-overview-2.0-draft-03.pdf
[Stand: 20.02.2005, Abfrage: 11.01.2007]

[OASIS06a] OASIS (2006):
Reference Model for Service Oriented Architecture 1.0.
URL: http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf
[Stand: 02.08.2006, Abfrage: 11.01.2007]

[OASIS06b] OASIS (2006):
OASIS Service Provisioning Markup Language (SPML) Version 2.
URL: http://www.oasis-open.org/committees/download.php/17708/pstc-spml-2.0-os.zip
[Stand: 17.04.2006, Abfrage: 11.01.2007]

[OASIS06c] OASIS (2006):
OASIS Service Provisioning Markup Language (SPML) v2 - DSML v2 Profile.
URL: http://www.oasis-open.org/committees/download.php/17708/pstc-spml-2.0-os.zip
[Stand: 17.04.2006, Abfrage: 11.01.2007]

[OASIS06d] OASIS (2006):
OASIS Service Provisioning Markup Language (SPML) v2 - XSD Profile.
URL: http://www.oasis-open.org/committees/download.php/17708/pstc-spml-2.0-os.zip
[Stand: 17.04.2006, Abfrage: 11.01.2007]

[OASIS06e] OASIS (2006):
OASIS Service Provisioning Markup Language (SPML) v2 - SAML 2.0 Profile.
URL: http://www.oasis-open.org/archives/provision/200608/doc00000.doc
[Stand: 01.04.2006, Abfrage: 29.04.2007]

[OASIS07] OASIS (2007):
OASIS eXtensible Access Control Markup Language (XACML) TC.
URL: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
[Abfrage: 06.04.2007]

[OIO07] OIO - Orientation in Objects (2005):
Die WS- Spezifikationen.*
URL: http://www.oio.de/public/xml/web-service-specifications.pdf
[Stand: 03.05.2007, Abfrage: 07.05.2007]

[OpenLDAP07] OpenLDAP (2007):
OpenLDAP - community developed LDAP software.
URL: http://www.openldap.org/
[Abfrage: 29.04.2007]

[OpenSPML07] OpenSPML (2007):
OpenSPML.org.
URL: http://www.openspml.org/
[Abfrage: 29.04.2007]

[Orac06] Oracle (2006):
Oracle Identity Manager - An Oracle White Paper.
URL: http://www.oracle.com/technology/products/id_mgmt/oxp/pdf/identity_manager_wp_10gr3.pdf
[Stand: 26.06.2006, Abfrage: 17.11.2006]

[Panc99] Panchapagesan, Bhujanga u.a. (1999):
The INEEL Data Integration Mediation System.
URL: http://infolab.stanford.edu/LIC/ERIS/erisii.ps
[Abfrage: 27.12.2006]

[Post69] Hans Joachim Postel (1969):
Die Kölner Phonetik - Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse.
IBM-Nachrichten, Jahrgang 19, S. 925-931

[PostgreSQL07] PostgreSQL (2007):
PostgreSQL – The world's most advanced open source database.
URL: http://www.postgresql.org/
[Abfrage: 29.04.2007]

[RaDo00] Rahm, Erhard / Do, Hong-Hai (2000):
Data Cleaning: Problems and Current Approaches.
IEEE Bulletin of the Technical Committee on Data Engineering, Jahrgang 23, Heft 4

[SaGi84] Salton, Gerard / McGill, Michael J. (1984):
Introduction to modern information retrieval.
2. Auflage, Auckland u. a.: McGraw-Hill International

[ScCa05] Scavo, Tom / Cantor, Scott (2005):
Shibboleth Architecture - Technical Overview.
URL: http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf
[Stand: 08.06.2005, Abfrage: 07.01.2007]

[Sun06] Sun Microsystems (2006):
Sun Java System Identity Manager.
URL: http://www.sun.com/software/products/identity_mgr/ds_identity_mgr.pdf
[Stand: 17.08.2006, Abfrage: 17.11.2006]

[Voss00] Vossen, Gottfried (2000):
Datenmodelle, Datenbanksprachen und Datenbankmanagement-Systeme.
4., korrigierte u. ergänzte Auflage, München, Wien: R. Oldenbourg Verlag

[VSV06] Valkenburg, Peter / Stals, Bert / van Vooren, Thomas (2006):
Federated Identity Management in Higher Education - Scenarios, services and solutions.
URL: http://www.surfnetters.nl/eva/Federated%20Identity%20Management%20in%20Higher%20Education%20-%20V1.0.ENG-1.pdf
[Stand: 02.10.2006, Abfrage: 02.03.2007]

[W3C02a] W3C - World Wide Web Consortium (2002):
XML-Signature Syntax and Processing.
URL: http://www.w3.org/TR/xmldsig-core/
[Stand: 12.02.2002, Abfrage: 11.01.2007]

[W3C02b] W3C (2002):
XML Encryption Syntax and Processing.
URL: http://www.w3.org/TR/xmlenc-core/
[Stand: 10.12.2002, Abfrage: 11.01.2007]

[Wahl97] Wahl, M. u.a. (1997):
Lightweight Directory Access Protocol (v3).
Request for Comments (RFC) 2251
URL: http://www.ietf.org/rfc/rfc2251.txt
[Abfrage: 07.01.2007]

[WFMC07] Workflow Management Coalition (2007):
WFMC.org Homepage.
URL: http://www.wfmc.org/
[Abfrage: 08.02.2007]

[Wied92] Wiederhold, Gio (1992):
Mediators in the Architecture of Future Information Systems.
IEEE Computer, Jahrgang 25, Heft 3, Seite 38-49

[Wind05] Windley, Phillip J. (2005):
Digital Identity.
Beijing u. a.: O'Reilly Media

[Yeon93] Yeong, W. u.a. (1993):
X.500 Lightweight Directory Access Protocol.
Request for Comments (RFC) 1487
URL: http://www.ietf.org/rfc/rfc1487.txt
[Abfrage: 07.01.2007]

[Yeon95] Yeong, W. u.a. (1995):
Lightweight Directory Access Protocol.
Request for Comments (RFC) 1777
URL: http://www.ietf.org/rfc/rfc1777.txt
[Abfrage: 07.01.2007]

8 Verzeichnis der Gesetze, Verordnungen, Richtlinien und Chartas

8.1 Gesetze

BDSG Bundesdatenschutzgesetz in der Fassung der Bekanntmachung vom 14. Januar 2003 (BGBl. I S. 66), zuletzt geändert durch Artikel 1 des Gesetzes vom 22. August 2006 (BGBl. I S. 1970).
URL: http://www.gesetze-im-internet.de/bundesrecht/bdsg_1990/gesamt.pdf
[Abfrage: 05.12.2006]

HRG Hochschulrahmengesetz in der Fassung der Bekanntmachung vom 19. Januar 1999 (BGBl. I S. 18), zuletzt geändert durch Artikel 2 des Gesetzes vom 12. April 2007 (BGBl. I S. 506).
URL: http://www.gesetze-im-internet.de/bundesrecht/hrg/gesamt.pdf
[Abfrage: 29.04.2007]

SigG Gesetz über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz) vom 16. Mai 2001 (BGBl. I S. 876), zuletzt geändert durch Artikel 4 des Gesetzes vom 26. Februar 2007 (BGBl. I S. 179).
URL: http://www.gesetze-im-internet.de/sigg_2001/index.html
[Abfrage: 29.04.2007]

TKG Telekommunikationsgesetz vom 22. Juni 2004 (BGBl. I S. 1190), zuletzt geändert durch Artikel 3 des Gesetzes vom 18. Februar 2007 (BGBl. I S. 106).
URL: http://www.gesetze-im-internet.de/bundesrecht/tkg_2004/gesamt.pdf
[Abfrage: 29.04.2007]

TMG Telemediengesetz vom 26. Februar 2007 (BGBl. I S. 179).
URL: http://www.gesetze-im-internet.de/bundesrecht/tmg/gesamt.pdf
[Abfrage: 29.04.2007]

BlnDSG Berliner Datenschutzgesetz in der Fassung vom 17. Dezember 1990 (GVBl. 1991 S. 16, 54, zuletzt geändert durch Artikel V des Gesetztes vom 2. Oktober 2003 (GVBl. 2003 S. 486).
URL: http://www.datenschutz-berlin.de/recht/bln/blndsg/blndsg_nichtamt.htm
[Abfrage: 05.12.2006]

HG NRW Hochschulgesetz Nordrhein-Westfalen in der Fassung des Gesetzes zur Weiterentwicklung der Hochschulreformen (Hochschulreformweiterentwicklungsgesetz)
- HRWG - vom 30.11.2004 (GV. NRW S. 752).
URL: http://www.innovation.nrw.de/hochschulen_in_nrw/Recht/HG.html
[Abfrage: 05.12.2006]

8.2 Verordnungen

(EG) Nr. 45/2001 (2000) Verordnung (EG) Nr. 45/2001 des Europäischen Parlaments und des Rates vom 18. Dezember 2000 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten durch die Organe und Einrichtungen der Gemeinschaft und zum freien Datenverkehr.
URL: http://www.datenschutz-berlin.de/recht/eu/rv/verord_richtlinie/ds_eu_einrichtung.htm
[Abfrage: 05.12.2006]

SigV Verordnung zur elektronischen Signatur (Signaturverordnung) vom 16. November 2001 (BGBl. I S. 3074), geändert durch Artikel 2 des Gesetzes vom 4. Januar 2005 (BGBl. I S. 2)
URL: http://www.gesetze-im-internet.de/bundesrecht/sigv_2001/gesamt.pdf
[Abfrage: 05.12.2006]

8.3 Richtlinien

[UN1990] Richtlinien betreffend personenbezogene Daten in automatisierten Dateien (von der Generalversammlung der Vereinten Nationen am 14. Dezember 1990 beschlossen).
URL: http://www.datenschutz-berlin.de/recht/int/uno/gl_pbdde.htm
[Abfrage: 05.12.2006]

2000/31/EG (2000) Richtlinie 2000/31/EG des Europäischen Parlaments und des Rates vom 8. Juni 2000 über bestimmte rechtliche Aspekte der Dienste der Informationsgesellschaft, insbesondere des elektronischen Geschäftsverkehrs, im Binnenmarkt ("Richtlinie über den elektronischen Geschäftsverkehr").
URL: http://www.datenschutz-berlin.de/recht/eu/rv/verord_richtlinie/rl_eu_ecomm.htm
[Abfrage: 05.12.2006]

2002/58/EG (2002) Europäische Datenschutzrichtlinie für elektronische Kommunikation vom 12.07.2002, Richtlinie 2002/58/EG des Europäischen Parlaments und des Rates vom 12. Juli 2002 über die Verarbeitung personenbezogener Daten und den Schutz der Privatsphäre in der elektronischen Kommunikation (Datenschutzrichtlinie für elektronische Kommunikation)(ABl. EG L 201 vom 31.07.2002, S. 37).
URL: http://www.datenschutz-berlin.de/recht/eu/rv/tk_med/tkdsr_de.htm
[Abfrage: 05.12.2006]

95/46/EG (1995) I. Richtlinie 95/46/EG des Europäischen Parlaments und des Rates vom 24. Oktober 1995 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr.
URL: http://www.datenschutz-berlin.de/infomat/heft24/dde.htm
[Abfrage: 05.12.2006]

98/0191(COD) (1999) Richtlinie des Europäischen Parlaments und des Rates über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen vom 18.11.1999.
URL: http://www.datenschutz-berlin.de/recht/eu/rv/tk_med/signatur.htm
[Abfrage: 05.12.2006]

8.4 Chartas

2000/C 364/01 (2000) Charta der Grundrechte der europäischen Union v. 07.12.2000.
URL: http://www.datenschutz-berlin.de/recht/eu/ggebung/charta.htm
[Abfrage: 05.12.2006]

9 Abbildungsverzeichnis

10 Tabellenverzeichnis