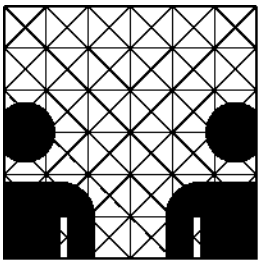

Dissertation zur Erlangung des
Doktorgrades der Naturwissenschaften
vorgelegt am
Fachbereich Informatik der
Universität Hamburg am 23. Juni 2005

Persönliches digitales Identitätsmanagement

Untersuchung und Entwicklung von Konzepten und Systemarchitekturen für die kontrollierte Selbstdarstellung in digitalen Netzen

Tobias Baier geboren am 17.10.1974 in Hamburg

Telefon: +49 40 4291 0502
E-Mail: baier@informatik.uni-hamburg.de

Betreuer: Prof. Dr. W. Lamersdorf

Zusammenfassung

Digitales Identitätsmanagement unterstützt Anwender und Dienste bei der Verwaltung, Kontrolle und Übermittlung persönlicher Daten in digitalen Kommunikationssystemen. Die Vielfalt dieser komplexen und sicherheitsrelevanten Aufgaben ist bisher auf verschiedene Anwendungen und Systeme verteilt. Durch digitales Identitätsmanagement werden diese unterschiedlichen Strukturen in einem gesamtheitlichen Konzept zur systemtechnischen Komponente, die anderen Kommunikationsanwendungen zur Verfügung steht. Eine gemeinsame Bedienschnittstelle hilft bei der Verwaltung der Identitätsinformationen sowie bei der Auswahl und Übertragung der persönlichen Daten.

Bei *persönlichem Identitätsmanagement* liegt die Kontrolle über die eigenen Daten im Gegensatz zum *Identitätsmanagement in Unternehmen* beim Benutzer selbst. Dadurch verändert sich sowohl die Systemarchitektur als auch der mögliche Anwendungskontext. Viele Konzepte und Mechanismen sind jedoch in beiden Arten vorhanden und werden in dieser Arbeit allgemein betrachtet.

Dazu werden zunächst die dem digitalen Identitätsmanagement zugrunde liegenden allgemeinen Konzepte untersucht und erläutert, woraufhin Definitionen von *digitaler Identität*, *Identitätsmanagement* und *Identitätsmanagement-Systemen* entwickelt werden. Dann wird eine Klassifizierung zur Einordnung von Identitätsmanagement-Systemen vorgestellt. Diese unterscheidet sich dadurch von anderen Klassifizierungen solcher Systeme, dass sie die *Intention* und damit die Zielgruppe des Systems berücksichtigt; Der funktionelle Unterschied zwischen unternehmensorientierten und persönlichen Systemen wird so verdeutlicht. Nachdem wichtige Identitätsmanagement-Systeme aus Praxis und Forschung untersucht und nach der Klassifizierung eingeordnet werden, wird eine Lücke dieses Rasters deutlich: Es fehlt ein persönliches, am Anwender orientiertes System, das die Selbstdarstellungsmöglichkeit des Benutzers und nicht den Datenschutz in den Vordergrund stellt. Diese Arbeit entwickelt deshalb ein eigenes Konzept für ein solches System. Dazu wird auch die Konstruktion eines diesen Konzepten folgenden Prototypen vorgestellt und dessen Einsatz in praktischen Beispielen erläutert. Abschließend wird das vorgestellte, prototypische System in das zuvor ausgemachte Umfeld eingeordnet und dessen Einfluss auf mögliche weitere Entwicklungen abgeschätzt.

Abstract

Digital identity management helps users and service providers to manage, control and transmit personal data within digital communication systems. The diversity of these complex tasks that always involve security issues, is currently distributed on many different applications and systems. Digital identity management ties these tasks into one single concept for a system level component which can be utilized by other communication applications. A common user interface helps with the management of the information as well as with the transmission of personal data to communication partners.

In contrast to *organisational*, identity management *personal* identity management puts the user into control of his data. This influences both the system architecture and the possible application context. Nevertheless, both share many concepts and mechanisms which will be considered conjointly in this work.

First the fundamental concepts of digital identity management are explained, which leads to definitions of *identity*, *identity management* and *identity management systems*. Then a classification of identity management systems is introduced. It differs from former approaches by considering the *intention* and thereby the target group of the system. With it, the functional differences between organisational and personal identity management systems become apparent. After applying this classification to important identity management systems which are currently developed and even deployed by commercial industry and research, it becomes clear that there is a gap: there is no personal digital identity management system which concentrates on self-portrayal instead of data security. This leads to this works main goal: the development of a concept for self-portrayal oriented digital identity management. A prototype following this concept is also introduced. Practical examples show how the prototype is used and what difference it makes to existing communication systems. Finally, the system is classified itself and its position in the field is elaborated.

Inhaltsverzeichnis

Abbildungsverzeichnis

Listings

1. Einleitung: Selbstdarstellung und Datenschutz bei digitaler Kommunikation

Anwendern wird im Internet in vielen Bereichen ein durch *Personalisierung* angepasster Dienst oder eine individualisierte Umgebung geboten. *Portale* sind ein Beispiel für Web-Seiten, bei denen Benutzern die Möglichkeit gegeben wird, durch eigene Einstellungen das Erscheinungsbild und sogar die Funktionalität anzupassen. *Online Communities* sind Gesellschaften, die sich über digitale Medien wie Online-Foren und E-Mail-Verteilerlisten organisieren. Viele Dienste wie zum Beispiel das *Einkaufen über das Internet* lassen sich mittels persönlicher Informationen des Benutzers verbessern, indem personalisierte Werbung für bestimmte Produkte platziert wird – bereits bevor die persönlichen Daten zum Abschluss des Kaufvertrags benötigt werden. Andere Dienste, wie zum Beispiel *Online Banking*, sind ohne Informationen über den Benutzer gar nicht möglich, weil Zugriffskontrolle gewährleistet sein muss. Bei *Online Spielen* geht es entweder um Errungenschaften des Anwenders, die im System gespeichert werden (beispielsweise seine Spielstärke im Schach), oder im Falle von Rollenspielen auch um eine fiktive Identität, die entwickelt und gepflegt wird. Personalisierung – die Berücksichtigung der Persönlichkeit des Anwenders – ist also ein wichtiges Thema verteilter Systeme.

1.1. Digitale Identitäten

In personalisierten Umgebungen muss ein Benutzer persönliche Informationen preisgeben. Die dabei entstehenden digitalen Abbilder persönlicher Informationen nennt man *digitale Identitäten* (vgl. Definition 2.1). Sie werden heute zumeist nur auf Seiten des Anbieters gespeichert; Der Nutzen dieser Daten für den Anwender selbst wird dadurch gemindert. Der Eigentümer der Daten verliert zudem die direkte Möglichkeit, selbst über seine persönlichen Daten zu bestimmen und sie zu seinen Zwecken einzusetzen.

1.1.1. Getrennte digitale Identitäten

Sind mehrere digitale Identitäten einer Person über verschiedene Dienste verteilt, so sind sie zwar in den Instanzen *disjunkt*, nicht aber nicht in den Inhalten. Das bedeutet, dass verschiedene digitale Identitäten einer Person durchaus die gleichen Daten beinhalten können (beispielsweise immer gleich bleibende Adressdaten), eine logische Verknüpfung dieser Daten in den verschiedenen Identitäten jedoch nicht möglich ist. Für viele Bereiche ist diese Verknüpfung aus Gründen des Datenschutzes ausdrücklich nicht wünschenswert – der Anwender würde tiefe Einschnitte in die Privatsphäre erleiden, wären alle seine digitalen Identitäten logisch verknüpfbar und dadurch Rückschlüsse auf seine Tätigkeiten und Eigenschaften in anderen Kontexten möglich. Für viele Situationen gilt jedoch, dass die Trennung der digitalen Identitäten für den Anwender von Nachteil ist. In vielen digitalen Identitäten steckt für den Besitzer ein großer Wert. Dieser Wert kann unterschiedlicher Gestalt sein: Beispielsweise kann ein der Identität anhaftender guter Ruf, oder viel Zeit und Arbeit, die in die Entwicklung der Identität geflossen sind, dem Besitzer viel bedeuten. Der Wert kann ideell oder finanziell sein: während die Energie, die in die Entwicklung eines Charakters in einem Rollenspiel geflossen ist, im wesentlichen Zeit kostet und keinen direkten finanziellen Vorteil bietet, ist eine gute Reputation in einem Online-Handelszentrum finanziell wertvoll. Neben der fehlenden direkten Kontrolle über eigene persönliche Daten ist also ein weiterer großer Nachteil, dass der Wert dieser Daten nicht außerhalb des Systems nutzbar ist, an das die digitale Identität gebunden ist.

1.1.2. Unpersönliche Personalisierbarkeit versus „gefühlte“ Anonymität

Bei der Kommunikation über das Internet bestehen zwei Widersprüche: erstens zwischen *Anonymitätsempfinden* und tatsächlicher *Anonymität*, zweitens zwischen *Personalisierbarkeit* und tatsächlich abgebildeten *Persönlichkeiten*. Normalerweise fühlen sich Benutzer des Internets sehr anonym: Gerade in nicht personalisierten Bereichen des WWW, wo für den Benutzer keine sichtbaren Spuren auf besuchten Servern hinterlassen werden und auch keine anderen Benutzer wahrgenommen werden können, kommt man sich allein und unbeobachtet vor. Die meisten Benutzer wissen nicht, das auf dem Server möglicherweise Zugriffe samt der IP-Adressen, über die später die tatsächliche Identität des Benutzers herausgefunden werden könnte, protokolliert werden. Darüber hinaus kann über Cookies (siehe Abschnitt 3.2.3.2) und versteckte Session-IDs (siehe Abschnitt 2.4) ein Benutzerprofil angelegt werden, das Informationen darüber enthält, welche Seiten der Benutzer wie lange besucht hat und wo er zuvor war. Insbesondere kann Verweisen in das WWW, die durch E-Mail-Werbung verteilt werden, ein eindeutiger Identifikator beigefügt werden, so dass Benutzer, die diesen Verweisen folgen, ein-

deutig identifiziert werden können, ohne dass ihnen dies bewusst gemacht werden müsste[1]. Trotz des aktuellen Anonymitätsempfindens können also viele personenbezogene Daten gesammelt werden. Diese mangelhafte Behandlung der Privatsphäre hat auch Auswirkungen auf den Datenschutz. In vielen Situationen muss oder möchte ein Benutzer persönliche Daten über das Internet übertragen, beispielsweise beim Online Banking, Einkaufen im Internet oder Kennenlernen neuer Freunde. Während im ersten Fall die Daten meistens verschlüsselt übertragen werden, ist dies beim Einkaufen nicht immer und bei persönlicher Kommunikation fast nie der Fall. Das mangelnde Wissen der Benutzer über die Struktur und Funktionsweise des Internet bringt sie in die Gefahr, *unwissentlich* unbefugten Personen Einsicht in ihre Daten zu gewähren.

Dagegen kann die Personalisierbarkeit, die im vorigen Abschnitt erwähnt worden ist, mangels der Abbildung einer wirklichen Persönlichkeit durchaus als „unpersönlich“ bezeichnet werden. Eine reale Persönlichkeit kann ihre Eigenschaften und Fähigkeiten in mehreren Kontexten und verschiedenen Situationen einsetzen: das kann eine digitale Identität im heutigen Sinne nicht.

1.1.3. Persönliche digitale Identitäten

Zur besseren Unterstützung persönlicher Kommunikation und Selbstdarstellung fehlt also eine genauere Abbildung des Konzepts der Persönlichkeit in digitalen Medien oder der *digitalen Identität*. Darüber hinaus werden Techniken und Systeme erforderlich, die den Umgang mit diesen digitalen Identitäten ermöglichen oder erleichtern, sowie den praktischen Einsatz in Kommunikationssystemen unterstützen. Die Entwicklung von Identitätsmanagement-Systemen findet sowohl in der Wirtschaft als auch in der wissenschaftlichen Forschung statt. Dabei werden verschiedene Blickwinkel deutlich: die kommerziellen Entwickler sehen Identitätsdaten wie Reputation oder Benutzerinteressen sowie Personalisierung als Potenzial für Kostenersparnis, Umsatzsteigerung und Kundenbindung. Wissenschaftliche Ansätze gehen dagegen mehrheitlich in die Richtung, dass dem Nutzer die Kontrolle über seine Daten gegeben werden kann, damit er einen größeren Nutzen bei gesteigerter Sicherheit aus den Internet-Technologien ziehen kann. Ein allgemeines System, welches alle diese Aspekte abdeckt, könnte das Potenzial haben, zu einer neuen Standard-Technologie im Internet zu avancieren. Ziel dieser Arbeit ist es, die grundlegenden Konzepte des digitalen Identitätsmanagements zu untersuchen und dadurch ein allgemeines Modell für Identitätsmanagement-Systeme zu entwickeln, die bisherige Umsetzung in Identitätsmanagement-Systemen zu kategorisieren und analysieren, sowie eine eigene Architektur zum persönlichen, digitalen Identitätsmanagement zu entwickeln.

[1] Der Verweis kann zum Beispiel durch ein Bild dargestellt werden, welches den eindeutigen Identifikator verbirgt.

1.2. Verschiedene Bereiche des Identitätsmanagements

Identitätsmanagement in digitalen Systemen gibt es in mehreren Bereichen. Die in der Einleitung genannten Punkte betreffen das persönliche Identitätsmanagement, den Schwerpunkt dieser Arbeit. Die Gesamtheit des digitalen Identitätsmanagements lässt sich grob in drei Bereiche einteilen:

1. Beim *Identitätsmanagement in Organisationen* werden innerhalb einer Organisation Identitätsdaten verwaltet. Dies geschieht meistens zentral zum Zwecke der Mitarbeiter- oder Kundenverwaltung.

2. *Föderiertes Identitätsmanagement* erlaubt das Teilen von Identitätsinformationen über Organisationsgrenzen hinweg. Modelle für diese Art des Identitätsmanagements (zum Beispiel Liberty Alliance, Hodges und Wason 2003, siehe dazu auch Abschnitt 7.2) sehen vor, dass Anwender es den föderierten Organisationen erlauben können, Daten über ihn auszutauschen. Das können Authentifizierungsdaten sein, wodurch ein Single Sign-On erreicht wird, oder auch weiterführende persönliche Daten, die eine bessere Personalisierung der angebotenen Dienste erlauben.

3. Beim *persönlichen Identitätsmanagement* steht der Anwender selbst im Zentrum der Kontrolle. Er kann seine Identitätsdaten selbst verwalten und zu unterschiedlichen Zwecken einsetzen.

Eine genauere Klassifizierung, die allerdings die Auswirkung auf persönliche Aspekte betont, wird in Kapitel 6 vorgenommen. In dieser Arbeit steht das persönliche Identitätsmanagement im Vordergrund – die beiden anderen Bereiche werden berührt, jedoch nicht ausführlich behandelt. Insbesondere Identitätsmanagement in Organisationen spielt für persönliches Identitätsmanagement eine untergeordnete Rolle. Die dafür wichtigen Technologien werden im folgenden Abschnitt kurz beschrieben, im weiteren Verlauf der Arbeit jedoch nicht explizit berücksichtigt. Die Konzepte des föderierten Identitätsmanagements werden in den Abschnitten 2.2.2 und 7.2 weitergehend betrachtet.

1.2.1. Identitätsmanagement in Organisationen

Die Verwaltung von Identitätsdaten in Organisationen und Unternehmen hat sich aus den Anfängen der Benutzerverwaltung in Time-Sharing Computersystemen entwickelt. Mit Beginn der Multi-User Systeme wurde eine Abbildung eines Benutzers im Computersystem notwendig, um Aktionen im System einzelnen Personen zuordnen zu können. Zunächst war dabei keine Authentifikation vorgesehen, der „Log In“ entsprach dem „Clock In“ – also dem Betätigen einer Stechuhr – und erzeugte lediglich eine Information über den Beginn der Arbeit eines Benutzers (vgl. Corbate 1962). Die Authentifizierung wurde erst später hinzugefügt. Die

Benutzer mussten sich mittels eines Passwortes dem System gegenüber authentifizieren, also ihre Identität belegen. Dieser Vorgang des Log Ins ist noch heute gebräuchlich. Außer dem Benutzernamen waren hier aber zunächst keine persönlichen Attribute des Benutzers hinterlegt. Dies erfolgte erst im nächsten Schritt, dem Einsatz von Verzeichnissen für Nutzerauthentifizierung und Informationsverwaltung.

Verzeichnisdienste

Verzeichnisse werden immer dann gebraucht, wenn klar strukturierte Datensätze häufig angefragt werden. Eine einfache Form des Verzeichnisses ist der Namensdienst, bei dem kein strukturierter Datensatz sondern lediglich ein Name angefragt werden kann. Ein Beispiel aus der nicht digitalen Welt ist das Telefonbuch: Anhand eines Namens kann man leicht die Telefonnummer herausfinden. Die Daten (Name, optional Adresse, Telefonnummer) sind klar strukturiert, der Zugriff ist einfach und schnell. Elektronische Verzeichnisse (oft auch englisch *Directory* genannt) sind ähnlich aufgebaut. Häufig ist der Zugriff sogar so stark vereinfacht, dass es normale Anwender gar nicht merken, wenn sie auf das Verzeichnis zugreifen. Das beste Beispiel dafür ist das Domain Name System DNS, welches Suchanfragen in Form von Domain-Namen mit deren IP-Adresse beantwortet. Solche Anfragen werden zum Beispiel von Web-Browsern intern genutzt, so dass der Nutzer des Browsers den Verzeichniszugriff als solchen gar nicht wahrnimmt.

Damit bei digitalen Verzeichnissen der Zugriff standardisiert sein kann, ist das Datenschema meistens vorgegeben und nicht dynamisch. Der anfragende Client muss die interne Struktur der Daten nicht kennen, wohl aber die Form, in der sie abrufbar sind, weil es sonst schwierig ist, eine Anfrage überhaupt zu formulieren. Chadwick beschreibt die Problematik wir folgt:

> All good databases have a schema. This is the set of rules which controls all aspects of what can be put into the database. The Directory is no exception to this. Anarchy does not rule OK in the world of databases. Users are not allowed to store whatever they want to in the system or the system would be full of user specific information (not usable by anyone else), and duplicate information (perhaps held in different formats). [...] The disc/memory space would soon be exhausted. (Chadwick 1994)

Bekannteste Beispiele für Verzeichnissysteme sind X.500 und das davon abgeleitete LDAP, auf dem unter anderem auch das Microsoft Active Directory aufsetzt. Sie werden in vielen Unternehmen und Organisationen zur Verwaltung der Mitglieder oder Kunden verwendet und bilden ein gutes Beispiel für Identitätsmanagement in Organisationen. Da dieses Thema in dieser Arbeit eher am

Rande behandelt wird, sollen die beiden Systeme hier kurz beleuchtet werden, damit der Unterschied zum persönlichen Identitätsmanagement deutlich wird.

X.500

Die X.500-Familie ist eine Reihe von ITU[2]-Standardisierungen für Verzeichnisse beliebigen Inhalts, der von der ISO[3] entwickelt worden ist. Es ist eng verknüpft mit dem ISO/OSI-Schichtenmodell (siehe Kerner 1995; Tanenbaum 1996) und bildet ein komplexes und umfassendes Rahmenwerk. Das System unterscheidet zwischen *Directory User Agents* (DUAs) und *Directory System Agents* (DSAs), denen unterschiedliche Protokolle zur Verfügung stehen. DUAs greifen mittels des *Directory Access Protocols* (DAP) auf DSAs zu. Dieses Protokoll umfasst Möglichkeiten zum Lesen, Schreiben (Ändern und Hinzufügen), Löschen und Suchen von Einträgen des Verzeichnisses.

Spezifiziert sind sowohl die Architektur als auch die Protokolle, nicht jedoch das Datenschema der Verzeichnisinhalte. Meistens werden in X.500-Verzeichnissen jedoch Personendaten und ihre Kontaktinformationen hinterlegt (hierfür existieren auch Referenzdatenschemata). Jedes X.500 Verzeichnis hat ein festes Schema.

Gespeichert werden die Daten bei X.500 im *Directory Information Tree* (DIT), einer baumförmigen Struktur, die dem Schema folgt. Die Gesamtheit der Daten in diesem Baum heißt *Directory Information Base* (DIB). Alle Datensätze einer DIB folgen einem gemeinsamen Schema. Dieses ist zwar erweiterbar, aber redundante Datendefinitionen sollten vermieden werden. Daher ist auch eine Übersetzung zwischen äquivalenten Schemata nicht vorgesehen. X.500 eignet sich daher nicht für verteilte, heterogene Datenschemata.

LDAP

Das *Lightweight Directory Access Protocol* LDAP (eine Sammlung der Definitionen von LDAP findet sich in Loshin und McCarthy 2000) wurde als schlanker Nachfolger des mächtigeren X.500-Protokolls DAP entworfen. Es lässt selten genutzte Elemente des Protokolls aus und macht damit die Entwicklung von effizienteren Clients und Servern möglich. LDAP hat sich aufgrund der einfacheren Technologie schnell weit verbreitet und ist in der quelloffenen Implementierung „OpenLDAP“[4] kostenlos erhältlich.

Der Hauptunterschied zu X.500 ist die Loslösung vom OSI-Schichtenmodell: LDAP baut direkt auf dem viel weiter verbreiteten TCP/IP-Stack auf. Ursprünglich waren LDAP-Server nur als Schnittstelle zwischen LDAP-Clients, die den

[2] ITU: International Telecommunication Union, zuvor bekannt als CCITT (Comité Consultatif International Téléphonique et Télégraphique)
[3] International Organization for Standardization
[4] `http://www.openldap.org`

LDAP-Server über TCP/IP ansprechen konnten, und OSI-basierten X.500-Servern gedacht. Mittlerweile lassen sich jedoch beliebige Datenquellen hinter LDAP-Server einhängen, die nicht OSI-basiert sein müssen. Es muss allerdings das in X.500 definierte Datenmodell befolgt werden.

Einen großen Schub in der Verbreitung von LDAP gab es durch den Einsatz solcher Verzeichnisse zum Verwalten von Netzwerkinformationen. Dazu gehören nicht nur Benutzer- und Gruppeninformationen, sondern auch IP-Dienste (bei Unix-Systemen in der Datei `/etc/services` notiert), Einhängepunkte für Dateisysteme (`/etc/fstab`) oder Boot-Informationen (siehe Howard 1998). Kombiniert mit der Möglichkeit, personenbezogene Daten im Verzeichnis zu speichern, ergibt sich daraus ein mächtiges Identitätsmanagement-System, welches vor allem innerhalb von Organisationen eingesetzt wird. Aufgrund dieser Möglichkeiten wird LDAP mittlerweile in vielen lokalen Netzen als Single-Sign-On-Lösung verwendet.

Defizite von Verzeichnisdiensten für persönliches Identitätsmanagement

Sowohl LDAP als auch das mächtigere X.500 bieten gute Grundlagen für Identitätsmanagement in Organisationen. Die mangelnde Flexibilität der Schemata und die Ausrichtung auf zentrale Datenbasen widersprechen jedoch der Anforderung an persönliches Identitätsmanagement, dem Benutzer größtmögliche Kontrolle zu überlassen – und dabei ist die Kontrolle des Speicherortes und des Datenschemas inbegriffen.

Darüber hinaus sind Verzeichnisdienste im Wesentlichen auf viele lesende und wenige schreibende Zugriffe ausgelegt. Beim persönlichen Identitätsmanagement kann es jedoch zu sehr vielen schreibenden Zugriffen kommen, wenn sehr dynamische oder kurzlebige Daten verwaltet werden. Durch viele schreibende Zugriffe werden die Zwischenspeicher (*Caches*) der Verzeichnisse schnell ungültig und behindern das schnelle und zuverlässige Funktionieren eher als dass es gefördert wird.

Daraus folgt, dass für persönliches Identitätsmanagement andere Formen der Identitätsdatenverwaltung notwendig sind. Ebenso ist fraglich, ob die bestehenden Kommunikationsprotokolle für den Einsatz im privaten, dezentralen Bereich geeignet sind.

1.3. Motivation für persönliches, digitales Identitätsmanagement

Das Verwalten, Pflegen, Auswählen und Übertragen digitaler Identitätsdaten ist nur selten ein Selbstzweck. Es erfolgt meistens in einem Kontext, der diese Aktivitäten erfordert, oder in dem die Aktivität einen Vorteil bringt. Viele der Aspekte

des Identitätsmanagement werden schon intensiv umgesetzt, beispielsweise die Übertragung von Adressdaten an Kommunikationspartner im Internet beim Online Shopping. Der Einsatz eines technischen Systems zur Unterstützung dieser Aufgaben bringt jedoch große Erleichterung und sogar neue Möglichkeiten. In diesem Abschnitt sollen *Beweggründe zum Einsatz eines persönlichen Identitätsmanagement-Systems* herausgestellt werden. Grob lassen sich diese Gründe in zwei Kategorien fassen: Zum einen gibt es den Wunsch nach Selbstdarstellung und Entwicklung von Online-Identitäten mit Wiedererkennungswert, Reputation und automatischer Personalisierung. Zum anderen besteht die Notwendigkeit zum Schutz der persönlichen Daten, die über das digitale Medium verbreitet werden. Während also einerseits persönliche Daten veröffentlicht werden sollen, soll andererseits darauf geachtet werden, dass dies nur kontrolliert geschieht. Diese Zweischneidigkeit entspricht dem Widerspruch des Anonymitätsempfindens, der in der Einleitung erklärt wurde.

Die eigentliche Motivation für digitales Identitätsmanagement besteht also in den Anwendungen, die darauf aufbauend entwickelt und eingesetzt werden, die Verbesserung schon bestehender Anwendungen durch Identitätsmanagement, sowie in den gesellschaftlichen Auswirkungen, die sich durch den Einsatz solcher Identitätsdaten-angereicherten Anwendungen ergeben könnten. Die Infrastruktur allein hat noch keine dem Benutzer oder dem Gesamtsystem hilfreiche Funktion. Lediglich vom Schutz-Effekt, der sich durch die Integration von Sicherheitsmechanismen in ein Identitätsmanagement-System erreichen ließe, könnte ohne weitere Anwendungen profitiert werden.

Im folgenden werden einige der wesentlichen konkreten Motivationen für persönliches Identitätsmanagement in digitalen Netzen angeführt. Sie stellen heraus, dass die Motivation nicht im Datenschutz oder dem Schutz der Privatsphäre liegt, sondern in der Erleichterung der Selbstdarstellung, welche in der Folge Datenschutz notwendig macht.

1.3.1. Bequemlichkeit / Convenience

Der Automatisierungsaspekt bei digitalen Identitätsmanagement-Systemen erleichtert die Nutzung von Diensten, die entweder Autorisierung oder Personalisierung erfordern. Dadurch, dass sich der Benutzer gegenüber dem Identitätsmanagement-System authentifiziert hat, sind ihm alle Authentifizierungen zugänglich, die einer seiner dort gespeicherten Teil-Identitäten zugewiesen sind. Dies befreit den Benutzer davon, sich für verschiedene Dienste mehrere, möglicherweise verschiedene Benutzerkennungen und Passwörter zu merken. Man nennt diesen Mechanismus „Single Sign-On“ (siehe auch Abschnitt 5.1), und er kann auf verschiedene Weisen realisiert werden.

Personalisierung ist das Anpassen einer Oberfläche oder einer Struktur an die Möglichkeiten oder Wünsche eines bestimmten Benutzers. Je nach Benutzer kann

also ein Dienst oder eine Schnittstelle unterschiedlich gestaltet sein. Um dem System mitzuteilen, in welcher Art man Personalisierung wünscht, muss man bestimmte Attribute übermitteln (z.B. gewünschtes Farbschema einer Web-Seite, bevorzugte Literaturrichtung bei einem Online-Buchladen, vereinfachte oder vergrößerte Darstellung für Menschen mit Sehschwäche).

1.3.2. Vertrauen und Reputation

Für viele Anwendungen und Kommunikationssituationen im Internet ist es unabdingbar, dem Kommunikationspartner zu vertrauen. Dazu gehören Handelsabwicklungen des E-Commerce genau so wie private Situationen, in denen sensible Daten (beispielsweise die Adresse) ausgetauscht werden oder professionelle Beratung wie ärztliche Betreuung oder Beratung durch einen Rechtsanwalt. Das Vertrauen, dass man sich gegenüberbringt, hängt im Internet zu einem großen Teil vom Ruf oder der Reputation des Individuums oder des Dienstanbieters ab (vgl. Fahrenholtz und Bartelt 2001) . Hat der andere einen guten Ruf, so fällt die Vertrauensentscheidung in den meisten Fällen positiv aus. Ein Ruf oder eine Reputation kann jedoch nicht einer anonymen Quelle oder einem unbekannten Pseudonym anhängen, sondern muss prinzipiell eindeutig zugeordnet werden können. Auch dazu sind digitale Identitäten notwendig. Ohne sie kann keine Reputation aufgebaut werden, um Vertrauensentscheidungen zu unterstützen. Das Vertrauen kann durch weitere Faktoren wie Verschlüsselung und Signierung unterstützt werden. So muss auch die Identitätsbehauptung, aus der die Reputationsannahme resultiert, durch eine Signatur belegt werden. Ohne digitale Identitäten lässt sich eine solche Behauptung jedoch gar nicht realisieren.

1.3.3. Wiedererkennbare Online-Persönlichkeiten

Die komplexeste Motivation für den Einsatz von digitalen Identitätsmanagement-Systemen ist nicht die Vereinfachung von schon bestehenden und verwendeten Mechanismen wie Schutz oder Single Sign-On, sondern die Möglichkeit neuer Funktionalität im Internet, nämlich das Erschaffen von Online-Persönlichkeiten und Wiedererkennbarkeit dieser Persönlichkeiten über Grenzen von beschränkten Anwendungen und Diensten hinweg. Die derzeitige automatische und unvermeidliche Pseudonymisierung von Teil-Identitäten durch die serverseitige Trennung von Benutzerdaten bewirkt, dass erarbeitete Persönlichkeiten in keinem anderen als dem Ursprungskontext eingesetzt werden können. Darüber hinaus ist es nahezu unmöglich, Kontakte, die man im einen Kontext aufgebaut hat, in einem anderen Kontext überhaupt wiederzuerkennen, auch wenn beide Kommunikationspartner dies wünschen würden. Dieser Negativ-Effekt von Datenschutz und erzwungener (aber für die Dienstanbieter unwirksamer) Privatsphäre ist für die Entwicklung einer Internet-Gesellschaft ein großes Hindernis. Als McLuhan 1964

davon sprach, dass durch neue, elektrische Medien ein „Dorf“ den ganzen Globus umspannen wird, hat er sicherlich noch nicht die heutigen Ausmaße des Internets erahnt, jedoch die Geschwindigkeit, mit der man heutzutage Nachrichten austauschen kann, richtig antizipiert. Zum Gestalten eines Dorfes gehört jedoch eben nicht die gesichtslose Kommunikation, wie sie heute oft im Internet praktiziert wird, sondern es ist ein gut bekannter Kommunikationspartner vonnöten. Bei der Benutzung von E-Mail ist es noch so, dass anhand der E-Mail-Adresse oder des entsprechenden Eintrags im eigenen Adressbuch auf die Identität der Gegenseite geschlossen werden kann – es sei denn, der Kommunikationspartner besitzt mehrere E-Mail-Adressen, von denen nicht alle bekannt sind. Kommunikation über digitale Netze hat sich jedoch so stark diversifiziert, dass die Identitäten, welche bei der Benutzung von E-Mail kaum zu verstecken sind, nicht in jedem Kontext anwendbar sind, wodurch den „Bewohnern des globalen Dorfes“ oft Masken aufgesetzt werden, die niemand abstreifen kann. Diese Masken abzusetzen und sich seinen Bekannten zeigen zu können ist eines der Ziele, die mit digitalem Identitätsmanagement erreicht werden können. Nur durch den Einsatz wiedererkennbarer Identitäten können sich im Internet bessere Abbildungen des Konzepts „Gesellschaft“ entwickeln.

1.3.4. Online-Communities

Eine der bekanntesten und am besten untersuchten Formen der Abbildung des Konzeptes „Gesellschaft“ in der digitalen Welt ist das der *Online-Communities* (vgl. Donath 1997; Koch und Wörndl 2001; Koch 2001; Schlichter u. a. 1998; Wellman und Gulia 1999). Online-Communities sind Zusammenschlüsse von Internet-Benutzern, die verschiedene Kommunikationsmechanismen benutzen können, um die Community zu gestalten. Waren in den Anfangstagen des Internets noch die USENET Newsgroups der am weitesten verbreitete Weg, eine Community zu betreiben, setzen heutzutage viele Projekte auf Web-Foren, die den Benutzern weitergehende Möglichkeiten der Kommunikation, aber auch der Selbstdarstellung bieten. So können in einem Web-Forum zum Beispiel aussagekräftige Profilinformationen (zum Beispiel Wohnort, Alter, Geschlecht, Hobbys, Kontaktdaten aus Instant-Messenger-Systemen) hinterlegt werden, damit andere mehr über einen selbst erfahren können. Auch bietet ein Web-Forum meistens verschiedene Rollen, so dass verschiedene Benutzer unterschiedliche Rechte und Pflichten zur Pflege und Administration des Forums haben. Über Logging-Mechanismen ist es möglich, auf komfortable Weise alle Beiträge eines Benutzers zu finden, meist wird auch die Anzahl der Beiträge eines Nutzers in seinem Profil angezeigt. Eine beliebte Modifikation ist es, anhand der Anzahl der Beiträge verschiedene Titel oder Ränge zu verleihen, die in jedem Beitrag des Nutzers angezeigt werden.

Dieses System der Ränge ist für einige Community-Mitglieder eine Motivation, sich stark an Diskussionen zu beteiligen. Es führt allerdings auch zum so genann-

Abbildung 1.1.: Grafische Darstellung einer digitalen Identität. Ausschnitt eines Beitrags in einem Web-Forum mit Name, Rang („Greater Obsidian Drake“ mit zwei Sternen), Avatar und weiteren Attributen

ten „Spammen“ oder „Schinden“, bei dem die Benutzer eine große Anzahl von sinnlosen Beiträgen abschicken, um einen höheren Rang zu erreichen. Daher ist der Reputationsgewinn durch einen hohen Rang begrenzt.

Die Teilnahme an einer Online-Community unterliegt den im vorigen Unterabschnitt genannten Einschränkungen: das innerhalb einer Community aufgebaute Persönlichkeitsbild samt Identitätsattributen, Reputation und Beitragshistorie ist an genau diese eine Community gebunden – eine Übertragung zum Beispiel der Reputation in eine andere Community ist nicht ohne weiteres möglich, man kann höchstens auf die erstere Community verweisen. Dies entspricht jedoch nicht dem Verständnis einer Gesellschaft, in der sich Reputation und andere Eigenschaften über verschiedene Gruppen hinweg entwickeln kann, wenn sich die Gruppenmitglieder wiedererkennen. Der Einsatz einer Identitätsmanagement-Infrastruktur könnte helfen, dieses Hindernis zu beseitigen.

1.3.5. Gruppierung von Kommunikationsteilnehmern und Äußerungen

Konkrete Situationen, in denen Selbstdarstellung oder Schutz der Privatsphäre eine Rolle spielen, haben meist einen weitergehenden Sinn, der über den Austausch oder das Verstecken persönlicher Daten hinausgeht. In diesen Situationen geht es selten lediglich darum, sich dem Kommunikationspartner vorzustellen. Vielmehr ist die Selbstdarstellung eine Bereicherung einer weiteren Kommunikation, nämlich einer *inhaltlichen* Kommunikation. Zum Beispiel teilt man einem Online-Buchhandel nur deshalb seinen bevorzugten Kriminalroman-Autor mit, weil man anschließend ein personalisiertes Angebot erwartet und den Katalog des Buchhandels durchstöbern, vielleicht sogar ein Buch kaufen möchte. Das Anfordern des Katalogs und der Kaufvorgang sind Kommunikationsakte, die keine Selbstdarstellung beinhalten, damit nicht in den Bereich des Identitätsmanagements fallen, jedoch von diesem beeinflusst und bereichert werden. Um die Teilnehmer der Kommunikation zu ihren Äußerungen sowie diese zu ihrem Ursprung zuordnen zu können, ist ein Gruppierungsmechanismus sinnvoll und wünschenswert. Die Identitätsbehauptungen und Selbstdarstellungsakte sollen mit den dazugehörigen inhaltlichen Kommunikationsakten in einen *gemeinsamen Kontext* gebracht werden. Die verschiedenen Kommunikationsakte der Selbstdarstellung und der inhaltlichen Kommunikation kann man samt der Teilnehmenden unter dem Begriff einer *Sitzung* zusammenfassen. Diese Sitzung kann beschreibende Attribute über den Verlauf oder den Inhalt der Kommunikationsakte beinhalten, wie zum Beispiel die gewünschte Verschlüsselung der Übertragung oder die Datenschutzvereinbarung.

Es wird deutlich, dass dieses Konzept notwendig ist, um die dynamischen Aspekte des Identitätsmanagements zu umfassen. Ohne die konsequente Einführung einer Sitzung verliert das Identitätsmanagement einen großen Teil seiner Wirkung im dynamischen Einsatz. Durch diese enge Verzahnung der Sitzung mit dem Identitätsmanagement ergibt sich, dass auch eine systemtechnische Unterstützung beider Konzepte kombiniert werden sollte.

1.3.6. Schutz

Der Schutz der Privatsphäre sowie der persönlichen Daten bei der Kommunikation über digitale Netze ist ein schwieriges aber sehr erstrebenswertes Ziel. Systeme, welche Benutzer bei der Selbstdarstellung für Kommunikationsprozesse in solchen Netzen unterstützen, sollten ebenso beim Schutz vor Beobachtung, Verletzung der Privatsphäre und beispielsweise Identitätsdiebstahl helfen. Die Verwendung von sog. *privacy enhancing technologies* (PET) ist zum großen Teil deshalb so schwach, weil diese Programme schlecht in das System eingebunden sind und schwierig zu bedienen sind. Der Benutzer hätte also großen Nutzen, wenn

Schutzmechanismen in das Kommunikationssystem teilweise unsichtbar (wie beispielsweise im Falle von Verschlüsselung per SSL) aber effektiv eingebunden wären und mit minimalem Zusatzaufwand einsetzbar wären. Eine große Motivation für den Einsatz von Identitätsmanagement-Systemen ist also der Sicherheitsgewinn, der durch die einfach zu bedienende Integration von Schutzmechanismen gewonnen werden kann.

1.4. Ziele der Arbeit und Gang der Untersuchung

Ziel der Arbeit ist es, das Feld der Identitätsmanagement-Systeme zu analysieren und zu beschreiben, sowie die dabei identifizierte Lücke im Bereich des auf Selbstdarstellung ausgerichteten persönlichen Identitätsmanagement konzeptuell und prototypisch zu schließen. In Kapitel 7 wird aufgezeigt, dass sich bisherige Ansätze des persönlichen Identitätsmanagements im Wesentlichen auf Datenschutz und den Schutz der Privatsphäre konzentrieren – der Bereich der Selbstdarstellung wird nur am Rande betrachtet. Die Hauptmotivation für persönliches Identitätsmanagement aber ist wie in der einleitenden Motivation gezeigt die Selbstdarstellung, und Datenschutz lediglich eine zwingende Folge davon. Die Aspekte des Datenschutzes werden erst dadurch wichtig, dass persönliche Daten in digitalen Systemen behandelt werden – ohne digitale Selbstdarstellung wäre auch kein Schutz der Privatsphäre in digitalen Systemen von Nöten. Bei der Entwicklung der Konzepte sollte Datenschutz daher eine wichtige Rolle spielen – er ist aber nicht die Hauptmotivation.

Neben der Kategorisierung der bestehenden Systeme ist die Entwicklung eines Konzeptes und das Darstellen der Wege zur Konstruktion eines Systems für diesen bisher vernachlässigten Bereich des Identitätsmanagements ein wesentliches Element dieser Arbeit. Es soll gezeigt werden, dass die Konzeption und Konstruktion eines solchen Systems nach modernen Entwicklungsaspekten möglich ist, bestehende Kommunikationsanwendungen entscheidend verbessert und ganz neue Arten von Kommunikationssystemen ermöglicht. Dazu wird auch die Vorstellung eines im Rahmen dieser Arbeit entstandenen technischen Prototypen herangezogen, an dessen Beispiel Kernpunkte der Systemarchitektur und Systementwicklung gezeigt werden. Anhand dieses Beispiels wird auch die Auswirkung des Einsatzes eines Identitätsmanagement-Systems auf Anwendungssysteme abgeschätzt.

Der Kern der Arbeit gliedert sich in drei Teile.

- Zunächst werden die grundlegenden Konzepte für digitales Identitätsmanagement identifiziert und analysiert. Dazu gehören die Begriffe der *Identität* in verschiedenen Kontexten und des *Identitätsmanagements*. Eine Begriffsbestimmung ist notwendig, damit die vieldeutigen Begriffe eindeutig benutzt werden können. Diese Konzepte werden im ersten Teil des vorlie-

genden Textes bearbeitet: Kapitel 2 untersucht die grundlegenden Konzepte „Identität“ und „Selbstdarstellung“. Das darauf folgende Kapitel 3 untersucht die verschiedenen Mechanismen des Identitäts*managements*. Dabei wird sowohl auf die statische Verwaltung von Identitätsdaten als auch auf den dynamischen Einsatz der Identitäten in Kommunikationssituationen eingegangen. Aus diesen Mechanismen ergeben sich die Anforderungen an bestimmte Elemente von Identitätsmanagement-*Systemen*, die in Kapitel Kapitel 4 aufgeführt und erklärt werden.

- Teil II beschäftigt sich mit dem Stand der Forschung und Praxis des Gebiets der Identitätsmanagement-Systeme. Um weiterhin zu motivieren, wozu diese unterstützenden Systeme eingesetzt werden können, sind in Kapitel 5 einige Anwendungsszenarien für Identitätsmanagement-Systeme aufgeführt und erläutert. Es gibt bereits einige Systeme, die Unterstützung beim digitalen Identitätsmanagement bieten. Diese müssen analysiert, kategorisiert und verglichen werden. Dazu wird eine Klassifizierung für Identitätsmanagement-Systemen eingeführt (Kapitel 6), nach der vorhandene Systeme nach Intention und Funktion unterschieden und eingeordnet werden können. In Kapitel 7 werden dann Systeme aus Forschung und Praxis genauer beschrieben und eingeordnet. Dabei wird deutlich, dass ein auf Selbstdarstellung ausgerichtetes, persönliches Identitätsmanagement-System fehlt.

- Der dritte Teil der Arbeit umfasst die konkrete Konzeption und Konstruktion von persönlichen Identitätsmanagement-Systemen am Beispiel eines im Rahmen dieser Arbeit entstandenen Konzepts und technischen Prototypen. Die einzelnen Komponenten werden identifiziert, konzipiert und modelliert (Kapitel 8). Zusätzlich werden in Kapitel 9 Beispielanwendungen demonstriert, die den technischen Einsatz des Systems zeigen.

Der wissenschaftliche Anspruch, den sich diese Arbeit stellt, verteilt sich auf mehrere Bereiche. Zunächst ist es ein wichtiger Fortschritt, dass die Motivation und Intention für Identitätsmanagement-Systeme als kategorisierenden Faktor aufgegriffen und der beherrschenden Motivation des Datenschutzes diejenige der Selbstdarstellung als gestaltenden Faktor für Identitätsmanagement-Systeme gegenüberstellt. Die daraus resultierende Kategorisierung solcher Systeme (siehe Kapitel 6) unterscheidet sich dadurch stark von bisherigen Ansätzen. Die grobe Unterteilung in Identitätsmanagement für Organisationen, föderiertes und persönliches Identitätsmanagement ist insofern nicht neu, dass sie schon seit Jahren praktiziert worden ist – in der expliziten Form, wie sie hier vorliegt, ist sie jedoch neu. Sie hilft, die Motivation für ein selbstdarstellungsorientiertes Identitätsmanagement-System zu argumentieren. Bei der Konzeption und exemplarischen Konstruktion eines selbstdarstellungsorientierten, persönlichen Identitätsmanagement-System ist die enge Verknüpfung von Sitzungen, Sitzungsverwaltung

und Identitätsmanagement zu betonen, wobei ein laufender Prototyp zu diesem Konzept vorgewiesen werden kann, der die Struktur der Internet-Kommunikation durch Einführung einer zusätzlichen Netzwerkschicht – der Sitzungsschicht – entscheidend verändert. Eine ähnliche Schicht ist zwar im OSI-Schichtenmodell (vergleiche Kerner 1995) vorgesehen, im Internet jedoch bisher nicht umgesetzt. Nicht zuletzt ist auch der Einsatz von untereinander unabhängigen Teil-Ontologien zum Semantikerhalt bei der Identitätsdatenkommunikation sowie der Mechanismus der automatischen Übersetzung der Daten verschiedener Teil-Ontologien in dieser Weise bisher nicht vorgeschlagen oder implementiert worden.

Danksagung

Diese Arbeit wurde ohne fremde Hilfe und eigenverantwortlich angefertigt. Dennoch möchte ich einigen Menschen persönlich danken, die mich auf dem Weg zu diesem großen Ziel begleitet und jeder auf seine Weise unterstützt haben. Die Reihenfolge der Nennungen hat keine Bedeutung für die Wichtigkeit.

Mein Dank gilt meinem Betreuer Prof. Dr. Winfried Lamersdorf. Durch die Mitarbeit an seinem Arbeitsbereich VSIS und seine vielen Beiträge zu meinen Themen wurde diese Arbeit erst möglich. Seine große Erfahrung im wissenschaftlichen Betrieb war für mich von sehr großem Wert. Ebenso möchte ich Prof. Dr. Michael Koch danken, der in der Endphase sehr wertvolle Kommentare zur Arbeit und damit eine Einordnung in das Sachgebiet gegeben hat.

Sehr wichtig war der Zusammenhalt und die gegenseitige Unterstützung der wissenschaftlichen Mitarbeiter des Arbeitsbereichs VSIS. Vor allen möchte ich meinen (teils ehemaligen) Kollegen Christian Zirpins, Christian P. Kunze, Harald Weinreich und Frank Wienberg für viele ergebnisreiche Diskussionen und ausführliches Feedback danken. Ebenso gilt mein großer Dank Anne Hansen-Awizen und Volker Nötzold für die technische, organisatorische und seelische Unterstützung während meiner Arbeit.

Das diese Arbeit begleitende Forschungsprojekt onefC wurde von einigen Diplomarbeiten gestützt, die hier unbedingt Erwähnung bedürfen. Die Gespräche mit meinen Studenten Frank Wollenweber, Bernd Claasen, Thoralf Rickert, Gordian Kaulbarsch und Alexander Sack haben es mir immer wieder erleichtert, meine Ideen zu formulieren. Ihnen gebührt großer Dank für die gute Zusammenarbeit.

Ohne die Unterstützung durch meine Familie wäre diese Arbeit am wenigsten möglich gewesen. Meine Eltern ermöglichten mir mein Studium und ermutigten mich zur Weiterführung bis zur Promotion. Meine Frau Stephanie und meine Tochter Mareile ertrugen nicht nur meine nervliche Belastung – vor allem in der Endphase der Arbeit – sondern gaben mir viel Energie und Durchhaltevermögen, indem sie nicht von meiner Seite wichen. Dank und Liebe.

Teil I.

Digitale Identitäten und Identitätsmanagement

Zusammenfassung

Nachdem in der Einleitung die grundlegende Motivation für persönliches digitales Identitätsmanagement dargelegt worden ist, sollen in diesem ersten Teil der Arbeit die elementaren Konzepte digitaler Identitäten und des Identitätsmanagements im Allgemeinen aufgeführt werden. Dazu werden in Kapitel 2 zunächst die Konzepte der Elemente „digitale Identität“ als Kern-Element und „Sitzung“ als gruppierender Strukturmechanismus für den dynamischen Einsatz digitaler Identitäten erklärt. Im darauf folgenden Kapitel 3 werden sowohl statische als auch dynamische Aspekte des Umgangs mit digitalen Identitäten erläutert, was sich zum Konzept des digitalen Identitätsmanagement zusammensetzt. Am Ende des Teils wird aus den sich ergebenden Anforderungen ein Grundgerüst für technische Identitätsmanagement-Systeme abgeleitet (Kapitel 4).

2. Grundlegende Konzepte persönlicher Kommunikation in digitalen Netzen

Technischen Systeme, die dem Benutzer digitales Identitätsmanagement ermöglichen, sollten ihn bei dieser schwierigen und umfangreichen Aufgabe so gut wie möglich unterstützen. Um eine geeignete Abbildung der Konzepte des Identitätsmanagements auf IT-Systeme zu finden, müssen diese zuerst gründlich untersucht und beschrieben werden. Das erste wesentliche Konzept dabei ist sicherlich das der Identität – was genau ist eine Identität, was macht sie aus, welche Eigenschaften hat sie? Diese Untersuchung erfordert Ausflüge in andere Disziplinen als die Informatik, da es sich um ein gesellschaftliches und psychologisches Phänomen handelt. Ebenso wird das Konzept der Selbstdarstellung (oder Selbstenthüllung) im Internet untersucht: welche Besonderheiten bietet das digitale Medium im Gegensatz zur gewohnten Kommunikation und welche Mechanismen werden gebraucht, um das Konzept der Selbstdarstellung oder Rollenauswahl möglichst gut zu unterstützen? Daraufhin werden bestehende Konzepte der Identität in Informations- und Kommunikationssystemen untersucht und eine Definition entwickelt, die den Gebrauch des Begriffs *digitale Identität* im Verlauf dieser Arbeit festlegt.

Im Zusammenhang mit digitalen Identitäten mit anhaftenden persönlichen Daten werden anschließend die Themen Datenschutz und Schutz der Privatsphäre betrachtet. Der Schutz persönlicher Daten in digitalen Systemen hat eine besondere Rolle, da hier ein Missbrauch der Daten auf einfachere Weise und in größerem Umfang möglich ist als in der nicht-digitalen Welt. Elektronische Datenverarbeitung und Datenübermittlung bergen große Risiken für die Privatsphäre. Daher muss dieser Aspekt schon in der Untersuchung der grundlegenden Elemente berücksichtigt werden.

Desweiteren wird in diesem Kapitel das Konzept der *Sitzung* untersucht. Sowohl Selbstdarstellung als auch die Sicherung der persönlichen Daten spielen zumeist in Situationen eine Rolle, in denen inhaltliche Kommunikation angereichert oder abgesichert werden soll. Diese Kommunikationsakte und die Selbstdarstellungsakte gemeinsam fasst man unter dem Begriff der Sitzung zusammen, der in Abschnitt 2.4 untersucht wird.

2.1. Identitäten und Identitätsbildung

Um das Konzept der *digitalen Identität* zu greifen, lohnt es sich, das allgemeine Konzept der Identität genauer zu betrachten (siehe hierzu auch Kunze 2003). Identität spielt in vielen Bereichen eine große Rolle. Grob unterscheiden kann man zwischen *logischen* und *sozialen* Aspekten der Identität. Bei letzteren spricht man häufig auch nicht von Identität sondern von Persönlichkeit, denn hier umfasst der Begriff deutlich mehr als in der Logik. Auch muss zwischen verschiedenen Identitätsattributen unterschieden werden: einige äußere Merkmale dienen der Identifizierbarkeit (zum Beispiel Name, Alter, Wohnort oder Augenfarbe), andere werden zur Selbst-Definition herangezogen (Gruppenzugehörigkeit, Familienaffinität).

Sprachlich wird im Deutschen der Begriff „Identität“ häufig ungenau benutzt. Während es inhaltlich einen Unterschied zwischen *Gleichheit* und *Identität* gibt, wurde diese Unterscheidung durch die Wörter „das Gleiche“ und „das Selbe“ wegen zu häufiger Missachtung aufgehoben (Duden 1996, S. 318). Während bei gleichen Dingen lediglich eine Vielzahl an Attributen gleich sind (jedoch nicht unbedingt die Identität, es kann sich durchaus um verschiedene Dinge handeln), ist bei der Bezeichnung zweier Dinge als „das Selbe“ tatsächlich immer nur ein Ding gemeint.

2.1.1. Der logische Identitätsbegriff

In der Logik, das heißt in der philosophischen und mathematischen Logik, bedeutet Identität die absolute Gleichheit zweier Einheiten. Eine Einheit ist immer nur identisch zu sich selbst. In der Mathematik spricht man von einer *identischen Abbildung* als eine Funktion, die mit ihrem Argument „nichts tut“, das heißt das Ergebnis ist immer das gleiche wie das Argument. Für eine Menge M, die zugleich Definitionsbereich und Wertebereich der Funktion ist, lautet die identische Abbildung (vgl. Wikipedia 2004):

$$id_M : M \to M, \forall x \in M : id_M(x) = x$$

Schon Aristoteles hat sich an einer Definition von Identität versucht, indem er die Ununterscheidbarkeit von Objekten herangezogen hat (siehe Mittelstraß 1984). Leibniz definierte die numerische Identität, um jedem Objekt eine eindeutige Nummer zuweisen zu können und damit die Welt berechenbar zu modellieren (Henrisch 1976). Die Leibnizsche (strikte) numerische Identität setzt voraus, dass wirklich alle Eigenschaften gleich sein müssen, das schließt das Alter ein. Diese extreme Definition der Identität hat entscheidende Nachteile, denn jede Zustandsänderung eines Objekts bedeutet Identitätsverlust. Eine Zustandsänderung tritt aber schon durch den bloßen Verlauf der Zeit ein, womit das Konzept der numerischen Identität auf einzelne Zeitpunkte beschränkt ist und auch keinen Zusammenhang zwischen zwei (identischen, aber in der Zeit verschiedenen) Objekten mehr bilden kann. Aus diesem Grund wurden verschiedene Begriffe der

gemäßigten numerischen Identität entwickelt, welche den Zustandswechsel erlauben.

In Mathematik und Philosophie geht es also eher um die Identifikation als um Identitäten im Sinne von Persönlichkeiten, denen weitere Eigenschaften zugesprochen werden können. Die Identifikation ist eine Identitätsfeststellung *a priori*, das heißt es muss keine Verifikation einer Identitätsbehauptung durchgeführt werden. Diese Eigenschaft der eindeutigen Identifikation ist ein entscheidender Faktor für digitale Identitäten in Computersystemen, wie später gezeigt wird.

2.1.2. Identität in Soziologie und Sozial-Psychologie

Identität bedeutet im sozialen Kontext meistens das gleiche wie *Persönlichkeit* oder *Individuum*. Identität ist hier häufig die Kurzform von *Selbst-* oder *Ich-Identität* (Brockhaus 1989). Gemeint ist das Bewusstsein, sich selbst von anderen Personen zu unterscheiden (vgl. Erikson 1956/1966, S. 107):

> Das Gefühl der Ich-Identität ist also das angesammelte Vertrauen darauf, dass der Einheitlichkeit und Kontinuität, die man in den Augen anderer hat, eine Fähigkeit entspricht, eine innere Einheitlichkeit und Kontinuität (also das Ich im Sinne der Psychologie) aufrecht zu erhalten.

Auch hier ist also nicht die Menge an Attributen gemeint, mittels derer Außenstehende eine Identität feststellen oder gar überprüfen können, sondern eher ein Konzept, das einem ein inneres Gefühl einer Einheit vermittelt. Es wird unterschieden in mehrere *Teil-Identitäten*, die kontextabhängig bewusst oder unbewusst aktiviert werden. Dabei ist nicht das klinische Störungsbild der *multiplen Persönlichkeit* gemeint, sondern das unterschiedliche Empfinden und Auftreten in sehr unterschiedlichen Situationen (zu hause oder bei der Arbeit, beim Arzt oder bei einer Freizeitaktivität) gemeint (siehe Döring 2003, Kap. 6.1).

Die Aufteilung der eigenen Selbst-Aspekte in Teil-Identitäten hilft bei der Kategorisierung der Identitätsattribute, die auch zu Selbstdarstellungszwecken gebraucht werden. Dabei muss nicht jedes Identitätsattribut einer bestimmten Teil-Identität exklusiv zugeordnet werden, diese Teilmengen sind nicht disjunkt. So kann zum Beispiel das Identitätsattribut „selbstbewusst und extrovertiert“, das im Beispiel von Abbildung 2.1 der Teil-Identität „Fußballfan“ anhaftet, ebenso der Teil-Identität „Angestellter“ anhaften, wenn die Arbeit dies erfordert oder ermöglicht.

2.1.3. Selbstdarstellung

Selbstdarstellung ist ein wichtiger Faktor in zwischenmenschlicher Kommunikation, aber auch bei der Benutzung von elektronischen Diensten. Es gibt verschiedene Formen der Selbstdarstellung. Nach Jones und Pittman (1982) sind diese

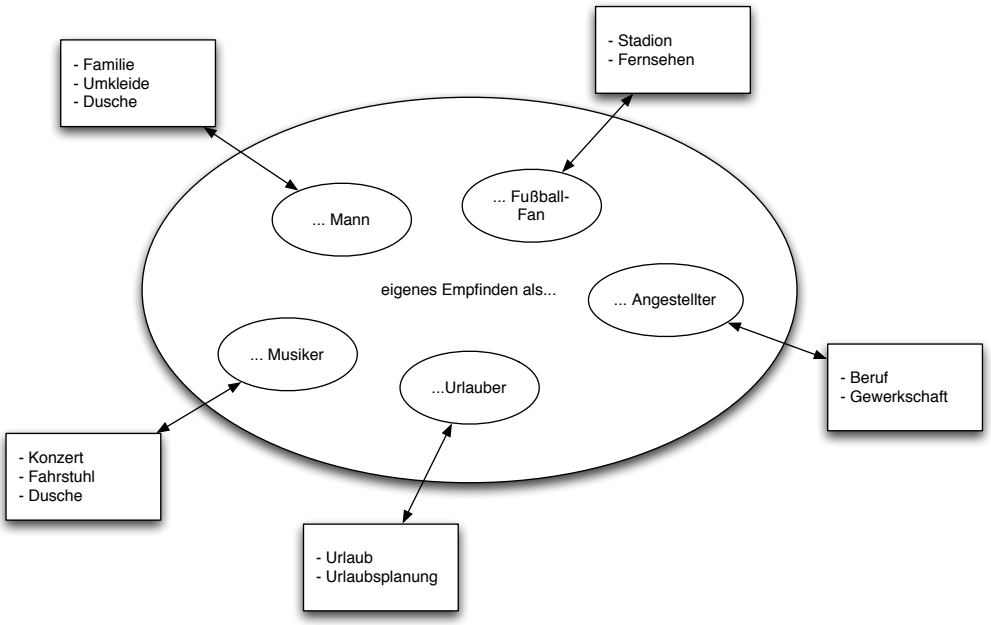

Abbildung 2.1.: Verschiedene Rollen und ihre assoziierten Teil-Identitäten

gegliedert in *sich beliebt machen* (ingratiation), *sich als kompetent darstellen* (self-promotion), *sich als vorbildlich darstellen* (exemplification), *andere einschüchtern* (intimidation) und *sich als hilfsbedürftig darstellen* (supplication). Alle Formen der Selbstdarstellung haben die Präsentation von persönlichen Eigenschaften gemein, um ein bestimmtes Ziel zu erreichen. Die Absicht der Selbstdarstellung hat also Einfluss auf die Auswahl der zu präsentierenden Eigenschaften.

Eine weitere Untergliederung der Selbstdarstellung stammt von Luft und Ingham (1955), in der die Persönlichkeit des Menschen in vier Bereiche geteilt wird, je nachdem ob die betreffende Information selbst oder den anderen bekannt ist. Daraus ergibt sich ein so genanntes Johari-Fenster[1] (vgl. Abbildung 2.2). Auf die digitale Kommunikation und Selbstdarstellung übertragen bleibt diese Unterteilung sehr interessant: der erste Sektor, die „Arena“, ist sowohl einem selbst als auch den anderen bekannt, er spiegelt die derzeit verwendete Persönlichkeit wider. Sektor 3, die „Facade“, ist nur einem selbst bekannt und ist der Datenbereich, der vor Zugriff geschützt werden muss (falls er überhaupt digital vorhanden ist). Der zweite Sektor dagegen bezeichnet die Eigenschaften, die nicht einem selbst sondern nur den Kommunikationspartnern bekannt sind. Es könnte sich hierbei etwa um Informationen, die mittels Data-Mining gewonnen worden sind, handeln.

[1] Der Name „Johari“ ist aus den Vornamen der Erfinder Joseph Luft und Harry Ingham abgeleitet

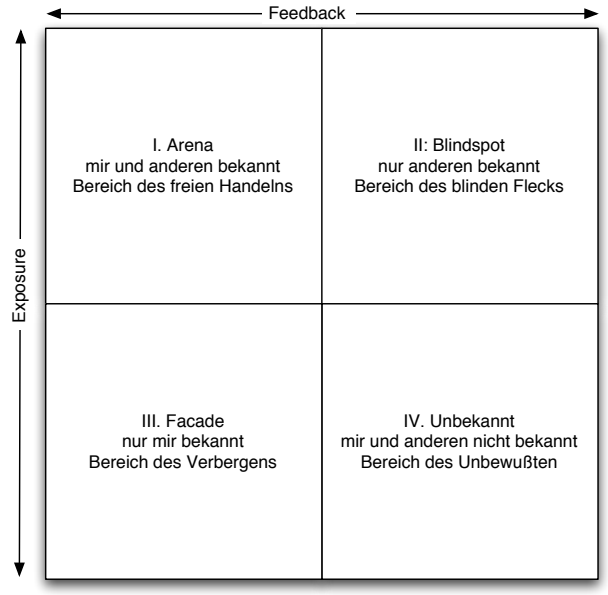

Abbildung 2.2.: Das Johari-Fenster, nach Luft und Ingham (1955)

Als konkretes Beispiel könnte man den Online-Buchhandel Amazon[2] nennen, die aus dem Verhalten der Benutzer Buchvorschläge entwickeln, und das Wissen, dass diese Bücher den Benutzer interessieren könnten, hat dieser selbst vorher noch nicht gehabt.

2.1.4. Rollenauswahl und Rollengestaltung

Mit dem Persönlichkeitsbild, das man bei den Kommunikationspartnern erzeugt hat, übernimmt man für die Dauer der zusammengehörigen Kommunikation (oder *Sitzung*) eine bestimmte Rolle, die sich jedoch im Verlaufe der Sitzung ändern kann. Die Rolle ist abhängig von den eigenen Wünschen und den Möglichkeiten, die einem die Umgebung gibt. Die Rolle des Administrators eines Community-Portals etwa ermöglicht einem Benutzer, neue News-Einträge einzupflegen oder weiteres. Diese Rolle kann jedoch nur eingenommen werden, wenn sich der Nutzer dem System gegenüber als Administrator authentifizieren kann. Eine Rolle ist also mit bestimmten Rechten und Pflichten verbunden. Die Auswahl der passenden Rolle kann eine schwierige Aufgabe sein, wenn man mehrere Rollen zur Auswahl hat und jede Rolle unterschiedliche Möglichkeiten bietet.

In verteilten Systemen wird das Gebiet der Rollen-Rechte-Systeme schon länger thematisiert. Rollen können innerhalb von Organisationssystemen und auch offenen Diensten verschiedene Aufgaben erledigen, wie zum Beispiel den Zugriff

[2] `http://www.amazon.de`

auf bestimmte Ressourcen ermöglichen, Funktionalität bereitstellen oder Verantwortlichkeiten zu organisieren (siehe zum Beispiel Lupu und Sloman 1997).

Das Konzept der Rolle ist in diesem Zusammenhang sehr ähnlich zum Konzept der Teil-Identität, wie es in Unterabschnitt 2.1.2 dargestellt ist. Eine Rolle ist allerdings eine nicht personengebundene Zusammenfassung der oben angeführten Eigenschaften, während Teil-Identitäten in der Soziologie natürlich personengebunden sind, weil sie vor allem innerhalb einer Persönlichkeit wichtig ist. Rollen können dynamisch von Teil-Identitäten eingenommen werden und gelten nur in bestimmten Situationen. Desweiteren bietet die Rolle einen Blick von außen auf die Person, Teil-Identitäten können auch privat entwickelt und ein Blick auf sich selbst sein. Auf Rollenauswahl wird näher in Unterabschnitt 3.3.2 eingegangen.

2.2. Identitäten in Computersystemen – digitale Identitäten

Auch im informations- und kommunikationstechnischen Bereich spricht man häufig von Identitäten. Hier wird meistens eine eindeutig identifizierbare Einheit gemeint, also ähnlich des logischen Identitätsbegriffs. Wenn es um Personen in digitalen Netzen geht, meint man mit Identität zumeist einen mittels digitaler Signatur oder anderen so genannten *credentials* authentifizierten Benutzer. Dass hinter Identitäten im Internet auch Persönlichkeiten stehen, wurde in der jüngsten Sozialforschung und im Datenschutz entdeckt. Während die erstere sich auf den Identitätsbegriff der Soziologie bezieht, steht im Bereich des Datenschutzes die Eigenschaft des Kommunikationspartners im Vordergrund. Hansen definiert eine Identität wie folgt: „Bei der Identität, die im Fokus [der Forschung an Identitätsmanagement-Systemen, Anm. des Verf.] steht, geht es um die *kommunikativ zugängliche Repräsentanz einer Person*. In der Regel wird es sich dabei um eine natürliche Person und Informationen über sie handeln, doch trifft man auf die Bezeichnung [Identitätsmanagement-System] auch bei juristischen Personen." (Hansen u. a. 2003, S. 551, Hervorhebung hinzugefügt). Mont u. a. (2003) definieren digitale Identitäten als „a view on the identity information associated to an entity, at a specific point of time". Letztere Definition beschreibt die digitale Identität indirekt als das, was der Benutzer zum Zeitpunkt der Kommunikation seinem Gegenüber zeigt, zuzüglich dessen was diesem schon bekannt ist.

Seit Juli 2000 wird von Pfitzmann und Köhntopp ein Dokument entwickelt, das eine Terminologie für Anonymität, Unbeobachtbarkeit und Pseudonymität bereitstellen soll. Seit Juni 2004 wird auch der Bereich des Identitätsmanagement berücksichtigt. Hier wird aufbauend auf den soziologisch geprägten Teil-Identitäten eine digitale Identität wie folgt definiert:

> Digital identity denotes attribution of properties to a person, which are immediately operationally accessible by technical means. More to the point, the identifier of a digital partial identity can be a simple e-mail address in a news group or a mailing list. Its owner will attain a certain reputation. More generally we might consider the whole identity as a combination from „I“ and „Me“ where the „Me“ can be divided into an implicit and an explicit part: Digital identity is the digital part from the explicated „Me“. Digital identity should denote all those personally related data that can be stored and automatically interlinked by a computer-based application. (Pfitzmann und Hansen 2004, S. 13)

Diese Definition des Begriffs „digitale Identität“ ist recht ausführlich erklärend. Sie beinhaltet den Bezug zur logischen Identität, indem ein (alphanumerischer) Identifikator eingeführt wird (hier wird das Beispiel „E-Mail-Adresse“ genannt), anhand dessen die Identifikation durchgeführt werden kann, und die zusätzliche Erweiterung des Identitätsbildes durch persönliche Daten. Die Unterscheidung zwischen „I“ und „Me“ stammt von Mead (1932). Er bezeichnete den Teil der eigenen Identität, der nur einem selbst zugänglich ist, als „I“[3], und den außen wahrgenommenen Teil der Persönlichkeit als „Me“[4].

Eine weitere Definition digitaler Identitäten (hier als „digital persona“ bezeichnet) stammt von Clarke (1994):

> The digital persona is a model of an individual's public personality based on data and maintained by transactions, and intended for use as a proxy for the individual. (Clarke 1994)

Hier steht das Persönlichkeitsbild, also das Abbild einer realen Person wie es von *anderen* empfunden wird, sowie der Zweck als Kommunikationsschnittstelle (*proxy*) im Vordergrund. Die Person hat selbst kein Abbild, sondern die digitale Identität existiert nur bei den Kommunikationspartnern oder anderen, die auf eine digitale Darstellung der Person getroffen sind. Clarke führt diesen Begriff vor allem dazu ein, die Angriffsmöglichkeit auf die Privatsphäre von Personen im Internet darzustellen. Er stellt jedoch ein differenziertes Konzept der digitalen Identität vor (inklusive eines passiven und eines aktiven Teils) und zeigt auch mögliche Anwendungsszenarien, die nicht vom Datenschutz geprägt sind.

Die Definition, die für diese Arbeit gültig sein soll, lautet:

Definition 2.1. Eine *digitale Identität* ist definiert als ein eindeutiger, digitaler Identifikator, dem personifizierende Attribute zugeordnet sein können.

[3] engl.: ich

[4] engl.: mich, aber teilweise eben auch „ich“: „Who is it?“ wird mit „Me“ beantwortet, nicht mit „I“!

Der eindeutige Identifikator ist zugleich ein Hilfsmittel zum Erkennen von Identitäten und eine logische Beschreibung eines Kommunikationsendpunktes. Im Gegensatz zu E-Mail-Adressen, die eine Referenz zu dem (technischen) Server beinhalten, auf dem der Account gespeichert wird, ist der eindeutige Identifikator in dieser Definition der digitalen Identität eine rein logische, inhaltliche Adresse. Diese Vorgehensweise ist zum Beispiel in Peer-to-Peer Netzen (zum Beispiel JXTA Peer-IDs, (siehe Oaks und Gong 2002)) oder anderen IP-Overlays (wie zum Beispiel *i3* (siehe Stoica u. a. 2004)) üblich. Die Übersetzung der logischen Adressen in physikalische wird vom System übernommen. Zu beachten ist, dass in dieser Definition die Bedingung für Verifikation der Identität (also Authentifizierung) nicht gegeben ist. Authentifizierung muss über Credentials oder andere Mechanismen realisiert werden, die in den Attributen der digitalen Identität abgelegt sind.

Personifizierend bedeutet in diesem Zusammenhang, dass dem (unpersönlichen) Identifikator Attribute angehängt werden, die im Gesamtzusammenhang eine digitale *Persönlichkeit* darstellen. Die Attribute müssen weder die gesamte Persönlichkeit des Eigentümers der digitalen Identität widerspiegeln (dies ist wegen der Komplexität und großen Subjektivität und damit schlechten digitalen Abbildbarkeit menschlicher Attribute auch nicht möglich) noch überhaupt realen Bezug haben; Sie personifizieren nur die digitale Identität.

Obwohl der soziologische oder psychologische Faktor des Identitätsbegriffs für die kommunikativen Eigenschaften – und damit für die Gestaltung eines die Kommunikation unterstützenden Systems – gewichtiger erscheint, bietet der logische Begriff der Identifizierbarkeit einen entscheidenden Anhaltspunkt für digitale Kommunikation, nämlich die Wiedererkennbarkeit. Nur wenn digitale Kommunikationspartner in verschiedenen Kontexten eindeutig identifizierbar sind, werden sie auch wiedererkennbar, und erst dann können sich längerfristige und kontextübergreifende Beziehungen zwischen den Kommunikationspartnern entwickeln. Der Ruf oder die Reputation eines Individuums ist nur dann nachvollziehbar, wenn die Person auch als die zu diesem Ruf gehörende erkannt werden kann. Während die eindeutige Identifizierbarkeit im wirklichen Leben von der Betrachtung einzelner, verschiedener Identitätsmerkmale abhängt, kann dieser Aspekt in digitalen Systemen durch einen generischen Identifikator gelöst werden. Identifiziert wird in diesem Fall eine digitale Identität, die zunächst keinen Bezug zu einer realen Person haben muss. Nur wenn dieser digitalen Identität weitere Daten anhaften, die berechtigte Rückschlüsse auf eine reale Identität haben, kann man eine reale Identifikation substituieren. Dies könnte beispielsweise durch die Verwendung von Personendaten geschehen, die von vertrauenswürdigen oder bekannten digitalen Identitäten (beispielsweise der digitalen Identität eines Einwohnermeldeamtes) signiert worden sind.

2.2.1. Selbstdarstellung im Internet

Selbstdarstellung findet im Internet meist in ungeordneter Form statt, also in Freitext anstatt in strukturierten Daten. Beispielsweise beschreibt man sich selbst in einem USENET- oder Web-Forum, um sich der Community vorzustellen, oder trägt bestimmte Daten über sich selbst in ein Profil eines Instant Messenger Systems ein (beispielsweise das Geburtsdatum und den Wohnort sowie die beherrschten Sprachen bei ICQ). Eine weit verbreitete Form der Selbstdarstellung im Internet ist das Erstellen und Anbieten einer *persönlichen Homepage*, auf der private Informationen zum Abruf für alle Internet-Nutzer angeboten werden. Persönliche oder auch private Homepages wurden schon von mehreren Soziologen auf ihre Selbstdarstellungsaspekte untersucht (Döring 2003; Machilek u. a. 2004). Dabei wurde festgestellt, dass Daten, die auf solchen Homepages veröffentlicht werden, meistens der Realität entsprechen. Durch gezielte Auswahl der Eigenschaften und Tatsachen, von denen auf den Homepages berichtet wird, versuchen die Anbieter der Homepages das gewünschte Persönlichkeitsbild beim Betrachter zu erwecken. Allerdings liegen auch die Daten auf Homepages unstrukturiert und ohne maschinenlesbare Zusatzinformationen vor, die sie definieren und beschreiben.

Im Gegensatz zu Homepages wird in spielerischen Umgebungen wie MUDs[5] oder MMORPGs[6] ganz bewusst und von allen Teilnehmern akzeptiert meist eine unwahre Identität vorgetäuscht. Die Spieler versuchen, durch ihr Verhalten und ihre Äußerungen den Eindruck einer neuen Persönlichkeit zu erzeugen und damit das Spiel zu bereichern. Zum Umfeld der Internet-Spiele gehört auch der Ruf, den sich ein Spieler in der Internet-Gemeinde erarbeiten kann: besonders begabte Internet-Rollenspieler sind in der gesamten Community bekannt und geachtet. Einen guten Ruf kann man sich auch durch Ausarbeitung und Bereitstellung von Hilfsinformationen zum Spiel erarbeiten. So ist zum Beispiel die Autorin einer Internet-Seite namens EQTraders[7], die nur unter dem Pseudonym Niami Denmother bekannt ist, großen Teilen der Spieler des MMORPGs EverQuest[8] ein Begriff. Auf ihrer Web-Seite stellt sie umfangreiche Informationen zum Herstellen von Gegenständen im Spiel bereit.

Selbstenthüllung oder *Selbstoffenbarung* (engl: self-disclosure) nennt man Selbstdarstellungsvorgänge, die nicht eigeninitiiert, sondern vom Kommunikationspartner angefordert sind. Eigene Persönlichkeitsmerkmale, die identifizierend sein können (aber nicht müssen), werden auf Anfrage preisgegeben. Ein Beispiel dafür wäre die Nennung des eigenen Namens in einem Hotel, in dem man übernachten

[5] MUD: Multi-User-Dungeon, eine Gattung textbasierter Internet-Computerspiele
[6] Massively Multiplayer Online Role Playing Games
[7] http://www.eqtraders.com
[8] http://www.everquest.com

möchte, oder die Äußerung der bevorzugten Literaturrichtung in einem Buchladen zwecks besserer Beratung.

Im Internet findet Selbstoffenbarung deutlich öfter und in geordneter Form statt als Selbstdarstellung. Jedes Einloggen in einen Internet-Dienst ist ein Akt der Selbstoffenbarung, weil man dadurch seine Identität preisgibt. Beim Einkaufen in einem Online-Shop muss man Selbstoffenbarung betreiben, damit der Händler die Adresse des Kunden weiß und ihm die Ware schicken kann. Das Herausgeben der Kreditkartennummer zum Bezahlen (oder teilweise auch nur zur Identitäts-Verifikation) gehört ebenfalls dazu.

Alles Aspekte der Selbstdarstellung im Internet sollten durch ein Identitätsmanagement-System gestützt werden. Dadurch hat der Anwender den Vorteil, dass er diese Selbstdarstellungsakte in konsistenter Form verüben kann und seine Identitätsdaten in einem abgeschlossenen System behandeln kann. Für die Kommunikation selbst ergibt sich aus der normierten oder standardisierten Identitätsdatenkommunikation eine leichtere Gestaltung der Kommunikationswerkzeuge. Die Zusammenfassung von Selbstdarstellungsaspekten im Internet in einer Identitätsmanagement-Infrastruktur ermöglicht es Anwendern und Diensten gleichsam, die aus dem normalen Leben gewohnten Vorgänge im digitalen Leben nachzubilden.

2.2.2. Föderierte Identitäten

Im betrieblichen Umfeld haben digitale Identitäten meist eine eingeschränktere Bedeutung. Hier wird unter diesem Begriff lediglich ein Benutzer-Account in einem digitalen Informationssystems oder der Datensatz eines Mitarbeiters in einer Datenbank verstanden. Auch diese Daten gehören zu einer digitalen Identität im Sinne von Mont u. a., aber der Aspekt der Kommunikation, wie er bei Pfitzmann und Hansen im Vordergrund steht, fehlt. Auch ist die Blickrichtung eine andere: während in Identitätsmanagement-Systemen aus der Forschung der Nutzer im Mittelpunkt steht und dieser Ausdrucksmöglichkeiten zur Selbstdarstellung gewinnen soll, liegen in kommerziellen Systemen die Interessen eher darin, einen zentralen Datenspeicher mit Personendaten zu füllen und auszuwerten. Die Kontrolle über die Daten liegt hier eher nicht beim Benutzer selbst.

Im Gegensatz zu diesem Identitätsmanagement in Organisationen wird der organisationsübergreifende Identitätsbegriff als *föderierte Identität* (engl: federated Identity) bezeichnet (vgl. Norlin und Durand 2002). Hier steht die Übertragbarkeit der Identitätsdaten im Vordergrund, bei der Sicherheits- und Vertrauensaspekte stärker berücksichtigt werden müssen. Die Notwendigkeit föderierter Identitäten ergibt sich aus der immer stärkeren Zusammenarbeit verschiedener Unternehmen in einem globalen Kontext. In diesem Zusammenhang werden außer den Sicherheitsaspekten zusätzlich noch die Interoperabilitätsaspekte sehr wichtig. Ohne gemeinsame Repräsentationen von Identitäten und Identitätsda-

tenaussagen ist eine föderierte, von mehreren Unternehmen gemeinsam genutzte digitale Identität nicht denkbar.

Eine föderierte Identität ist also eine digitale Identität, die innerhalb einer Föderation unter den Organisationen ausgetauscht werden kann. Dadurch wird sowohl organisationsübergreifende Authentifizierung als auch weiterführende Personalisierbarkeit möglich. Die Organisationen können der föderierten Identität weitere, situationsabhängige Attribute hinzufügen, die von den Partnern zur Personalisierung hinzugezogen werden. Offen bleibt dabei die Frage der Kontrolle über die Identitätsdaten: auch wenn der Benutzer vor der Föderierung der Identität um Zustimmung gebeten wird, kann er möglicherweise nicht nachvollziehen, in welchem Umfang Daten ausgetauscht worden sind. Die sich daraus ergebenden Probleme des Datenschutzes sollten vor dem Einsatz einer solchen Struktur geprüft werden.

2.2.3. Semantik von Identitätsattributen

Während die Persönlichkeiten von Menschen in der Wirklichkeit ihr ganzes Leben lang Eigenschaften und Werte sammeln und sich so zu komplexen Gebilden entwickeln, findet die Identitätsbildung im Internet auf einer anderen Ebene statt. Wenn sie nicht vom Benutzer selbst eingegeben worden sind, können Eigenschaften digitalen Identitäten direkt zugesprochen werden, oder sie entwickeln sich implizit aus Kommunikationssitzungen (siehe auch Unterabschnitt 3.3.3). Wenn der Aspekt der Kommunikation von Identitätsdaten im Vordergrund steht, dann trifft man jedoch bald auf das Problem der Semantik: Wie kann sichergestellt werden, dass die Kommunikationsgegenseite unter dem übertragenen Attribut auch das versteht, was gemeint war? Und wie kann bei der Zusprechung von Attributen sichergestellt sein, dass der Empfänger weiß, was ihm dort zugesprochen wird?

Um semantischen Zusammenhalt zu gewährleisten, werden in Computersystemen immer häufiger *Ontologien* verwendet. Der Begriff „Ontologie“ kommt aus dem griechischen und bedeutet „Lehre vom Sein“. In Informationssystemen meint man damit eine Kategorisierung des Systemwissens, so dass eine strukturierte Basis der Dinge vorliegt, welche im System vorkommen können. Gruber definiert Ontologien wie folgt:

> „A conzeptualization is an abstract, simplified view of the world that we wish to represent for some purpose. ... An ontology is an explicit specification of a conceptualization.“ (Gruber 1993)

Entsprechend der Aufgabe der Wissensrepräsentation und Wissensstrukturierung findet die Forschung bezüglich Ontologien in Computersystemen seine Wurzeln in der Künstlichen Intelligenz. Aber auch in verteilten, heterogenen Systemen wird diese Technologie gebraucht, denn hier kann nicht garantiert sein, dass alle

Kommunikationspartner „die gleiche Sprache sprechen“. Ein Beispiel für Wissensrepräsentation in verteilten Umgebungen sind Multi-Agenten-Systeme: Es wird eine applikationsabhängige Ontologie erstellt, anhand derer die Agenten bei Kommunikationsakten sicher sein können, worüber kommuniziert wird (Tamma und Bench-Capon 2002). Der Einsatz von Ontologien variiert dabei. In einigen Systemen bildet die Ontologie sozusagen das Modell der Wissensbasis, und die Einheiten der Kommunikation sind Objekte, die von der Ontologie abgeleitet sind. In anderen Systemen werden die Daten *nachträglich* mit Metadaten versehen, um den Bezug zur Ontologie herzustellen. Ein Beispiel dafür ist das Semantic Web, welches das schon bestehende WWW mit Hilfe der RDF-Technologie um semantische Informationen erweitert (z. B. kurz beschrieben in Swick 1999). In RDF kann man Metadaten verfassen, die dann zu einer bestehender Web-Seite assoziiert werden. Die Kommunikationsgegenstände, nämlich die Web-Seiten, sind hier anders als bei den Agentenkommunikationssprachen nicht direkt von der Ontologie abgeleitet (im Sinne von objektorientierter Vererbung), sondern lediglich assoziiert.

Eine dritte Möglichkeit für den Gebrauch von Ontologien ist die Nutzung zur *Definition von Datenstrukturen* und nicht für die Daten selbst. Während bei Verwendung von RDF für das World Wide Web den Web-Seiten direkt Metadaten zugeordnet werden, kann bei dieser Anwendungsweise einem strukturierten Datentyp seine Bedeutung per Ontologie zugeordnet werden. Alle Daten, die diesem strukturierten Datentyp entsprechen, sind dann per Ontologie definiert. Dies ist ein sehr mächtiges Konzept, wenn es viele verschiedene Datentypen mit jeweils vielen möglichen Daten gibt, wie es bei digitalen Identitäten der Fall ist. Man stelle sich den Inhalt der Bibliothek einer Person als Identitätsdaten vor. In diesem Fall ist es viel effektiver, den Datentypen mit Hilfe einer Ontologie zu hinterlegen, als alle vorkommenden Daten, da diese sehr umfangreich sein können (sowohl in der Hinsicht, dass eine Person eine sehr große Bibliothek haben kann, als auch in der Hinsicht, dass sehr viele Menschen eine Bibliothek haben).

Allerdings ist der Gebrauch von Ontologien zur semantischen Anreicherung von Inhalten nicht unumstritten (siehe Marshall und Shipman 2003). Ontologien werden zwar schon seit einigen Jahren vor allem in Bereichen der Künstlichen Intelligenz eingehend betrachtet, aber der Umgang mit ihnen ist durch viele Hürden versperrt. Nur in sehr eingeschränkten Umgebungen (beispielsweise in fest umrissenen Agentenszenarien) ist es möglich, eine für diesen Realitätsausschnitt allumfassende Ontologie zu erstellen und diese dann auch plattformübergreifend zu nutzen. Es wird allseits als quasi unmöglich betrachtet, eine Ontologie für die gesamte Welt zu erstellen, wie sie allerdings für die Erfassung des gesamten WWW (und eben auch für die Erfassung aller möglicher Identitätsdaten!) von Nöten wäre.

Das Konzept der Ontologie kann auch für die Attribute digitaler Identitäten hilfreich genutzt werden. Ist jedes Identitätsattribut einer Ontologie zugeordnet,

so kann es nach Bedeutung strukturiert gespeichert, dargestellt und zu Kommunikationspartnern übertragen werden. Allerdings trifft man dabei auf Probleme, welche die grundsätzlichen Probleme beim Umgang mit Ontologien noch erweitern: digitale Identitäten werden von ihren Eigentümern, also meistens Individuen verwaltet. Diese haben unterschiedliche Anforderungen an ihre Daten und wollen vielleicht ganz unterschiedliche Bereiche damit abdecken. Eine gemeinsame Ontologie für alle diese Ansprüche ist also kaum vorstellbar oder wäre zumindest unhandhabbar groß. Der logische Schluss daraus ist die Verwendung von *Teil-Ontologien*, die jeweils nur ein bestimmtes Gebiet umfassen. Damit könnte jeder Anwender selbst entscheiden, mit welchen Teil-Ontologien er seine Daten hinterlegen möchte. Dies jedoch führt automatisch zu Inkompatibilitäten, wenn zwei Nutzer Daten aus dem gleichen Gebiet nach unterschiedlichen, unabhängig voneinander entstandenen Teil-Ontologien strukturieren. In diesem Fall sind *Abbildungen* der Attribute notwendig, um sinnbehaftete Kommunikation zu ermöglichen. Diese Abbildungen können in Übersetzungsregeln zusammengefasst werden, die dynamisch auf Kommunikationsfragmente angewendet werden können.

2.3. Datenschutz und Privatsphäre

Datenschutz ist der Schutz persönlicher Belange bzw. des Persönlichkeitsrechts sowie des Rechts auf informationelle Selbstbestimmung. Das heißt, dass nicht die Daten selbst geschützt sind, sondern die Rechte an ihnen. Personenbezogene Daten sind in diesem Zusammenhang „Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener)“[9]. Das Recht auf informationelle Selbstbestimmung hat in Deutschland durch das so genannte „Volkszählungsurteil“ an Bedeutung gewonnen. Während der Volkszählung in Deutschland 1983 gab es große Bedenken seitens der Bevölkerung, ob die erhobenen Daten nicht für weitere als den eigentlichen Zweck der Volkszählung verwendet werden: Die Menschen fürchteten sich vor Missbrauch ihrer persönlichen Daten. Das Volkszählungsurteil schlug sich damals im Volkszählungsgesetz (VZG) nieder, welches das Nachteilsverbot aussprach: „Angaben der Volkszählung nach § 2 Nr. 1 und 2 können mit den Melderegistern verglichen und zu deren Berichtigung verwendet werden. Aus diesen Angaben gewonnene Erkenntnisse dürfen nicht zu Maßnahmen gegen den einzelnen Auskunftspflichtigen verwendet werden.“[10]. Das Recht auf informationelle Selbstbestimmung wird dabei aus dem Grundgesetz abgeleitet: „Jeder hat das Recht auf die freie Entfaltung seiner Persönlichkeit, soweit er nicht die Rechte anderer verletzt und nicht gegen die verfassungsmäßige Ordnung oder das Sittengesetz verstößt“[11].

[9] aus §3 Abs. 1 Bundesdatenschutzgesetz (BDSG)
[10] Nachteilsverbot: §9 Abs. 1 Satz 2 VZG 1983

Datenschutz spielt bei der Kommunikation in digitalen Netzen eine besondere Rolle. Personenbezogene Daten können in digitalen Netzen viel einfacher und effektiver gesammelt und verarbeitet werden als in der analogen Welt; somit ließe sich hier mit deutlich geringerem Aufwand ein sehr großer Schaden anrichten, wenn diese Daten falsch behandelt werden. Deshalb muss im virtuellen Leben noch mehr auf Datenschutz geachtet werden als in der realen Welt. Dies betrifft sowohl den statischen Teil des Identitätsmanagements, also die Speicherung und Verwaltung der Daten, als auch die Übertragung der Daten zu Kommunikationspartnern, also der dynamische Einsatz digitaler Identitäten.

Das Bundesdatenschutzgesetz bestimmt, dass digitale Dienste auf die Datensparsamkeit achten *müssen* und nicht mehr Daten erheben dürfen, als für die Erbringung des Dienstes notwendig ist: „Gestaltung und Auswahl von Datenverarbeitungssystemen haben sich an dem Ziel auszurichten, keine oder so wenig personenbezogene Daten wie möglich zu erheben, zu verarbeiten oder zu nutzen. Insbesondere ist von den Möglichkeiten der Anonymisierung und Pseudonymisierung Gebrauch zu machen, soweit dies möglich ist und der Aufwand in einem angemessenen Verhältnis zu dem angestrebten Schutzzweck steht.“[12]. Allerdings ist bei digitalen Dienstleistungen streitbar, welche Daten für die Erbringung derselben notwendig ist: Der Dienstleister kann argumentieren, dass der Dienst durch umfangreichere Datenerhebungen verbessert werden könnte – wenn auch nur durch Verbesserung der Werbung für den Dienst.

Der Schutz der Privatsphäre (engl.: privacy) hängt eng mit dem Datenschutz zusammen. Unter Privatsphäre versteht man einen Bereich, in dem Privatpersonen unbeobachtet und unbehelligt leben, handeln und kommunizieren können. In diesem Bereich sollen sie selbst bestimmen können, wer sie sieht und ihnen zuhören kann. Privatsphäre wird häufig mit Datenschutz verwechselt, so dass das teilweise berechtigte Bedürfnis von Menschen, ihre Privatsphäre mit anderen zu teilen, durch zu starken Datenschutz eingeschränkt wird. Beim Schutz der Privatsphäre sollte im Gegensatz zum Datenschutz stärker im Vordergrund stehen, dass der Betroffene nicht automatisch unsichtbar werden soll, sondern die Kontrolle darüber erhalten sollte, wer welche Eigenschaften von ihm sehen kann. Diese Unterscheidung zwischen dem Schutz der Privatsphäre und dem Datenschutz sollte in einem Identitätsmanagement-System besondere Beachtung finden.

2.4. Sitzungen als Gruppierungsmechanismus von Kommunikationsvorgängen

In Unterabschnitt 1.3.5 wurde motiviert, dass Kommunikationsteilnehmer sowie deren Äußerungen – insbesondere ihre Äußerungen persönlicher Daten – gruppiert

[11] Art. 2 Abs. 1 GG in Verbindung mit Art. 1 Abs. 1 GG
[12] BDSG § 3a (Datenvermeidung und Datensparsamkeit)

werden sollten. Dieser dynamische Aspekt des Einsatzes oder Gebrauchs digitaler Identitäten spielt für das Gestalten eines Identitätsmanagement-Systems eine zentrale Rolle. Ein geeignetes Konstrukt zur Gruppierung der Teilnehmer und Aussagen sowie zur Schaffung eines Kontextes, in dem sich die Teilnehmer wiederfinden und bewusst selbst darstellen können, ist die *Sitzung*. Das Konzept der *Sitzung* wird in der Informatik vielfach verwendet, jedoch selten explizit definiert. Im WWW wird hauptsächlich das zustandslose Protokoll HTTP (definiert und beschrieben in Berners-Lee u. a. 1996; Fielding u. a. 1999) verwendet, welches kein Konzept einer Sitzung einbindet. Da diese jedoch von vielen Anbietern im Netz benötigt wird, um die Funktionalität ihrer Web-Seiten zu gewährleisten, wurde nachträglich eine Sitzung eingeführt, die auf verschiedene Arten implementiert werden kann. Dadurch sind diese Sitzungen untereinander nicht kompatibel, und eine dienstübergreifende Sitzung ist nur in wenigen Fällen möglich. Eine weitere Anwendung des Konzepts Sitzung ist im *Session Initiation Protocol* (definiert in Rosenberg u. a. 2002) beschrieben, welches den Aufbau von Multimedia-Sitzungen auf Anwendungsebene beschreibt. Innerhalb einer solchen Sitzung können die Teilnehmer mittels spezieller Anwendungen eine Audio- oder Videokonferenz abhalten. Auf das entsprechende Medium einigt man sich mit Hilfe des Protokolls. Der Anwendungsbereich dieser Sitzungen ist also scharf umrissen und nicht offen für weitere Anwendungen.

Selbstdarstellung ist in den seltensten Fällen ein Selbstzweck. Im realen wie im digitalen Leben versucht man meistens deshalb ein bestimmtes Persönlichkeitsbild beim Kommunikationspartner zu erzeugen, um die eigentliche Kommunikation, die neben dem Identitätsdaten-Austausch stattfindet, zu beeinflussen. Das Persönlichkeitsbild, das man beim Gegenüber erzeugt hat, besteht mindestens so lange wie die dazugehörige Kommunikation zusammenhängend stattfindet. Zusammengehörige Kommunikationsakte und dazu assoziierte Teil-Identitäten werden auch hier in Sitzungen zusammengefasst (vgl. Baier und Kunze 2004a). Diese Sitzungen sollten applikations-, dienst- und verbindungsunabhängig sein, damit verschiedene Kommunikationspartner in verschiedenen Kommunikationsnetzen mit verschiedenen Anwendungen an der gleichen Sitzung teilnehmen können. Jede Kommunikationsapplikation sollte auf der gleichen Sitzungs-Infrastruktur aufbauen, damit die Sitzungen selbst für alle Applikationen die gleichen Dienste, insbesondere die Identitätsdienste, bieten können. Aber auch Sicherheitsaspekte wie Verschlüsselung und Datenschutz könnten so in einem Konzept gebündelt werden. Eine Datenschutzvereinbarung könnte den Geltungsbereich einer Sitzung haben, so dass nicht bei jeder Kommunikation über eine solche *policy* abgestimmt werden muss. Auch sollten möglichst alle benutzten Dienste über diese Sitzungs-Infrastruktur erreichbar sein, damit Informationsaustausch zwischen den Diensten erleichtert oder teilweise auch erst ermöglicht wird.

Definition 2.2. Eine *Sitzung* ist eine Bündelung von Kommunikationsakten und Verweisen auf die Darstellungen der Kommunikationspartner. Ferner haben Sitzungen verschiedene, kontextabhängige Eigenschaften wie Anfangszeitpunkt und Dauer, Verschlüsselungsart und Verschlüsselungsgrad oder Beitrittsregelung. Eine Liste der Kommunikationspartner samt ihrer Endpunkte ist die einzige Eigenschaft, die auf alle Sitzungen zutrifft.

Diese Definition der Sitzung bietet ein Konzept, bei dem mehrere Sitzungsteilnehmer über verschiedene Anwendungen und verschiedene Transport-Protokolle an einer Sitzung teilnehmen können. Die Sitzungsteilnehmer sind dabei als Teil-Identitäten zu denken, die nur für diese Sitzung bestehen – die Sichtbarkeit ist also an diese Sitzung gekoppelt. Wiedererkennbarkeit und Wiederverwendung von Teil-Identitäten wird durch das Binden „fester“ Teil-Identitäten an Sitzungs-Identitäten erreicht.

3. Konzepte und Mechanismen des Identitätsmanagements

Nachdem in Kapitel 2 die grundlegenden Konzepte des digitalen Kommunikationslebens als digitale Identitäten und Sitzungen in digitalen Netzen identifiziert und erläutert worden sind, soll in diesem Kapitel die Verwaltung, Pflege, Kontrolle und der Umgang mit diesen Konzepten beleuchtet werden. Zusammenfassend werden diese Aktivitäten als *digitales Identitätsmanagement* bezeichnet. Sie werden grob in statische und dynamische Aufgaben gegliedert, wobei die Verwaltung und Pflege zu den statischen Aufgaben gehören, während Kontrolle und Umgang eher in dynamischen Kontexten zu sehen sind. Auf beiden Seiten findet man den Schutz der sensiblen Daten: sowohl bei der Verwaltung, Speicherung und Pflege als auch beim Umgang damit, also dem Versenden und Empfangen, Signieren und Verschlüsseln der Daten ist der Schutz der Privatsphäre und damit der persönlichen Daten von äußerster Wichtigkeit.

Zunächst soll in diesem Kapitel eine allgemeine Definition der Begriffe „Identitätsmanagement“ und „digitales Identitätsmanagement“ gefunden werden, wobei im folgenden Text beide Begriffe synonym für die digitale Variante gebraucht werden. Anschließend werden die Mechanismen und Konzepte beleuchtet, die zusammengesetzt das digitale Identitätsmanagement ergeben. Einige davon sind heute schon gebräuchlich, werden aber getrennt voneinander eingesetzt. Am Ende steht eine Zusammenfassung der vorgestellten Mechanismen, was eine Überleitung zum nächsten Kapitel ergibt, in dem die grundsätzlichen Komponenten eines digitalen Identitätsmanagement-Systems erläutert werden sollen.

3.1. Definition: Identitätsmanagement

Die Persönlichkeit des Menschen ist meist ein komplexes Gebilde, welches Eigenschaften und unterschiedliches Verhalten für verschiedenste Situationen und Umgebungen bereithält und entwickelt. Wie in Unterabschnitt 2.1.2 beschrieben setzt sie sich genau wie das eigene Persönlichkeitsbild bei anderen Menschen also aus mehreren inhaltlich nicht disjunkten Teil-Identitäten zusammen. Der Prozess der Auswahl oder Aktivierung einer jeweiligen Teil-Identität geschieht im realen Leben größtenteils unbewusst und automatisch, er hat jedoch auch bewusste Ausprägungen wie die Auswahl einer bestimmten Visitenkarte oder Kleidung, eines besonderen Sprachstils oder einer ausdrucksvollen Mimik. Das Prinzip der Teil-

Identitäten findet sich im digitalen Leben wieder – gerade in digitalen Netzen wird durch die Kanalreduktion sehr deutlich, dass nur bestimmte Aspekte der jeweiligen Person abgebildet werden können. Dennoch werden etliche Eigenschaften und Verhaltensweisen von Personen in der digitalen Welt verwendet. Diese digitalen Teil-Identitäten sind nicht mit den Teil-Identitäten der wirklichen Welt gleichzusetzen, wohl aber vergleichbar. Die Unterschiede zeigen sich vor allem in der einfachen informationstechnologischen Möglichkeit zur Bearbeitung der digitalen Teil-Identitäten im Gegensatz zu ihren realen Analogen. Dieser Unterschied hat Einfluss auf den Einsatz der Attribute sowie auf die Möglichkeiten zum Schutz, zur Weiterverarbeitung und zur Verbreitung.

Definition 3.1. *Identitätsmanagement* fasst die statischen Aufgaben der Verwaltung und Pflege sowie die dynamischen Aufgaben der bewussten, situations- und kontextabhängigen Auswahl und kontrollierten Übermittlung persönlicher Daten an Kommunikationspartner zusammen.

Definition 3.2. Werden beim Identitätsmanagement *digitale Identitäten* (siehe Definition 2.1) verwaltet, so spricht man von *digitalem Identitätsmanagement*.

Eine ähnliche Definition stammt aus Pfitzmann und Hansen (2004) und betont die Rollenauswahl und die Rollengestaltung:

> Identity Management means managing the various partial identities, i.e. their valuation as 'applicable to oneself' (role taking) or forming them (role making). A prerequisite to choose the appropriate partial identity is to recognize the situation the person is acting in.“ (Pfitzmann und Hansen 2004, S. 13)

Dort wird der Schutz persönlicher Daten in einer gesonderten Definition von „*privacy enhancing* identity management“ vorgenommen, wobei ein Identitätsmanagement „*perfectly* privacy enhancing“ heißt, wenn ein Angreifer durch die Verwendung der Pseudonyme nicht mehr Verknüpfungen unter den Teil-Identitäten herausfinden kann als wenn die Pseudonyme weggelassen werden würden. Für den einfachen „privacy enhancing“-Status wird gefordert, dass die Verknüpfbarkeit nicht wesentlich steigt.

3.2. Mechanismen zum Verwalten von Identitätsdaten

„Management“ bedeutet direkt übersetzt „Verwaltung“. Insofern kann man den in der englischen Fachliteratur gebräuchlichen Begriff „identity management“ als Verwaltung von Identitäten beziehungsweise Identitätsdaten übersetzen, auch wenn dies die dynamischen Aspekte des Identitätsmanagements außen vor lässt.

Verwalten beinhaltet verschiedene Aufgaben, im Wesentlichen das Eingeben, Speichern, Aktualisieren und Abrufen der Daten. Diese Aufgaben werden durch verschiedene Mechanismen unterstützt. In diesem Abschnitt soll betrachtet werden, welche Konzepte zum Verwalten von Identitätsdaten gehören sowie welche davon schon Anwendung finden.

3.2.1. Datenhaltung und Codierung

Das vielleicht grundlegendste Konzept des Identitätsmanagement ist die Datenhaltung. Um Identitätsdaten zu verwalten, muss man sie auf irgendeine Weise speichern und zugreifbar machen. Während dies für Identitätsdaten von Kontakten sehr oft in strukturierter Weise passiert (zum Beispiel in Adressbuch-Programmen), werden Personendaten, welche die eigene Identität betreffen, eher selten zusammenhängend und strukturiert gespeichert. Vielmehr werden einzelne Datensätze über verschiedene Systeme verteilt. Während also die eigenen Kontaktdaten (Adresse, Telefonnummer u.a.) meistens noch neben den fremden Kontaktdaten im Adressbuch-Programm gespeichert werden, sind weiterführende Identitätsdaten selten dort aufgeführt. Dazu gehören alle Arten von persönlichen Daten, zum Beispiel WWW-Bookmarks, Passwörter und Benutzerkennungen, Listen von erworbenen Büchern oder Audio-Daten, Präferenzen jeglicher Art oder Aussagen anderer über die eigene Person. Die Codierung dieser Daten ist zur Zeit anwendungsspezifisch und schlecht austauschbar. Für Bookmarks gibt es zwar Import-Möglichkeiten in Nachfolgeversionen des gleichen Browsers, für den Austausch von Bookmarks zwischen verschiedenen Browsern (beispielsweise Netscape Navigator und Microsoft Internet Explorer) werden spezielle Konvertierungstools gebraucht.

3.2.2. Teil-Ontologien zur semantischen Strukturierung und Beschreibung von Identitätsdaten

Während schon die syntaktische Strukturierung der Daten zum eigenen Gebrauch Schwierigkeiten bereitet, wird das Problem bei der Übertragung der Daten noch deutlicher: Wie kann sichergestellt werden, dass der Empfänger einer Identitätsdatenaussage oder -anfrage versteht, welches Datum gemeint ist?

Wie in Unterabschnitt 2.2.3 dargestellt werden Probleme dieser Art in der Informatik häufig durch den Einsatz von Ontologien gelöst. Im Zusammenhang mit Identitätsmanagement bedeutet dies, dass die Daten einer Teil-Identität einzelnen Teil-Ontologien zugeordnet werden und nicht unstrukturiert in einem Verzeichnis abgelegt werden. Durch die Teil-Ontologie lässt sich voraussagen, welcher Struktur die Daten folgen und was sie bedeuten. Dies erleichtert den Zugriff und die Darstellung ebenso wie die Übertragung der Daten. Die Aufteilung der semantischen Definition der Daten in Teil-Ontologien hat den Vorteil, dass jede

Anwendungsdomäne eine eigene Teil-Ontologie definieren kann, die in Systemen von Benutzern, die keinen Bezug zu dieser Anwendungsdomäne haben, nicht vorkommen müssen (siehe dazu auch Kaulbarsch 2005).

3.2.3. Anwendungen der Identitätsdatenverwaltung

Das Konzept der Identitätsdatenverwaltung wird heute in verschiedensten Anwendungen um- und eingesetzt. Diese Anwendungen teilen sich in den seltensten Fällen die Identitätsdaten, woraus sich eine mehrfache Verwaltung und damit die Gefahr der Inkonsistenz ergibt. In diesem Abschnitt sollen zwei der derzeit eingesetzten Mechanismen betrachtet werden. Dabei betrifft der erste Mechanismus die lokale Speicherung der Daten, der zweite dagegen die Speicherung und Assoziierung der Daten auf Seiten des Servers.

3.2.3.1. Passwort-Speicher

Eine spezielle Ausprägung der Datenhaltung ist der Passwort-Speicher. Hier werden nicht beliebige Identitätsdaten gehalten, sondern nur Daten, die zur Authentifizierung von digitalen Diensten erforderlich sind.

Authentifizierungsdaten sind sehr sensible Identitätsdaten. Benutzer authentifizieren sich gegenüber Internetdiensten meist mit einer Benutzerkennung und einem Passwort. Mit dieser Kombination kann jeder den entsprechenden Dienst unter dem Namen des Benutzers in Anspruch nehmen. Fortschrittlichere und sicherere Methoden, wie sie zum Beispiel durch Biometrie, also dem Auswerten physischer Daten des Benutzers (beispielsweise Fingerabdruck oder Irisgeometrie sowie Gesichtserkennung), finden erst beschränkten Einsatz in sicherheitskritischen Bereichen wie dem Flugverkehr oder Militäranlagen. Selbst hier werden diese Methoden eher für den körperlichen Zugang zu bestimmten Orten (z.B. dem Flughafenterminal) verwendet als für digitale Dienste (vgl. Woodward u. a. 2002). Desweiteren können Chipkarten oder Smartcards samt einer PIN verwendet werden, um sich einem System gegenüber zu authentifizieren. Da Chipkartenleser für den Heimgebrauch jedoch noch zu teuer sind und zu selten verlangt werden, hat sich diese Methode bisher nicht durchgesetzt.

Benutzername und Passwort sind also die wichtigsten Authentifizierungsmerkmale in digitalen Systemen. Da teilweise viele verschiedene Dienste angeboten und auch genutzt werden (wie zum Beispiel im Internet, wo man sich quasi auf jeder größeren Web-Site mit Benutzerkennung und Passwort einloggen kann), und es verständlicherweise als unsicher erachtet wird, wenn man in jedem System die gleiche Nutzerkennung und das gleiche Passwort benutzt, müssen die Benutzer von digitalen Diensten mit sehr vielen dieser Wertpaare umgehen. Zusätzlich gilt, dass Passwörter keine Wörter aus einem Wörterbuch sein sollten, weil diese besonders einfach herauszufinden sind.

Es gilt also, sich viele verschiedene, möglichst kryptische Passwörter zu merken. Um diese Aufgabe zu erleichtern, hat sich das Konzept des Passwort-Speichers an vielen Stellen durchgesetzt. Ein Passwort-Speicher merkt sich für verschiedene Dienste das Kennung/Passwort-Paar und setzt es automatisch ein, sobald der Benutzer den Dienst in Anspruch nehmen will. Teilweise erfordern diese Speichersysteme selbst ein Passwort, um darauf zugreifen zu können, aber einige Systeme bieten diese Sicherheit nicht.

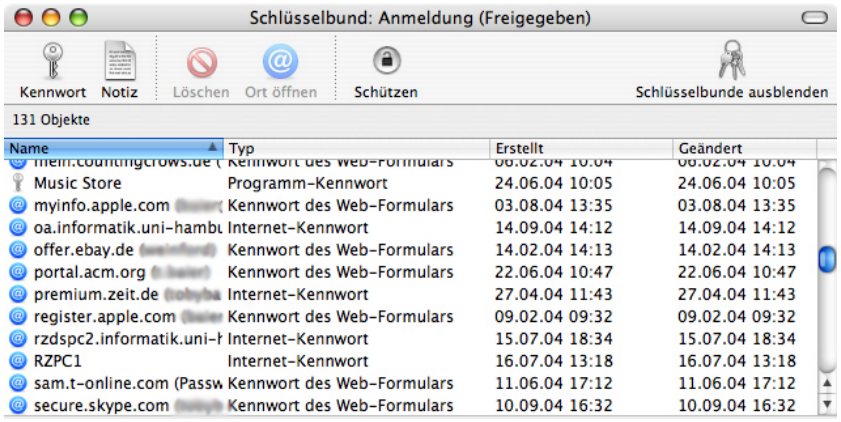

Abbildung 3.1.: Screenshot eines Passwort-Speichers

Passwort-Speicher können auf verschiedenen Ebenen angesiedelt sein. Sie können in einen Web-Browser integriert sein, in diesem Fall sind sie selten selbst Passwort-geschützt[1], außer durch das Passwort des Benutzeraccounts auf Betriebssystem-Ebene, welches gebraucht wird, um Zugang zum Browser und den darin gespeicherten Passwörtern zu haben. Hier werden Passwörter auch lediglich für solche Dienste gespeichert, die über den Browser genutzt werden können, also zum Beispiel Anmeldungen in Internet-Foren, Online-Shops oder FTP-Server.

Auf einer anderen Ebene sind Standalone-Passwort-Speicher angesiedelt. Der Zugriff zu diesen wird ebenfalls über den Account im Betriebssystem geregelt, zusätzlich kann bei einigen Systemen ein weiteres Passwort vergeben werden, oder es wird bei Zugriff auf den Passwort-Speicher das System-Passwort erneut abgefragt. Hier können nicht nur Web-Passwörter gespeichert werden, sondern auch solche für allgemeine Dienste im Netz wie Drucken, Netzwerklaufwerke zum Speichern von Daten und Kennworte für den Zugriff auf bestimmte Programme wie beispielsweise Instant Messenger.

Eine weitere Ebene für Passwort-Speicher ist das Betriebssystem selbst, zumindest der Teil, der das Einbinden von Netzwerk-Druckern und -Laufwerken ermög-

[1] Mozilla bietet die Option eines Master-Passwortes, ohne das nicht auf die gespeicherten Passwörter zugegriffen werden kann.

licht. Auch hier können Benutzerkennungen und Passwörter gespeichert werden, damit sie nicht jedesmal neu eingegeben werden müssen.

Das Konzept des Passwort-Speichers ist nicht zu verwechseln mit dem Konzept des Single Sign-Ons (siehe dazu auch Abschnitt 5.1). Beim Single Sign-On gibt es nur eine Kennung und ein Passwort. Die Stelle, bei der man sich authentifiziert hat, überträgt diese Authentifizierung auf andere Dienste, die damit einen gemeinsamen Raum der Authentifizierung bilden (vgl. Abbildung 3.2). Bei Single Sign-On Systemen ist es notwendig, dass alle eingebundenen Dienste einem bestimmten Standard des automatischen Log-Ins folgen, dies ist bei der Benutzung von Passwort-Speichern nicht unbedingt notwendig. Anders gesagt muss man Dienste für den Gebrauch mit Passwort-Speichern nicht anpassen, da beim Einloggen das Passwort automatisch eingetragen wird – bei Single Sign-On hingegen muss der Dienst ebenso wie das Zugriffsprogramm angepasst werden.

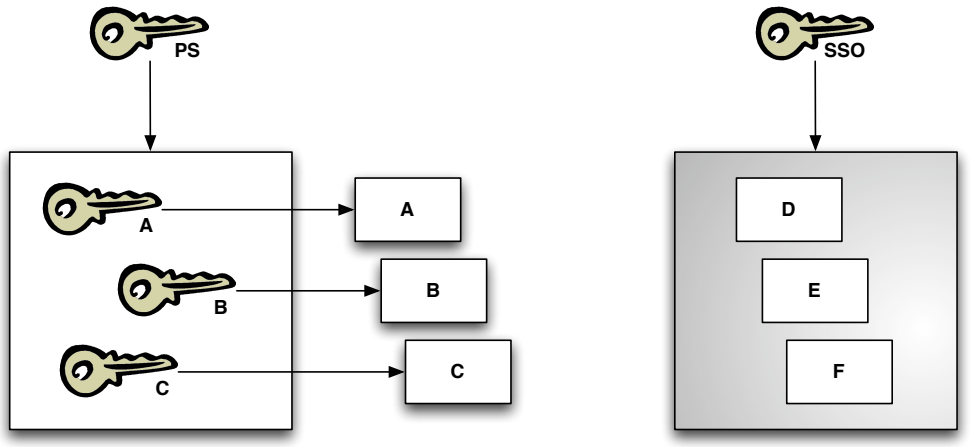

Abbildung 3.2.: Links wird ein Passwort-Speicher dargestellt, der drei Schlüssel für weitere Dienste enthält. Die Schlüssel für A, B und C können unterschiedlicher Art sein. Rechts ist ein Single Sign-On System mit einem Schlüssel dargestellt, der Authentifizierung für mehrere Dienste gleichzeitig bietet.

3.2.3.2. Cookie-Management

Beim Surfen im World Wide Web erstellen Web-Server häufig Profile der Besucher, die sie auf dem Server speichern. Dazu werden häufig kleine Datensätze, genannt *Cookies*, auf dem Computer des Clients gespeichert[2]. Dieser Cookie kann

[2]Das Speichern der Cookies ist browserabhängig: einige Browser legen einzelne Dateien an, andere speichern alle Cookies in einer großen Datei oder Datenbank. Eigentlich wird aber keine Datei übertragen, sondern lediglich eine weitere Zeile im HTTP-Header.

vom selben Server bei einem späteren Besuch des Clients von jenem wieder ausgelesen werden. Dieser Mechanismus dient dazu, dass der Web-Server sich trotz des *zustandslosen* HTTP-Protokolls bestimmte Werte über den Client merken kann. In einem Cookie kann zum Beispiel eine Benutzerkennung gespeichert sein, damit der Web-Server auch bei gewechselter oder mit anderen Nutzern geteilter IP-Adresse einen wiederkehrenden Benutzer erkennen kann. Die Eigenschaften eines Cookies sind in der Tabelle 3.1 aufgeführt.

Tabelle 3.1.: Eigenschaften von Cookies

Attribut	Funktion
Name	Der Name des Cookies. Er kann frei gewählt werden. Beispiel: Benutzerkennung
Wert	Der Wert des Cookies. Beispiel: baier42
Verfalldatum	Datum, ab dem der Cookie nicht mehr gültig ist.
Pfad	Der (relative) Pfad, für den der Cookie gültig ist. Beispiel: /informationen/news
Domäne	Web-Domäne, für die der Cookie gültig ist. Beispiel: www.uni-hamburg.de
Sicherheit	Einstellung, ob für die Übertragung des Cookies eine sichere Verbindung (HTTPS) erforderlich ist.

Während Cookies nicht die Integrität des Client-Rechners gefährden, weil kein übertragener Code ausgeführt werden kann, sind sie doch eine Gefahr für die Privatsphäre des Web-Surfers. Dieser kann zwar bei den meisten Web-Browsern selbst entscheiden, ob er einen Cookie annehmen will oder nicht (entweder fallweise oder generell), aber das spätere Auslesen des Cookies geschieht immer unbeobachtet. Der Benutzer merkt also nicht, dass Daten, die zuvor bei ihm gespeichert worden sind, wieder herausgegeben werden. Hinzu kommt die Problematik, dass Daten zwar nur an Domänen herausgegeben werden, von denen sie ursprünglich gespeichert worden sind, die Einbindung von Fremd-Domänen in die Website jedoch das site-übergreifende Verfolgen von Benutzern erlaubt. Bekanntestes Beispiel für dieses Vorgehen ist die Werbung des Drittanbieters DoubleClick[3]. Web-Site-Betreiber binden bestimmte Code-Fragmente in ihre Web-Seiten ein, die Werbungsinhalte vom Server `ads.doubleclick.com` laden. Bindet ein anderer Web-Site-Betreiber auch ein solches Fragment ein, wird wieder ein Werbebanner vom DoubleClick-Server geladen, der dann den Cookie auswerten kann. DoubleClick kann also feststellen, wann ein Nutzer welche Seiten seiner Kunden

[3] `http://www.doubleclick.com`

aufgerufen hat. Dies ist ein Eingriff in die Privatsphäre des Nutzers, weil dieser so beim Surfen beobachtet werden kann. Deshalb hat DoubleClick als Reaktion auf heftige Proteste der Internet-Nutzer sein System so umgestellt, dass man sich von der Verfolgung ausnehmen kann.

Das Verwalten von Cookies kann also auch das Verwalten von Identitätsdaten sein, wenn personenbezogene Daten im Cookie gespeichert werden. Cookies können von verschiedenen Systemen verwaltet werden. In der Regel bieten Web-Browser eine einfache Möglichkeit, die Cookies zu Verwalten: zusätzlich zu den Optionen, Cookies generell anzunehmen oder abzulehnen, kann man auswählen, dass Cookies nur vom eigentlich besuchten Server akzeptiert werden (um zum Beispiel ein Abspeichern von Cookies von Dritten wie DoubleClick zu verhindern), oder dass bei jedem Versuch, einen Cookie zu speichern, der Benutzer gefragt wird, ob er akzeptiert werden soll. Bei einigen Browsern kann man in diesem Fall aus der Wahl eine Regel erstellen, so dass Cookies von dieser Domäne immer akzeptiert oder abgelehnt werden.

Weiterführende Produkte versuchen, die Gefährdung der Privatsphäre durch Cookies weiter zu mindern. Das Produkt „CookieCooker“[4] zum Beispiel tauscht die Cookies verschiedener Benutzer zufällig untereinander aus, um fehlerhafte Daten bei den Betreibern zu generieren. Diese können die erhobenen Daten dann eigentlich nicht mehr zu ihren Zwecken (meistens optimiertes Marketing) nutzen, jedoch ist ihnen die Fehlerhaftigkeit der Daten gar nicht bewusst, weil das Austauschen der Daten von ihnen unbeobachtet geschieht.

Allerdings ist das korrekte Funktionieren vieler Web-Sites von der korrekten Verwendung von Cookies abhängig. So ist zum Beispiel der Inhalt des virtuellen Warenkorbs bei vielen Online-Shops in Cookies gespeichert[5]. Durch sofortiges Setzen und Lesen eines Cookies beim ersten Zugriff auf eine Web-Site kann der Betreiber feststellen, ob Cookies auf dem Client-Browser aktiviert sind und falls nötig eine Warnmeldung anzeigen.

Abschließend kann gesagt werden, dass das Verwalten von Cookies nur indirekt zum Identitätsmanagement gehört, da die verwalteten Identitätsdaten nur teilweise in den Cookies selbst abgelegt sind. Größtenteils besteht das Risiko für die Privatsphäre nur aus den Meta-Informationen, die den Cookies anhaften und nur auf Web-Server-Seite sichtbar sind. Das Verwalten der Meta-Informationen selbst ist den Dienstanbietern vorbehalten.

[4] `http://www.cookiecooker.de`

[5] Eigentlich ist nicht der Inhalt des Warenkorbs im Cookie gespeichert, sondern nur eine Kennung, über die der Warenkorbinhalt im Datenbanksystem des Anbieters gefunden werden kann.

3.2.4. Wiedererkennbarkeit

Ein wichtiges Konzept des Identitätsmanagements ist die Wiedererkennbarkeit. Wie schon in Unterabschnitt 3.4.2 erwähnt, können Pseudonyme und damit identifizierbare Teil-Identitäten mit Unterstützung des Systems halbautomatisch ausgewählt werden. In einigen Situationen möchte man dies nutzen, um die Privatsphäre zu schützen und dem Kommunikationspartner möglichst wenig über die eigene Identität oder die eigenen persönlichen Daten zu verraten. In bestimmten Fällen ist jedoch darauf zu achten, dass man eine wiedererkennbare Teil-Identität auswählt. Dies trifft auf alle Situationen zu, in denen man eine bestimmte, beispielsweise mit Reputation oder anderen Eigenschaften ausgestattete Rolle einnehmen möchte. Dies ist nur möglich, wenn die Teil-Identität oder zumindest das Pseudonym eindeutig wiedererkennbar ist und von anderen Kommunikationspartnern als solche wahrgenommen werden kann. Dieser Aspekt der *linkability* wird in der Identitätsmanagement-Forschung größtenteils ignoriert, da hier eher auf die *unlinkability* geachtet wird (ein gutes Beispiel für das Bestreben nach unlinkability findet sich bei Li u. a. (2003)).

Auch im privaten Bereich ist Wiedererkennbarkeit von großem Nutzen, wenn man aufgebaute Kontakte weiterhin nutzen möchte. Erst dadurch wird ein spontanes Treffen zweier Internet-Nutzer auf einer Web-Site durch Wissen über den jeweils anderen angereichert, welches man vorher schon bekommen haben kann.

3.3. Dynamische Aspekte des Identitätsmanagements

Nachdem im vorigen Abschnitt im Wesentlichen die Aspekte der statischen Identitätsdatenverwaltung beleuchtet worden sind, sollen hier die dynamischen Aspekte des Einsatzes von Identitätsmanagement in Kommunikationssystemen genau untersucht werden. Es geht also um konkrete Kommunikationssituationen, in denen persönliche Daten ausgetauscht werden. Untersucht wird der Kontext und der Zweck des Identitätsdatenaustauschs, sowie die Notwendigkeit, bestimmte Kommunikationsssschritte zu koppeln und zu gruppieren.

3.3.1. Übermittlung von Identitätsattributen

In vielen Situationen ist es wünschenswert oder erforderlich, Kommunikationspartnern persönliche Eigenschaften mitzuteilen. Sei es zur Personalisierung eines Dienstes oder zur genaueren Selbstdarstellung gegenüber einem privaten Kontakt: In jedem Fall geht es um die digitale Übermittlung personenbezogener Daten. Dabei kann die Übertragung eigeninitiiert sein und somit der geplanten Selbstdarstellung entsprechen, oder eine Antwort auf eine Identitätsdatenanfrage der Gegenstelle sein. Die Übermittlung von Identitätsattributen ist die Grundlage für Selbstdarstellung, Identifikation und Authentifikation. Genau wie Selbstdar-

stellung im realen Leben kann sie in der digitalen Welt bewusst und gezielt oder unbewusst und ungezielt geschehen. Während ersteres leicht nachzuvollziehen und der Regelfall ist, hilft bei letzterem ein Beispiel zum Verständnis: eine unbewusste, ungezielte Übermittlung von Identitätsdaten kann die automatische Preisgabe einer indirekt verwalteten Eigenschaft sein. So kann das System so konfiguriert sein, dass zum Beispiel die Ortszeit beim Anwender automatisch allen Kommunikationspartnern – also ungezielt – zur Abfrage freigegeben wird. Dieses Attribut gehört zur Teil-Identität dazu, ist jedoch nicht direkt eingegeben oder verwaltet. Durch die generelle Freigabe geschieht die konkrete Übermittlung zum Kommunikationspartner für den Anwender unbewusst.

3.3.1.1. Identifikation und Authentifikation

Für viele digitale Dienste muss man sich als berechtigter Nutzer authentifizieren. Dies unterscheidet sich von einer einfachen Identifikation durch den Zusatz der Verifikation der Identität, also des Beweises, dass man tatsächlich der vorgegebenen Identität entspricht. Dieser Vorgang gestaltet sich in digitalen Situationen deutlich schwieriger als im realen Leben, da die hier häufig verwendete persönliche Bekanntheit wegen fehlender direkter Sichtbarkeit nicht anwendbar ist. Stattdessen muss auf Authentifikation über Passworte oder Credentials vertraut werden, was den Ausweisen im realen Leben entspricht.

Eine wichtige Unterscheidung hierbei ist zwischen der Authentifikation einer digitalen Identität und einer realen Identität zu treffen. Die Authentifikation einer digitalen Identität lässt sich anhand einer digitalen Signatur verifizieren, die mit einem einer digitalen Identität zuordnenbaren Schlüssel erstellt ist. Die Authentizität einer realen Identität ist jedoch deutlich schwieriger zu belegen. Hier ist der direkte, persönliche Austausch der Schlüssel oder ein von einer sehr vertrauenswürdigen dritten Instanz (beispielsweise einem Amt oder einer Bank) signierter Schlüssel von Nöten.

Identifikation kann mittels Identitätsmanagement auf zwei verschiedenen Ebenen realisiert werden: zunächst können Benutzernamen oder andere identifizierende Attribute als Identitätsdaten in Teil-Identitäten gespeichert werden. Dies bietet sich an, wenn bestehende Systeme oder schon vorhandene Benutzerkennungen in das Identitätsmanagement-System eingebunden werden sollen. Desweiteren könnte aber auch die eindeutige Kennung der Teil-Identität selbst (siehe Definition 2.1) als Identifikator für den Dienst genutzt werden. Dieser Weg sollte von Diensten verwendet werden, die explizit zur Nutzung mit dem Identitätsmanagement-System entwickelt werden.

Authentifikation kann wie erwähnt über jegliches bestehendes Authentifikationsschema geschehen. Das notwendige Passwort oder Credential wird als Attribut einer Teil-Identität zugeordnet. Am Beispiel von Kerberos (definiert von Kohl und Neuman 1993) kann dies bedeuten, dass der geheime Schlüssel, der mit dem

Kerberos Authentication Server (AS) geteilt wird, in einer Teil-Identität abgelegt wird, und sobald man auf einen Kerberos-Dienst zugreift, wird automatisch ein Kerberos *Ticket-Granting Ticket* angefragt. Andererseits kann während einer Sitzung ein Ticket-Granting Ticket in einer Teil-Identität abgelegt werden, wobei dieses natürlich eine geringere Lebensdauer hat.

3.3.2. Rollenauswahl

Den Akt der Präsentation von eigenen Identitätsmerkmalen gegenüber Kommunikationspartnern zum Zwecke des Erstellens eines bestimmten Persönlichkeitsbildes nennt man auch Selbstdarstellung. Die Auswahl der zu übertragenden Merkmale bestimmt die Rolle, welche man in der Kommunikationssitzung einnehmen kann, also kann hier von einer Rollenauswahl (siehe auch Unterabschnitt 2.1.4) gesprochen werden. Die gewünschte Rolle des Benutzers kann sowohl vom Kontext als auch vom Bestreben des Nutzers abhängen – welche Aufgabe ist zu bewerkstelligen, in welcher Stimmung ist der Nutzer, welche anderen Kommunikationspartner sind beteiligt? Die Auswahl der Rolle kann bestimmend sein für das Ergebnis der Sitzung, insofern kann es manchmal von entscheidender Wichtigkeit sein, die richtige Rolle zu wählen. Der Nutzer sollte vom System dahingehend unterstützt werden, dass ihm verdeutlicht wird, welche Rollen zur Verfügung stehen (zum Beispiel in einer Online-Community: Moderator des Forums, normales Mitglied oder anonymer Besucher) und wie der Kontext (beispielsweise die anderen Kommunikationspartner) beschaffen ist. Dies ist eine sehr anspruchsvolle Aufgabe für ein Identitätsmanagement-System, da es wenig über das Bestreben des Nutzers wissen kann.

Sehr deutlich ist, ja sogar wörtlich genommen wird das Konzept der Rollenauswahl bei den so genannten *Rollenspielen*. Diese werden sowohl in der Psychotherapie und (Sozial-)Pädagogik eingesetzt als auch zum Zeitvertreib gespielt – jeweils mit verschiedenen Vorgehensweisen und Zielsetzungen. Bei letzterem wählt ein Spieler eine meist fiktive Rolle aus, die er während der Dauer des Spiels innehält. Auch im Internet wird diese Form des Spiels gespielt: hier hilft die schnelle digitale Kommunikation dabei, Spielpartner auf der ganzen Welt miteinander zu verbinden. Zunächst gab es lediglich textbasierte Systeme (so genannte MUDs[6], ausführlich dokumentiert von Döring (2003)), bei denen die Spieler sich in einer virtuellen Welt bewegen und agieren konnten, die ihrerseits Geschehnisse anzeigen konnte. Heutzutage gibt es Massively Multiplayer Online Role Playing Games (MMORPGs), bei denen sowohl die Welt als auch die Spieler visuell dargestellt werden. Diese Spiele erfreuen sich großer Beliebtheit, vor allem im asiatischen Raum sind Rollenspiele wie „Lineage“ ein großer kommerzieller Erfolg. Bei diesem Spiel kann man sich eine Figur aus der Fantasy-Welt ausdenken und mit

[6]MUD (Abk.): Multi-User-Domain oder Multi-User-Dungeon

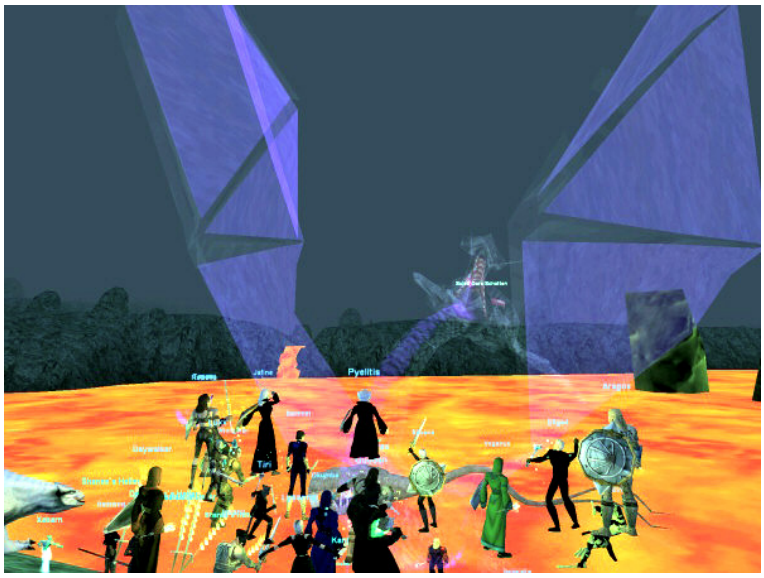

Abbildung 3.3.: Screenshot aus dem MMORPG „EverQuest“ mit spielergesteuerten Figuren im Vordergrund und einem rechnergesteuertem Drachen

ihr eine Rolle gestalten. Diese kann sich im Beruf, in den Fertigkeiten und in einer Gruppenzugehörigkeit (Gilde) ausdrücken. Die Auswahl der Rolle findet hier sowohl bei der Wahl des *Characters* (so wird eine virtuelle Figur in einem MMORPG genannt) statt als auch beim Zusammenschließen mit anderen Spielern, denn jede Figur hat ihrerseits verschiedene Fertigkeiten, die in einer Gruppe sinnvoll eingesetzt werden können.

Das Konzept der Rollenauswahl ist also bei Rollenspielen sehr ausgeprägt und wird bewusst eingesetzt. Die Mechanismen, die den Spielern zur Unterstützung angeboten werden, könnten bei der Gestaltung von Identitätsmanagement-Systemen genutzt oder weiterentwickelt werden, da diese Systeme ebenfalls stark auf Rollenauswahl basieren.

3.3.2.1. Anwendung: Automatisches Ausfüllen von Formularen

Eine Anwendung, die WWW-Benutzer bei der Rollenauswahl unterstützen, ist das automatische Ausfüllen von Web-Formularen. Bei diesem auch als *form filling* bezeichneten Vorgang merkt sich der Browser oder ein Zusatzprogramm die Eingaben in Web-Formulare und füllt zukünftige Formulare automatisch mit diesen Werten. Eine Rollenauswahl ist dann möglich, wenn das Programm mehrere Datensätze – beispielsweise eine Privat- und eine Geschäftsadresse – verwalten kann, die situationsabhängig gewählt werden können.

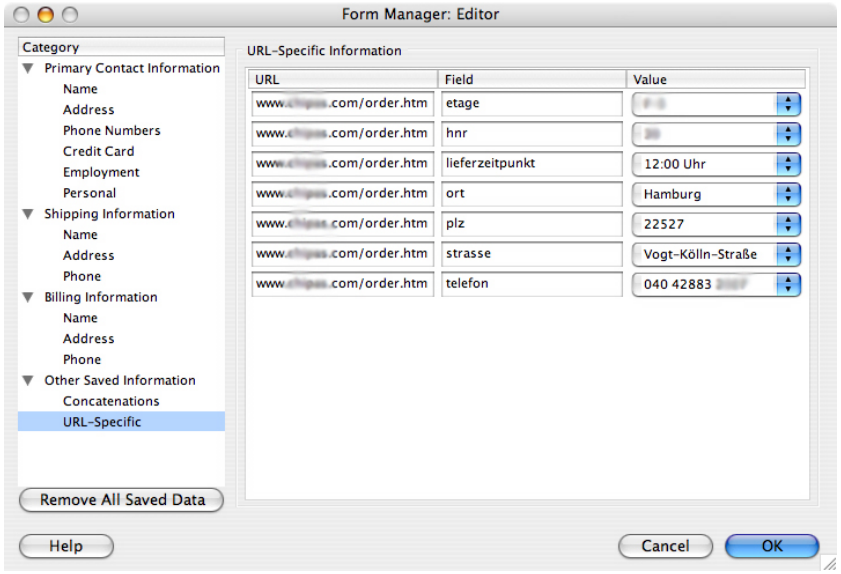

Abbildung 3.4.: Mit dem Web-Browser Mozilla kann man Eingaben in Web-Formulare verwalten

So genannte *Form Manager* bieten diese Funktion, sich Eingaben in Web-Formulare zu merken und diese bei einem später angezeigten Formular automatisch einzutragen (siehe Abbildung 3.4). Dieses Konzept dient der Bequemlichkeit der Nutzer, indem eine Arbeitserleichterung erreicht wird: der Benutzer muss weniger Daten von Hand eingeben. Insbesondere findet Form Management Anwendung mit Adressdaten, die in Web-Formularen eingegeben werden müssen. Allerdings ist der Nutzer darauf angewiesen, dass das Programm die richtige Adresse einträgt, denn viele Nutzer haben mehrere E-Mail-Adressen, oder es gibt eine Unterscheidung zwischen dienstlicher und privater (sowohl elektronischer als auch postalischer) Adresse. Hier ist eine starke Ähnlichkeit zum Konzept der Teil-Identitäten erkennbar: für verschiedene Kontexte werden unterschiedliche Daten des gleichen „Datentyps“ oder gleicher Art verwendet. Die Auswahl einer Teil-Identität beinhaltet auch die Zuordnung einer Rolle.

In Bezug auf den Datenschutz birgt dieses Konzept eher Gefahren als Vorteile. Das automatische Ausfüllen eines Formulars kann vom Benutzer übersehen werden, wodurch mehr Daten preisgegeben werden als gewünscht war. Dies kann sogar von Web-Formular-Gestaltern ausgenutzt werden, indem sie Formularfelder weit unten auf einer Web-Seite positionieren, die nur durch „scrollen“ sichbar werden, den Schalter zum Übertragen der Daten jedoch weiter oben. Auf diese Weise könnten Adressdaten ausspioniert werden. Es ist also notwendig, dass Form Manager dem Anwender jeden automatischen Eintrag in ein Formular anzeigen und

möglicherweise auch protokollieren, um später bei Missbrauch eine Verfolgung der Geschehnisse und Datenflüsse möglich zu machen.

3.3.3. Identitätsbildung

Bei allen Überlegungen, wie man Identitätsdaten verwaltet oder sicher überträgt, darf die Frage nach dem Ursprung dieser Daten nicht außer acht gelassen werden. Mit welchen persönlichen oder fiktiven Daten wird eine Teil-Identität gefüllt, oder besser ausgedrückt angereichert? Hier gibt es drei verschiedene Möglichkeiten (dargestellt auch in Abbildung 3.5):

- Übernahme von Daten aus bestehenden Systemen
- Eigenhändige Eingabe
- Übernahme von Aussagen anderer über die eigene Identität

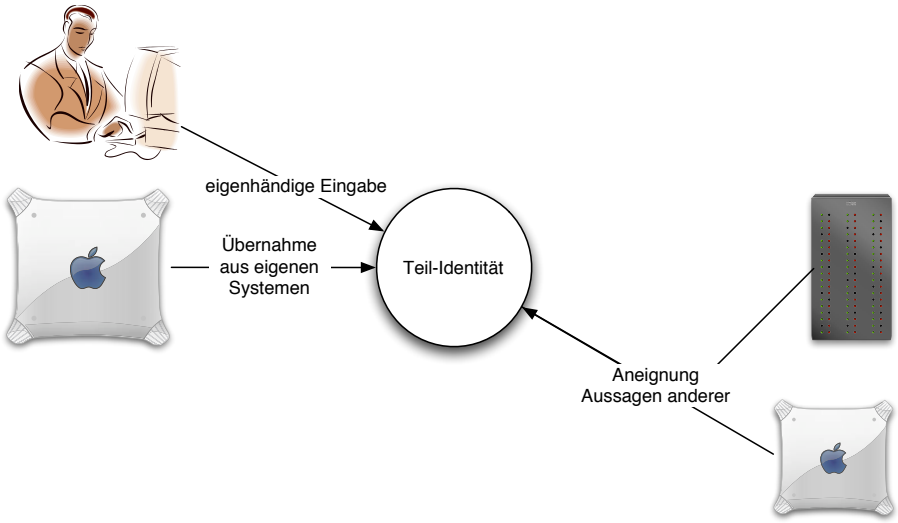

Abbildung 3.5.: verschiedene Ursprünge von Identitätsdaten

Zunächst können Daten aus anderen Systemen übernommen werden, in denen Identitätsdaten schon vorliegen. Dazu gehören andere Identitätsmanagement-Systeme im engeren Sinne, aber auch alle Systeme, die nur entfernt mit Identitätsmanagement zu tun haben. Zum Beispiel könnten die Lesezeichen[7] eines Web-Browsers Aussagen über die Präferenzen von Web-Seiten oder sogar spezielle Interessen ausdrücken. Dies sind persönliche Daten, die durchaus für Anwendungen

[7] im MS Internet Explorer heißen diese „Favoriten“, das englische „Bookmarks“ ist mittlerweile auch im Duden verzeichnet.

nützlich sein können, die auf eine Identitätsmanagement-Infrastruktur aufsetzen. Ein Beispiel dafür ist das *CoInternet-System*, welches in Abschnitt 5.4 beschrieben wird.

Persönliche Daten, die noch nicht erfasst worden sind oder in einer neuen Form vorhanden sein sollen, können vom Benutzer selbst eingegeben werden. Dazu sollte das System entsprechende Eingabehilfen in Form von Masken, Formularen oder geführte Dialoge (so genannte *Wizards*) bieten, die einen bei der Eingabe von einer Stufe zur nächsten führen und kontextsensitive Unterstützung anbieten. Hierbei ist vor Eingabe der Daten die Auswahl eines Schemas oder einer Taxonomie zu treffen, damit strukturierte, wiederverwendbare Daten vorliegen. Alternativ könnten solche Systeme mittels eines speziellen Parsers die Daten aus freitextähnlichen Passagen (beispielsweise persönlichen Homepages) extrahieren.

Zuletzt wird eine digitale Teil-Identität auch von Aussagen anderer geprägt. Dazu gehört der Aufbau einer Reputation genauso wie die Entgegennahme von Auszeichnungen oder Urkunden jeglicher Art. Beispielsweise könnte ein Einwohnermeldeamt eine signierte Adressinformation hinterlegen, mit welcher der Eigentümer der Teil-Identität eine belegbare Aussage über seinen Wohnort machen kann. Eine Auszeichnung kann auch eine abgeschlossene Ausbildung, ein akademischer Grad oder ein erreichtes Ziel in einem Online Spiel sein (beispielsweise eine ELO-Zahl[8] auf einem Schach-Server). Der Benutzer sollte die Möglichkeit haben, solche Aussagen in seine Teil-Identität einzubinden oder abzulehnen.

Zu beachten ist, dass eine Teil-Identität nicht unbedingt aussagekräftiger ist, je mehr Daten sie enthält. Hier herrscht das Prinzip: Qualität vor Quantität. Am Beispiel der erwähnten Web-Lesezeichen ist dies besonders deutlich: sammelt jemand alle Web-Seiten, auf denen er jemals irgendetwas bemerkenswertes gefunden hat, so ist dies eine eher große, unübersichtliche Informationsmenge. Wählt er jedoch gezielt genau die Web-Seiten aus, die für diese Teil-Identität die größte Rolle spielen, lassen sich dadurch viel einfacher Rückschlüsse auf seine Interessen ziehen.

3.4. Mechanismen zum Schützen von Identitätsdaten

Identitätsdaten müssen in ganz besonderem Maße vor unbefugtem Zugriff und Kompromittierung geschützt werden. Dies betrifft im Wesentlichen den Bereich der Kommunikation, aber auch die gespeicherten Daten müssen geschützt werden. Wie in Abschnitt 2.3 beschrieben unterliegen personenbezogene Daten einem besonderen rechtlichen Schutz. Auch die Privatsphäre von Personen ist nicht nur moralisch, sondern auch juristisch vor Eindringlingen geschützt (siehe dazu auch Abschnitt 2.3). Um diesen jedoch nicht in Anspruch nehmen zu müssen, soll-

[8] Ranglistensystem im Schachspiel, nach Prof. Arpad Emrick Elo (* 25. August 1903 bei Pápa (Ungarn), † 5. November 1992)

ten technische Vorkehrungen dafür sorgen, dass der Missbrauch möglichst stark eingeschränkt wird. Dafür gibt es verschiedene Vorgehensweisen, die in diesem Kapitel beleuchtet werden sollen. Burkert unterscheidet diese so genannten *Privacy Enhancing Technologies* in organisatorische und technische Systeme (Burkert 1997). Erstere regeln dabei den Informationsfluss durch organisatorische Maßnahmen, beispielsweise durch die Einführung von Datenschutzvereinbarungen oder Siegeln für Web-Seiten, welche die Privatsphäre der Besucher achten (zum Beispiel TrustE[9]). Technische Systeme stellen dagegen durch zum Beispiel Zugriffskontrolle, Verschlüsselung oder Anonymisierung sicher, dass persönliche Daten (beziehungsweise Informationen über Aktivitäten identifizierbarer Personen) nicht an unbefugte Dritte gelangen.

Während Datensparsamkeit ein größtenteils eigenverantworteter Bereich ist, kann auch hier technische Unterstützung erfolgen (siehe das Problem der Form Manager, welches in Abschnitt 3.3.2.1 beschrieben ist). Der Bereich der teilautomatischen Pseudonymität ist genau wie Verschlüsselungstechnologien vom System zu gewährleisten und durch den Nutzer nur kontrollierbar.

3.4.1. Datensparsamkeit

Mit dem Begriff *Datensparsamkeit* meint man den auf das nötige Maß beschränkten Umgang mit persönlichen Daten: es sollen so wenig persönliche Daten preisgegeben beziehungsweise an Kommunikationspartner übermittelt werden wie möglich. Hier ist zunächst der Benutzer gefordert. Er sollte darauf achten, dass er so wenig persönliche Daten wie möglich im Internet preisgibt. Aber ebenso sind Dienstanbieter gefordert, so wenig Daten wie möglich zu erheben. Wie schon in Abschnitt 2.3 beschrieben ist dies im Bundesdatenschutzgesetz festgeschrieben. Werden technische Systeme eingesetzt, um digitale Identitätsdaten zu übertragen, können diese so eingerichtet werden, dass sie nur genau die Daten übertragen, die für die Diensterbringung notwendig sind.

Datensparsamkeit ist im Zusammenhang mit Selbstdarstellung ein schwieriges Ziel: Einerseits will sich der Anwender selbst darstellen, um seine Ziele besser zu erreichen, andererseits sollen keine Daten übertragen werden, die zu diesen Zielen nicht erforderlich sind. Es muss also genau die Grenze getroffen werden. Wird zu sehr auf Datensparsamkeit geachtet, leidet die Qualität der Kommunikation – wird Selbstdarstellung zu weit getrieben, leidet der Datenschutz.

Zusätzlich können an dieser Stelle Mechanismen zum „Aushandeln“ der zu übertragenen Daten eingesetzt werden. Dabei kann die Menge der zu übertragenen Daten dynamisch an die Datenschutzbedürfnisse des Anwenders abhängig von der Datenschutzrichtlinie des Anbieters angepasst werden. Dieser Mechanismus wird in Unterabschnitt 3.4.3 beschrieben.

[9] `http://www.truste.com`

3.4.2. Anonymität und teilautomatische Pseudonymität

Quasi das höchste Maß an Datensparsamkeit ist die *Anonymität*. In diesem Zustand gibt man keinerlei personenbezogene Daten über sich Preis. Dadurch ist es für niemanden ersichtlich, wer kommuniziert oder wo man gewesen ist. Dies kann wünschenswert sein, wenn man sehr viel Wert auf seine Privatsphäre legt und keinerlei Personalisierung und keine bestimmte Rolle benötigt, auch wenn hier eigentlich nicht von „Identitätsmanagement“ gesprochen werden kann.

Man kann Anonymität auf mehreren Ebenen definieren. Auf Anwendungsebene bedeutet Anonymität, dass die Anwendungen keine Identitätsdaten austauschen, was beispielsweise durch Login auf einer Web-Seite geschehen könnte. Auf Transportebene bedeutet Anonymität, dass die Kommunikationspartner die Zieladresse nicht kennen. Dies ist schwieriger zu erreichen, da das Problem des Routings besteht: wie sollen die Antworten auf eine Anfrage ihr Ziel finden? Ein Lösungsansatz dazu sind Mix-Netze (siehe Chaum 1981; Díaz und Serjantov 2003), die Datenpakete über mehrere so genannte Mixe zum Ziel leiten. Ein *Mix* ist dabei ein Netzwerkknoten, der Datenpakete von mehreren Kommunikationspartnern entgegennimmt, kurze Zeit puffert und erst dann in veränderter Reihenfolge weiterleitet. Dabei sollten mehrere Mixe hintereinandergeschaltet werden, wodurch die Pakete nicht rückverfolgbar sind. Durch das Puffern werden die Pakete allerdings verzögert, wodurch die Latenz steigt. Um dieses Problem zu beheben und dadurch die Nutzbarkeit für Anwender zu erhöhen werden verschiedene Modifikationen vorgeschlagen (zum Beispiel von Rennhard und Plattner 2003).

Sobald Personalisierung gebraucht wird, ist zumindest Pseudonymität erforderlich. *Pseudonymität* nennt man den Zustand, unter Benutzung eines Pseudonyms aufzutreten, also eines nicht auf eine reale Identität abbildbaren Identifikators.

David Chaum hat 1985 beschrieben, wie Pseudonyme benutzt werden können, um unerkannt sichere Transaktionen durchzuführen. Damit wurde damals der Grundstein des datenschützenden Identitätsmanagement gelegt; alle folgenden Arbeiten bezogen sich auf diesen Artikel. Das darin vorgeschlagene System sieht vor, dass ein kleines, elektronisches Gerät (ähnlich einem heutigen PDA) die Verwaltung von Pseudonymen für einen Benutzer übernimmt und situationsabhängig ein passendes Pseudonym herausgibt. Dieser Vorgang der teilautomatischen Pseudonymität ist ein sehr effektiver Mechanismus, um die Privatsphäre bei Kommunikationsvorgängen im Internet zu schützen. Allerdings muss es sich dabei nicht um ein eigenständiges Gerät handeln, der Dienst der Pseudonymverwaltung kann auch auf dem Arbeitsrechner oder einem entfernten Server vorgenommen werden. Wichtig ist, dass abhängig vom Kontext und vom Wunsch nach Wiedererkennung (auch Pseudonyme, deren Inhaber nicht identifizierbar sind, können wiedererkannt werden und so einen Mehrwert bringen) ein Pseudonym entweder aus dem Speicher ausgewählt oder neu generiert wird. Benutzte Pseudonyme müssen gemerkt und vor Wiederverwendung geprüft werden. Während Chaum diesen

Mechanismus lediglich für Bezahlvorgänge und ähnliche Transaktionen vorgesehen hatte (zu der Zeit spielte persönliche Kommunikation über das Internet noch keine entscheidende Rolle), wird der Mechanismus heute dadurch erweitert, dass den Pseudonymen persönliche Daten anhaften können.

3.4.3. Aushandlung von Datenschutzvereinbarungen

Ein wichtiger Mechanismus zum Schutz vor unvorsichtiger Preisgabe privater Daten ist das Aushandeln von Datenschutzvereinbarungen, so genannten *privacy policies*. In diesen Vereinbarungen wird festgehalten, was der Empfänger mit persönlichen Daten, die während einer Sitzung übertragen worden sind, tun darf. Dazu gehört das persistente Speichern oder Übertragen zu dritten, aber auch der Zweck dieser Speicherung oder Übertragung. Beispielsweise könnte ein Anbieter einen Kunden nach seiner Adresse fragen. Diese Adresse könnte er speichern wollen, um später selbst noch einmal darauf zugreifen, dem Kunden Waren schicken oder sie an Werbetreibende verkaufen zu können.

Zwar geben immer mehr Web-Sites eine privacy policy an (einer Studie von 2002 zufolge haben 77% aller Web-Sites eine privacy policy, vgl. Adkinson u. a. 2002), jedoch ist die Struktur und der Inhalt dieser Dokumente sehr unterschiedlich und für den Anwender keine große Hilfe (Jensen und Potts 2004). Technische Mechanismen, die sicherstellen, dass sich der Empfänger an die von ihm übertragene policy hält, gibt es derzeit nicht – hier ist im Zweifelsfall der Rechtsweg einzuschlagen.

Im Internet wird diese Datenschutzvereinbarung bisher nicht verhandelt, sondern lediglich vom Dienstanbieter bekanntgegeben. Der Dienstnutzer muss sie also akzeptieren, wenn er den Dienst nutzen möchte. Darüber hinaus ist die gesamte Datenschutzvereinbarung meistens sowohl sehr lang als auch in schwieriger, juristischer Sprache verfasst, wodurch es den meisten Dienstnutzern erschwert wird, diese Vereinbarung zu lesen und zu verstehen. So wird von vielen Dienstnutzern aus Bequemlichkeit oder Unverständnis eine Datenschutzvereinbarung akzeptiert, die sie gar nicht kennen – dies birgt eine große Gefahr für die Privatsphäre.

Sinnvoller für den Anwender und den Datenschutz wäre eine Datenschutzvereinbarung, die aufgrund von Regeln, die der Anwender für seine Daten angegeben hat, ausgehandelt wird. Zunächst könnte die vom Dienstanbieter angebotene Datenschutzrichtlinie anhand der Regeln ausgewertet werden. Werden die Anforderungen erfüllt, so wird die Richtlinie akzeptiert. Werden sie jedoch nicht erfüllt, so kann eine Verhandlung folgen, bei welcher der Dienstnutzer selbst eine Richtlinie vorschlägt, die wiederum vom Dienstanbieter auf seine Anforderungen untersucht werden müsste.

Einen solchen Ansatz bietet das Projekt „Platform for Privacy Preferences“[10], kurz P3P (vgl. Cranor u. a. 2002; Berthold und Köhntopp 2001). In diesem Projekt wurde eine XML Sprache (die ebenfalls P3P heißt) entwickelt, mit der standardisierte Datenschutzrichtlinien formuliert werden können. Außerdem wurde eine weitere Sprache namens APPEL („A Privacy Policy Evaluation Language“) entwickelt, in der Regeln für die persönlichen Daten verfasst werden können. Ein Appel Evaluator kann anhand einer P3P policy, eines APPEL Ausdrucks und einer Datenanfrage feststellen, ob die angegebene *policy* den Anforderungen des APPEL Ausdrucks gerecht wird. Das Ergebnis beinhaltet sowohl ein Verhalten, welches bei der Antwort eingenommen werden soll (z.B. Antworten oder Blockieren) als auch eine finale Policy.

3.4.4. Multilaterale Sicherheit

In vielen Situationen im Internet sind mehr als zwei Kommunikationspartner an einer Sitzung beteiligt. In diesen *multilateralen* Beziehungen muss auf besondere Weise für Sicherheit gesorgt werden: *Multilaterale Sicherheit* bedeutet, dass für jeden Kommunikationspartner Sicherheit gewährt wird (nach Clauß und Köhntopp 2001). Von jedem Kommunikationspartner darf nur ein Minimum an Vertrauen verlangt werden (siehe Pfitzmann 2001). Dabei ist zu berücksichtigen, dass jeder Kommunikationspartner eigene Sicherheitsziele verfolgt, über die sich verständigt werden muss – am Ende dieser Verhandlung sollte es zu einem Kompromiss kommen. Über ein Identitätsmanagement-System können sowohl die eigenen Sicherheitsbedürfnisse als auch die Bedingungen für eine solche Verhandlung verwaltet werden. Für jede Teil-Identität oder sogar für jedes persönliche Attribut können eigene Sicherheitsanforderungen formuliert und gespeichert werden. Darüber hinaus kann angegeben werden, unter welchen Bedingungen die Daten herausgegeben werden, um einen größeren Schutz für die Privatsphäre zu bieten.

3.4.5. Kryptographische Methoden: Verschlüsselung und digitale Signaturen

Kryptographische Methoden, die zur Verschlüsselung und zum Signieren von digitalen Daten eingesetzt werden, sind ein großes und komplexes Forschungsgebiet. Es werden sowohl Algorithmen zum Verschlüsseln von Daten betrachtet als auch die notwendigen Protokolle und andere Mechanismen, um die Schlüssel auszutauschen. Einen guten Überblick über moderne Kryptographie bietet Beutelspacher u. a. (2004). In dieser Arbeit kann nur ein kleiner Überblick gegeben werden, der helfen soll, die verschiedenen Mechanismen im Zusammenhang mit Identitätsdatenkommunikation zu verstehen.

[10] siehe `http://www.w3.org/P3P`

Während mit den zuvor genannten Mechanismen zum Schützen von Identitäten im Internet meist die Identitäten direkt geschützt werden, ist im Falle der Kryptographie eher die Kommunikation *zwischen* den Identitäten geschützt (vgl. Tavani und Moor 2001). Sie betrifft also den dynamischen, aktiven Teil des Identitätsmanagements und bildet dort die Grundlage der *Privacy Enhancing Technologies* (PET).

Grundsätzlich gibt es zwei verschiedene Arten der Verschlüsselung, die symmetrische und die asymmetrische Verschlüsselung. Bei der symmetrischen Verschlüsselung gibt es nur einen Schlüssel, der auf beiden Seiten der Kommunikation vorhanden sein muss, und der sowohl für die Verschlüsselung als auch für die Entschlüsselung verwendet wird. Beide Vorgänge können sehr effizient und sogar auf Hardware-Ebene implementiert werden, was für kommunikationsintensive Situationen (zum Beispiel längere Sitzungen) wertvoll ist. Schwieriger ist dagegen der Schlüsselaustausch. Hier werden für spontane Kommunikation zwischen zuvor unbekannten Partnern häufig dynamisch Schlüssel erstellt, die nur für eine Kommunikationssitzung gültig sind. Diese Schlüssel müssen auf sicherem Wege zwischen den Partnern ausgetauscht werden, damit niemand anders Zugriff darauf und damit auf die Möglichkeit zur Entschlüsselung bekommt. Es gibt Protokolle (Diffie und Hellman 1976), die einen solchen Austausch ermöglichen.

Bei der asymmetrischen Verschlüsselung hat jeder Kommunikationspartner ein Schlüsselpaar: einen öffentlichen Schlüssel (*public key*), der frei verfügbar sein kann, und einen privaten Schlüssel (meistens *secret key*, seltener *private key*), auf den nur der Besitzer Zugriff haben darf. Daten, die mit dem einen verschlüsselt sind, lassen sich nur mit dem jeweils anderen entschlüsseln. Das ermöglicht zwei interessante Möglichkeiten: zum einen kann natürlich nur der Besitzer des Schlüsselpaars Daten entschlüsseln, die jemand anders mit seinem öffentlichen Schlüssel verschlüsselt hat, zum anderen aber können alle anderen Daten entschlüsseln, die der Besitzer mit seinem geheimen Schlüssel *ver*schlüsselt hat. Dadurch lässt sich der Sender einer Nachricht verifizieren. In der Praxis wird dies in Verbindung mit Hash-Werten verwendet: Der Sender einer Nachricht erstellt von dieser mittels eines beiden Seiten bekannten Algorithmus einen Hash-Wert und fügt diesen, verschlüsselt mit seinem geheimen Schlüssel, zur Nachricht hinzu. Der Empfänger kann diesen Anhang entschlüsseln und seinerseits den Hash-Wert der Nachricht erstellen. Sind diese beiden Werte gleich, so stammt die Nachricht offenbar vom Besitzer des geheimen Schlüssels. Zusätzlich kann sicher ausgesagt werden, dass niemand die Nachricht geändert hat, während sie übertragen worden ist – die Integrität der Daten ist gewährleistet. Der verschlüsselte Hash-Wert einer Nachricht ist eine *digitale Signatur*, mit der durch Entschlüsselung mit dem öffentlichen Schlüssel des Senders sichergestellt werden kann, dass die Nachricht von ihm stammt und nicht verändert worden ist. Die digitale Identität des Senders ist damit belegbar.

Durch Verschlüsselung von Daten kann sichergestellt werden, dass nur befugte Personen, die den zum Entschlüsseln notwendigen Schlüssel besitzen, diese Daten lesen können. Dieser Mechanismus schützt die Privatsphäre bei Kommunikation über das Internet, weil diese naturgemäß über verschiedene Knoten des Internets zum Ziel geleitet werden. Würden die Daten nicht verschlüsselt, könnte jeder Zwischenknoten die Daten mitlesen.

3.5. Benutzbarkeit

Durch falschen Einsatz von Identitätsmanagement entstehen Sicherheitsrisiken und Gefährdungen der Privatsphäre. Aus diesem Grund muss die Benutzbarkeit der Systeme und Applikationen sehr hoch sein, damit der Benutzer sich nicht durch mangelhafte Bedienung des Systems gefährdet. Dieser Aspekt wurde ausführlich von Jendricke (2002) behandelt. Durch die Integration von Identitätsmanagement zur Selbstdarstellung und Datenschutz-Mechanismen wird zwar der Gebrauch letzterer vereinfacht beziehungsweise attraktiver, jedoch muss die Bedienschnittstelle des Systems auch von Laien korrekt einsetzbar sein, damit die gewonnene Sicherheit nicht wieder verloren geht.

Die Integration von Sicherheitsmechanismen wie Verschlüsselung und Signatur von persönlichen Daten trägt zur Benutzbarkeit insofern bei, dass der Anwender keine zusätzlichen Programme installieren, einrichten und pflegen muss, um diese Mechanismen nutzen zu können. Derzeit sind einzelne Sicherheitsmechanismen in Kommunikationswerkzeuge integriert, beispielsweise Verschlüsselungsoptionen in E-Mail-Programmen[11] oder Instant Messengern[12]. Jedoch ist die Handhabung dieser häufig als Plug-In realisierten Mechanismen meist derart kompliziert, dass unerfahrene Anwender diese Möglichkeiten gar nicht ausnutzen, oder schlimmer noch: sie falsch benutzen und damit ihre Sicherheit gefährden. Wären diese Mechanismen mit sinnvollen Standardeinstellungen in eine Identitätsmanagement-Infrastruktur integriert, ließe sich die Benutzbarkeitshürde deutlich senken. Die Verwendung der Infrastruktur wäre vielleicht aus anderen Gründen motiviert, brächte jedoch trotzdem die Vorteile der Sicherheitsmechanismen.

Desweiteren liegt die Integration einer Sitzungs-Infrastruktur nahe (wie in Abschnitt 2.4 beschrieben), denn dadurch ließe sich die Benutzbarkeit weiter steigern. Da es für viele verschiedene Kommunikationsplattformen ein Sitzungskonzept gibt, lohnt es sich, eine allgemeine Infrastruktur dafür zu verwenden, deren Bedienung nur einmal erlernt werden muss und deren Einstellungen auf einer konsistenten Bedienoberfläche vorgenommen werden können.

[11] Für das E-Mail-Programm „Thunderbird“ (`http://www.mozilla.org/products/thunderbird`) gibt es ein Plug-In (Enigmail, `http://enigmail.mozdev.org`), welches asymmetrische Verschlüsselung mit GnuPG (`www.gnupg.org`) ermöglicht.

[12] Für einige Instant Messenger, beispielsweise „Fire“ (`http://fire.sourceforge.net`) oder „Gaim“ (`http://gaim.sourceforge.net`) gibt es ebenfalls GnuPG-Plugins.

4. Basis-Elemente eines generellen Identitätsmanagement-Systems

Im vorangegangen Kapitel wurden die wichtigsten Mechanismen des Identitätsmanagement genannt. Diese Mechanismen können im digitalen Kontext durch technische Systeme unterstützt werden. Diese Systeme nennt man Identitätsmanagement-*Systeme*.

Definition 4.1. Ein *Identitätsmanagement-System* ist ein technisches System zur Unterstützung von Benutzern und Diensten beim digitalen Identitätsmanagement (siehe Definition 3.2).

Daraus ergibt sich, dass sich Identitätsmanagement-Systeme auf einer tieferen Ebene der Systemarchitektur befinden: Sie unterstützen Benutzer und Dienste bei der Verwaltung ihrer Persönlichkeitsbilder und bei der Verwendung dieser in Kommunikationssituationen untereinander. Auf Anwendungsebene findet sich lediglich die Bedienschnittstelle, die auch in die eigentlichen Kommunikationswerkzeuge – beispielsweise einen Web-Browser – integriert sein kann.

Aus der Beschreibung der Mechanismen des Identitätsmanagements ergeben sich prinzipielle Elemente (oder *Komponenten*) von Identitätsmanagement-Systemen. Sie kapseln die unterschiedlichen Aufgabengebiete und verbergen so die Komplexität vor Entwicklern und Anwendern; außerdem erleichtert das Konzept der Komponenten die Austauschbarkeit der einzelnen Mechanismen (siehe Griffel 1998). In diesem Kapitel sollen die wesentlichen Komponenten eines Identitätsmanagement-Systems abstrakt dargestellt werden. Dabei werden sowohl Elemente für organisatorisches oder föderiertes als auch für persönliches Identitätsmanagement berücksichtigt. Sie lassen sich grob in die Bereiche der *statischen Verwaltung* und des *dynamischen Einsatzes* der Identitätsdaten in Kommunikationssystemen gliedern, in die auch schon die Mechanismen aufgeteilt worden sind.

4.1. Elemente zur Verwaltung von Identitätsdaten

Die zentrale Komponente eines Identitätsmanagement-Systems hilft bei der Verwaltung der eigenen und fremden Identitätsdaten, die in Teil-Identitäten organisiert sind. Zumeist wird sie als „Identitätsmanager“ bezeichnet, obwohl zu einem

Identitätsmanagement-System noch weitere Elemente oder Komponenten gehören. An dieser zentralen Stelle sollte die Einbindung der anderen Komponenten geschehen: Sowohl die Speicherung als auch die kontrollierte Übertragung der Daten ist von der Verwaltung abhängig. Über eine Bedienschnittstelle kann ein Anwender Einsicht in eigene und fremde Identitätsdaten erhalten (soweit sie ihm bekannt sind), die eigenen pflegen (anlegen, modifizieren, löschen) und weitere Aussagen über fremde treffen. Hier wird also Kontrolle über die Identitätsdaten ausgeübt. Die Entscheidung, wer diese Kontrolle ausüben kann, bestimmt auch die Ausrichtung des Systems: Hat der Benutzer selbst die Kontrolle, so handelt es sich wahrscheinlich um persönliches Identitätsmanagement; liegt die Kontrolle bei einer zentralen Instanz, so handelt es sich um Identitätsmanagement in Organisationen (die Unterscheidung in diese groben Kategorien wurde in Abschnitt 1.2 vorgenommen).

4.1.1. Datenpflege

Identitätsdaten können statisch sein wie zum Beispiel das Geburtsdatum, oder sehr dynamisch wie ein momentanes Navigationsziel. Statische Daten können zu Anfang oder bei der ersten Verwendung des Attributs eingegeben werden. Um die dynamischen Attribute in einem Identitätsmanagement-System gut pflegen zu können, muss eine Komponente angeboten werden, die den Benutzer oder Dienst dabei unterstützt. Es muss Bedienschnittstellen mit ergonomischen Eingabemasken oder anderen Möglichkeiten geben, Daten direkt einzugeben, aber auch die automatische Übernahme von Daten aus anderen Systemen sollte unterstützt werden (vgl. Unterabschnitt 3.3.3). Transiente Daten wie oben angeführte Navigationsziele sollten gegebenenfalls (beispielsweise beim Erreichen des Ziels) automatisch als ungültig markiert oder sogar gelöscht werden.

4.1.2. Persistente Datenhaltung

Damit die eigenen und auch fremden Identitätsdaten langfristig verfügbar sind, müssen sie persistent gespeichert werden. Diesen statischen Aspekt übernimmt häufig auch die zentrale Identitätsmanager-Komponente, sie kann jedoch auch ausgelagert werden. Die Datenhaltung für persönliches Identitätsmanagement kann lokal im Anwendersystem oder entfernt bei einer *Trusted Third Party* geschehen. In beiden Fällen sollte sie den Sicherheitsanforderungen im Bezug auf Verschlüsselung und Sicherung des Zugriffs entsprechen. Das Speichermedium kann auch in der lokalen Lösung getrennt vom eigentlichen Kommunikationssystem liegen, beispielsweise auf einer Diskette, einer Speicherkarte oder einem angebundenen PDA. Dadurch werden die eigentlichen Daten mobil und können sowohl vom Benutzer mitgeführt als auch an verschiedenen Orten eingesetzt werden. Beim Identitätsmanagement in Organisationen erfolgt die Datenhaltung zen-

tral auf einem Server – hier spielt die Kontrolle des physischen Zugriffs auf die Speichermedien eine Rolle, sowie die Datensicherung für Notfälle. Bei vielen parallelen Zugriffen kann ein Clustering-System, wie es auch bei großen Web-Servern bekannt ist, schnellere Antwortzeiten auf die Datenanfragen versprechen.

4.1.3. Verwaltung von Ontologien

Zur Sicherstellung der Bedeutung von Identitätsattributen ist es notwendig, dass die attributsdefinierenden Ontologien – in welcher Form sie auch immer vorliegen – dem System langfristig zur Verfügung stehen. Ebenso kann der Benutzer zur Erweiterung seiner Teil-Identitäten oder zur Beantwortung einer Identitätsdaten-Anfrage Informationen über eine weitere Teil-Ontologie benötigen (siehe Unterabschnitt 3.2.2). Hierfür ist eine Komponente zur Verwaltung der einzelnen Teil-Ontologien vorzusehen. Teil-Ontologien sollten nicht nur über ihren Namen referenzierbar, sondern auch über Suche nach Eigenschaften auffindbar sein. Zu diesen Eigenschaften gehören in erster Linie die durch die Ontologie definierten Attributstypen, aber auch die Herkunft oder das Erstellungsdatum können relevant sein. Ebenso kann die Information wichtig sein, zu welchen Attributstypen in welchen Teil-Identitäten Daten vorliegen. Diese Information kann zwar dynamisch ermittelt werden – dies ist jedoch bei großen Datenbeständen recht zeitaufwändig. Im persönlichen Identitätsmanagement kann mit dieser Information zum Beispiel eine passende Transformationsregel für eine Anfrage nach einem unbekannten Identitätsattribut gefunden werden, in Identitätsmanagement-Systemen in Organisationen lässt sie sich für verschiedene *Data Mining* Aktivitäten nutzen, bei denen Zusammenhänge von Datensätzen gesucht werden, die nicht explizit vorhanden sind.

4.2. Elemente für die dynamische Nutzung von Identitätsdaten

Für den dynamischen Einsatz von Identitätsdaten in Kommunikationssituationen sind weitere Elemente notwendig, die Benutzer und Systeme bei dieser anspruchsvollen Aufgabe unterstützen. Dabei sind die konträren Ziele der Selbstdarstellung und des Schutzes der Privatsphäre gleichermaßen zu beachten. Primäre Voraussetzung für Selbstdarstellung ist, die einzelnen Kommunikationsakte an bestimmte Teil-Identitäten zu knüpfen. Zum Schutz der Privatsphäre darf jedoch erstens keine weiterführende Information über andere Teil-Identitäten an den Kommunikationspartner sowie gar keine Information an Dritte gelangen.

Ein wesentliches Element der Identitätsdatenkommunikation sind die *Protokolle*, die zum Übertragen der Daten und zu organisatorischen Zwecken eingesetzt werden. Anhand der unterstützten Protokolle entscheidet sich grundlegend, ob

eine Identitätsdatenkommunikation mit der Gegenseite überhaupt oder vielleicht nur eingeschränkt möglich ist. Es sollten also möglichst universelle oder vielleicht auch mehrere Protokolle unterstützt werden, damit eine große Zahl an Kommunikationspartnern erreicht werden kann.

4.2.1. Aufbau und Kontrolle von Kommunikationssitzungen

Zur generellen Assoziation von Teil-Identitäten zu Kommunikationsakten ist eine Erweiterung der bisherigen Kommunikationsprotokolle notwendig. So könnte man jeder Nachricht ein identifizierendes Merkmal hinzufügen. Eleganter wäre jedoch der Weg, eine initiale oder sich während der Kommunikation ergebende Identifizierung zur gesamten Menge der zusammengehörigen Kommunikationsakte zu assoziieren. Durch diese Bündelung von Kommunikationsakten und den dazugehörigen Teil-Identitäten ergibt sich das Konzept der Kommunikationssitzung (siehe Abschnitt 2.4). Eine Komponente, die dynamische Aspekte des digitalen Identitätsmanagements unterstützt, sollte also den Aufbau und die Kontrolle von Sitzungen unterstützen. Dadurch wird Selbstdarstellung erleichtert, weil man durch inkrementelles Freilegen der eigenen Identitätsdaten und auch Annehmen fremder Attribute eine Identitätsbildung erreicht, wie sie in Unterabschnitt 3.3.3 gezeigt worden ist. Die Sitzungsverwaltung hat die Aufgabe, Sitzungen mit anderen Teilnehmern aufzubauen, Sitzungsanfragen zu bearbeiten (annehmen oder ablehnen), sowie Teil-Identitäten den Sitzungen zuzuordnen, um zusammengehörige Kommunikationsakte und die dazugehörigen digitalen Identitäten zu gruppieren und mit Nachrichten der Anwendungsprotokolle zu assoziieren.

Auch für diese Aufgaben ist eine Bedienschnittstelle erforderlich. Diese sollte mit der Bedienschnittstelle des Identitätsmanagers kombiniert sein, weil vor allem innerhalb von Sitzungen Identitätsdaten ausgetauscht werden, und Sitzungen ohne Selbstdarstellung auf anderer Ebene gelöst werden. Durch die Kombination der Bedienschnittstellen ist eine einheitliche Verwendung der Teil-Identitäten innerhalb von Sitzungen unterschiedlicher Applikationen möglich.

4.2.2. Pseudonymisierung

Ein wichtiger Aspekt der Identitätsdatenkommunikation ist das Verwenden von Pseudonymen: Nicht in jeder Situation ist eine sofortige Preisgabe einer wiedererkennnbaren Teil-Identität erforderlich oder wünschenswert. Um dennoch Identitätsdaten austauschen zu können, hat sich das Prinzip der Pseudonymität bewährt. Ein Pseudonym ist in diesem Fall eine transiente Teil-Identität. Es kann Attribute tragen wie eine normale digitale Identität, ist jedoch nur für einen vom Anwender bestimmbaren Zeitraum gültig.

Im Falle eines Systems, das auf Sitzungen aufbaut, kann der Sitzungsmanager die Pseudonymisierung übernehmen: Bei Aufbau einer Sitzung wird dann auto-

matisch ein neues Pseudonym erzeugt, welches die Person in der Sitzung repräsentiert. Der Anwender kann während der Sitzung entscheiden, ob er das Pseudonym an eine schon bestehende Teil-Identität bindet oder eine neue Teil-Identität erstellt. Die Preisgabe dieser gewählten Teil-Identität ist wiederum Aufgabe des Identitätsmanagers. Sie führt zum Verlust der Pseudonymität, bietet aber das Potential der Wiedererkennung. Attribute, die vom Anwender oder dem Kommunikationspartner einem Pseudonym zugesprochen worden sind, können nach der Bindung in eine Teil-Identität übernommen werden (wiederum Identitätsbildung, siehe Unterabschnitt 3.3.3).

4.2.3. Auswahl von Teil-Identitäten

Bei der Identifizierung gegenüber Kommunikationspartnern ist eine wichtige Frage, welche Teil-Identität man zeigt. Diese Entscheidung wird von diversen Faktoren beeinflusst. Zum Beispiel spielt das gewünschte erzielte Persönlichkeitsbild eine große Rolle – in organisatorischen Identitätsmanagement-Systemen hängt oft der Zugriff auf bestimmte Ressourcen von der gewählten Identität ab. Ein anderer wichtiger Aspekt ist der Umfang der Daten, den man dem Kommunikationspartner offenbaren möchte oder muss. Die Identifizierung mit einer Teil-Identität kann die Wiedererkennbarkeit und damit Verknüpfbarkeit von vergangenen Äußerungen mit dieser Identität ermöglichen, daher muss sorgfältig abgewogen werden, ob durch eine Identifizierung nicht die Privatsphäre in einem diese Sitzung übersteigenden Rahmen preisgegeben wird.

Das Identitätsmanagement-System sollte den Anwender bei der Auswahl einer günstigen Teil-Identität unterstützen. Dies kann durch Anzeige der notwendigen und optionalen Identitätseigenschaften, die von der Gegenseite erwartet werden, und den dazu passenden Teil-Identitäten geschehen. Ebenso kann aufgezeigt werden, welche Informationen implizit durch die Wahl einer bestimmten Identität offenbart werden könnten. Es handelt sich also um einen Aspekt der Erleichterung der Selbstdarstellung und Benutzbarkeit, der jedoch große Auswirkung auf Datenschutz und Privatsphäre hat.

4.2.4. Privatsphäre bei Identitätsdatenkommunikation

Werden Identitätsdaten über digitale Netze übertragen, so stellt sich immer die Frage nach dem Schutz der Privatsphäre. Das bedeutet, dass ein Anwender nach dem Recht auf informationelle Selbstbestimmung selbst unter Kontrolle haben sollte, wem er private Daten mitteilt (vgl. Abschnitt 2.3). Dabei gibt es zwei Aspekte der Privatheit: Erstens sollte dem Kommunikationspartner nicht mehr mitgeteilt werden, als beabsichtigt ist und zweitens sollte Dritten kein unbefugter Zugriff auf die übermittelten Daten möglich sein. Im Idealfall ist hier die Unbe-

obachtbarkeit (engl: *unobservability*) zu erreichen, bei der kein Dritter feststellen kann, dass überhaupt kommuniziert wird.

Schutz vor unvorsichtiger Preisgabe persönlicher Informationen an Kommunikationspartner, deren Datenschutzrichtlinie nicht den eigenen Privatsphärenansprüchen genügt, kann durch eine automatische Überprüfung dieser Eigenschaften erreicht werden. Eine Datenschutzkomponente hat die Aufgabe zu überprüfen, ob die einer Sitzung oder einem Kommunikationspartner anhaftende Datenschutzrichtlinie den eigenen Anforderungen entspricht. Dazu sollte jedem eigenen Identitätsattribut eine Beschreibung anhängen, unter welchen Bedingungen dieses Attribut übertragen werden darf. Der Datenschutzkomponente werden dann sowohl die Datenschutzrichtlinie des Fragenden als auch die eigene Bedingung, woraufhin die Konformität überprüft und letztendlich eine Verhaltensweise vorgeschlagen wird (siehe Unterabschnitt 3.4.3). Der Datenschutzkomponente werden nicht die Identitätsdaten selbst übergeben. Dadurch ist hier keine Verletzung der Privatsphäre zu befürchten, sollte die Datenschutzkomponente auf einem entfernten Rechner genutzt werden, der nicht unter der eigenen Kontrolle steht. Die Kommunikation mit der Datenschutzkomponente kann weiterhin verschlüsselt werden, damit Dritte nicht herausfinden können, welche Datenschutzrichtlinie auf welche Bedingungen geprüft wird.

4.2.5. Authentifizierung gegenüber Kommunikationspartnern und Authentizität von Identitätsdatenaussagen

Ein wichtiger Aspekt der Identifizierung gegenüber anderen ist die Belegung der vorgegebenen Identität. Dieser Vorgang wird *Authentifizierung* genannt: Es wird bewiesen, dass man tatsächlich derjenige ist, der man vorgibt zu sein. Darüber hinaus können nicht nur Identifizierungsaussagen belegt werden, sondern beliebige Aussagen über Identitätseigenschaften. Damit wird die *Authentizität* der Aussage belegt. Die Signierung und Verifizierung von Aussagen kann im zentralen Verwaltungselement vorgenommen werden. Eine Aussage wird dabei beispielsweise mit dem privaten Schlüssel einer digitalen Identität signiert, woraufhin der Empfänger diese Aussage verifizieren kann. Man beachte, dass damit im Falle von Authentifizierungen lediglich die *digitale* Identität des Senders verifiziert werden kann – um die tatsächliche, reale Identität des Senders zu verifizieren, müssten weitere, von vertrauenswürdigen Dritten (beispielsweise eines unabhängigen Instituts oder einer Behörde) signierte, identifizierende Daten übermittelt werden, mit denen die digitale Identität einer realen Identität zugeordnet werden kann. Damit sind dann alle von dieser digitalen Identität signierten Aussagen dieser realen Identität zusprechbar.

4.2.6. Datentransformation

In offenen Identitätsmanagement-Systemen ist eine dynamische Weiterentwicklung der attributsdefinierenden Ontologien möglich. Wird in einer Kommunikationssitzung ein Attribut angefragt, das in der eigenen Teil-Identität in einer anderen Teil-Ontologie vorliegt als der angefragten (jede Anfrage beinhaltet einen Verweis auf die Teil-Ontologie, der die Antwort entsprechen soll), so muss sowohl die Anfrage übersetzt werden, damit das Identitätsmanagement-System sie interpretieren kann, als auch die Antwort, damit sie der angefragten Teil-Ontologie entspricht. Dazu wird zunächst herausgefunden, nach welcher Ontologie entsprechende Quelldaten vorliegen. Anhand einer Liste der möglichen Übersetzungen muss entschieden werden, welche durchgeführt werden soll. Die Durchführung übernimmt wiederum der Transformationskomponente. An dieser Komponente wird deutlich, dass viele Funktionen in die Identitätsverwaltungskomponente integriert sein können, wenn diese leistungsfähig genug ist. Zum Beispiel kann das Durchführen der Transformation auch vom Identitätsmanager selbst ausgeführt werden, wenn eine Verteilung dieser Aufgabe aus Performanz-, Latenz- oder Datenschutzgründen nicht gewünscht ist.

Zu beachten ist, dass sowohl zum Auffinden neuer Ontologien in der Komponente zum Verwalten der Ontologien (Unterabschnitt 4.1.3) als auch zum Finden von Transformationsregeln ein Zugriff auf das Internet notwendig ist. Dieser sollte unabhängig vom Identitätsmanagement stattfinden, um Rückschlüsse auf die Identitätsdatenkommunikation zu vermeiden.

4.3. Zusammenfassung

Die in diesem Kapitel genannten Komponenten bilden den Grundstock eines generellen, umfassenden Identitätsmanagement-Systems. Wie in Kapitel 7 beschrieben ist, wird jedoch nicht in jedem aktuellen System jede Komponente eingesetzt, was zum Verlust oder zur Einschränkung der entsprechenden Funktionalität führt. Zusätzlich zu den genannten Komponenten können in Einzelfällen auch noch weitere hinzugefügt werden, die einem speziellen Anwendungszweck dienen oder erweiterte Möglichkeiten bieten. Mit den angeführten Komponenten allein wird jedoch schon ein umfassendes System beschrieben. Abbildung 4.1 zeigt eine Übersicht über die aufgeführten Komponenten und ihren Zusammenhang. Dabei ist die Unterteilung in Module ein Vorschlag, der die Umsetzung und Struktur eines konkreten Systems vereinfachen könnte, weil er die essentiellen Mechanismen des Identitätsmanagements bündelt. Auswahl und Ausprägung der einzelnen Komponenten sind entscheidend für Einsatzmöglichkeiten des jeweiligen Systems.

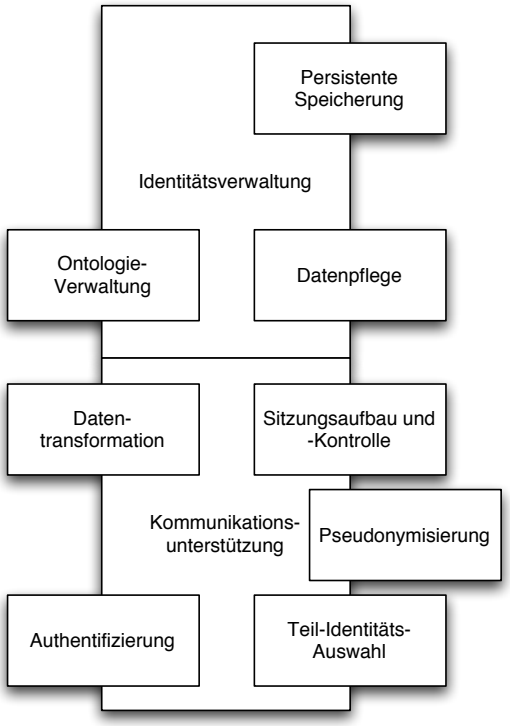

Abbildung 4.1.: Übersicht der generellen Elemente eines Identitätsmanagement-Systems

Teil II.

Identitätsmanagement-Systeme: Übersicht und Einordnung

Zusammenfassung

Dieser Teil der Arbeit beschäftigt sich mit in der Entwicklung befindlichen und bereits bestehenden Identitätsmanagement-Systemen. Zunächst wird ein Überblick über den Einfluss gegeben, den diese Systeme auf praktische Anwendungssysteme bereits haben und bekommen könnten. Mit diesem Blick in die Zukunft soll die Entwicklung der verschiedenen Identitätsmanagement-Systeme motiviert und weiterhin erklärt werden. In Kapitel 6 wird eine Klassifizierung entwickelt, nach der Identitätsmanagement-Systeme kategorisiert werden können. Die Klassifizierung berücksichtigt systemarchitektonische, funktionelle und intentionelle Aspekte. In Kapitel 7 werden dann Identitätsmanagement-Systeme sowohl aus der aktuellen Forschung als auch bereits in Betrieb genommene Systeme der Praxis beschrieben und in die zuvor entwickelte Klassifizierung eingeordnet.

5. Anwendungsszenarien von Identitätsmanagement-Systemen

Dieses Kapitel schlägt eine Brücke zurück zur Einleitung und Motivation, in der verschiedene Gründe zum Einsatz von digitalen Identitätsmanagement-Systemen genannt worden sind. Dort wurde bereits erwähnt, dass Identitätsmanagement-Infrastrukturen keinen Selbstzweck haben, sondern zur Anreicherung anderer Kommunikationsvorgänge dienen. Hier soll nun gezeigt werden, wie diese Einsätze tatsächlich aussehen. Dazu wird teilweise auf das Identitätsmanagement-System „onefC“ (*open network environment for Citizens*) vorgegriffen, welches im letzten Teil der Arbeit beschrieben wird. Allerdings sind die Beispiele auch auf andere Identitätsmanagement-Systeme übertragbar. Dabei wurden bewusst Beispiele gewählt, die vor allem durch organisatorisches oder föderiertes Identitätsmanagement denkbar sind (Abschnitt 5.1), aber auch Beispiele, die eher durch persönliches Identitätsmanagement erreichbar sind (Abschnitt 5.3 und Unterabschnitt 5.4.1). Die aufeinander aufbauenden Beispiele „Reputationssysteme“ (Abschnitt 5.2) und „Vertrauenssysteme“ (Unterabschnitt 5.2.1) sind gleichermaßen mit allen Identitätsmanagement-Arten denkbar.

5.1. Authentifizierungssysteme und „Single Sign-On“

Für viele digitale Dienste muss ein Anwender seine Identität belegen. Diesen Vorgang nennt man Authentifizierung (siehe dazu auch Abschnitt 3.2.3.1). Ein Authentifizierungssystem ist eine Systemkomponente, die in Anwendungen eingebunden wird, um Authentifizierung zu ermöglichen. Es erfordert vom Anwender eine Art „Beweis“ für seine Identität. Meistens ist dies ein Passwort, denkbar ist jedes Attribut, das nur diesem Anwender anhängen kann (Biometrie, signiertes Credential) oder von diesem geheim gehalten werden kann.

Ein allgemeines Identitätsmanagement-System kann persönliche Daten aller Art verwalten. Dazu gehören auch Identifikations- und Authentifizierungsdaten. Credentials, die in digitaler Form vorliegen, können als strukturierte Daten in Identitätsattributen gespeichert und abrufbar gemacht werden. Auch ein asymmetrisches Schlüsselpaar lässt sich in Identitätsattributen speichern, um damit die Signaturfunktion einer PKI zu benutzen. Die Identitätsmanagement-Infrastruktur onefC sieht in der Kern-Ontologie (siehe Listing A.1 im Anhang) ein

asymmetrisches Schlüsselpaar für jede digitale Identität vor, mit dem die Signierung von persönlichen Daten oder zusätzliche Verschlüsselung möglich ist[1].

Um diese identifizierenden Daten nutzen zu können, muss sich der Benutzer nur dem Identitätsmanagement-System gegenüber authentifizieren; alle darin gespeicherten Passwörter und Benutzernamen können dann automatisch benutzt werden. Die Funktionalität von Single Sign-On ist damit gegeben, wenn auch teilweise als indirekt genutzter Passwort-Speicher. Die Gegenseite der Kommunikation fragt dann das Identitätsmanagement-System des Anwenders nach den authentifizierenden Daten; von dort können die Anfragen direkt beantwortet werden, ohne dass die Daten erneut eingegeben werden müssen.

5.2. Reputationssysteme

Die Reputation oder der „gute Ruf“ einer Person sind häufig die Grundlage für andere, mit dieser Person Transaktionen (Informationsaustausch, Geschäfte oder weiteres) durchzuführen. Reputation ist immer an Personen (oder Institutionen) gebunden und ist somit sinnvoll in eine Identitätsinfrastruktur einzubinden. Die Daten, aus denen die Reputation eines Teilnehmers berechnet wird, können entweder signiert in der Identität des Reputationstragenden gespeichert oder vor der Transaktion von anderen Teilnehmern im Systemen zusammengetragen werden. Eine Herausforderung an Reputationssysteme ist immer die Integrität und Authentizität der Daten.

In einem Identitätsmanagement-System können sowohl eigene als auch fremde Identitätsdaten gespeichert werden. Eigene Identitätsdaten können vom Anwender selbst stammen oder von Fremden dieser Teil-Identität zugesprochen worden sein (vergleiche Unterabschnitt 3.3.3). Im letzteren Falle können die Daten vom Zusprechenden signiert werden, sollte ihre Integrität wichtig sein. So kann eine Bewertung eines Dritten, die signiert in einer Teil-Identität abgelegt wird, zur Berechnung der Reputation eines Nutzers abgefragt werden.

Beispiele für Reputationssysteme im Internet sind Beispielsweise das „Karma“ der Online-Community „Slashdot“[2] oder das kumulierende Bewertungssystem des Auktionshauses eBay[3]. Die Nachrichten, die auf der Slashdot-Seite erscheinen, werden von den Lesern der Seite selbst eingereicht. Karma erhalten jedoch nur die Leser, die zu diesen Nachrichten „wertvolle“ Kommentare schreiben. Ein Kommentar kann dabei als „wertvoll“ erachtet werden, wenn er informativ, erhellend oder besonders lustig ist. Abzüge beim Karma kann es für unnötige, störende oder

[1] Die Identitätsdaten, die über onefC ausgetauscht werden, sind durch die Verschlüsselung des Sitzungskanals in jedem Fall mit einem symmetrischen Schlüssel verschlüsselt.

[2] `http://slashdot.org`, mit dem Slogan „News for nerds, stuff that matters“ eine der wichtigsten Nachrichtenquellen für IT-Themen im englischsprachigen Internet geworden und in der Szene häufig als „/.“ abgekürzt

[3] `http://www.ebay.de`

vom Thema zu stark abweichende Beiträge geben. Die Bewertungen nehmen dabei die Mitglieder der Community selbst vor: wer ein gutes Karma hat, bekommt vom System gelegentlich Karma-Punkte zugeteilt, die nach eigenem Bemessen verteilt werden dürfen.

Bei eBay bekommen die Nutzer nach jeder Transaktion die Möglichkeit, ihren Handelspartner entweder positiv, neutral oder negativ zu bewerten. Möchte ein Nutzer etwas kaufen oder verkaufen, erscheint auf der Auktionsseite neben dem Benutzernamen ein farbiger Stern, der den Ruf des Benutzers symbolisiert, sowie die Anzahl der bisherigen Bewertungen; bei Verkäufern sieht man direkt auch den Anteil der positiven Bewertungen. Auf weiteren Seiten kann man sich die Bewertungen eines Händlers oder Käufers auch mit den Kommentaren der Bewertenden ansehen.

In beiden Fällen (Slashdot und eBay) kann man von digitalen Identitäten nach Definition 2.1 sprechen: der Benutzername fungiert als für das jeweilige System eindeutiger Bezeichner, und Karma sowie Bewertungen sind personifizierende Attribute. Für beide Systeme gilt jedoch, dass die Reputationsinformationen auf diese eine Community (auch die Gruppe der eBay-Nutzer kann man als Online-Community bezeichnen) beschränkt sind – außerhalb dieser Community sind sie schwerlich nutzbar. Um zum Beispiel seine eBay-Reputation auf einer anderen Auktions-Internetseite als eBay zu nutzen, könnte man seine dortige Profilseite verlinken; dies beweist jedoch nicht, dass das verlinkte Profil tatsächlich zu einem gehört. Betrüger könnten fremde Profile verlinken und dadurch eine Reputation vortäuschen. Wenn jedoch eBay die Reputationsinformation signiert und assoziiert zu einer Teil-Identität des Nutzers im Identitätsmanagement-System (sowohl dem eigenen als auch dem des Anwenders) speichern würde, dann könnte der Besitzer diese Teil-Identität auch in anderen Kontexten nutzen und dort seine eBay-Reputation vorzeigen. Da diese nun von eBay signiert ist, ist ein Betrug ausgeschlossen.

Dieser Vorgang mag für den Fall eBay unwahrscheinlich sein, da eBay gerade durch die Reputationsinformation seiner Benutzer einen Wettbewerbsvorteil gegenüber anderen Internet-Auktionsplattformen sieht. Im Falle Slashdot, einer Online-Community die erstens kostenlos und zweitens lediglich zum Informationstausch und informellen Gespräch gedacht ist, ist die Preisgabe der Reputationsinformation jedoch durchaus denkbar. Das Slashdot-Karma könnte auf anderen Community-Portalen mit ähnlichen Themen sichtbar gemacht werden, damit die Anwender sich untereinander darüber informieren können, wer welches Karma hat. Diese Reputationsinformation können die Anwender zum Beispiel dazu nutzen, Beiträge in Online-Foren danach zu sortieren oder gezielt nach Beiträgen von Benutzern mit hoher Reputation zu suchen.

Reputation ist zwar wandelbar, die Information darüber jedoch statisch an eine Identität gebunden. Es geht in diesem Zusammenhang also nicht um Kommunikationssituationen, sondern um die Gestaltung der Daten in den Informationssys-

temen. Das angesprochene Hauptproblem der Reputation digitaler Identitäten lässt sich auf ihre Trennung zurückführen, die schon in der Einleitung der Arbeit erwähnt worden ist. Durch die Zusammenführung der digitalen Identitäten lässt sich auch ein Zusammenhang der Reputation erkennbar machen. Die Reputation wird also genau wie die digitale Identität selbst vom Anwendungssystem gelöst und für den Anwender auch in anderen Kontexten einsetzbar – ebenso können nun mehrere Kontexte Einfluss auf die Reputation nehmen.

5.2.1. Vertrauenssysteme

„Vertrauen ist ein Werkzeug zur Reduktion von Komplexität“ ist eine Definition des Begriffs „Vertrauen“ von Luhmann (1989). Gemeint ist dabei die *soziale* Komplexität, die sich dabei ergibt, wenn Entscheidungen über Kooperationen und Interaktionen getroffen werden müssen. Vertrauen basiert dabei im wesentlichen auf Reputation (vgl. Dasgupta 1988; Fahrenholtz und Bartelt 2001). Da diese wiederum zwingend einen Reputationsträger, also eine Identität braucht, kann man Identitäten auch als notwendige Voraussetzung für Vertrauen bezeichnen. Vertrauen kann man in verschiedene Dinge haben (zum Beispiel Vertrauen in einen Zustand, eine Fähigkeit, eine Organisationsform). Am wichtigsten jedoch bei der Entscheidung, ob man mit einer Person interagieren möchte, ist das Vertrauen in diese Person. Im Internet müssen Personen digital abgebildet werden, damit man Vertrauen in sie haben kann. Daher ist eine Identitätsmanagement-Infrastruktur eine wesentliche Unterstützung für Vertrauensbildung im Internet.

Wenn Vertrauen als Werkzeug betrachtet wird, dann wird es offensichtlich für Aktionen benötigt. Hier wird also im Gegensatz zur Reputation der dynamische, aktive Teil des Identitätsmanagements in Anspruch genommen. Vertrauen spielt bei konkreten Entscheidungen eine Rolle und sollte situations- und kontextabhängig einsetzbar sein.

Vertrauen spielt im Internet an vielen Stellen eine Rolle. Beim zuvor genannten Beispiel der Auktionsplattform eBay muss man seinen Handelspartnern vertrauen. Wenn man ein Gebot für einen Artikel abgibt, kennt man zuvor meistens lediglich den Spitznamen des Anbieters auf der eBay Plattform sowie die Bewertungen, die andere über diesen abgegeben haben. Bei kommerziellen Anbietern kennt man häufig noch den Namen des Geschäfts sowie dessen WWW- und postalische Adresse – bei privaten Anbietern ist dies sehr häufig nicht der Fall. Anhand dieser Informationen muss man entscheiden, ob man ein Gebot für einen Artikel abgibt. Diese Gebote sind bindend. Nach Abschluss einer Auktion muss der Gewinner den ersteigerten Artikel erst bezahlen, bevor dieser versandt wird. Einige Anbieter lassen auch eine Transaktion über einen Treuhand-Dienst zu, der erst die Zahlung des Käufers entgegennimmt, diese Information dann dem Händler zukommen lässt, welcher daraufhin den Artikel ausliefert. Nach Empfangsbestätigung des Käufers reicht der Treuhänder das Geld weiter an den Ver-

käufer. Es wird deutlich, dass in jedem Fall Vertrauen gegenüber dem Anbieter oder dem Treuhänder notwendig ist. Dieses Vertrauen wird bei eBay im Wesentlichen durch die Bewertungen anderer Käufer erworben, bei kommerziellen Anbietern kommen weitere Faktoren (wie Bekanntheit der Marke oder Seriösität des Internet-Auftritts) hinzu. Vertrauen, Reputation und die damit behafteten Identitäten spielen also sehr dicht zusammen, was für eine Integration dieser Konzepte spricht.

Aber auch in nicht kommerziellen Situationen spielt Vertrauen im digitalen Netz eine große Rolle. Jedesmal, wenn man eine E-Mail empfängt und zum Lesen öffnet, trifft man heutzutage eine Vertrauensentscheidung: Dem Absender der E-Mail wird vertraut, dass er erstens seine angegebene Identität nicht vortäuscht, dass der Inhalt der E-Mail relevante Informationen enthält sowie dass durch Öffnen der E-Mail dem eigenen System kein Schaden zugefügt wird. Das Aufkommen unerwünschter Werbe-E-Mails nimmt immer noch stark zu und führt mittlerweile zu volkswirtschaftlichen Schäden durch Verschwendung wertvoller Arbeitszeit, die das Aussortieren der so genannten Spam-Mails kostet (siehe Cranor und LaMacchia 1998; Gburzynski und Maitan 2004). Auch ist E-Mail immer noch ein wichtiges Transportmittel für Viren und Würmer im Internet (vgl. Kienzle und Elder 2003). Vor dem Öffnen einer E-Mail sieht der Empfänger lediglich die Absenderadresse, die Betreffzeile und den Zeitpunkt des Versands. Aus diesen Informationen, die alle vom Absender stammen und somit gefälscht sein können, muss der Empfänger das Vertrauen ziehen. Davon gehört die Absenderadresse zu den persönlichen Daten des Senders und somit zu seiner digitalen Identität. Ein Beleg dieser digitalen Identität, wie es die in dieser Arbeit vorgeschlagene Identitätsmanagement-Infrastruktur ermöglichen würde, könnte das Vertrauen in E-Mail stärken und den Missbrauch dieses wichtigen Kommunikationsmediums erheblich einschränken.

Ein weiteres wichtiges Einsatzgebiet für Vertrauen im Internet ist die elektronische Verwaltung, das so genannte E-Government (siehe Riedl 2004). Hier müssen beide Seiten (öffentliche Verwaltung und Bürger) dem System und sich gegenseitig Vertrauen aufbringen, wenn sie über das digitale Netz Dienstleistungen anbieten beziehungsweise nutzen wollen. Denn im Gegensatz zu Slashdot geht es hierbei nicht um „reine“ digitale Identitäten, sondern um solche, bei denen der Bezug zur realen Identität des Benutzers eine große Rolle spielt. Zum Beispiel ist die Korrektheit der Identitätsinformationen beim Ummelden eines Wohnsitzes von großer Wichtigkeit. Demgegenüber stehen allerdings diverse Informationsdienste aus dem Bereich des E-Government, bei dem Pseudonymisierung und Zugriffskontrolle essentiell sind. In allen Bereichen kann mit einer Identitätsmanagement-Infrastruktur geholfen werden.

5.3. Online-Community-Support

Online-Communities bilden eine neue Gesellschaftsform, da sich ihre Mitglieder teilweise nicht persönlich kennen, sondern nur ihre digitalen Abbilder untereinander austauschen. Organisiert werden Online-Communities häufig über Online-Foren (Blackboard-Systeme) oder Portale, bei denen mehrere Informationsaustausch-Möglichkeiten bestehen (vgl. Unterabschnitt 1.3.4). Es gibt einige technische Systeme, die beim Aufbau einer Infrastruktur für eine Online-Community helfen. In diesem Abschnitt soll gezeigt werden, wie die Integration einer Identitätsmanagement-Infrastruktur in ein Community-Support-System die Interoperabilität und den Anwender-Austausch zwischen den Communities stützt und dadurch den Communities selbst hilft.

5.3.1. Aktuelle Community-Support-Systeme

Dazu werden zunächst drei exemplarische Community-Support-Systeme kurz vorgestellt und anschließend die technische Integration eines Identitätsmanagement-Systems besprochen. Alle drei Systeme haben eigene Mechanismen zur Abbildung digitaler Identitäten, die untereinander nicht kompatibel oder austauschbar sind, obwohl sie im Wesentlichen die gleichen Daten umfassen.

phpBB

Das Foren-System phpBB ist seit seiner Entstehung von einer reinen Blackboard-Lösung zu einem ausgereiften Portal weiterentwickelt worden, das den Community-Betreibern eine große Vielfalt an Möglichkeiten bietet, Zusatzfunktionen (wie zum Beispiel Bildergalerien, Up-/Download-Bereiche oder Kalender) in das entstandene Portal einzubinden[4]. Das System setzt auf die Programmiersprache PHP[5], mit der serverseitige Online Anwendungen entwickelt werden können. Der Web-Server interpretiert den PHP-Code und liefert dynamische, möglicherweise personalisierte Web-Seiten aus.

CommSy

Das Projekt CommSy[6] ist an der Universität Hamburg entstanden und bietet eine Plattform zur vernetzten Projektarbeit. Es können Projektgruppen eingerichtet werden, die untereinander Materialien (zum Beispiel PDF Dokumente) und andere Informationen austauschen. Das System ist auch im Quelltext frei verfügbar

[4] Das phpBB-System ist kostenfrei unter `http://www.phpbb.com` verfügbar. Hier sind auch etliche Modifikationen erhältlich, die das Forum um weitere Funktionalitäten erweitern.
[5] `http://www.php.net`
[6] `http://www.commsy.de`

(Lizenz: GPL) und wird vielfach eingesetzt. Es basiert ebenso wie phpBB auf der Programmiersprache PHP.

CoBricks

Das an der Technischen Universität München entstandene Projekt CoBricks[7] bietet allgemeine Unterstützung für Online-Communities. Es ist vollständig in Java entwickelt und stellt mit Hilfe der Servlet-Technologie eine Vielzahl an Funktionen für Communities bereit. Es ist öffentlich verfügbar und leicht erweiterbar. Im Mittelpunkt steht bei CoBricks der Aspekt des Portals sowie der Kollaboration über dieses Portal. Es existieren mehrere Beispielprojekte, die auf CoBricks aufsetzen – unter anderem ein Informationssystem für die TU München („Drehscheibe“, siehe Koch 2003) und ein Teilnehmer-Informationssystem für Veranstaltungen, bei denen die essentiellen Daten der anderen Teilnehmer eventuell nicht bekannt sind („Meeting Mirror“, vgl. Koch 2004). Dies trifft zum Beispiel auf Fachkonferenzen zu, auf denen das Bewusstsein (engl: *awareness*) über Interesse und Aktivitäten sowie auch derzeitiger Aufenthaltsort anderer Konferenzteilnehmer gesteigert werden soll.

5.3.2. Integration von Identitätsmanagement in Community-Support-Systeme

Allen Community-Support-Systemen ist gemein, dass sie Gruppen im Internet eine Plattform zum Informationsaustausch bieten. Dabei ist diese Information meistens persönlich: Man kann nachvollziehen, wer welche Information eingestellt hat (beispielsweise ist an jedem Beitrag im Forum der Autor notiert) und detaillierte Informationen über die Mitglieder der Community ist verfügbar. Dafür ist in den heutigen Systemen eine Abbildung der Benutzer auf der Seite des Servers vorhanden, die alle Informationen über ihn hält – eine digitale Identität, über die der Benutzer selbst nur indirekte Kontrolle hat. Der Einsatz einer allgemeinen Identitätsmanagement-Infrastruktur würde für diese Systeme bedeuten, dass neue Anwender ihre Daten aus anderen Systemen übernehmen und sogar die digitalen Identitäten in verschiedenen Communities verknüpfen könnten. Folgen die Community-Support-Systeme einem gemeinsamen Standard wie zum Beispiel den onefC-Protokollen, so ist eine einfach Interoperabilität möglich. Wichtig wäre dabei, dass das Support-System – und nicht das Community-System selbst – die Unterstützung für die Identitätsmanagement-Infrastruktur bietet, weil dann automatisch alle Communities, die auf das entsprechende Support-System aufbauen, die Vorteile der Identitätsmanagement-Infrastruktur nutzen könnten.

Dies würde auf technischer Seite bedeuten, dass die Anwender ihre digitalen Identitäten mit einem eigenen Identitätsmanagement-System verwalten müssen,

[7] `http://www.cobricks.de`

und die Web-Server um sitzungsbasierte Identitätsmanagement-Komponenten erweitert werden müssten. In Abschnitt 9.2 wird prototypisch gezeigt werden, wie dies für Java-basierte Systeme möglich ist, ohne die bestehenden Browser und Web-Server vollständig austauschen zu müssen. Für das Beispiel CoBricks bedeutet dies die geringsten Änderungen, da es wie der onefC-Prototyp auf Java basiert und die schon bestehenden Servlet-Erweiterungen nutzen kann. Für phpBB und CommSy wäre ein entsprechender Adapter für PHP von Nöten.

5.4. Navigation in digitalen Räumen mit sozialer Unterstützung

Unter dem Stichpunkt *Social Navigation* werden in der Literatur Navigationsstrategien und -verhaltensweisen bezeichnet, bei denen direkte und indirekte Hinweise anderer Personen die eigene Navigation beeinflussen (Dourish und Chalmers 1994). Da der Mensch ein soziales Wesen ist, lassen sich durch Rückschlüsse auf Hinweise anderer Vorteile im eigenen Verhalten erzielen. Beim Navigieren in Informationsräumen wie dem WWW treten sehr viele Probleme auf: die Informationsmenge übersteigt menschliches Fassungsvermögen (*information overload*, vgl. Gerhards und Mende 2003), die Orientierung in digitalen Welten ist aufgrund ihrer anderen Struktur schwieriger als in der realen Welt (vgl. Conklin 1987) und die Qualität der gefundenen oder angebotenen Navigationsziele lässt sich schwer überprüfen. Social Navigation kann dabei helfen, diese Schwierigkeiten abzuschwächen: Die Orientierung an Hilfestellungen anderer Personen erleichtern die Navigation. Dabei hat Social Navigation allerdings auch wieder Nachteile und Schwächen. Es stellt sich die Frage, welchen Navigationshinweisen man folgen soll, um die eigenen Ziele zu erreichen – in einem jungen System ist die Anzahl der Hinweise oder Hilfestellenden möglicherweise zu gering, in einem gewachsenen System eventuell schon zu groß, um geeignete Beiträge zu finden. Möglicherweise weiß man nichts über die Ziele der anderen, dann kann man die Relevanz der Hilfestellung nicht beurteilen, oder man kann die Qualität nicht beurteilen, weil die Reputation des Hilfestellenden nicht aussagekräftig ist. Bei diesen Nachteilen kann die Integration einer Identitätsmanagement-Infrastruktur zu einem gewissen Grade Abhilfe schaffen (Baier u. a. 2004). Je mehr über die Reputation und die Interessen der Hilfestellung Anbietenden in Erfahrung gebracht werden kann, desto besser lässt sich der Nutzen der Hilfestellung für die eigenen Ziele einschätzen. So kann man die Hilfestellungen selbst – und nicht nur die Gesamtheit der Möglichkeiten – zu eigenen Zwecken filtern.

Das Problem von Social Navigation Systemen der anfänglich geringe Nutzerzahl mit wenigen, anwendungsabhängigen Daten, und dadurch spärlich gesäten Informationsdichte, aus der Ähnlichkeiten gefunden werden müssen, ließe sich dadurch mindern, dass viele Social Navigation Systeme auf die gleiche Identitäts-

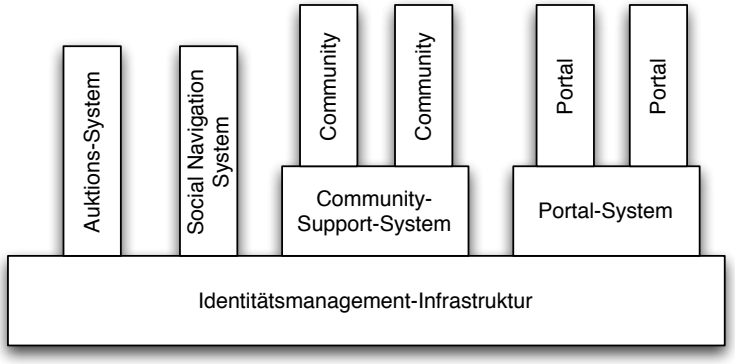

Abbildung 5.1.: Unterlegung mehrerer Anwendungen und Systeme mit einer Identitätsmanagement-Infrastruktur

management-Infrastruktur aufsetzt. Die Einstiegshürde für neue Nutzer würde sinken, neue Benutzer hätten bereits Daten aus anderen Anwendungskontexten, die für die neue Anwendung nutzbar wären und Vertrauensbeziehungen aus anderen Anwendungen wären wiederverwendbar. Dieser Vorteil potenziert sich bei der Integration der Identitätsmanagement-Infrastruktur in Community-Support-Systeme oder Portal-Systeme, denn dadurch stünde die Identitätsmanagement-Funktionalität allen darauf aufsetzenden Communities und Portalen zur Verfügung (siehe Abbildung 5.1).

5.4.1. Empfehlungssystem *CoInternet*

Eine Kombination eines Empfehlungssystems mit Identitäts-bewusstem Filtern ist prototypisch in dem System *CoInternet* entstanden (Wollenweber 2004). In diesem System können die Anwender Web-Seiten auf einer Skala von -2 (sehr schlecht) bis +2 (sehr gut) bewerten. Nun werden Links im WWW nach einem Mechanismus bewertet, der die eigene Historie der Bewertungen in Relation zu den Bewertungen der anderen Benutzer setzt. Haben andere Benutzer die Seite schon bewertet, auf die verlinkt wird, so kann anhand der Korrelation der schon getätigten Bewertungen errechnet werden, ob die Seite relevant ist. Im Gegensatz zu identitäts-unbewussten Empfehlungsystemen, bei denen alle fremden Bewertungen akkumuliert werden, wird bei diesem System die Ähnlichkeit der gesamten bekannten digitalen Identität der Bewertenden untereinander mit hineingezogen. Zusätzlich können anderen Benutzern direkt Vertrauensaussagen gemacht werden. Dies wird wiederum als persönliches Merkmal der digitalen Identität angehaftet. Bewertungen von Benutzern, denen man explizit vertraut, werden dann höher in die Voraussage eingerechnet. Insgesamt können über die Identitätsmanagement-Infrastruktur beliebige weitere persönliche Attribute in die Bewertung einbezo-

gen werden; beispielhaft wurde dies mit Bookmark-Sammlungen durchgeführt, die auf Ähnlichkeiten untersucht werden.

Ein sehr mächtiges Merkmal des CoInternet-Systems ist das Umsortieren von Suchergebnissen. Es wurde eine Schnittstelle zur Google-SOAP-API[8] entwickelt, über die ein Suchbegriff zur Suchmaschine geschickt wird. Die zurückkommenden Ergebnisse werden durch das Bewertungssystem geleitet, wodurch eine individuelle Umsortierung der Ergebnisse erreicht werden kann. Ein gutes Beispiel dazu ist die Suche nach dem Begriff „Jaguar“. Ohne individuelle Umsortierung erhält man eine gemischte Liste, die Links zu zoologischen Informationen, Automobilherstellern und Betriebssystem-Hilfestellungen enthält, denn „Jaguar“ ist ein Homonym, das sowohl für ein Tier, eine Automarke als auch eine Betriebssystemversion von Apple[9] steht. Aber auch innerhalb eines Themenbereiches gibt es Homonyme: In der Informationstechnologie steht zum Beispiel der Begriff „Robot“ sowohl für ein technisches System aus der Robotik, als auch für eine automatische Software im Internet (Web-Robot). Kann das System das eigentliche Interesse des Nutzers durch Analyse des persönlichen Profils einschätzen, so werden die Links, die von Interesse sind (beispielsweise `http://www.jaguar.de` für Auto-Fans) nach oben sortiert und mit einem grünen Symbol versehen, und eher als uninteressant eingestufe Links nach unten sortiert und mit einem roten Symbol versehen (vgl. Abbildung 5.2).

Insgesamt hat sich bei der Umsetzung von CoInternet gezeigt, dass die Integration eines Identitätsmanagement-Systems sowohl die Konstruktion erleichtert, da der gesamte Bereich des Benutzermanagements ausgegliedert ist, als auch weiterführende Möglichkeiten erlaubt, die im isolierten, von anderen Anwendungen getrennten Betrieb unmöglich sind. Die Einbindung von Anwendungsfremden Informationen in die Berechnung der Navigationsempfehlung bildet einen großen Vorteil gegenüber herkömmlichen Empfehlungssystemen.

5.5. Zusammenfassung

Anhand der in diesem Kapitel genannten Beispiele wird deutlich, dass es eine Vielzahl an Anwendungssystemen gibt, die von der Integration eines Identitätsmanagement-Systems profitieren können. Dabei ist heute noch unklar, welche weiteren Anwendungsfelder zu diesen Beispielen hinzukommen könnten. Durch die immer weitere Verbreitung digitaler Netze auch im privaten Bereich, der nicht von herkömmlichen PCs geprägt ist, könnte diese Entwicklung auch eher im Hintergrund geschehen. Die Entwicklung des Ubiquitous Computing – also der „allgegenwärtigen“ Rechnerunterstützung (vgl. Weiser 1993a, b) – und der fortschreitenden Digitalisierung der Unterhaltungselektronik durch zum Beispiel

[8] `http://www.google.com/apis`
[9] „Jaguar“ war der Codename für Mac OS X v. 10.2

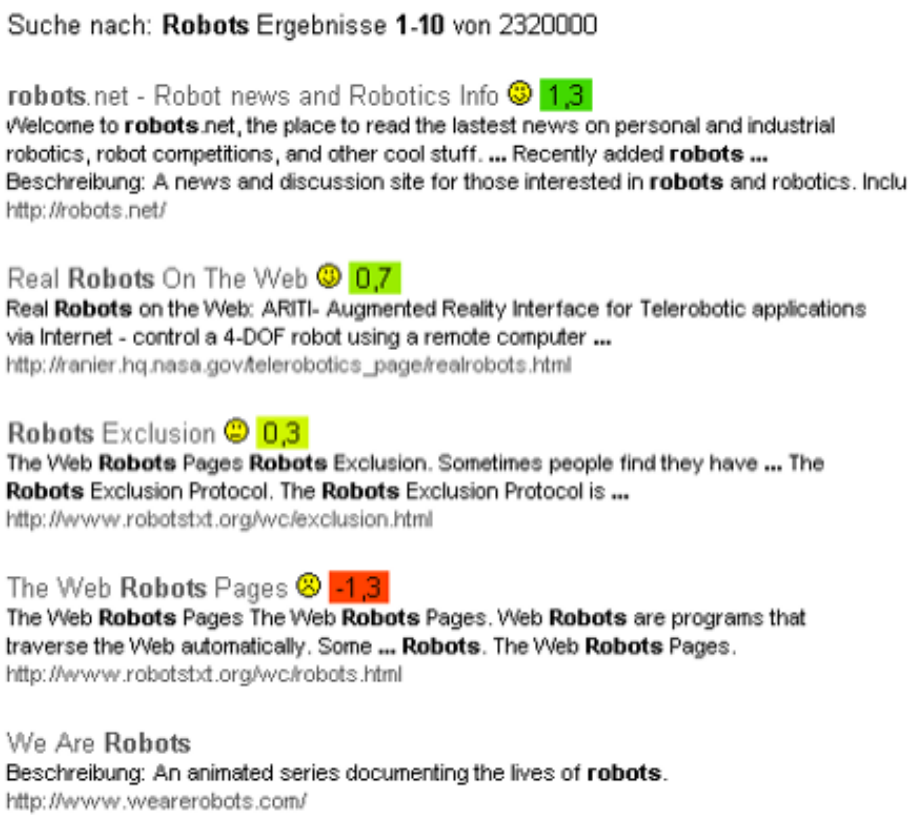

Abbildung 5.2.: Vom CoInternet-System für einen Robotik-Interessierten umsortierte Suchergebnisse nach dem Homonym „Robots“.

Set-Top-Boxen für den Fernseher eröffnen weitere große Bereiche, in denen digitale Identitäten eine wichtige Rolle spielen. Letztendlich heißt aber auch der PC „Personal Computer“, weil er den persönlichen Anforderungen des Endanwenders entsprechen soll – auch hier wäre eine größere Berücksichtigung der Persönlichkeit des Anwenders sehr hilfreich.

6. Kriterien für die Klassifizierung von Identitätsmanagement-Systemen

Der Begriff des Identitätsmanagements wird heutzutage in den Medien bereits als *Buzzword* bezeichnet (zum Beispiel in Kuri 2004). Daran ist zu erkennen, dass viele kommerzielle und nicht-kommerzielle Organisationen das Thema aufgegriffen haben und an Entwicklung von Technologien und Standards beteiligt sein wollen. Allerdings kommt es dadurch zu einer Verwischung der Bedeutung des Begriffs „Identitätsmanagement“, denn jede Partei möchte vor allem die eigenen Ziele verfolgen, die selten alle Konzepte umfassen. Die Entwicklung einer Identitätsmanagement-Infrastruktur wird stark von den Anwendungen getrieben, die der jeweilige Entwickler eigentlich vorantreiben will. Dieses Vorgehen ist nicht grundsätzlich zu kritisieren, da Identitätsmanagement immer eine Hilfsfunktionalität ist, lässt jedoch andere Anwendungen, die auch von Identitätsmanagement profitieren könnten, außer Acht. Die konstruierten Identitätsmanagement-Infrastrukturen sind teilweise nicht interoperabel und verlieren dadurch essentielle Vorteile, wie zum Beispiel die gegenseitige Ergänzung der Anwendergruppen und den Aufbau einer systemunabhängigen Identität der Nutzer (vgl. Baier u. a. 2004).

Demzufolge gibt es wesentliche Unterschiede der derzeit entwickelten Identitätsmanagement-Systeme. Sie unterscheiden sich in der Systemarchitektur, der angebotenen Funktionalität, ihrer Einsatzgebiete und vor allem in ihrer Intention, die Auswirkungen auf die erstgenannten Unterschiede hat. Es fehlt ein gemeinsames Verständnis des Identitätsbegriffs, aber auch des Begriffs des Identitätsmanagements und der Identitätsmanagement-Infrastruktur. Nachdem die Begrifflichkeiten für diese Arbeit in den vorangegangenen Kapiteln dargelegt worden sind, soll hier die grundsätzliche Klassifizierung von möglichen und vorhandenen Identitätsmanagement-Systemen vorgenommen werden. Sie ergänzt die grobe Unterteilung in persönliches, föderiertes und organisatorisches Identitätsmanagement aus Abschnitt 1.2.

6.1. Systemarchitektonische Unterscheidung

Ein technischer Unterschied von Identitätsmanagement-Systemen, der für Anwender keine direkte Rollen spielen sollte, aber Einfluss auf verschiedene andere Eigenschaften hat, ist die Systemarchitektur. Dazu gehören verschiedene Faktoren. Als erstes kann unterschieden werden in *lokale* und *verteilte Systeme*. Ein Identitätsmanagement-System ist nur dann verteilt, wenn es aus mehreren unabhängigen Computern besteht, „die dem Benutzer wie ein einzelnes, kohärentes System erscheinen“ (Tanenbaum und van Steen 2003, S. 18). Agiert das System also nur lokal und nimmt keine Verbindung zu anderen Computern, die auch zum Identitätsmanagement (und nicht etwa zum Web-Server) gehören, handelt es sich um ein lokales System, das die persönlichen Daten verwalten und schützen sowie den Anwender bei der Auswahl der gewählten Teil-Identität unterstützen kann.

Handelt es sich um ein verteiltes System, kann man weitere systemtechnische Unterscheidungen fällen. Das System kann nach dem bewährten *Client/Server-Prinzip* (Tanenbaum und van Steen 2003, S. 61ff) aufgebaut sein oder einem *Peer-to-Peer-Modell* (vgl. Schollmeier und Hermann 2003) folgen. Bei letzterem können die Peers identitätsbezogene Dienste sowohl anbieten als auch nutzen. Die *Datenhaltung* kann beim Client/Server-Modell zentral oder dezentral sein. Diese unterschiedlichen Architekturen haben Einfluss darauf, wieviel Kontrolle die Anwender über ihre Daten haben können, und welche Kommunikationsmöglichkeiten sie haben: Client/Server-Systeme können den Austausch zwischen den Clients steuern und nur bestimmte Verbindungen zulassen.

Zusätzlich kann man unterscheiden, wie *invasiv* ein System ist. Ein Identitätsmanagement-System kann Änderungen sowohl an Applikationen als auch an Systemtechnik (insbesondere dem Netzwerkzugriff) auf einer oder allen Seiten der Kommunikation erfordern. Ein wenig invasives System hat den Vorteil, dass es leichter und schneller eingesetzt werden kann, ein invasives System kann dafür mehr Funktionalität, Sicherheit oder Bedienkomfort bringen.

Ein weiteres Unterscheidungsmerkmal ist das Datenschema, das für die Darstellung der Identitätsdaten verwendet werden kann. Ist dieses fest vorgegeben, kann das System schlecht an neu entstehende Anwendungskontexte angepasst werden. Sind dynamische Datenschemata möglich, kann Semantik-Erhalt zu einem Problem werden (siehe Unterabschnitt 2.2.3), dafür wird mehr Ausdrucksstärke gegeben.

Das Unterscheidungskriterium der *Interoperabilität mit anderen Systemen* hängt mit der Systemarchitektur, den Datenschemata aber auch mit der Betreiberpolitik zusammen. Ist ein Identitätsmanagement-System mit anderen Systemen interoperabel, so können Anwender dieses Systems Dienste und Kontakte der anderen Systeme möglicherweise ohne weiteren Aufwand nutzen. Die Interoperabilität kann verschiedene Formen haben, so kann sie durch serverseitige Zusatzdienste

erreicht werden oder in die gesamte Systemstruktur integriert sein (vgl. Liberty Alliance 2003).

Die Eigenschaften der Invasivität, des Datenschemas und der Interoperabilität können Einfluss auf die Akzeptanz der Nutzer und damit auf die mögliche Marktdurchdringung haben. Die Marktdurchdringung ist ein wichtiger Faktor für die Nützlichkeit des Systems, denn die kritische Masse an Mitgliedern, die ein Identitätsmanagement-System nützlich macht, ist mindestens so groß wie die kritische Masse für eine Online-Community, da diese durch ein solches System unterstützt werden soll. Für Online-Communities gibt sowohl eine untere Grenze, ab der die Community sinnvolle Ergebnisse liefern oder nützlich sein kann, als auch eine obere Grenze, ab der die Anzahl der Mitglieder und die Beiträge zu unübersichtlich wird und damit an Nutzen verliert (siehe Preece 2000, S. 91ff und S.170ff). Da Identitätsmanagement-Systeme jedoch nicht eine einzelne Online-Community unterstützen sondern idealerweise eine große Zahl davon, zwischen denen sich die Mitglieder wiedererkennen können sollen, addiert sich die Zahl zu einer weitaus größeren kritischen Masse und öffnet die obere Grenze (siehe Wollenweber 2004; Baier u. a. 2004).

6.2. Intentionelle Unterscheidung

Ein wichtiger Anhaltspunkt um ein Identitätsmanagement-System zu klassifizieren ist dessen Intention, also die Absicht, mit der es entwickelt und verbreitet wird. Anhand der Intention kann man auf verschiedene Aspekte eines solchen Systems schließen. Diese wirkt sich auf verschiedene Bereiche aus, unter anderem auf die Funktionalität, aber auch auf die Akzeptanz durch die Anwender und damit wiederum auf die mögliche Marktdurchdringung.

Bei der intentionellen Unterscheidung gibt es meistens zwei gegensätzliche oder sogar widersprüchliche Absichten, von denen jeweils verschiedene Systeme getrieben werden. Bei anderen Systemen können diese Absichten weniger deutlich oder ausgeprägt sein. Hier sollen die Intentionen betrachtet werden, die einen starken Einfluss auf Funktionalität oder Systemarchitektur haben.

6.2.1. Anwenderorientierung versus Anbieterorientierung

Die erste und deutlichste Unterscheidung bei der Absicht eines Identitätsmanagement-Systems ist die Gruppe, die den größeren Nutzen daraus ziehen soll: die Anwender oder die Anbieter von Diensten und Inhalten. Bei einem *anwenderorientiertem* System stehen die Punkte im Vordergrund, die den Anwendern der Dienste Vorteile bei der Erreichung ihrer Ziele (siehe Abschnitt 1.3) bringen. Dabei kann entweder die Sicherheit oder die Personalisierung im Mittelpunkt stehen, seltener beides (zu dieser Unterscheidung siehe Unterabschnitt 6.2.2). Die Interessen der Anbieter, die im wesentlichen Kunden binden und Umsätze steigern

wollen, stehen hierbei im Hintergrund. Charakteristisch für diese Systemen ist, dass der Anwender große Kontrolle über seine Daten hat, sie möglicherweise selbst organisiert und speichert. Dies resultiert in einer *dezentralen Systemarchitektur*.

Andere Systeme stellen die Interessen der Dienstanbieter in den Vordergrund und geben den Nutzen des digitalen Identitätsmanagement nur im geringen Maße an den Anwender weiter. Die Dienstanbieter sollen sowohl einfachere Möglichkeiten bekommen, Daten der Anwender zu erfassen und verarbeiten zu können, als auch durch Personalisierung und Geheimhaltung der gewonnenen Daten die Nutzer an den Dienst zu binden. Ein Beispiel dafür ist der Online-Buchhandel Amazon[1], der Profile über die Kunden erstellt, indem sowohl gekaufte als auch betrachtete Bücher gemerkt und ausgewertet werden. Diese Profile werden mit anderen Kunden verglichen, wodurch Voraussagen über mögliche Buchinteressen getroffen werden. Diese Voraussagen werden dem Kunden jedoch nur auf der Web-Seite von Amazon angeboten und nicht in ausführlicher Version zum eigenen Gebrauch in möglicherweise einem anderen Online-Buchhandel offeriert. Hier hat der Kunde weder Kontrolle über die eigenen persönlichen Daten, noch hat er Einsicht darin, was der Dienst über ihn weiß und was damit geschieht.

Anbieterorientierte Dienste nutzen eher *zentrale Systemarchitekturen*, die dem Anbieter Kontrolle über die Daten bieten. Dabei wird darauf geachtet, dass keine Interoperabilität mit anderen Systemen gegeben ist, wodurch verhindert wird, dass die (eventuell unter hohen Kosten) gewonnenen Daten von nicht autorisierten Dritten genutzt werden. Durch die Investition in Identitätsmanagement wird so eine Kundenbindung erwirkt.

6.2.2. Datenschutz versus Selbstdarstellung

Wie im vorigen Unterabschnitt erwähnt, können anwenderorientierte Identitätsmanagement-Systeme entweder den Datenschutz und den Schutz der Privatsphäre als Motivation haben, oder sie haben die Absicht, den Nutzer bei Personalisierung und Selbstdarstellung zu unterstützen. Diese Unterscheidung ist nicht unbedingt exklusiv, es kann durchaus Systeme geben, die beides zum unbedingten Ziel haben, dennoch ist häufig ein Schwerpunkt festzumachen. Systeme, die den Datenschutz in den Vordergrund stellen (*Privacy-Driven*), vermindern teilweise die Selbstdarstellungsoptionen des Anwenders und verhindern so auch die Vorteile, die der Anwender in Form von besserer Personalisierung, möglicher Wiedererkennung anderer und Aufbau einer ausgefeilten digitalen Identität haben könnte. Andererseits können selbstdarstellungs-orientierte Systeme (*Self-Portrayal-Driven*) den Datenschutz gefährden, wenn sie den Anwender zu wenig bei der Auswahl der darzustellenden (beziehungsweise der wegzulassenden!) Identitätsattribute unter-

[1] `http://www.amazon.de`

stützen oder Sicherheitsmechanismen vernachlässigen (zur Unterscheidung in diese Kategorien siehe auch Baier und Kunze 2004b).

Ein Beispiel für ein „Privacy-Driven Identity Management System“ ist das bereits erwähnte „CookieCooker“-Programm (siehe Unterunterabschnitt 3.2.3.2), das für eine stärker gesicherte Privatsphäre sorgt, indem es verhindert, dass Web-Site-Betreiber die Navigation der Anwender verfolgen. Allerdings ist es für die Anwender mit diesem System schwieriger, personalisierte Angebote zu bekommen, da genau dafür der Einsatz von Cookies, die vom CookieCooker verfälscht werden, häufig verwendet wird. Zwar unterstützt CookieCooker das authentische Beibehalten ausgewählter Cookies, aber diese Auswahl zu treffen ist für die Anwender Mehraufwand und teilweise eine schwierige Aufgabe.

Auf der anderen Seite gibt es Entwicklungen, die den Vorteil der Personalisierung bei besserer Selbstdarstellungsmöglichkeit im Netz nutzen wollen (siehe zum Beispiel Koch 2002; Koch und Wörndl 2001). Auch hier stehen die Interessen des Anwenders im Vordergrund, nur dass in diesem Fall die (kontrollierte) Verbreitung der Daten erleichtert und nicht behindert werden soll. Eine Untersuchung hat ergeben, dass Anwendern, die das Internet zu sehr viel persönlicher Kommunikation nutzen, ein selbstdarstellungs-orientiertes Identitätsmanagement-System von großem Nutzen wäre (siehe Baier und Kunze 2004b).

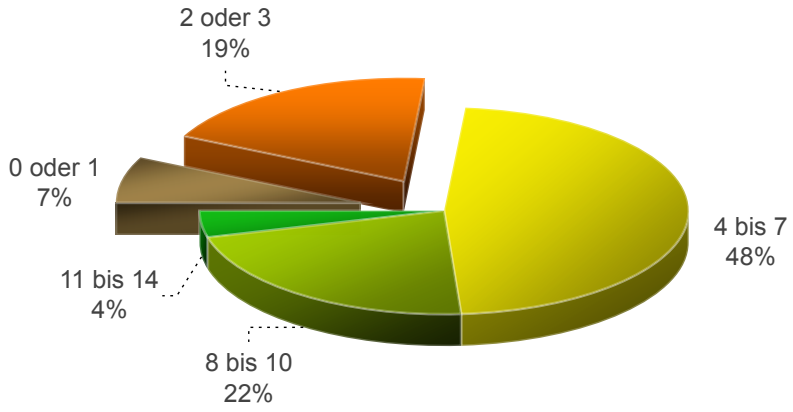

Abbildung 6.1.: Anteil der Nutzer geordnet nach Anteil der über mehrere Dienste preisgegebenen persönlichen Daten. Hervorgehoben sind die Gruppen, für die ein Identitätsmanagement-System wenig Nutzen zur Selbstdarstellung bringen würde.

Dabei wurde mittels eines Online-Fragebogens untersucht, welchen Teil der persönlichen Daten die Anwender über mehrere Dienste hinweg preisgeben würden. Es ergab sich, dass 74% der Befragten vier oder mehr der 14 zur Auswahl

gestellten persönlichen Attribute über mehrere Dienste hinweg teilen würden, davon 26% mehr als die Hälfte der Attribute[2]. Für diese Benutzergruppen würde ein Identitätsmanagement-System eine große Erleichterung beim Verwalten und Übertragen persönlicher Daten über das Internet bringen (siehe Abbildung 6.1).

6.3. Funktionelle Unterscheidung

Zusätzlich zur intentionellen Unterscheidung der Identitätsmanagement-Systeme gibt es die funktionelle Unterscheidung, bei der die Art und Mächtigkeit der angebotenen Funktionen des Systems betrachtet und verglichen werden. Die möglichen Funktionen eines solchen Systems sind in Kapitel 3 aufgeführt.

Eine grundsätzliche Klassifizierung von Identitätsmanagement-Systemen kann anhand des Schwerpunktes der angebotenen Funktionalität erfolgen. Hier decken sich die Ergebnisse ansatzweise mit der Klassifizierung nach der Intention, insbesondere bei der Unterscheidung zwischen „Self-Portrayal-Driven“ und „Privacy-Driven“. Anwendungen, die Selbstdarstellung unterstützen wollen, legen mehr Wert auf Funktionen, die diese unterstützen, also beispielsweise das Ausfüllen von Formularen und das Ermöglichen einer Wiedererkennung in anderen Kontexten (Pseudonym-Management) inklusive des Single Sign-Ons. Auf der anderen Seite werden Anwendungen, die den Schutz der Privatsphäre und den Datenschutz am meisten hervorheben wollen, ihr Gewicht auf Sicherheitsfunktionen wie Verschlüsselung, Anonymisierung und auch Pseudonym-Management legen. Hier wird Pseudonym-Management allerdings mit der Absicht betrieben, die Pseudonyme nicht an anderer Stelle wiederzuverwenden.

Eine pragmatische Einteilung und Bewertung von Identitätsmanagement-Systemen nach Funktionalität und Datensicherheit liegt in einer von der EU beauftragten Studie vor (IMS-Study 2003). In dieser Studie werden zunächst die grundlegenden Anforderungen an Identitätsmanagement-Anwendungen herausgearbeitet. Die Mechanismen, die diese Anforderungen erfüllen können, werden vorgestellt. Danach wird eine umfangreiche Liste aller bekannter Anwendungen und Dienste dargestellt, die im weiteren Sinne zum Identitätsmanagement gehören. Zu diesen Diensten werden folgende Eigenschaften angegeben: Name, Hersteller, Ursprungsland, Status (verfügbar, Prototyp, eingestellter Prototyp, Konzept/Vision), Interoperabilität, Systemarchitektur und Funktionalität. Bei der Funktionalität wird in dieser Liste lediglich nach fünf grundsätzlichen Funktionen unterschieden: Access Management (Zugriff auf und Verwaltung von Identitätsdaten), Form Filling (siehe Unterunterabschnitt 3.3.2.1), automatische Rollenauswahl, Pseudonymverwaltung, Erreichbarkeitsmanagement und in einer Freitextspalte

[2] Die Auswahl der 14 angebotenen Attribute war so getroffen, dass sie von „sehr persönlich“ bis „sehr öffentlich“ variierten, was sich auch in der Anzahl der Nennungen durch die Befragten widerspiegelte.

eventuelle Zusatzfunktionalität. Eine genauere Untersuchung folgt im Anschluss, betrachtet allerdings nur die Systeme, die in Forschung oder Praxis eine relevante Rolle spielen. Die Studie ist sehr umfangreich (sie umfasst 327 Seiten) und kann als Referenzwerk für die Untersuchung bestehender Identitätsmanagement-Systeme gewertet werden. In dieser Arbeit kann nur ein Ausschnitt davon dargestellt werden.

6.4. Zusammenfassung der Kategorisierung

Im folgenden Kapitel sollen ausgewählte Identitätsmanagement-Systeme auf die zuvor erklärten Unterschiede untersucht und danach kategorisiert werden. Dabei kann nicht jedes System in jeder der Kategorien eingeordnet werden – zum Beispiel lässt sich bei einem lokalen System keine Unterscheidung zwischen Peer-to-Peer oder Client/Server treffen, und ein anbieterorientiertes System kann nicht Privacy-Driven sein (auch wenn die Anbieter Wert auf die Privatsphäre des Anwenders legen, ist dies nicht die Hauptintention des Systems). Zur Übersicht sind die Unterschiede in Tabelle 6.1 aufgelistet. Dabei werden die funktionellen Unterschiede zunächst außer Acht gelassen. Sie werden während der Untersuchung im einzelnen aufgelistet, da sie zu vielfältig sind als dass sie in weitere Kategorien aufgeteilt werden könnten.

Tabelle 6.1.: Unterscheidungen bei Identitätsmanagement-Systemen

lokal	verteilt
Client/Server	Peer-to-Peer
zentrale Datenhaltung	dezentrale Datenhaltung
stark invasiv	wenig invasiv
statisches Datenschema	dynamisches Datenschema
interoperabel	nicht interoperabel
anwenderorientiert	anbieterorientiert
Privacy-Driven	Self-Portrayal Driven

Beim Entwurf eines Identitätsmanagement-Systems spielt die Entscheidung für die angestrebte Kategorie eine wichtige Rolle für die Ausprägung der in Kapitel 4 genannten Basis-Elemente des Systems. Ist zum Beispiel ein statisches Datenschema vorgesehen, so ist die Verwaltung verschiedener Ontologien nicht nötig. Ist das System auf Datenschutz ausgerichtet, wird der Hauptaugenmerk auf den Elementen für die dynamische Nutzung und dabei auf der Pseudonymisierung und der Sicherung der Privatsphäre liegen; die Unterstützung beim Aufbau von

Kommunikationssitzungen zum Austausch von Identitätsdaten tritt dabei in den Hintergrund.

7. Identitätsmanagement-Systeme in Praxis und Forschung

Einige Identitätsmanagement-Systeme werden bereits im praktischen Umfeld eingesetzt. Diese sollen in diesem Kapitel analysiert und anschließend in die zuvor erarbeiteten Kategorien eingeordnet werden. Dabei liegt der Hauptaugenmerk auf den Systemen, die eine große Marktrelevanz haben, also auf Microsofts *.NET Passport*-System und der *Liberty Alliance*. Zusätzlich werden hier einige Systeme aus der Forschung betrachtet, deren Prototypen interessante Varianten in Ausrichtung und Systemarchitektur aufweisen.

7.1. Microsoft .NET Passport

Microsoft hat das Bedürfnis vieler Internet-Anwender nach einer einfachen Single Sign-On Lösung aufgegriffen und im Zuge der Entwicklung der .NET Plattform den „Microsoft .NET Passport“ entwickelt. Ziel der Entwicklung ist es, mit einem Benutzernamen und einem Passwort alle Dienste, die .NET Passport unterstützen, nutzen zu können. Die Authentifizierung läuft immer über den .NET Passport Server und wird dann per HTTP-Redirect an den Ziel-Server weitergeleitet. Genutzt wird das System hauptsächlich von Microsoft selbst: alle Benutzer des kostenlosen E-Mail-Dienstes HotMail[1] sowie alle Benutzer jeglicher Dienste des Microsoft Networks (wie beispielsweise des MSN Messengers, eines sehr beliebten Instant Messengers) müssen seit 2002 den .NET Passport zur Authentifizierung nutzen. Dazu wurden ihre Accounts automatisch umgestellt, so dass sich weder Benutzername noch Passwort geändert haben. Es besteht dadurch die berechtigte Befürchtung, dass technisch weniger versierte Benutzer die Umstellung gar nicht bemerken, sich wie gewohnt in ihren Dienst einloggen und nicht merken, dass sie nach Verlassen der entsprechenden WWW-Seite immer noch eingeloggt und somit erkennbar für alle anderen .NET Passport Dienste sind. Desweiteren sind schon mehrere schwere Sicherheitslücken aufgetreten, die unter anderem dazu geführt haben, dass Benutzer nach der Anmeldung als andere Benutzer authentifiziert waren und somit deren E-Mails lesen konnten. Alle privatsphärenrelevanten Bedenken werden auf der WWW-Seite „Sign Out of Microsoft Passport“[2] gesammelt.

[1] `http://www.hotmail.com`
[2] `http://www.epic.org/privacy/consumer/microsoft/`

Durch das automatische Generieren von .NET Passport Nutzern durch die Umstellung der Authentifizierung zu Microsofts HotMail und MSN-Produkten ist die Nutzerzahl des .NET Passports laut Microsoft bereits auf über 200 Millionen gewachsen, während die Anzahl der Authentifizierungen durch .NET Passport bei insgesamt 3,5 Milliarden liegt. Durch die vielen Anwender wird es auch für andere Anbieter interessant, die Technologie zu lizenzieren. Bekanntester Lizenznehmer in Deutschland ist bisher der elektronische Auktionshandelsplatz eBay[3]. Allerdings ist seit Anfang 2005 bekannt, dass auch eBay sich wegen verschiedener Sicherheitsprobleme von .NET Passport verabschieden wird.

Zusätzlich zur Authentifizierung können auf dem .NET Passport Server weitere Informationen hinterlegt werden. Dazu gehören eine E-Mail-Adresse (dies ist eigentlich der Identifikator des Passports), Vor- und Nachname, Land oder Region, Geschlecht, Geburtsdatum und Beruf / Beschäftigung. Zu diesen Daten kann man die Sichtbarkeit einrichten: in der Grundeinstellung wird vom Microsoft-Server nur die E-Mail-Adresse an andere .NET Passport-nutzende Dienste herausgegeben. Zusätzlich kann man den Namen freigeben und mit einer weiteren Einstellung alle weiteren Daten auf einmal. Diese Freigaben sind nicht dienstspezifisch – sie gelten für alle Dienste, die mit dem Passport genutzt werden.

Bis März 2003 konnte auch die Kreditkartennummer im so genannten „Passport Wallet“ gespeichert werden. Diese Funktion hat sich als zu unsicher erwiesen, weswegen sie abgestellt wurde. Es ist allerdings laut Microsoft nicht sicher, dass alle Dienste, die über das Passport Wallet die Kreditkartennummer erhalten haben, diese auch gelöscht haben. Hier wird das Problem deutlich, dass es keinen technischen Weg gibt, einmal herausgegebene Daten sicher wieder zu löschen.

Desweiteren gibt es die Option eines „Kids Passport“. Dieser soll nicht als Filter ungeeigneter Inhalte dienen, wie es andere Produkte zum Schutz von Kindern im Internet anbieten, sondern vielmehr die Möglichkeit bieten, die Eltern der Kinder um Zustimmung zur Sammlung und Weitergabe personenbezogener Daten der Kinder zu befragen: „Kids Passport is not a filtering service that blocks your child from visiting Web sites on the Internet. It is simply a service used by some Web sites to obtain parental consent for the collection, use, and disclosure of children's personal information.“ (aus dem „.NET Passport Kids Privacy Statement“, unter `http://kids.passport.com` abrufbar (Stand: April 2004)).

7.1.1. Systemarchitektur

Das Konzept hinter dem Passport ist ein zentraler Server, der die Profile aller Nutzer speichert, Authentifizierungsanfragen beantwortet und weitere Daten preisgibt.

[3] `http://www.ebay.de`

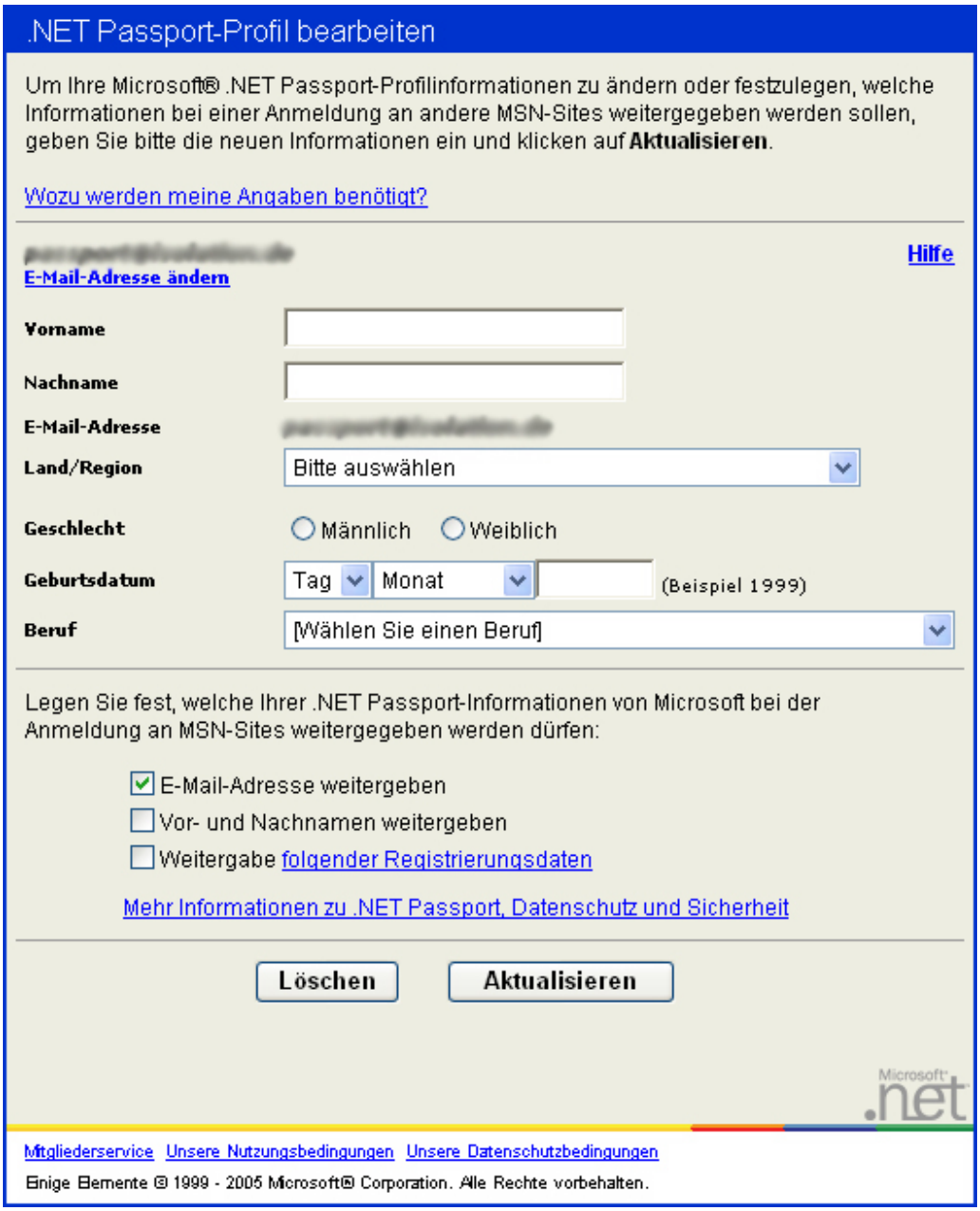

Abbildung 7.1.: Auf dieser Web-Seite kann ein .NET Passport-Profil editiert und die Sichtbarkeit der Elemente eingestellt werden.

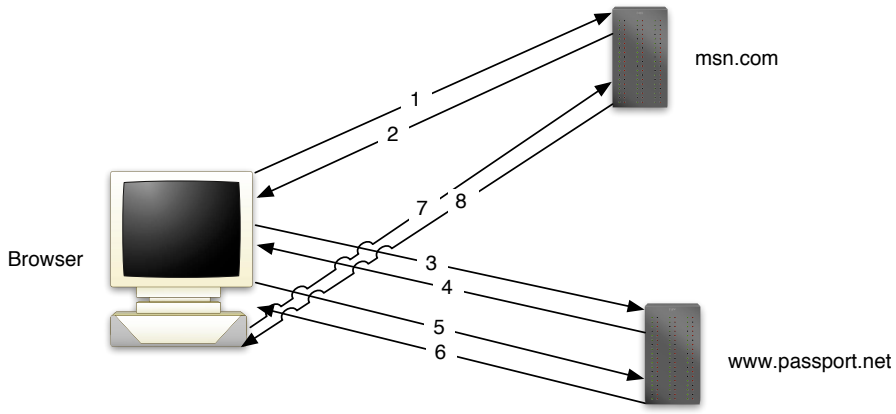

Abbildung 7.2.: Schema des Passport-Protokolls, Beschreibung siehe Text

Das Protokoll folgt dem in Abbildung 7.2 dargestelltem Schema: 1. Der Browser des Anwenders stellt eine Anfrage an einen Server (hier: `msn.com`), für die Autorisierung und damit Authentifizierung notwendig ist. 2. Der Server schickt einen HTTP-Redirect-Befehl an den Browser, der ihn in 3. an den allen bekannten `www.passport.net` Server weiterleitet. 4. Dieser fragt den Benutzer nach Authentifizierung (.NET Passport Benutzerkennung und Passwort), welche in 5. gesendet wird. 6. ist ein HTTP-Redirect zurück zum ursprünglichen Server, wobei die Authentizität des Anwenders im HTTP-Header vermerkt wird. Dieser Vermerk ist mit einem Schlüssel verschlüsselt, der zuvor zwischen Server (`msn.com`) und Passport-Server ausgetauscht worden ist. In Schritt 7 wird diese Information an den Server weitergeleitet, welcher in 8. einen (ebenfalls verschlüsselten) Cookie setzt, der den Anwender als authentifiziert auszeichnet.

Dieses System bietet einige Angriffspunkte, welche die Sicherheit der Nutzer gefährden (vgl. Kormann und Rubin 2000). Diese setzen nicht nur auf Schwachstellen im Passport-Protokoll auf, sondern auch auf den verwendeten Technologien SSL und Cookies. Während der Gebrauch ausschließlich schon vorhandener Technologien den Einsatz von Passport ohne Veränderung der Clients oder der Server ermöglicht, eröffnet dies Angreifern auch die Möglichkeit, vorhandene Sicherheitslücken auszunutzen. Im Falle von SSL wird vornehmlich das Benutzbarkeitsproblem erwähnt: Der Anwender bemerkt im Normalfall die Verwendung von SSL nicht. Sobald ein SSL-Schlüssel von einer der vielen akzeptierten Root-Schlüssel zertifiziert ist, wird dem Anwender keinerlei Informationen über die benutzten Server aufgezeigt. Kormann und Rubin argumentieren, dass SSL sich nicht zur Delegation eignet, aber gerade diese Funktion wird von .NET Passport genutzt.

7.1.2. Klassifizierung

Microsoft .NET Passport ist ein anbieterorientiertes System mit zentraler Datenhaltung nach dem Client/Server-Aufbau. Es ermöglicht Single Sign-On bei Anbietern, die das System von Microsoft lizenzieren sowie geringe, grobaufgelöste Identitätsmanagement-Funktionen mit statischem Datenschema. Der Anwender bekommt wenig Möglichkeiten, den verschiedenen Anbietern unterschiedliche Daten aus dem Passport zukommen zu lassen und gar keine Möglichkeit, diese Daten anderen als Microsofts Partnern zu zeigen. Das schließt auch den Gebrauch für private Kommunikation aus. Das System ist mit keinem anderen System interoperabel. Auf der Seite des Anwenders muss keine zusätzliche Software installiert werden, damit das System benutzt werden kann, der Dienstanbieter muss sein System dagegen anpassen.

7.2. Liberty Alliance

Die Liberty Alliance[4] ist ein von Sun Microsystems initiiertes Konsortium sehr vieler Firmen, die nicht notwendigerweise alle aus dem IT-Bereich kommen, aber ein Interesse an digitalen Identitäten und den Austausch dieser haben. Gemeinsam wurde eine Spezifikation entwickelt, die aufbauend auf bestehenden Mechanismen wie SAML[5] (siehe Madsen 2004) und Web Services (Web Services Architecture 2004) den Austausch von Identitätsdaten innerhalb einer so genannten „Föderation“ (engl: *federation*) ermöglicht. Eine Föderation ist ein Zusammenschluss von Diensten, die sich auf den Austausch von Identitätsdaten verständigt haben. Nutzt ein Anwender einen dieser Dienste und möchte anschließend einen weiteren in Anspruch nehmen, so können diese Dienste nach Zustimmung des Anwenders die relevanten Daten austauschen. Diese Sichtweise auf Identitätsdaten wird auch „föderierte Identität“ (engl: *federated identity*) genannt, um den Gegensatz zur abgeschlossenen, nur innerhalb eines Unternehmens sichtbaren digitalen Identität zu verdeutlichen (siehe dazu auch Abschnitt 1.2 und Unterabschnitt 2.2.2).

Ein Beispiel dafür ist eine Föderation aus einer Fluglinie und einer Autovermietung. Hat ein Anwender bei der Fluglinie einen Flug gebucht und wünscht anschließend ein Auto am Zielort zu reservieren, so kann er der Fluglinie erlauben, der Autovermietung die Ankunftszeit am Zielort mitzuteilen, damit sie dort nicht erneut eingegeben werden muss. Zusätzlich können identifizierende Daten weitergegeben werden, wodurch ein Single Sign-On erreicht wird.

[4] `http://www.project-liberty.org`
[5] Security Assertion Markup Language

7.2.1. Systemarchitektur

Das System bei Liberty-Alliance-Anwendungen kann verschiedenen Architekturen folgen, jedoch ist es prinzipiell dezentral, denn einen zentralen Server, der alle Authentifizierungen erledigt wie bei .NET Passport, gibt es nicht. Innerhalb eines „Vertrauenskreises“ (engl: *circle of trust*, gemeint ist oben erwähnte Föderation plus den Anwender) kann es einen oder auch mehrere so genannte *Identity Provider* geben, die sowohl Identifizierungen als auch Identitätsdaten anbieten (siehe Hodges und Wason 2003). Jeder Partner einer Föderation kann diese Aufgabe übernehmen. Benutzer können wählen, welchen Identity Provider sie einsetzen wollen – dabei kann man eine digitale Identität auch bei anderen Diensten der Föderation einsetzen, die selbst einen Identity Provider anbieten. Hat sich ein Anwender gegenüber einem der Identity Provider authentifiziert, kann ein Cookie gesetzt werden, der von einem allen Anbietern der Föderation zugänglichen Web-Server stammt und damit für alle Mitglieder sichtbar ist.

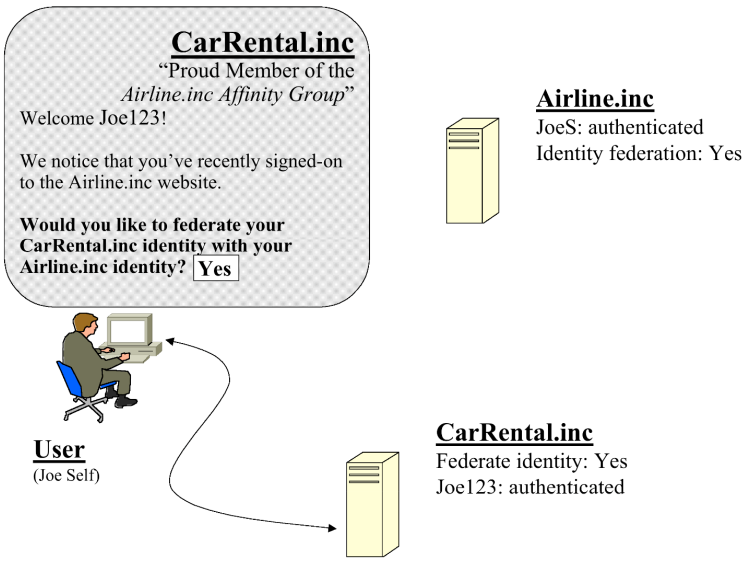

Abbildung 7.3.: Ein Liberty Alliance Benutzer wird aufgefordert, seine Identität zu föderieren (aus Hodges und Wason 2003).

Mit dieser Technik sind mehrere Szenarien möglich – eines davon entspricht dem Protokoll von .NET Passport, nur dass der Server, der die Authentifizierung vornimmt, möglicherweise durch den Benutzer wählbar ist. Ein weiteres, interessantes Szenario tritt auf, wenn ein Benutzer sich schon gegenüber einem Mitglied einer Föderation authentifiziert hat und danach eine Web-Seite eines weiteren Mitglieds besucht. Diese kann anhand des gemeinsamen Cookies feststellen, dass der Benutzer schon authentifiziert ist und fragen, ob der Anwender

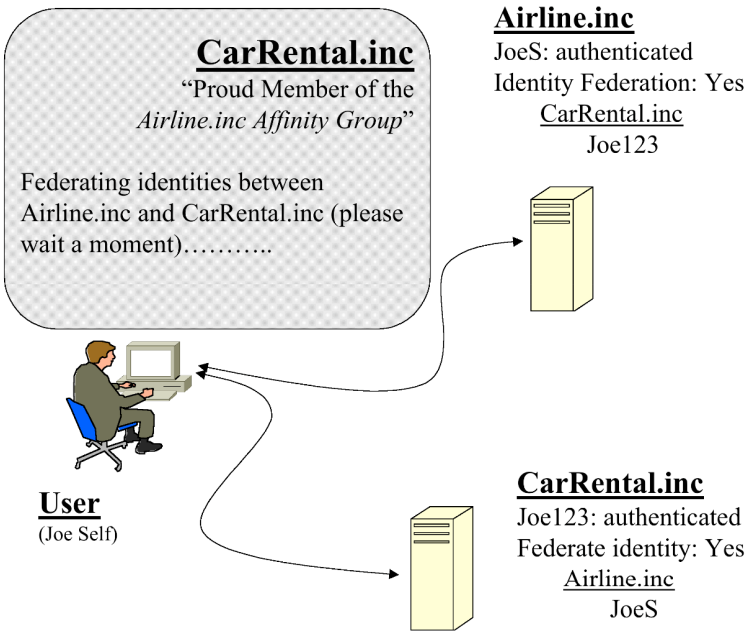

Abbildung 7.4.: Die Identitätsdaten werden ausgetauscht (aus Hodges und Wason 2003).

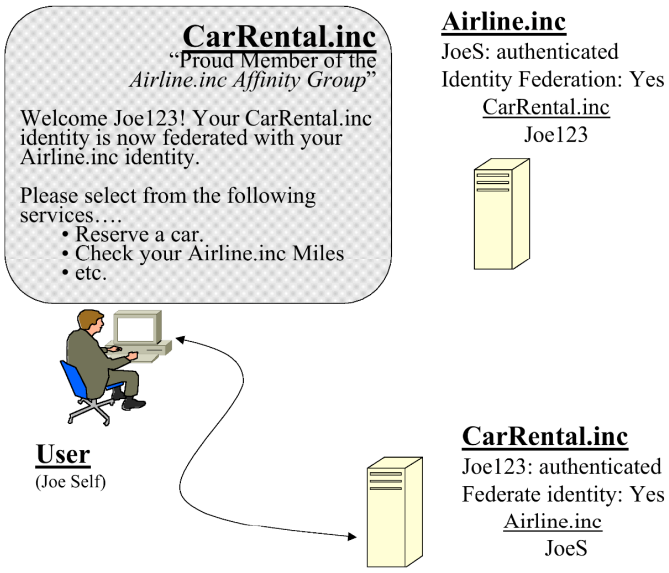

Abbildung 7.5.: Die Identität des Anwenders ist in der Föderation bekannt (aus Hodges und Wason 2003).

seine Identität „föderieren“ möchte, das heißt die Attribute, die er schon dem ersten Mitglied gezeigt hat, auch dem zweiten offenbaren (siehe Abbildungen 7.3, 7.4 und 7.5).

Auch Liberty Alliance erlaubt die Nutzung des Protokolls mit normalen Web-Browsern, was den Vorteil der einfacheren Verbreitung des Systems hat, jedoch den Nachteil, dass das System den Einschränkungen der für diese Funktionalität nicht gedachten Protokollen unterliegt. Für die Liberty Alliance ist daher auch die Nutzung mit „enabled clients“ vorgesehen, die auf Anwenderseite den Browser in seiner Funktionalität erweitert und eine Web-Service-basierte Kommunikation zwischen Client und Identity Provider ermöglicht. Eine Man-In-The-Middle-Angriffsmöglichkeit, die in der ersten Version dieses Teils der Spezifikation gegeben war, ist durch eine Aktualisierung beseitigt worden (vgl. Pfitzmann und Waidner 2002b).

7.2.2. Implementierung: SourceID

Die Liberty Alliance versteht sich lediglich als Organisation zur Definition von Protokollen zur Nutzung von föderierten Identitäten und Diensten, die diese föderierten Identitäten nutzen. Implementierungen oder Werkzeuge für den Einsatz dieser Protokolle werden nicht direkt angeboten. Neben etlichen, firmeninternen Implementierungen der Liberty Alliance Mitglieder existiert ein Open Source Werkzeug der Protokolle namens „SourceID“[6]. Angeboten wird eine vollständige Umsetzung der Liberty Alliance Protokolle und Komponenten sowie eine eigene SAML Implementierung, die jeweils für die Programmiersprache Java und Microsofts .NET Komponentenarchitektur vorliegen. SourceID wird von der mitgliedergetragenen Organisation PingID[7] kommerziell betrieben.

7.2.3. Klassifizierung

Die Liberty Alliance ist anbieterorientiert und nutzt ein Client/Server-Modell bei dezentraler Datenhaltung. Sie erlaubt es Anwendern nicht, ihre Identitätsdaten unter beliebigen Diensten zu teilen, sondern nur solchen, die gemeinsam einer Föderation angehören. Es ist nicht notwendig, eine einzige Online Identität für alle Zwecke einzusetzen. Dagegen ist es sogar möglich, innerhalb einer Föderation mehrere digitale Identitäten zu pflegen. Die offene Architektur der Liberty Alliance erlaubt interoperable Implementierungen zu anderen Systemen, allerdings kann eine digitale Identität immer nur innerhalb einer bestimmten Föderation genutzt werden, was den Interoperabilitätsaspekt einschränkt.

Es gibt eine weitere Spezifikation für föderierte Identitäten: die Internet2-Organisation[8], die unter anderem auch das Internetprotokoll Version 6 (IPv6)

[6] `http://www.sourceid.org`
[7] `http://www.pingid.net`

vorangetrieben hat, entwickelt ein System namens *Shibboleth* (vgl. Pfitzmann und Waidner 2002a), das ebenso wie Liberty Alliance auf SAML basiert und Angehörigen amerikanischer Universitäten Autorisierung für den Zugriff auf Ressourcen anderer Universitäten ermöglichen soll. Es wird derzeit untersucht, inwiefern Shibboleth-Systeme mit Liberty Alliance kompatibel sind (vgl. Liberty Alliance 2003).

Liberty Alliance Implementierungen können sehr wenig invasiv sein und nur Änderungen der Serveranwendungen erfordern, aber auch sehr invasiv inklusive Änderung der Client-Systeme.

Das Datenschema bei Liberty Alliance kann frei gestaltet werden und ist offen für beliebige Anwendungskontexte. Grundsätzlich lässt sich sagen, dass Liberty Alliance durch die Offenheit der Protokolle und der Anpassungsfähigkeit der Datenschemata mit sehr vielen anderen Identitätsmanagement-Systemen interoperabel sein kann, indem sie in Liberty Alliance integriert werden.

Die Einordnung nach funktioneller Kategorie ist bei der Liberty Alliance fallweise nach Implementierung vorzunehmen, weil auf Anwendungsebene verschiedene Funktionen angeboten werden können. Hauptsächlich ist die Liberty Alliance auf Single Sign-On und Identitätsdatenaustausch zwischen den Anbietern ausgelegt.

7.3. CookieCooker und JAP

Wie schon in Unterunterabschnitt 3.2.3.2 beschrieben ist das Programm *CookieCooker* ein System zum automatischen Verwischen von Spuren (in Form von Cookies), die beim Surfen im WWW von Servern auf dem Rechner des Anwenders hinterlassen werden. Dies soll die Privatsphäre des Anwenders erhöhen, indem Anbieter verfälschte Daten über die Historie des Anwenders präsentiert bekommen, sollte diese im Cookie gespeichert sein.

Zur Nutzung von Diensten, die ohne Cookie nicht sinnvoll genutzt werden können, weil die richtige Identität preisgegeben werden muss (beispielsweise ein Web-E-Mail-Account), ermöglicht CookieCooker eine Art Identitätsmanagement, bei dem der Anwender festlegen kann, welche (beziehungsweise von welchen Servern) Cookies nicht vertauscht werden dürfen.

Zusätzlich wird zur weiteren Erhöhung der Privatsphäre das Programm *JAP*[9] mit ausgeliefert. Dies ist ein lokal zu installierender Proxy für Web-Anwendungen, der alle Anfragen über ein MIX-Netz (siehe Chaum 1981) weiterleitet und damit Anonymität auf IP-Ebene bietet. Diese Kombination von Privatsphärenmechanismen auf Anwendungsebene (CookieCooker) und Transportebene (JAP) deckt einen großen Bereich der Angriffsfläche ab.

[8] `http://www.internet2.org`
[9] `http://anon.inf.tu-dresden.de`

Abbildung 7.6.: Die Oberfläche des CookieCooker-Clients

7.3.1. Klassifizierung

CookieCooker ist ein anwenderorientiertes System, insbesondere in Verbindung mit JAP. Die Intention des Systems ist es, dem Anwender mehr Privatsphäre zu sichern und vor unautorisierter Beobachtung durch Dienstanbieter mittels Cookies oder IP-Adressen-Nachforschung zu schützen. Der Austausch der Cookies geschieht über ein Peer-to-Peer-System. Die Identitätsmanagement-Funktionalität beschränkt sich auf die Auswahl der Cookies, die nicht verfälscht werden sollen, es gibt also kein eigenes Datenschema für Identitätsdaten. Da lediglich auf der Clientseite ein Proxy installiert werden muss, der nichts an der Transportschicht oder den Anwendungen ändert, ist das System sehr wenig invasiv.

7.4. ATUS: A Toolkit for Usable Security

An der Universität Freiburg wird im Rahmen des Projektes ATUS[10] (A Toolkit for Usable Security) das Identitätsmanagement-System iManager entwickelt. Es hat das Ziel, die Privatsphäre der Anwender zu verbessern, indem es ihn bei der Auswahl der verwendeten Teil-Identitäten unterstützt. Dabei liegt der Hauptaugenmerk auf der Benutzbarkeit: Bei der Gestaltung des Systems werden moderne Software-Entwicklungsprozesse eingesetzt (siehe Gerd tom Markotten 2002). Grundlegende Anonymisierung, Verschlüsselung, aber auch Zertifizierung und damit Zurechenbarkeit von Aussagen sollen auf anwenderfreundliche Art und Weise in einem integriertem System angeboten werden. Die Anwenderfreundlichkeit ist dabei nicht mit der Anwenderorientierung gleichzusetzen, wie sie in Unterabschnitt 6.2.1 dargestellt ist: obwohl sie diese unterstützt, ist nicht jedes an-

[10] `http://tserv.iig.uni-freiburg.de/telematik/forschung/projekte/kom_technik/atus/index.html`

wenderfreundliche System auch anwenderorientiert. Vielmehr soll durch die Anwenderfreundlichkeit erreicht werden, dass Benutzer die Sicherheitsmechanismen tatsächlich als Hilfe (und nicht als Hürde) empfinden und auch sinnvoll einsetzen können. Falscher Einsatz von Sicherheitsmechanismen kann zu einer Minderung der Sicherheit führen (vgl. Jendricke und Gerd tom Markotten 2000).

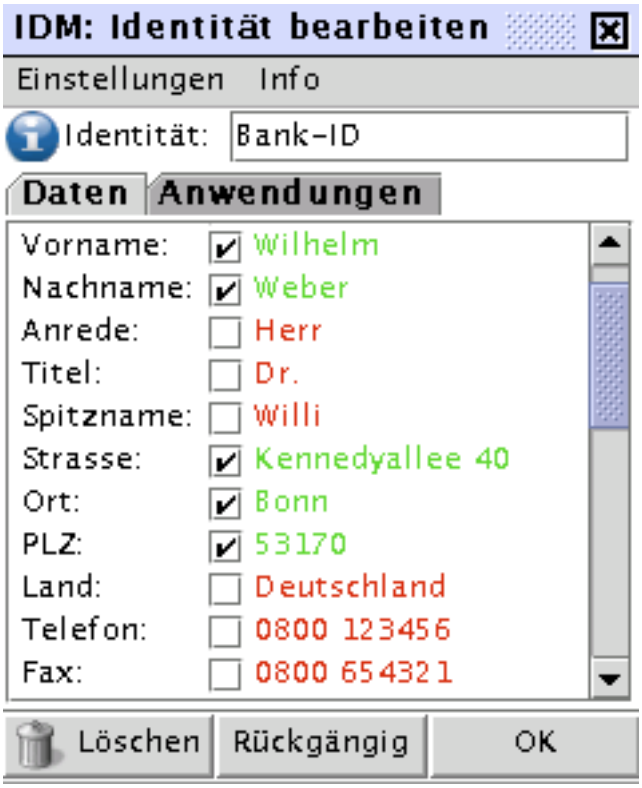

Abbildung 7.7.: Bedienoberfläche des ATUS iManager

7.4.1. Systemarchitektur

Der iManager fungiert als Proxy zwischen Anwendungen und dem Internet. Nach außen kommuniziert das System über einen Anonymisierungsdienst wie *JAP*. Die Personalisierung erfolgt auf Anwendungsebene. Dabei setzt er nicht wie der Identity Manager des DRIM-Projektes (siehe Abschnitt 7.5) auf veränderte Protokolle mit angepassten Servern, sondern verwendet einen Datenfilter, um Identitätsdatenkommunikation zu erkennen. Mit Hilfe von *P3P*[11], der „Platform for Privacy

[11] `http://www.w3.org/P3P`

Preferences“ wird anhand der Datenschutzpolitik des Dienstanbieters eine Teil-Identität ausgehandelt beziehungsweise dem Anwender vorgeschlagen (vgl. Cranor 2002). Die Datenhaltung der Identitätsdaten wird mit einer generischen Datenbank gelöst.

Insbesondere eignet sich der iManager zum Einsatz auf mobilen Geräten wie PDAs (*Personal Digital Agent*) (vgl. Jendricke u. a. 2002). Gerade beim mobilen Identitätsmanagement sehen die Entwickler des Systems bisher einen großen Mangel an Datenschutz. In Abbildung 7.7 ist die Bedienoberfläche des iManager auf einem PDA zu sehen.

7.4.2. Klassifizierung

Der ATUS iManager ist ein lokales, anwenderorientiertes System, das vor allem die anwendergerechte Benutzbarkeit (*usability*) der Software in den Vordergrund stellt. Es erfordert keine Systemänderungen auf Seiten der Dienstanbieter, wodurch es schon jetzt universell einsetzbar ist. Allerdings lässt sich eine Interoperabilität mit anderen Systemen nur schwer (über die Anpassung des Datenfilters) einrichten.

7.5. Dresden Identity Management

An der Technischen Universität Dresden wird ein Identitätsmanagement-System mit dem Namen *DRIM*[12] (Dresden Identity Management) entwickelt, das auf verschiedenen anderen Technologien aufbaut (Clauß und Köhntopp 2001; Köhntopp und Pfitzmann 2001). Dazu zählt auch der ebenfalls an der TU-Dresden entwickelte Anonymisierungsproxy *JAP* und die Bibliothek *SSONET* zum Verschlüsseln der Daten. Desweiteren gibt es einen Service *PKI*, mit dem man Pseudonyme beglaubigen kann.

Hauptziel von DRIM ist es, dem Anwender im Internet mehr Privatsphäre und gleichzeitig multilaterale Sicherheit (vgl. Pfitzmann 2001) zu gewähren. Dazu wird der Anwender zunächst mittels eines Chaum Mixes (siehe Chaum 1981) anonymisiert, um ihm anschließend durch den Einsatz von Pseudonymen innerhalb eines bestimmten Rahmens erkennbar zu machen. Alle Kommunikationsakte werden verschlüsselt.

7.5.1. Systemarchitektur

Die Systemarchitektur von DRIM sieht es vor, auf der Seite des Anwenders einen Proxy zu installieren, durch den alle Kommunikationsvorgänge geleitet werden. Dazu muss im Browser oder im Anwendersystem dieser lokale Proxy eingerichtet

[12] `http://drim.inf.tu-dresden.de`

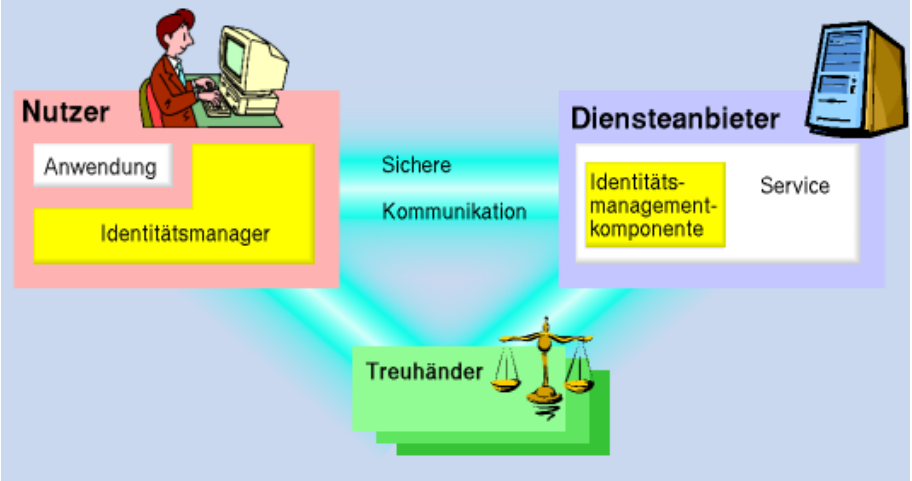

Abbildung 7.8.: Kommunikationsverbindung mit DRIM

werden. Der Proxy übernimmt die Anonymisierung mittels JAP, die Verschlüsselung sowie das Verwalten der Pseudonyme. Während die Anonymisierung auf der Routing-Ebene (IP) keine Änderung des Dienstes erfordert, ist dies jedoch notwendig für den Einsatz verschlüsselter Verbindungen oder Pseudonymisierung. Es liegt eine Anpassung des Web-Servers *JIGSAW*[13] vor, die mit den DRIM Pseudonymen umgehen kann. Die Übertragung der Identitätsdaten wird mittels eines angepassten P3P-Protokolls durchgeführt, wobei dieses eigentlich nur zur Aushandlung der Datenschutzregeln spezifiziert ist.

Es ist vorgesehen, dass Identitätsdaten nicht nur im lokalen System abgelegt werden können (dies wird von der Komponente *PSMAN* übernommen), sondern auch von *Identity Brokers* im Netz verwaltet werden können. Mit diesen *Trusted Third Parties* oder Treuhändern (siehe Abbildung 7.8) wird so eine *PKI* (Public Key Infrastructure) aufgebaut.

7.5.2. Klassifizierung

DRIM ist ein anwenderorientiertes System im prototypischen Status. Es hat den Datenschutz als höchste Priorität und will Anwender dabei unterstützen, ihre Kommunikation im Internet zu sichern. Allerdings ist die Benutzbarkeit der derzeitigen prototypischen Version noch nicht besonders weit fortgeschritten, wodurch im Moment keine Marktdurchdringung zu erwarten ist. Allerdings wird das Projekt aktiv weiterentwickelt und im Rahmen des von der EU gestützten

[13] `http://www.w3.org/Jigsaw/`

Integrierten Projektes PRIME[14] (Privacy and Identity Management for Europe) in einen größeren Zusammenhang gebracht.

Ähnlich wie CookieCooker beschränkt sich die Invasivität auf Clientseite auf die Installation eines Web-Proxys – damit die Identitätsmanagement-Aspekte jedoch nutzbar sind, ist ein tieferer Eingriff in die Server-Systeme nötig. Der Einsatz standardisierter Protokolle und Mechanismen erlaubt eine spätere Interoperabilität mit anderen Diensten. Allerdings könnte der Gebrauch des modifizierten P3P-Protokolls zu Inkompatibilitäten führen. Das Architekturmodell von DRIM ist prinzipiell Client/Server-basiert – es ist noch nicht klar, ob das System auch Peer-to-Peer-Szenarien unterstützen soll.

7.6. Zusammenfassung

An den vorangegangenen Betrachtungen kann man erkennen, dass es bereits eine Vielfalt an (teilweise prototypisch) existierenden Identitätsmanagement-Systemen gibt. Sie unterscheiden sich in mehreren Kategorien und sind daher nur teilweise miteinander vergleichbar. Es lassen sich jedoch grundlegend unterschiedliche Vorgehensweisen in allen genannten Kategorien erkennen. So existieren große Gegensätze beim stark zentralisierten .NET Passport System und dem stark dezentralsisierten System DRIM. Die Invasivität ist bei CookieCooker nur minimal gegeben, bei Liberty Alliance abhängig von der Anwendung. Die Gestaltung des Datenschemas, ein wichtiger Faktor zur Öffnung für neue Anwendungsgebiete, rangiert von fest und grob (.NET Passport) bis frei wählbar und dynamisch (DRIM). Die Ausrichtung oder Absicht ist bei den kommerziellen Systemen .NET Passport und Liberty Alliance prinzipiell anbieterorientiert, während sie bei CookieCooker und ATUS stark anwenderorientiert ist.

Es existieren also in allen Kategorien Extreme, obwohl nicht jede Kategorie in jedem System extrem ausgeprägt sein muss. In Tabelle 7.1 werden die für diese Arbeit wesentlichen Punkte zusammenfassend dargestellt. Es fällt jedoch auf, dass ein anwenderorientiertes System, das Selbstdarstellung als Intention hat, fehlt. Anwenderorientierte Systeme sind bisher vor allem durch Sicherheits- und Datenschutzaspekte motiviert. Zusammen mit der diese Arbeit einleitenden Motivation ergibt sich daraus, dass Konzepte und Systeme für diesen bisher vernachlässigten Bereich eine Bereicherung der Systemlandschaft wären. Teil III dieser Arbeit hat daher zum Ziel, solche Konzepte aufzuzeigen und die Implementierung eines solchen Systems zu demonstrieren.

[14] `http://www.prime-project.eu.org`

Tabelle 7.1.: Kategorisierung der vorgestellten Identitätsmanagement-Systeme

	.NET Passport	Liberty Alliance	Cookie-Cooker	ATUS	DRIM
Architektur	Client/-Server	Client/-Server	Peer-to-Peer	lokal	Client/-Server
Datenhaltung	zentral	dezentral (Server)	lokal	lokal	dezentral (lokal / TTP)
invasiv	wenig (Server)	je nach Anwendung	sehr wenig	wenig	nur Server
Datenschema	fest	serverseitig	–	fest	dynamisch ch
Interoperabilität	keine	möglich	keine	möglich (aufwändig)	möglich
Orientierung	Anbieter	Anbieter	Anwender	Anwender	Anwender
Intention	–	–	Privacy-Driven	Privacy-Driven	Privacy-Driven

Teil III.

Zur Realisierung eines selbstdarstellungsorientierten Identitätsmanagement-Systems

Zusammenfassung

Nachdem im ersten Teil dieser Arbeit die grundlegenden Konzepte des Identitätsmanagements betrachtet und in Teil zwei die darauf aufbauenden Identitätsmanagement-Systeme analysiert worden sind, sollen in diesem dritten Teil Konzepte und Lösungen zur Erstellung eines auf Selbstdarstellung ausgerichteten Identitätsmanagement-Systems vorgestellt werden. Dazu wird das System verwendet, welches im Rahmen des Projektes „onefC“ am Fachbereich Informatik der Universität Hamburg entstanden ist. Es unterstützt persönliches Identitätsmanagement und Identitätsdatenkommunikation mit dem Ziel der Selbstdarstellung weitgehend generisch und schließt damit die in Teil II gefundene Lücke der selbstdarstellungsorientierten Identitätsmanagement-Systeme. Kapitel 8 umfasst die technische Konzeption und Konstruktion der prototypischen onefC-Infrastruktur. In Kapitel 9 wird der praktische Einsatz des Systems an Beispielen erläutert. Dadurch wird die technische Umsetzung von auf onefC aufsetzenden Anwendungen demonstriert und mit bisherigen Ansätzen verglichen.

8. Technische Konzeption und Konstruktion des dezentralen Identitätsmanagement-Systems „onefC“

In diesem Kapitel soll die Konzeption und Konstruktion der Identitätsmanagement-Infrastruktur „onefC“ (open network environment for Citizens[1]) hergeleitet werden. Das Konzept konzentriert sich auf Kommunikation und Selbstdarstellung, ohne den Datenschutz und Schutz der Privatsphäre zu vernachlässigen. Dadurch wird ein persönliches Identitätsmanagement möglich, das den Nutzer dabei unterstützt, seine digitalen Identitäten sinnvoll einzusetzen.

Die Konzentration auf Selbstdarstellung ist dabei ein Alleinstellungsmerkmal, denn bisherige Systeme haben sich – wie in Teil II der Arbeit gezeigt — entweder auf Identitätsmanagement in Organisationen oder auf persönliches Identitätsmanagement mit dem Schwerpunkt des Datenschutzes ausgerichtet. Der Weg dahin führt auf die Ebene der Systemtechnik: Durch die Integration der Identitätsdatenkommunikation in die generelle Kommunikationsschicht soll die Infrastruktur ihre Funktionalität beliebigen Anwendungen zur Verfügung stellen, die mit digitalen Abbildungen von Personen arbeiten. Durch die Verwendung einer einheitlichen Identitätsmanagement-Infrastruktur in möglichst vielen Anwendungen soll allen Anwendern der Zugang zu diesen Systemen mit wiedererkennbaren Teil-Identitäten möglich gemacht werden (siehe dazu Kapitel 5). Dadurch können diese Anwender untereinander applikations- und systemunabhängige Beziehungen aufbauen. Auf diese Weise soll eine tatsächliche Internet-Gemeinschaft mit Individuen entstehen, die sich in verschiedenen Netz- und Anwendungskontexten wiedererkennen können (siehe Baier u. a. 2003).

Es sei nochmals erwähnt, dass ein solches System keinen Selbstzweck verfolgen, sondern immer nur ein Baustein in einer größeren Architektur sein kann. Selbstdarstellung an sich hat immer nur das Ziel, eine weitere Kommunikation anzureichern, insofern muss man ein auf Selbstdarstellung ausgerichtetes Iden-

[1] Der Begriff „Citizen“ wurde in Anlehnung an die Verwendung des Begriffs im Roman „Otherland“ (Williams 1998) gewählt, wo er für menschengesteuerte Avatare im Gegensatz zu künstlichen Intelligenzen steht.

titätsmanagement-System in diesem Zusammenhang als eine Komponente eines Kommunikationssystems sehen, die lediglich unterstützende Wirkung hat.

Das nachgelagerte Problem des Datenschutzes wird keineswegs vernachlässigt. Ein symbolischer Vergleich mit der Flugtechnik bietet sich an: Ohne vernünftige Flugsicherheit würden höchstens Testflugzeuge den sicheren Boden verlassen, niemals jedoch Passagiermaschinen. Ohne die allgemeine Flugzeugtechnik würde jedoch Flugsicherheit gar keinen Sinn machen – sie ist ein nachgelagertes aber notwendiges Thema. Genau so verhält es sich mit dem Datenschutz im Identitätsmanagement: Zum realen Betrieb einer Identitätsmanagement-Infrastruktur ist guter Datenschutz unabdingbar. Ohne Identitätsdatenkommunikation braucht man jedoch auch keinen Datenschutz.

Das onefC-System liegt als lauffähiger Prototyp vor und kann auf der Web-Seite `http://www.onefc.org` heruntergeladen und ausprobiert werden. Es umfasst eine Java-API, durch welche die notwendigen Protokolle genutzt werden können sowie eine als Browser-Plugin entwickelte grafische Oberfläche für die Verwaltung der Identitäten. Eine Hilfskomponente zum Überprüfen von Datenschutzrichtlinien ist enthalten und wird vom Browser-Plugin benutzt.

8.1. Konzeption

Das im Rahmen dieser Arbeit entwickelte Identitätsmanagement-System onefC soll konzeptionell und prototypisch den Bereich des persönlichen, selbstdarstellungsorientierten Identitätsmanagements abdecken, von dem in Teil II herausgefunden wurde, dass er bisher vernachlässigt wird. Es überwiegen die Systeme des Identitätsmanagements in Organisationen und des datenschutzorientierten Identitätsmanagements. Um die Ausrichtung auf persönliche Selbstdarstellung zu erreichen, müssen bestimmte systemarchitektonische Entscheidungen gefällt werden. Diese sollen in diesem Abschnitt erläutert werden. Dazu wird auf Kapitel 4 referenziert, in dem die generellen Basis-Elemente von Identitätsmanagement-Systemen aufgeführt sind. Diese Basis-Elemente werden hier konkretisiert und für den vorliegenden Fall ausgewählt.

Zunächst ist zu beachten, dass das Verwalten der Identitäten auf Anwendungsebene nicht Teil der Identitätsmanagement-Infrastruktur ist. Die Infrastruktur spezifiziert lediglich die Datenkodierung sowie die Protokolle und Mechanismen zum Austausch der Daten. Dadurch wird es möglich, verschiedene Identitätsmanagement-Anwendungen zu entwickeln, die unterschiedlichen Ansprüchen genügen. Beispielsweise sind auf mobilen Geräten andere Anforderungen zu erfüllen als auf Desktop-PCs oder Set-Top-Boxen für Fernsehgeräte. Ebenso kann Identitätsmanagement-Funktionalität in andere Anwendungen eingebunden werden und dadurch dem Anwender noch näher gebracht werden – er braucht in diesem Fall kein weiteres Programm zu benutzen. Durch diese Designentscheidung ist

es also möglich, dass das System viele Anwendungsbereiche und Systeme durchdringt und dadurch dem Anwender an vielen Stellen Nutzen bringt, weil er überall die gleichen Teil-Identitäten einsetzen kann.

Wichtiger Aspekt beim Entwurf von onefC ist der Einsatz eines *Sitzungskonzepts*. Wie in Abschnitt 2.4 beschrieben fasst eine Sitzung sowohl die daran teilnehmenden digitalen Identitäten (die auch pseudonym sein können) als auch die darin stattfindende Kommunikation zusammen. Um dieses generische Konzept für eine Identitätsinfrastruktur nutzbar zu machen, wurde zwischen Anwendungsebene und Kommunikationsschicht eine Sitzungsschicht eingefügt. Jegliche identitätsbezogene Kommunikation kann nur innerhalb einer Sitzung vorgenommen werden. Die Sitzungen sind unabhängig von der Verbindungsschicht und von der Anwendungsschicht. Dadurch können sie sowohl in verschiedenen Netzen als auch von verschiedenen Anwendungen genutzt werden. Durch dieses flexible Design lassen sich komplexe Szenarien aus mehreren Anwendungen und Kommunikationsnetzen entwerfen, die trotzdem mit konsistenten Identitätsdaten arbeiten können.

8.1.1. Aufteilung des Systems in Komponenten

In onefC werden die statischen, nicht-kommunikativen Aufgaben des Identitätsmanagements von einer Identitätsmanagement-Komponente gestützt, während die dynamischen, kommunikativen Aufgaben von einer Sitzungsmanagement-Komponente unterstützt werden. Der Identitätsmanager übernimmt die Verwaltung der eigenen und fremden Teil-Identitäten sowie die Einbindung der verschiedenen Hilfskomponenten des Systems. Die Teil-Identitäten müssen dazu in eine geeignete Abbildung gebracht werden, die sowohl die lokale Speicherung als auch die Übertragung zu Kommunikationspartnern ermöglicht. Definition 2.1 legt einen eindeutigen Bezeichner fest, dem beschreibende Attribute anhängen können. In onefC werden die Eigenschaften einer Teil-Identität als Graph von zusammenhängenden, durch Verweise auf Teil-Ontologien beschriebenen Attributen zu eindeutigen Bezeichnern assoziiert. So entsteht ein Baum mit dem eindeutigen Identifikator als Wurzel. Querverweise auf Teile dieses Baums zwischen verschiedenen Teil-Identitäten (also verschiedenen eindeutigen Bezeichnern) sind nicht vorgesehen. Dadurch besteht die Gefahr der Inkonsistenz durch Redundanz (zum Beispiel können Adressdaten mehrfach vorliegen), jedoch wird das Problem der Verwaltung der Zugriffsrechte vermieden und dem Schutz der Privatsphäre durch Vermeiden von unerwünschtem Offenbaren des Zusammenhangs zweier Teil-Identitäten Rechnung getragen (Teil-Identitäten mit zum Beispiel gleichen Adressdaten sind nicht unbedingt auf dieselbe Person zurückführbar). Die Beschreibung der Attributtypen durch Ontologien stellt einen semantischen Zusammenhang oder eine gemeinsame semantische Basis der Attribute dar. Verweisen zwei Attribute auf die selbe Ontologie, so haben sie die selbe Bedeutung (wenn auch vielleicht eine andere Ausprägung). Zum Beispiel könnten zwei Attribute auf eine Ontologie

verweisen, die Namen von Personen beschreibt. Beide Attribute stehen also für Namen, auch wenn die einzelnen Namen unterschiedlich sein können.

Im Sitzungsmanager wird das Konzept der Sitzung entsprechend den Anforderungen umgesetzt, die in den Abschnitten 1.3.5 und vor allem 2.4 vorgestellt worden sind. Jegliche identitätsdatenbereicherte Kommunikation ist sitzungsassoziiert. Durch die Sitzung wird ein Kontext aufgebaut, innerhalb dessen (inhaltliche oder identitätsorientierte) Aussagen einzelnen Personen zugeordnet werden können. Zusätzlich ermöglicht der Aufbau einer Sitzung für jeden Kommunikationsvorgang die automatische Pseudonymisierung. Dazu wird beim Sitzungsaufbau dynamisch ein Pseudonym erstellt, das im späteren Verlauf einer wiederverwendbaren, wieder*erkennbaren* Teil-Identität zugeordnet werden kann.

Als Hilfskomponenten sind in onefC ein Datenschutzdienst sowie ein Ontologie- und Transformationsdienst vorgesehen. Mit Hilfe des Datenschutzdienstes werden Datenschutzbestimmungen von Kommunikationspartnern auf Kompatibilität mit den eigenen Anforderungen an die Privatsphäre überprüft. Dies hilft bei der Entscheidung, ob bestimmte Identitätsdaten in einer Sitzung preisgegeben werden oder nicht. Dafür bestehen drei Voraussetzungen: Erstens muss den Attributdefinitionen in der Teil-Ontologie eine Kategorie zugeordnet sein, nach der das Attribut eingeordnet wird. Zweitens muss mit dem Kommunikationspartner eine Datenschutzrichtlinie vereinbart worden sein. Drittens muss den Identitätsdaten selbst eine Regel anhaften, wie diese Datenschutzrichtlinie beschaffen sein muss, damit einer Übertragung zugestimmt wird.

Der Ontologie- und Transformationsdienst verbindet die statische Aufgabe der Verwaltung von Teil-Ontologien mit der dynamischen Aufgabe der Übersetzung von Daten aus einer Teil-Ontologie in eine andere. Diese Funktionalitäten sind in einer Komponente zusammengefasst, da beide sowohl mit den Teil-Ontologien arbeiten als auch einen vom Identitätsmanagement-System abgekoppelten Zugriff auf das Internet Zwecks Suche nach Teil-Ontologien oder Übersetzungen haben sollten. Die Schnittstelle zur Komponente stellt Methoden zum Suchen nach Teil-Ontologien und Übersetzungen bereit. Bei der Suche nach Übersetzungen wird die gewünschte Ziel-Ontologie angegeben. Als Antwort wird eine Liste der möglichen Quell-Ontologien sowie die eigentliche Übersetzung geboten. Ist das gesuchte Objekt nicht in einem lokalen Cache enthalten, wird die Suchanfrage ins Internet propagiert.

Die Zusammenarbeit der Komponenten sowie der grobe Verbindungsaufbau und die Kommunikation zwischen zwei Kommunikationspartnern wird in Abbildung 8.1 beschrieben. In dieser Abbildung sind die Komponenten nur auf der einen Seite der Kommunikation gezeigt, die hier als *Client* bezeichnet ist. In einer dezentralen Peer-to-Peer Umgebung spielt die Unterscheidung zwischen Client und Server natürlich keine Rolle – hier soll lediglich verdeutlicht werden, dass die Implementierung und der Aufbau des Identitätsmanagement-Systems auf den unterschiedlichen Seiten der Kommunikation nicht unbedingt gleich sein muss. Die

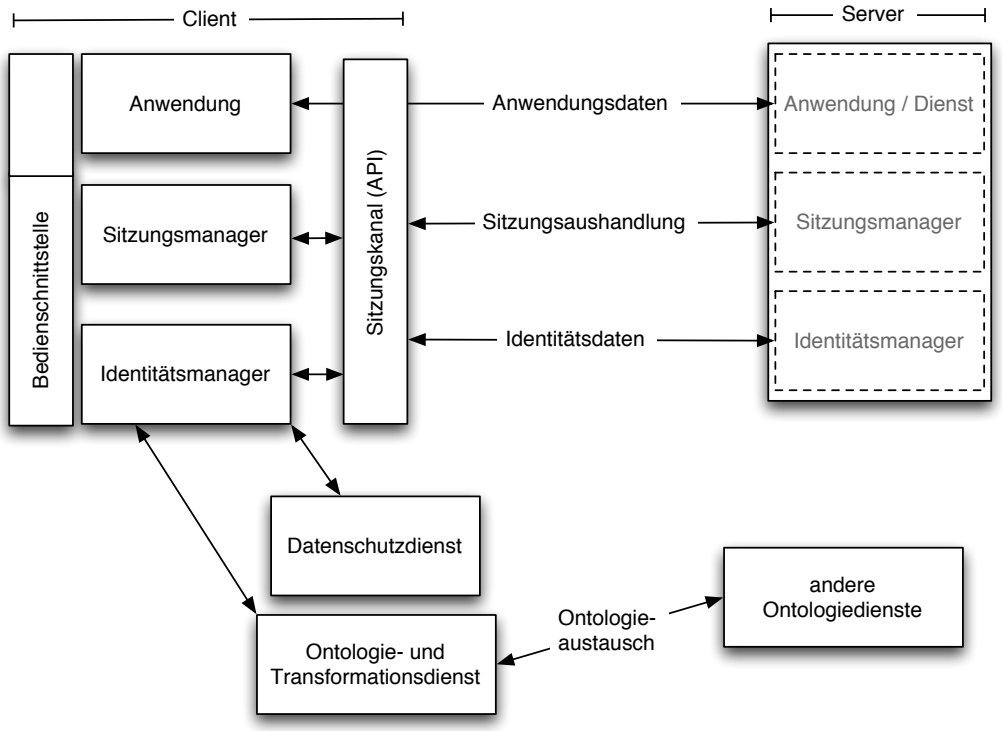

Abbildung 8.1.: Grobe Übersicht über die onefC-Komponenten mit Darstellung der grundlegenden Kommunikationsflüsse

Systeme müssen nur mit den gleichen Protokollen (siehe dazu Abschnitt 8.3) umgehen können. Ein Identitätsmanagement-System eines großen Dienstanbieters, der eventuell mehrere tausend Anfragen und Aussagen pro Sekunde bearbeiten muss, wird zum Beispiel eine deutlich weniger interaktive Bedienschnittstelle haben als das eines privaten Anwenders, der auf jede Situation gezielt reagieren möchte oder muss.

Zu beachten ist, dass die Bedienschnittstelle des Sitzungs- und des Identitätsmanagers ausdrücklich in diejenige der Anwendung integriert sein kann oder auch nicht. Dadurch wird ermöglicht, dass *identitäts-bewusste* Anwendungen (also solche, denen die Assoziation ihres Applikationsprotokolls zu einer mit Identitätsdaten behafteten Sitzung bekannt ist) die Verwaltung der ihnen zugehörigen Sitzungen samt der darin verwendeten Teil-Identitäten selbst verwalten können. Ebenso müssen die Anwendungsdaten nicht unbedingt durch den Sitzungskanal geleitet werden – es reicht aus, wenn sie zu diesem assoziiert sind. Das kann auf verschiedene Weisen geschehen, wie in Abschnitt 9.2 für das Applikationsprotokoll HTTP näher betrachtet wird.

Ein weiterer wichtiger Aspekt, der auf Abbildung 8.1 erkennbar ist, ist die mögliche Verteilung der Komponenten „Datenschutzdienst“ und „Ontologie- und Transformationsdienst“. Diese können auf dem Rechner des Clients laufen, müssen es aber nicht – aus Performanz-, Sicherheits- oder weiteren Gründen können diese Dienste ausgelagert und auf entfernten Systemen genutzt werden. Dies spielt zum Beispiel im Falle eines Identitätsmanagement-Systems auf einem mobilen Gerät mit geringer Rechenleistung und Speicherkapazität (beispielsweise einem Mobiltelefon) eine Rolle.

8.1.2. Verteilung des Systems

Um dem Benutzer größtmögliche Kontrolle zu ermöglichen und damit der zentralen Forderung eines *persönlichen* Identitätsmanagement-Systems nachzukommen, sieht die onefC-Architektur eine vollständige *Dezentralität* vor. Es ist kein zentraler Server oder Dienst zur Verwaltung der Identitätsdaten oder Ontologien vorgesehen, sondern die Anwender sollen ihre Daten selbst unter Kontrolle haben. Dies erschwert den Einsatz als Identifikations- oder Authentifikationsdienst, ermöglicht jedoch flexible Teil-Identitätsentwicklung und die schnelle Integration neuer Dienste und Anwendungen. Durch die Dezentralität wird auch das Problem der Semantik von Identitätsaussagen erschwert, denn mit einer zentralen Instanz, die festlegt, welche Daten zu Teil-Identitäten gehören können, könnte die Bedeutung dieser Daten mit festgelegt werden. Dafür kann in der dezentralen Variante jeder Nutzer selbst entscheiden, welche Attribute er verwenden möchte – inklusive der Option auf eigene Definitionen. Dieser Mechanismus macht die Ausrichtung des Systems auf Selbstdarstellung deutlich. Während andere Systeme eine fest vorgegebenen Satz an Identitätsattributen bieten, können die Anwender bei onefC alle digitalisierbaren Eigenschaften ihrer Teil-Identitäten einsetzen.

Die Dezentralität ermöglicht verschiedene Szenarien: Nicht immer muss der Identitätsmanager und die Speicherung der Daten lokal erfolgen. Es kann ebenso einen *Identitäts-Provider* geben, der diese Aufgaben übernimmt. In diesem Fall gibt der Anwender einen Teil der Kontrolle ab. Dies kann aus Gründen der *Sicherheit*, der *Konsistenz* oder der *Performanz* gewünscht sein. Identitäts-Provider haben andere Möglichkeiten als Privatanwender, Daten zu schützen. Dies betrifft sowohl den Schutz vor Zugriffen über das Netz durch Firewalls, Virenscanner und andere technische Mittel, sowie den physischen Schutz der Speichermedien und Server-Systeme selbst. Desweiteren kann die Wahl eines bekannten Identitäts-Providers das Vertrauen der Kommunikationspartner stärken.

Anwender, die ihre Identitätsdaten von verschiedenen Systemen aus nutzen wollen, ohne sie auf tragbaren Medien (beispielsweise Smartcards) tatsächlich bei sich haben zu wollen, können durch den entfernten Zugriff auf ihre Daten bei einem Identitäts-Provider größere Mobilität erreichen, ohne auf den verschiedenen Rechnern möglicherweise inkonsistente Daten zu halten. Ebenso gilt für viele mo-

bile Systeme, dass sowohl Rechenleistung als auch Speicherkapazität nicht ausreichen, um aufwändigere Teile des Identitätsmanagements selbst zu erledigen. Dazu kann das Überprüfen von Signaturen oder das Verschlüsseln der Kommunikation gehören. Durch das Auslagern dieser Funktionalität auf einen entfernten Rechner muss das mobile Gerät nur noch die Bedienschnittstelle bereitstellen. Auch hier gibt es Einschränkungen, da mobile Geräte häufig geringere Darstellungsmöglichkeiten und eingeschränkte Eingabegeräte haben – dieses Problem lässt sich jedoch durch Auslagerung nicht lösen.

Während der Identitätsmanager also entfernt genutzt werden kann, trifft dies für das Sitzungskonzept nur sehr bedingt zu. Da es sich bei den onefC-Sitzungen um eine Erweiterung des Netzwerkprotokollstapels handelt (siehe dazu Unterabschnitt 8.4.1), sollte diese Ebene unbedingt auch auf dem Rechner laufen, auf dem die Anwendung läuft. Da der Sitzungskanal sich als API präsentiert, ist dies in der Regel auch gar nicht anders möglich. Eine Ausnahme bilden entfernt laufende Proxys, wie sie in Abschnitt 9.2 beschrieben sind. Hier wird die Sitzung aber auch nur zwischen Proxy und dem Kommunikationspartner aufgebaut, nicht zwischen dem eigentlichen Anwender und der Gegenstelle. In diesem Spezialfall ist ein entfernter Sitzungskanal mit lokaler Bedienschnittstelle (Sitzungsmanager) denkbar.

In onefC ist eine Komponente für einen Ontologie- und Transformationsdienst sowie eine weitere für einen Datenschutzdienst vorgesehen. Da diese beiden Dienste nicht unbedingt nur für einen Anwender laufen müssen sondern geteilt werden können, sind diese Komponenten als *verteilte Dienste* modelliert, die von unterschiedlichen Benutzern über entfernten Aufruf genutzt werden können. Während sie auf einem leistungsfähigen System lokal installiert und genutzt werden können, entsteht so jedoch die Möglichkeit, sie von weniger leistungsfähigen Endgeräten (beispielsweise Mobiltelefonen oder PDAs) auszulagern.

8.2. Technische Aspekte von Identitäten, Attributen und Ontologien

Digitale Identitäten sind „eindeutige, digitale Identifikatoren, denen personifizierende Attribute zugeordnet sein können“ (siehe Definition 2.1). Die Attribute sind wiederum jeweils durch eine Teil-Ontologie beschrieben und können strukturiert sein (vergleiche Unterabschnitt 2.2.3). Für die strukturierte Darstellung von Daten, denen weitere Eigenschaften zugeordnet werden sollen, bietet sich XML an (siehe Mintert 2002), ein Standard des World Wide Web Konsortiums W3C[2]. XML bietet die Möglichkeit, durch Schemata Datendefinitionen festzulegen, nach denen die eigentlichen Daten validiert werden können. XML hat sich durch Offenheit, Klarheit der Datentypen, Erweiterbarkeit und auch der leichten Lesbarkeit

[2] `http://www.w3.org`

durch sowohl Menschen als auch Computersysteme zu einem Standard für Datenrepräsentation in vielen Anwendungen entwickelt. Bekanntestes Beispiel dafür ist die Definition von XHTML (Altheim u. a. 2001), einer auf XML festgelegten Spezialisierung der WWW-Hypertext-Sprache HTML (spezifiziert in Raggett u. a. 1998).

Allerdings ist mit XML nur syntaktische Korrektheit überprüfbar – für semantische Absicherung bedarf es weiterer Mechanismen. Ebenfalls vom W3C ist zu diesem Zweck das Resource Description Framework RDF (Manola und Miller 2004; Swick 1999) entwickelt worden. Grundlage von RDF ist das Treffen von Aussagen über Ressourcen. Eine Ressource ist dabei ein beliebiges Objekt, welches durch einen *Unified Resource Identificator* (URI) bezeichnet wird. Eine Aussage enthält immer ein Subjekt, ein Prädikat und ein Objekt. Subjekt und Objekt sind Ressourcen, Prädikate sind logische Beziehungen[3]. RDF-Daten müssen nicht unbedingt in XML vorliegen, *RDF/XML* ist aber eine gebräuchliche und für Computersysteme leicht zu verarbeitende Form. In Dokumentationen dagegen wird RDF häufig als grafische Baum- oder Netzstruktur dargestellt, bei denen Subjekte und Objekte durch Ovale repräsentiert werden, Prädikate durch Kanten, die diese Ovale sinnhaft verbinden. Abbildung 8.2 zeigt auf diese Weise ein Beispiel aus der onefC-Kern-Ontologie. Während die interne Repräsentation des RDF den Systemen selbst überlassen ist, bietet sich RDF/XML für die Verwendung in Protokollen zwischen den Kommunikationspartnern und auch zum Speichern der Daten an.

Werden Daten in RDF repräsentiert, so ist mit *RDF Schema* (RDFS, siehe Nejdl u. a. 2000) oder der *Web Ontology Language* (OWL, siehe Bechhofer u. a. 2004) ein nah verwandtes System zum Definieren der Ontologien vorhanden. Beides baut auf RDF auf und hat somit klare XML-Repräsentationen. Dabei ist OWL eine Erweiterung von RDFS um weitere Fähigkeiten der Ontologiebeschreibung, welche sie von ihren Vorgängern DAML (Berners-Lee u. a. 2000) und OIL (Fensel u. a. 1999) geerbt hat. Zum Aufspannen einer Ontologie werden dabei Klassen genutzt, die untereinander in verschiedenen Beziehungen stehen können (siehe auch Kaulbarsch 2005).

Die Gesamtontologie eines Benutzers setzt sich aus seinen Teil-Ontologien zusammen. Dabei basieren alle Teil-Ontologien auf einer Kern-Ontologie (ein Ausschnitt daraus ist als RDF-Baum in Abbildung 8.2 dargestellt, die vollständige Kern-Ontologie als RDF/XML findet sich in Listing A.1), welche zum System gehört und die Basis-Klassen für Identitäten, Pseudonyme (hier: *SessionIdentities*) und ihre Attribute in verschiedenen Ausprägungen zur Verfügung stellt.

In einer Teil-Ontologie, die Attributklassen für den Anwender bereitstellt, leiten sich alle Elemente von den Klassen *UserPropertyClass* und den davon er-

[3] Eigentlich sind Prädikate in RDF ebenfalls Ressourcen; sie werden jedoch genutzt, um Subjekt und Objekt in eine Beziehung zu setzen.

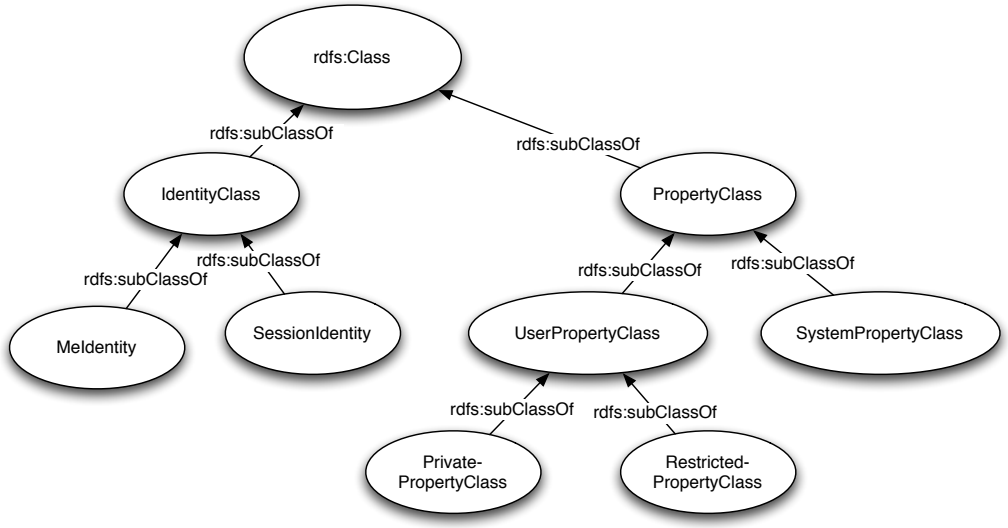

Abbildung 8.2.: Ausschnitt aus der Kern-Ontologie von onefC: Identitäten und Eigenschaften

benden *PublicPropertyClass*, *RestrictedPropertyClass* und *PrivatePropertyClass* ab. Dabei haben alle Attributklassen, die von RestrictedPropertyClass abgeleitet werden, eine Beschreibung nach P3P, zu welchen Kategorien entsprechende Daten gehören. Zum Beispiel wird „physical“ für physische Kontaktdatentypen oder „financial“ für Datentypen aus dem Bereich des Finanzwesens genutzt; eine vollständige Auflistung der Kategorien findet sich in der P3P-Spezifikation (Cranor u. a. 2002) und in Cranor (2002). Die Daten selbst haben jeweils eine Restriktion, die in der Sprache APPEL verfasst ist. Daten einer PrivatePropertyClass werden auf keinen Fall herausgegeben (hierzu gehören zum Beispiel *private keys* einer PKI), Daten einer PublicPropertyClass haben keine Restriktionen (zum Beispiel eine Datenschutzrichtlinie, die von jedermann einsehbar sein soll).

Es ist zu erwarten, dass sich für bestimmte Gebiete *de-facto* Standard-Teil-Ontologien entwickeln, die von fast allen Anwendern eingesetzt werden. So existieren schon jetzt Abbildungen des vCard Standards für Namens- und Kontaktinformationen (Iannella 2001) und des P3P-Standards für Datenschutzrichtlinien (McBride u. a. 2002) auf RDF/XML, wodurch die Einbindung dieser Formate in die onefC-Infrastruktur sehr einfach wird. Aber auch für neue Anwendungen ist es nicht zu erwarten, dass sich viele konkurrierende Teil-Ontologien mit den selben Themen befassen, da die meisten Anwender (und auch Anbieter) den Aufwand des Erstellens einer eigenen Ontologie und die dadurch erzeugte Gefahr der Inkompatibilität untereinander und daraus folgender Inkonsistenz der Daten eher scheuen werden.

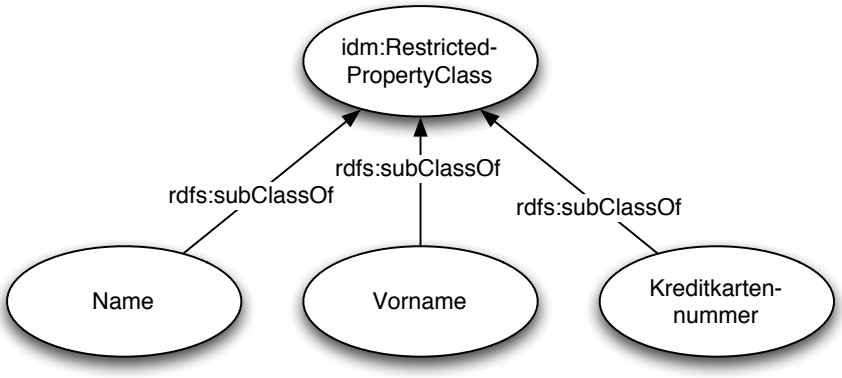

Abbildung 8.3.: Einfaches Beispiel einer Teil-Ontologie für Namen und Kreditkartennummern

Für das onefC-System wurde eine kleine Teil-Ontologie erstellt, die Namen und Kreditkarten-Informationen umfasst (siehe Abbildung 8.3 und Listing A.2). Anhand dieser Teil-Ontologie kann nun ein Beispiel von Identitätsdaten betrachtet werden, die in RDF vorliegen (siehe Abbildung 8.4). Sie beinhaltet je ein Datum für die Elemente der Teil-Ontologie. Der Wert des Datums ist zur einfacheren Lesbarkeit der Grafik in Klammern angegeben.

Ein anderes Beispiel wird in Listing 8.1 in RDF/XML-Format angegeben, damit das Zusammenspiel der Kern-Ontologie mit der Teil-Ontologie deutlicher wird[4]. Die APPEL-Regel der Kreditkartennummer ist hier zur besseren Lesbarkeit ausgelassen, weiterhin wurden die URIs aller Elemente gekürzt.

Listing 8.1: Darstellung einer onefC-Teil-Identität mit RDF/XML

```
<?xml version="1.0"?>
<RDF:RDF xmlns:IDM="http://www.jwsdot.com/rdf/ISE/IDM#"
         xmlns:IDMNames="http://www.jwsdot.com/rdf/ISE/IDMNames#"
         xmlns:RDFS="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:RDF="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:appel="http://www.w3.org/2002/04/APPELv1">

  <IDM:MeIdentity RDF:about="onefc:Bob-Brown-97781f15">
    <IDM:property RDF:resource="onefc:Bob-Brown-97781f15#Name"/>
    <IDM:property RDF:resource="onefc:Bob-Brown-97781f15#Nick"/>
    <IDM:property RDF:resource="onefc:Bob-Brown-97781f15#CreditCard"/>
  </IDM:MeIdentity>
  <IDMNames:CreditCard RDF:about="onefc:Bob-Brown-97781f15#CreditCard">
    <RDFS:label>5323-1234-9876-1234 (333)</RDFS:label>
 <IDM:Restriction
    RDF:resource="onefc:Bob-Brown-97781f15#CreditCardRestriction"/>
  </IDMNames:CreditCard>
    <IDM:Restriction
```

[4]Das Listing für dieses Beispiel wurde mit dem „onefC“-Prototypen erstellt.

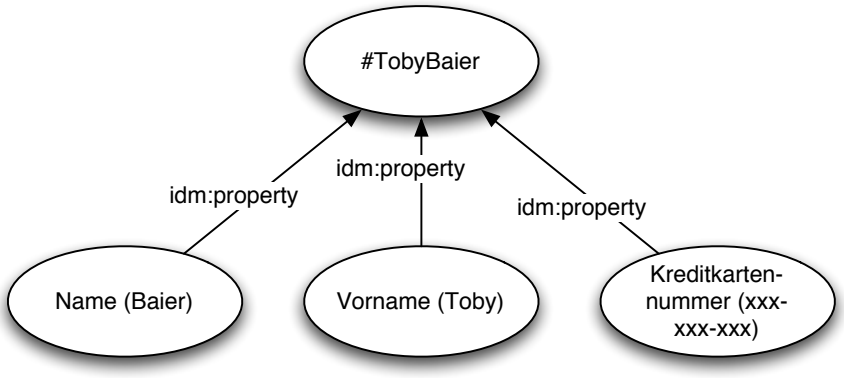

Abbildung 8.4.: Teil-Identität, abgeleitet aus der Teil-Ontologie aus Abbildung 8.3

```
      RDF:about="onefc:Bob-Brown-97781f15#CreditCardRestriction"
      RDFS:label="(APPEL␣Rule)" />
  <IDM:Nick RDF:about="onefc:Bob-Brown-97781f15#Nick"
    RDFS:label="Bob␣the␣Browny" />
  <IDMNames:Name RDF:about="onefc:Bob-Brown-97781f15#Name"
    RDFS:label="Brown" />
</RDF:RDF>
```

8.3. Protokolle

Neben den grundlegenden Komponenten eines Identitätsmanagement-Systems ist die Spezifikation der zu verwendenden *Protokolle* in einem dezentralen, verteilten System die wichtigste Aufgabe, um Interoperabilität verschiedener Implementierungen (beispielsweise für verschiedene Anwendungs- oder Betriebssysteme) zu gewährleisten. Während also die API des Sitzungskanals für verschiedene Systeme oder Sprachen syntaktisch unterschiedlich sein kann (soweit sie das selbe leistet), so muss das Protokoll zwischen zwei kommunizierenden Instanzen der gleichen Ebene im Netzwerkprotokollstapel unbedingt kompatibel sein.

Im hier beschriebenen Identitätsmanagement-System gibt es mehrere Protokolle, die definiert werden müssen. Dazu gehören

1. das Protokoll zur Sitzungsaushandlung,
2. das Protokoll zum Identitätsdatenaustausch (Datenanfragen und -äußerungen),
3. das Protokoll zum Ontologie- und Transformationsdienst sowie
4. das Protokoll zum Datenschutzdienst.

Dabei ist das erste Protokoll (Sitzungsaushandlung) auf technischer Ebene angesiedelt. Bei allen anderen Protokollen spielen bereits Identitätsdaten oder ihre Definition in Ontologien eine Rolle; daher ist hier eine große Nähe zum Identitätsmodell angebracht.

8.3.1. Das Sitzungsprotokoll

Eine Sitzung fasst – wie in Definition 2.2 beschrieben – die Sitzungsteilnehmer sowie die Aussagen, die innerhalb der Sitzung getroffen werden, zusammen. Auf technischer Ebene bedeutet dies, dass ein Kommunikationskanal geöffnet werden muss, der eindeutig identifiziert werden kann, damit er auch in Anwendungsprotokollen, die nicht über den Sitzungskanal laufen, referenziert werden kann (siehe dazu auch Sack 2005).

Das *Protokoll zum Sitzungsaufbau* ist an den *Transport Layer Security* Handshake (TLS, siehe Dierks und Allen 1999) angelehnt. Dabei wurde die *anonyme* Variante des TLS genutzt, da die Identitäten der Teilnehmer für den Sitzungsaufbau noch keine Rolle spielen und später per Identitätsprotokoll festgestellt werden sollen. Der Server überträgt nach Aufforderung des Clients im *Server hello* ein selbstsigniertes Zertifikat, aus dem der Client den öffentlichen Schlüssel des Servers extrahieren kann. Damit wird ein selbst erstelltes *master secret* verschlüsselt und an den Server zurückgeschickt. Anhand des *master secret* und der Zufallszahlen, die jeweils im *hello* integriert waren, können nun die Schlüssel zur Verschlüsselung der weiteren Kommunikation erzeugt werden. Anders als beim TLS werden jedoch über diese Verbindung keine Anwendungsdaten übertragen – hier werden lediglich weitere Nachrichten des Sitzungsprotokolls sowie die Nachrichten des Identitätsdatenprotokolls versandt.

Abbildung 8.5 zeigt das Protokoll in grafischer Darstellung. Die Werte „client nickname“ (das Sitzungs-Pseudonym des Clients) und „client random“ werden vor Beginn beim Client generiert. Entsprechend werden die „server“ Werte nach Empfangen eines „Client hello“ beim Server generiert.

Der wichtigste Aspekt des weiteren Sitzungsprotokolls ist die Aushandlung von Pseudonymen (oder *session nicknames*) der Kommunikationspartner. Diese Pseudonyme gelten nur für diese eine Sitzung und sind deshalb nicht mit digitalen Identitäten gleichzusetzen. Anders als diese sind Sitzungs-Pseudonyme nicht wiederverwendbar. Genau wie digitale Identitäten können sie jedoch Attribute tragen. Das aussagekräftigste Attribut ist dabei jenes, das einem Sitzungs-Pseudonym eine digitale Identität zuweist. Dadurch können Daten, die einem Pseudonym zugesprochen worden sind, in eine digitale Identität übernommen und in einer anderen Sitzung wiederverwendet werden.

Desweiteren umfasst das Sitzungsprotokoll die Aushandlung von *Transportkanälen*, über die Applikationsprotokolle genutzt werden können. Transportkanäle erhalten dabei automatisch die Sitzungskennung der Sitzung, über die sie verhan-

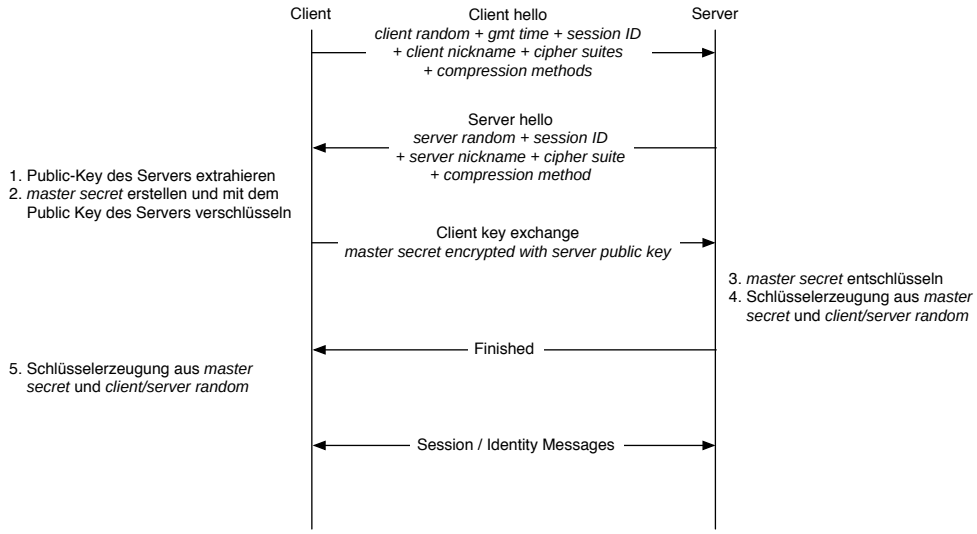

Abbildung 8.5.: Protokoll zur Sitzungsaushandlung, angelehnt an den TLS-Handshake

delt worden sind, als Attribut. Dadurch sind sie auch für die Anwendung als zu dieser Sitzung gehörend erkennbar. Auf diese Weise können Applikationen, die solche Transportkanäle verwenden, automatisch und transparent sitzungsassoziierte und damit identitätsassoziierte Kommunikation betreiben.

Es bleibt zu beachten, dass die Rollen bei der Sitzungsaushandlung nicht mit den Rollen in einem Applikationsprotokoll zusammenhängen. Die Bezeichnungen *client* und *server* im Protokoll stammen aus dem TLS-Protokoll und könnten hier in *initiator* und *acceptor* umbenannt werden. Die Rollen der auf eine Sitzung aufbauenden Anwendungsprotokolle können davon ganz verschieden sein – insbesondere können auch Peer-to-Peer Anwendungen auf diese Sitzungsinfrastruktur aufbauen.

8.3.2. Das Identitätsdatenprotokoll

Das *Identitätsdatenprotokoll* dient zum Austausch von personifizierenden Daten, die digitalen Identitäten anhaften. Es ist asynchron – einen genauen Ablauf wie beim Sitzungsaufbau gibt es nicht. Dies ergibt sich schon aus der Natur der Nachrichten, die oft von Menschen einzeln bearbeitet werden und dadurch eine große Laufzeit haben. Die Nachrichten des Protokolls sind:

1. Anfrage nach einem Identitätsattribut

2. Äußerung eines Identitätsattributs

3. Anfrage nach Identifizierung des Sitzungspseudonyms mit einer digitalen Identität

4. Identifizierung des Sitzungspseudonyms mit einer digitalen Identität

Dabei sind typische Kombinationen (erst Anfrage 1., dann Antwort 2.) denkbar, aber nicht zwingend vorgegeben. Anfragen müssen nicht beantwortet werden, und Äußerungen sowie Identifizierungen können auch ohne Anfrage gesendet werden. Wie bereits erwähnt ist die Identifizierung des Sitzungspseudonyms mit einer digitalen Identität eigentlich auch nur die Äußerung eines Identitätsattributs für dieses Sitzungspseudonym. Da es eine sehr zentrale Rolle spielt, hat dieser Vorgang jedoch einen eigenen Protokollnachrichtentyp bekommen.

Das Identitätsdatenprotokoll wird zwischen den Anwendungen (sowohl dem Sitzungs- und Identitätsmanager als auch den eigentlichen, identitätsangereicherten Anwendungen) und dem Sitzungskanal angewendet. Ist der Empfänger einer Nachricht ein entfernter Sitzungsteilnehmer, so leitet der Sitzungskanal die Nachricht dorthin weiter. Anwendungen können sich beim Sitzungskanal anmelden. Dabei gibt es einen *Master-Adapter*, an den alle Anfragen bezüglich des eigenen Sitzungspseudonyms geleitet werden. Alle anderen Mithörenden bekommen nur die Aussagen mitgeteilt, nicht jedoch die Anfragen.

Anfragen nach und Äußerungen von Identitätsdaten haben viele Eigenschaften gemeinsam. Beide Nachrichtentypen umfassen einen *Sender*, einen *Empfänger* sowie einen *Betreffenden* (*subject*) der Nachricht. Der Betreffende einer Identitätsdatenanfrage muss also nicht zwingend auch der Sender sein; Äußerungen können sowohl über das eigene als auch über ein fremdes Sitzungspseudonym getroffen werden, beispielsweise wenn ein Dienst einem Benutzer eine bestimmte Eigenschaft zuspricht. Der Sitzungskanal kann von einer Anwendung auch nach eigenen Attributen gefragt werden.

Desweiteren enthalten beide Nachrichtentypen einen Verweis auf eine Klasse einer Teil-Ontologie, um zu spezifizieren, was das Attribut bedeutet, das angefragt oder über das etwas ausgesagt wird. Die Attribute, die eine Nachricht einer Sitzung zuweisen und sie überhaupt als Identitätsdatennachricht ausweisen, gehören zu jeder Nachricht des Protokolls, auch den Identifizierungsanfragen und Identifizierungen.

Listing 8.2: Beispiel des Identitätsdaten-Protokolls: Äußerung eines eigenen Attributs

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:j.1="http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#" >
  <rdf:Description rdf:about="onefc://IDMProperty-c32408a0">
    <rdfs:label>Toby</rdfs:label>
  </rdf:Description>
```

```
  <rdf:Description rdf:about="onefc:1fd51a02/statement">
    <j.1:hasSubject rdf:resource="onefc:SESSIONNICK-23412dcb"/>
    <j.1:session rdf:resource="onefc:SESSION-44168d8b"/>
    <j.1:type>PropertyStatement</j.1:type>
    <j.1:hasSrc rdf:resource="onefc:SESSIONNICK-23412dcb"/>
    <j.1:object rdf:resource="onefc://IDMProperty-c32408a0"/>
    <j.1:ptype rdf:resource="http://www.jwsdot.com/rdf/ISE/IDMNames#Name"/>
    <rdf:type rdf:resource=
      "http://www.jwsdot.com/ISE/rdf/IDM/request/IdentityStatement"/>
    <j.1:hasDest rdf:resource="onefc:SESSIONNICK-b9c10804"/>
  </rdf:Description>
</rdf:RDF>
```

Listing 8.2 zeigt ein Beispiel einer Identitätsdaten-Nachricht. Es handelt sich dabei um eine Äußerung eines Namens nach der Teil-Ontologie `IDMNames`. Die URIs der Sitzung, der Sitzungs-Pseudonyme und der Nachricht selbst wurden zur besseren Lesbarkeit abgekürzt.

Bei Identifikationen und Aufforderungen dazu fehlt das Ontologie-Attribut, weil es nicht benötigt wird. Sitzungs-Pseudonyme haben eine Eigenschaft „identifiedBy“, welche eine URI einer digitalen Identität halten kann. Eine Identifikationsanfrage ist eine Identitätsdatenanfrage nach genau diesem Attribut. Anstatt es jedoch als normale Identitätsdaten-Anfrage mit entsprechendem Ontologie-Eintrag zu modellieren, wird ein eigener Nachrichtentyp eingeführt, damit die Wichtigkeit und Aussagekraft dieser Nachricht unterstrichen wird. Ein Beispiel für eine Aufforderung zur Identifikation ist in Listing 8.3 gegeben.

Listing 8.3: Beispiel des Identitätsdaten-Protokolls: Aufforderung zur Identifikation

```
<rdf:RDF
    xmlns:j.0="http://www.jwsdot.com/ISE/rdf/IDM/request/"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:j.1="http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#" >
  <rdf:Description rdf:about="onefc:070d33cd">
  <j.1:type>PleaseIdentify</j.1:type>
  <j.1:session rdf:resource="onefc:SESSION-f1a6c9cb"/>
  <rdf:type
   rdf:resource="http://www.jwsdot.com/ISE/rdf/IDM/request/PleaseIdentify"/>
  <j.1:hasDest rdf:resource="onefc:SESSIONNICK-32cee308"/>
  <j.1:hasSubject rdf:resource="onefc:SESSIONNICK-32cee308"/>
  <j.1:hasSrc rdf:resource="onefc:SESSIONNICK-b7e13132"/>
  </rdf:Description>
</rdf:RDF>
```

8.3.3. Protokoll zur Nutzung des Datenschutzdienstes

Zur Nutzung des Ontologie- und Transformationsdienstes sowie des Datenschutzdienstes sieht die onefC-Infrastruktur eigene Protokolle vor, da diese Dienste möglicherweise auf einem entfernten System genutzt werden. Als mittlerweile weit

verbreitete und erprobte Technologie bietet sich für die entfernten Komponenten die *Web Service* Technologie an (Web Services Architecture 2004). Web Services sind Dienste, für die eine Schnittstellenbeschreibung in WSDL (web service definition language) vorliegt und die über ein Web Service Protokoll erreichbar sind, beispielsweise SOAP (Mitra 2004). Als Transportprotokolle sind dabei Protokolle der Anwendungsebene vorgesehen, beispielsweise HTTP (Fielding u. a. 1999). Die Nutzung von Web Services ist aus vielen Systemen sehr einfach möglich. Dadurch wird diese Technologie sehr interessant zum Verbinden verteilter, heterogener Komponenten. Setzt eine Anwendungs-Architektur vollständig auf Web Services zur Verknüpfung ihrer Komponenten auf, so spricht man vom *service oriented computing* (SOC, siehe zum Beispiel Papazoglou und Georgakopoulos 2003; Alonso u. a. 2004).

Als Alternative zu Web Services wäre der Einsatz eines anderen RPC-Mechanismus (beispielsweise RMI, einem in Java integrierten Mechanismus zum Aufruf entfernter Methoden, vgl. Abts 2003) oder einer generelleren Middleware (beispielsweise CORBA, einer von der Object Management Group[5] spezifizierten Middleware zur systemübergreifenden Verbindung objektorientierter Systeme, vgl. Brose u. a. 2001) möglich gewesen (vergleiche dazu auch den Überblick über Verteilungsmechanismen in Boger 1999). Letztendlich wäre auch die Implementierung eines eigenen Protokolls, das über TCP/IP angesprochen wird, möglich gewesen, hätte aber einen deutlich höheren Aufwand und geringere Interoperabilität zu weiteren Systemen bedeutet. Die Entscheidung für den Einsatz von Web Services war in diesem Falle begründet durch die Heterogenität der Komponenten und der Einsatz von XML-Technologien in weiteren Komponenten des Systems und die dadurch erleichterte Umsetzung und Integration.

Natürlich bietet sich für den Fall der exklusiven lokalen Nutzung dieser Dienste auch die direkte Nutzung der API dieser Komponenten an. Dadurch werden die Nachteile der verteilten Architektur vermieden, welche sich als Performanzverlust und möglicherweise schlechtere Erreichbarkeit zeigen könnten.

Die Beschreibung des Protokolls zur Nutzung des Datenschutzdienstes ist weniger interessant als dessen Schnittstellenbeschreibung, denn das Protokoll besteht nur aus einem Aufruf und der entsprechenden Antwort. Der Datenschutzdienst bietet nur eine Funktion an: das Untersuchen einer P3P-Datenschutzrichtline auf Konformität zu einer APPEL-Regel. Übergabeparameter sind zwei XML-Dokumente. Zusätzlich wird ein Datenschema gebraucht, dem beide Dokumente entsprechen müssen. Der vierte und letzte Parameter ist der Name des Attributs, das angefragt wurde. Der Rückgabewert ist ein strukturierter Datentyp, der Informationen zum vorgeschlagenen Verhalten, eine Beschreibung der Auswertung, die letztendlich vorgeschlagene Datenschutzrichtlinie, eine mögliche Frage an den Benutzer sowie einen Wahrheitswert umfasst, ob der Benutzer gefragt werden

[5] OMG, siehe `http://www.omg.org`

soll. Desweiteren ist ein möglicher Fehlercode enthalten, falls die Auswertung fehlgeschlagen ist. Außerdem wird die Nummer der APPEL-Regel, die gewählt worden ist, mit angegeben. Die ausführliche Beschreibung der Schnittstelle im WSDL-Format findet sich im Anhang B als Listing B.1 (siehe dazu auch Rickert 2004).

8.3.4. Protokoll zur Nutzung des Ontologie- und Transformationsdienstes

Der Ontologie- und Transformationsdienst ist eine Zusammenfassung der Elemente, die in Unterabschnitt 4.1.3 und Unterabschnitt 4.2.6 eingeführt worden sind und hat die Aufgabe, dem Identitätsmanagement-System auf Anfrage Teil-Ontologien und Transformationsregeln zwischen verschiedenen Teil-Ontologien zu suchen. Die Transformation selbst kann vom Dienst oder auch vom Identitätsmanager selbst durchgeführt werden. Wegen des engen Zusammenhangs von Teil-Ontologien und Transformationsregeln ist eine Zusammenfassung dieser Elemente sinnvoll.

Ontologiedienst

Es gibt zwei Methoden zum Suchen nach Teil-Ontologien. Erstens kann nach einer Teil-Ontologie gesucht werden, deren Name[6] bekannt ist. Alternativ kann nach einem Attributtyp gesucht werden, der in der Ontologie enthalten sein soll. In beiden Fällen ist das Ergebnis eine Teil-Ontologie oder eine Fehlermeldung, falls keine entsprechende Teil-Ontologie gefunden werden konnte.

Transformationsdienst

Der Umgang mit Transformationen, also den Übersetzungen von Daten, die einer Ontologie entsprechen, in äquivalente Daten, die einer anderen Ontologie entsprechen, ist komplexer als der Umgang mit Ontologien. Das Identitätsmanagement-System muss zuerst herausfinden, welche der Daten es in das gewünschte Zielformat übersetzen kann.

Die Suche nach Transformationsregeln gestaltet sich wie folgt: Erhält das Identitätsmanagement-System eine Anfrage, deren Teil-Ontologie nicht bekannt ist oder für die keine Daten vorliegen, so kann das System nicht automatisch wissen, in welchem anderen Format die Daten möglicherweise vorliegen. Die erste Anfrage an den Transformationsdienst ist also der Name der Teil-Ontologie, nach der das System gefragt worden ist. Das Ergebnis ist eine Liste der Teil-Ontologien, für die Transformationsregeln mit der gesuchten Teil-Ontologie als Ziel der Transformation vorliegen. Aus dieser Liste wählt der Identitätsmanager eine Ontologie

[6] Mit dem Namen ist hier der eindeutige Bezeichner, also ein URI gemeint.

aus, für die Daten vorliegen. Soll der Identitätsmanager die Transformation selbst durchführen, so kann er nun den Transformationsdienst direkt nach dieser Transformationsregel fragen (diese Anfrage enthält die Namen der beiden Ontologien) und erhält sie als Antwort. Soll die Transformation dagegen beim Dienst erfolgen, so wird der Datensatz, der transformiert werden soll, und der Name der Ziel-Ontologie übergeben. Der Rückgabewert dieses Aufrufs ist der transformierte Datensatz, der als Antwort an den Kommunikationspartner geschickt werden kann.

8.4. Notwendige und optionale Komponenten der onefC-Identitätsmanagement-Infrastruktur im Detail

In Kapitel 4 sind auf abstrakte Weise die elementaren Bausteine eines Identitätsmanagement-Systems aufgeführt und erklärt worden. Es wurden die notwendigen, funktionalen Elemente und Komponenten eines solchen Systems benannt und in ihrer Funktionsweise beschrieben. In diesem Kapitel soll nun an einem konkreten Beispiel – der onefC-Identitätsmanagement-Infrastruktur – gezeigt werden, wie diese Komponenten im Detail konstruiert sein können. Dazu wird mit einem Hilfsmittel für Identitätsmanagement-Systeme angefangen: der Sitzungsebene. Die Sitzungsebene bietet Mechanismen zum Bündeln von Kommunikationsakten und dazugehörigen digitalen Identitäten, wie es in Unterabschnitt 1.3.5 gefordert und in Abschnitt 2.4 beschrieben worden ist. Sie bietet die Grundlage, auf der Identitätsdatenkommunikation stattfindet. Elementarer Baustein dieser Ebene ist der Sitzungskanal und seine Programmierschnittstelle (API). Aber auch die Sitzungsmanagement-Funktionalität im Identitäts- und Sitzungsmanager leistet einen wesentlichen Beitrag zum Identitätsmanagement: Da beim onefC-Konzept Pseudonymisierung automatisch durch die Einführung sitzungsabhängiger *session nicknames* gewährleistet ist, sollte die Verwaltung dieser Sitzungspseudonyme und der eigentlichen digitalen Identitäten gemeinsam betrachtet werden.

8.4.1. Die Sitzungskanal API

Ein zentraler Baustein der onefC-Infrastruktur ist die Sitzungsebene, welche zwischen Transportebene und Anwendungsebene in den Netzwerk-Protokollstapel eingefügt wird. Um diese Ebene zu benutzen, müssen Anwendungsprogramme anstelle der API der Transportebene (beispielsweise `Socket` und `ServerSocket` aus dem Paket `java.net` im Falle von Java-Programmen) nun die API der Sitzungsebene verwenden. Sie bietet die Methoden, die einen Sitzungsauf- und -abbau initiieren, Identitätsdaten übertragen und Transportkanäle aushandeln.

Die Sitzungskanal-API für onefC liegt prototypisch für die Programmiersprache Java[7] vor. Dieser Prototyp ist an TCP gebunden, dies ist für viele Anwendungsprogramme das wichtigste Transportprotokoll. Andere Transportprotokolle, auch solche die nicht zum Internet gehören, könnten ebenso umgesetzt werden – zum Zwecke des „Proof of Concept“ wurde lediglich das wichtigste implementiert.

Die Klasse `SessionChannel`

Das Protokoll zum Sitzungsaufbau (vgl. Unterabschnitt 8.3.1) wurde in der Klasse `SessionChannel` implementiert. Sie bietet folgende Methoden an:

- `getChannel` – Diese Methode initialisiert eine logische Verbindung zu einem Kommunikationspartner, dessen `URI` als Parameter übergeben werden muss. Es wird noch nicht notwendigerweise eine physikalische Verbindung aufgebaut. Dies ist eine statische Methode, die ein Objekt der Klasse `SessionChannel` zurückgibt, welches dann die Hauptschnittstelle für Anwendungsprogramme darstellt (in dieser Rolle, aber nicht in der Funktion vergleichbar mit der Klasse `java.net.Socket`).

- `probe` – Diese Methode testet einen `SessionChannel` auf Kompatibilität mit dem Protokoll der Gegenseite.

- `negotiate` – Diese Methode implementiert den Sitzungsaufbau, wie er in Unterabschnitt 8.3.1 beschrieben ist. Es werden *session nicknames* und eine Sitzungskennung ausgehandelt sowie kryptographische Schlüssel ausgetauscht. Es handelt sich hierbei um eine nicht blockierende Methode – soll der Programmfluss bis zum erfolgreichen Aufbau der Sitzung (oder bis zum auftretenden Fehler) unterbrochen werden, so kann stattdessen die Methode `negotiateBlocking` verwendet werden, der zusätzlich ein Timeout als Parameter übergeben wird (-1 als Parameter erlaubt zeitlich unbegrenztes Warten).

- `acceptTransportSlot` – Dies ist eine blockierende Methode, die einen TransportSlot zurückgibt, sobald von der Kommunikationsgegenseite eine Anfrage danach ankommt. Sie stellt die „Server“-Seite einer Transportkanal-Verhandlung dar.

- `negotiateTransportSlot` – Mit dieser Methode kann innerhalb einer Sitzung über einen Transportkanal verhandelt werden. Als Parameter wird ein `URI` benötigt, der sowohl den Endpunkt auf der Gegenseite als auch das Transportprotokoll festlegt. Der Rückgabewert ist ein Objekt der Klasse `TransportSlot`.

[7] `http://java.sun.com`

- `suspendSession` – Sitzungen können unterbrochen werden. In diesem Fall werden alle Daten der Sitzung (*session nicknames* und Verschlüsselungsdaten) persistiert.
- `resumeSession` – Diese Methode nimmt eine zuvor per `suspendSession` unterbrochene Sitzung wieder auf. Diese beiden Methoden sind zur einfacheren Wiedererkennung von Kommunikationspartnern gedacht, allerdings im derzeitigen Prototypen noch nicht implementiert.

Neben diesen für die Sitzungsverwaltung notwendigen Methoden gibt es einige weitere Methoden, welche die Anbindung des Identitätsmanagement sicherstellen. Sie werden von identitätsbewussten Anwendungen (*identity aware*) gebraucht.

- `registerIdentityMsgListener` – Objekte, die das Interface `IdentityMessageListener` implementieren, können sich mit dieser Methode beim `SessionChannel` registrieren, um über Identitätsdaten-Nachrichten informiert zu werden, die in dieser Sitzung ausgetauscht werden. Es wird das „Listener Pattern“ nach Gamma u. a. (1995) genutzt.
- `sendIDMsg` – Über diese Methode können Identitätsdaten-Nachrichten gesendet werden. Sie werden in Form eines `Model` aus dem Jena Framework[8] übergeben, welches ein RDF-Modell nach Unterabschnitt 8.3.2 enthält.

Über eine statische Methode eines nach dem Singleton-Pattern implementierten `SessionMasterAdapterManager` erhält ein `SessionChannel` einen Verweis auf einen `SessionMasterAdapter`. Dies ist die Schnittstelle für den Sitzungs- und Identitätsmanager, der über die Methode `processMessage` alle ein- und ausgehenden Identitätsdaten-Nachrichten bearbeitet. Dem Prototypen ist eine Implementierung eines `SessionMasterAdapters` beigefügt, der über eine integrierte Servlet-Engine das Management von Sitzungen per HTTP-Protokoll zulässt (`DefaultSessionMasterAdapter`). Dieser wird von der prototypischen Implementierung des Identitäts- und Sitzungsmanagers genutzt, die im Unterabschnitt 8.4.2 beschrieben wird. Eine andere `SessionMasterAdapter`-Implementierung kann dem Singleton `SessionMasterAdapterManager` über die statische Methode `setSessionMasterAdapterImpl` bekanntgegeben werden. Dies wird für den Einsatz eigener, nicht auf HTTP aufsetzender Identitäts- und Sitzungsmanager benötigt und ist beispielsweise in Unterabschnitt 9.2.1 beschrieben.

Die Klassen `SocketTransport` und `ServerSocketTransport`

Transportkanäle übernehmen innerhalb einer Sitzung die Übertragung von Daten der Applikationsprotokolle. Die implementierende Klasse des Prototypen für

[8] Das Jena-Projekt bietet ein mächtiges Framework zum Umgang mit RDF-Modellen in Java, siehe `http://jena.sourceforge.net`

TCP-Kommunikation heißt `SocketTransport` und ist wegen der ähnlichen Funktionalität sowie zur leichteren Verständlichkeit und auch Migration älterer Programme sehr stark an die Klasse `java.net.Socket` angelehnt.

- `connect` – Der Aufruf von `connect` versucht, mit einer als Parameter übergebenen URI eine TCP-Verbindung aufzubauen, über die Daten eines Applikationsprotokolls gesendet werden können. Als zweiter Parameter wird ein Sitzungskanal übergeben, mit dem dieser Transportkanal assoziiert sein soll.
- `getInputStream` – Als Rückgabewert dieser Funktion erhält man einen `java.io.InputStream`, aus dem eingehende Daten der Verbindung gelesen werden können.
- `getOutputStream` – Diese Methode gibt einen `java.io.OutputStream` zurück, in den Daten geschrieben werden können, die über die Verbindung geschickt werden sollen.

Analog gibt es die Klasse `ServerSocketTransport`, die mit der Methode `accept-SocketTransport` einen per `connect`-Aufruf gestarteten Verbindungsaufbau entgegennimmt und auf der Serverseite ein `SocketTransport`-Objekt zur Verfügung stellt.

8.4.2. Identitäts- und Sitzungsmanager

Der Identitäts- und Sitzungsmanager der prototypischen Implementierung wurde nicht in Java entwickelt, sondern als Erweiterung für den Web-Browser Firefox[9]. Dieser findet eine immer größere Verbreitung und bietet ein mächtiges Framework zum Entwickeln von Plugins (siehe Boswell u. a. 2002). Die Oberflächen der Plugins werden in der XML-Sprache XUL[10] beschrieben, während die Logik in JavaScript programmiert wird. Es steht eine große Menge fertiger Komponenten bereit, die beispielsweise den Umgang mit RDF oder digitalen Signaturen erleichtern. Eigene Komponenten können in JavaScript oder in C++ entwickelt werden, wobei JavaScript die Plattform-Unabhängigkeit sichert, C++ jedoch die Leistung steigert und das Entwickeln einfacher macht, da Debugger und andere Entwicklungshilfen genutzt werden können. Die Komponenten für den onefC-Prototypen wurden in JavaScript entwickelt. Der Einsatz der mitgelieferten Komponenten für RDF hat die Entwicklung stark vereinfacht, was ein wesentlicher Entscheidungsgrund für diese Plattform war.

Der onefC-Identitäts- und Sitzungsmanager trägt den Namen „mozidm“ (Mozilla Identitätsmanager[11]). Er präsentiert sich als eigenes Fenster, in dem drei Karteireiter zur Wahl stehen: „Profile Management“ für das Betrachten und Verwalten

[9] `http://www.mozilla.org/products/firefox`
[10] eXtensible User Interface Language, siehe `http://www.xulplanet.com`

der eigenen und fremden Identitätsdaten, „Session Management“ für das Sitzungsmanagement und damit verbundene Identitätsmanagement-Aufgaben, sowie „Vocabulary“ für das Verwalten und Betrachten der Teil-Ontologien.

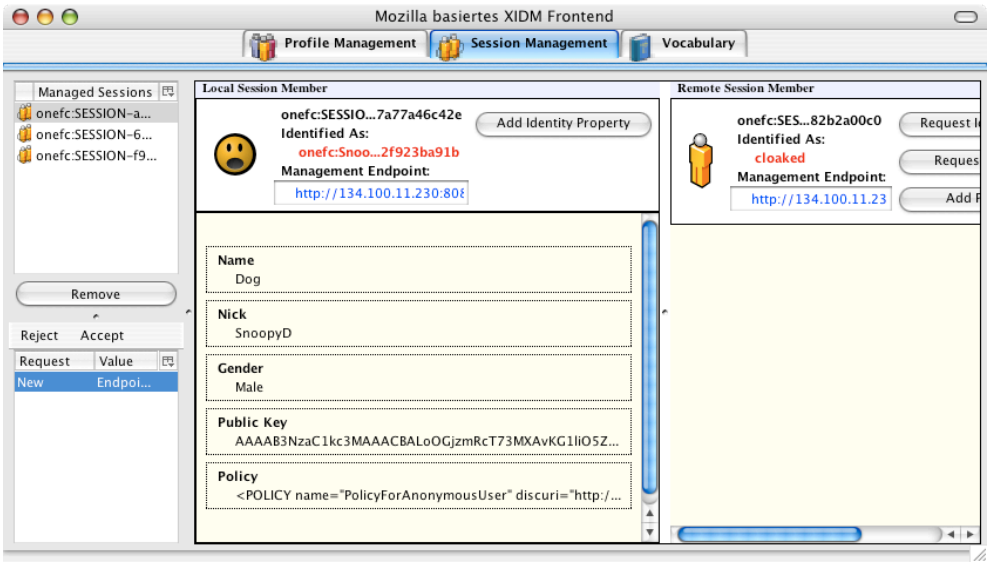

Abbildung 8.6.: Sitzungsverwaltung in „mozidm“, der onefC Identitäts- und Sitzungsmanager Bedienschnittstelle.

Die meisten Aktionen sind im „Session Management“-Bereich möglich (siehe Abbildung 8.6). Hier werden alle eingehenden Anfragen und Aussagen gesammelt (Liste unten links im Bild). Je nach Eintrag in dieser Liste erscheinen darüber die verschiedenen Aktionen, die der Benutzer damit ausführen kann. Als erstes erscheinen hier die Hinweise, dass neue Sitzungen aufgebaut werden. Die Verwaltung der Sitzungen kann akzeptiert oder abgelehnt werden – in letzterem Fall kann das System keine Identitätsnachrichten aus der Sitzung empfangen und keine darin senden. Bei Anfragen kann der Benutzer

- sich Details zur Anfrage anzeigen lassen (Herkunft, *session nicknames*, Sitzungskennung),
- sich die Sitzung, in der die Anfrage gestellt wurde, anzeigen lassen (im rechten Bereich des Fensters erscheinen Darstellungen der Sitzungsteilnehmer),
- die Anfrage beantworten (öffnet einen Dialog),

[11] Mozilla ist der Name des Projektes, aus dem Firefox stammt, sowie auch die der Name der ältesten Anwendung aus diesem Projekt. Sie umfasst nicht nur einen Browser, sondern auch einen E-Mail Client, einen IRC-Client und einen Kalender. Weitere Informationen unter `http://www.mozilla.org`.

- die Antwort verweigern (die Gegenseite erhält eine Nachricht darüber) oder
- die Anfrage ignorieren (die Gegenseite erhält keine Nachricht).

Aussagen können zu den Sitzungspseudonymen gespeichert (assoziiert) oder ignoriert werden; auch hier können Details angezeigt werden. Aussagen, die über das eigene Sitzungspseudonym getroffen worden sind, können über eine „Aquire“-Schaltfläche in die Teil-Identität, die diesem Sitzungspseudonym zugeordnet ist, übernommen werden – dies erlaubt die Identitätsbildung (vergleiche dazu Unterabschnitt 3.3.3). Die Zuordnung einer Teil-Identität zu einem Sitzungspseudonym lässt sich über die „Choose Identity“-Schaltfläche in der Darstellung des eigenen Sitzungspseudonyms (linke Hälfte der Darstellungen der Sitzungsteilnehmer, überschrieben mit „Local Session Member“) tätigen. Im dabei erscheinenden Dialog kann der Benutzer nicht nur aus einer Liste seiner eigenen Teil-Identitäten wählen, sondern auch eine neue anlegen. Mittels der Schaltflächen bei der Darstellung des Kommunikationspartners lassen sich Identitätsdaten-Anfragen, Aufforderungen zur Identifizierung und Aussagen über das (fremde) Sitzungspseudonym senden.

Der Sitzungsaufbau wird von „mozidm“ nicht initiiert. Dieses Frontend dient lediglich der Bedienung von Sitzungen, die von anderen Applikationen aufgebaut werden. Es kommuniziert dafür über HTTP mit dem im vorigen Abschnitt genannten `DefaulSessionMasterAdapter`.

Der Datenschutzdienst wird über SOAP/HTTP angesprochen. Auch dafür gibt es eine Firefox/Mozilla Komponente, die den Umgang mit Web Services erlaubt. Die Schnittstelle wird in dem Dialog bedient, der das Beantworten von Identitätsdaten-Anfragen erlaubt: wählt ein Benutzer die Schaltfläche „Check Policy“, so werden zunächst die notwendigen Daten gesammelt. Sollte die P3P-Policy des Kommunikationspartners noch nicht bekannt sein, so wird sie automatisch angefragt. Dann wird der Datenschutzdienst mit diesen Daten aufgerufen. Je nach Ergebnis wird dem Benutzer ein Warnhinweis angezeigt. Dieses rudimentäre Verhalten ist sicherlich noch verbesserungsfähig, reicht jedoch zur prototypischen Evaluation der Integration der Dienste aus.

8.4.3. Datenschutzdienst

Der Datenschutzdienst, der Benutzer bei der Entscheidung helfen soll, ob bestimmte persönliche Daten an den Kommunikationspartner gesendet werden sollen oder nicht, ist prototypisch als Web Service umgesetzt worden. Dabei wurde die Funktionalität in der Programmiersprache Java umgesetzt. Die Erstellung der Web Services Interfaces sowie der WSDL-Beschreibung des Dienstes wurde mit dem Werkzeug „Axis“[12] unterstützt, welches eine mächtige Bibliothek zum Erstellen und Nutzen von Web Services unter Java bietet. Axis ist ein Open-Source-

[12] `http://ws.apache.org`

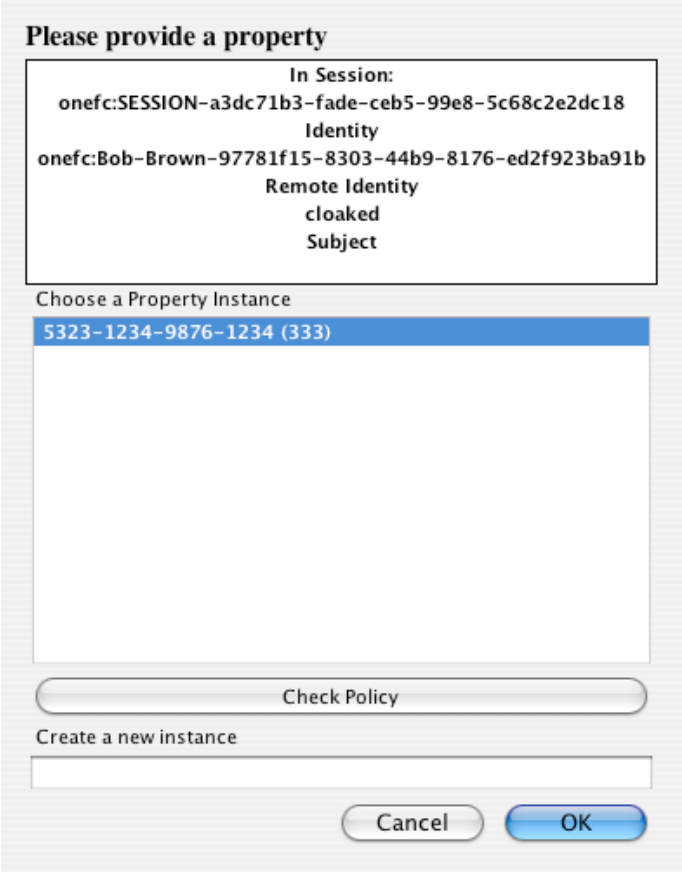

Abbildung 8.7.: Dialog zum Beantworten einer Identitätsdaten-Anfrage. Die „Check Policy“-Schaltfläche löst eine Anfrage an den Datenschutzdienst aus.

Projekt und durch den kostenlosen Bezug für die Erstellung wissenschaftlicher Prototypen gut geeignet.

Die Schnittstelle des Datenschutzdienstes bietet dabei zwei Methoden, um die Funktionalität zu nutzen (siehe Listing 8.4): Die P3P-Policy und die APPEL-Regel können entweder als `Document` nach dem Standard des W3C oder als einfache Zeichenkette (`String`) übergeben werden. Dies erleichtert die Nutzung aus Umgebungen, die den W3C Standard nicht unterstützen. Rückgabewert ist in jedem Fall ein strukturierter Datentyp, der alle möglichen Ergebnisse kapselt. Der Versuch, die Rückgabedaten als Zeichenketten (`String[]`) zurückzugeben, scheiterte an einem Fehler der SOAP-Erstellung von Axis.

Listing 8.4: Java-Interface des Datenschutzdienstes

```
import java.rmi.RemoteException;
import org.onefc.privacy.appelevaluator.EvalReturn;
import org.w3c.dom.Document;

public interface IAppelWS extends java.rmi.Remote  {
  public EvalReturn evaluateStrings(
    String appel,
    String policy,
    String dataschema,
    String url) throws RemoteException;

  public EvalReturn evaluate(
    Document appel,
    Document policy,
    Document dataschema,
    String url) throws RemoteException;
}
```

Die Implementierung der APPEL-Evaluationsfunktionalität ist von einer J-Script Implementierung des European Commission Joint Research Center[13] abgeleitet und vollständig in Java als Open Source (GPL[14]) unter `http://www.onefc.org` verfügbar. Ein solcher Datenschutzdienst ist permanent unter der Adresse `http://onefc.informatik.uni-hamburg.de/axis/services/AppelEvalService` erreichbar, kann aber auch lokal installiert werden. Durch eine Modifikation des Web Application Containers, der den Web Service zur Verfügung stellt, wäre auch eine Nutzung über das verschlüsselnde Protokoll HTTPS möglich; zur prototypischen Demonstration wurde darauf verzichtet.

8.4.4. Ontologie- und Transformationsdienst

Die Konstruktion des Ontologie- und Transformationsdienstes ähnelt der des Datenschutzdienstes: die Funktionalität ist in Java programmiert und per Axis Web Service verfügbar gemacht. Dabei ist der Dienst in drei Komponenten aufgeteilt: einen Ontologiedienst, der die Suche nach Ontologien aufnimmt, einen Transformations-Suchdienst sowie einen Transformationsdienst, der die Übersetzung letztendlich durchführt.

`IOntologyFinderWS`

Die Programmierschnittstelle des Ontologiedienstes (`IOntologyFinderWS`) hat zwei Methoden: `findOntologyByName`, die einen String als Parameter bekommt, der für den Bezeichner einer Ontologie steht. Rückgabewert ist die gefundene Ontologie oder ein Fehlercode, falls sie nicht gefunden worden ist. Die Methode `findOntologyByAttributeName` bekommt einen Attributnamen als Parameter,

[13] `http://p3p.jrc.it`
[14] `http://www.gnu.org/licenses/licenses.html#GPL`

der in der zu suchenden Ontologie enthalten sein soll. Rückgabewert kann hier eine Liste der gefundenen Ontologien oder wiederum ein Fehlercode sein.

Implementiert ist dieses Interface durch eine Klasse, die sowohl ein Speicher für Ontologien ist als auch eine Schnittstelle zu einem Suchsystem hat. Das Suchsystem wurde als Peer-to-Peer Netzwerk nach dem JXTA[15]-Framework entwickelt (vgl. Oaks und Gong 2002). JXTA bietet die Möglichkeit, so genannte Peer-Groups zu bilden, innerhalb derer verschiedenste Dienste angeboten werden, die von einem, mehreren oder allen Peers der Gruppe erbracht werden. Im Falle der Suche nach Ontologien wird der Dienst von allen Peers erbracht – es ist ein vollständig dezentraler Dienst. Jeder Teilnehmer bietet die ihm bekannten Ontologien im Netzwerk an und kann entsprechend auf alle Ontologien aller anderen Teilnehmer zugreifen.

`IMappingFinderWS`

Die Schnittstelle zum Transformations-Suchdienst (`IMappingFinderWS`) hat zwei Methoden: `listMappingsTo` erhält den Namen einer Ziel-Ontologie als Parameter (`String`) und gibt eine Liste der Namen der möglichen Quell-Ontologien zurück. Um eine Transformationsregel zu erhalten, ruft man `getMapping` mit den beiden Namen der Teil-Ontologien (Quell-Ontologie und Ziel-Ontologie) auf. Diese Methode wird nur in dem Fall vom Identitätsmanager aufgerufen, wenn dort die Transformation durchgeführt werden soll. Ansonsten wird sie vom Transformationsdienst aufgerufen. Dies kann direkt über die Java-API geschehen, wenn der Transformationsdienst und der Transformations-Suchdienst im gleichen Prozessraum laufen. Die Verfügbarkeit der Methode als Web Service ermöglicht es aber auch, diese beiden Dienste auf getrennten Systemen laufen zu lassen.

Die implementierende Klasse des Suchdienstes ähnelt dabei dem Ontologie-Suchdienst und nutzt ebenfalls eine JXTA-Peer-Group zum Suchen der Transformationen. Durch die große Ähnlichkeit bietet sich hier die Ableitung der beiden Klassen von einer gemeinsamen, abstrakten Klasse an, welche die Grundfunktionalitäten zum Speichern, Verwalten und Suchen der Elemente erledigt. Ontologien und Transformationsregeln werden dafür von einem gemeinsamen Interface abgeleitet.

`IMapperWS`

Das Interface des Transformationsdienstes (`IMapperWS`) hat nur eine Methode: `doMap` bekommt als ersten Parameter die zu übersetzenden Daten, und als zweiten Parameter den Namen der Ziel-Ontologie. Rückgabewert sind die übersetzten Daten. Den Namen der Quell-Ontologie kann sich die Implementierung aus den Quell-Daten extrahieren. Liegt die Transformationsregel noch nicht vor, so wird

[15] `http://www.jxta.org`

sie per `getMapping` vom Transformations-Suchdienst angefordert. Da Quell- und Ziel-Daten jeweils in RDF/XML vorliegen, ist die Transformation leicht mittels XSLT (eXtensible Stylesheet Language Transformations, siehe Clark 1999; Brink und Tauber 2004) zu erreichen.

8.5. Einordnung des onefC-Systems nach der Klassifizierung aus Kapitel 7

Bei technischen Kommunikationssystemen ist eine Vorhersage, wie dieses System nach dem Entwurf dann tatsächlich benutzt wird, besonders schwer. Ein Beispiel dafür ist das World Wide Web, das Anfang der neunziger Jahre dazu entworfen worden ist, wissenschaftliche Beiträge von Forschern verschiedener Hochschulen verknüpfen und abrufen zu können. Die technische Weiterentwicklung und sowie kommerzielle als auch kulturelle und soziale Nutzung des WWW, wie sie heute stattfindet, war damals weder absehbar noch vorbereitet.

Entsprechend ist es schwierig vorauszusagen, für welche Zwecke eine Identitätsmanagement-Infrastruktur tatsächlich eingesetzt werden wird. Einige Anwendungsgebiete sind prädestiniert, wie in Kapitel 5 beschrieben. Jedoch ist nicht absehbar, welche weiteren Anwendungsfelder die Internet-Gemeinschaft in den neuen Möglichkeiten sieht. Daher ist es möglich, dass der Schwerpunkt der auf Identitätsmanagement-Systemen basierenden Anwendungen in ganz anderen als den dort genannten Bereichen liegen. Während einerseits neue technische Möglichkeiten die Erfahrung des Internets erweitern, verändert die Internet-Kultur andererseits die technischen Systeme, indem neue Anforderungen der Benutzer von Werkzeug- und Dienstanbietern erfüllt werden. Die vorgestellten Anwendungsgebiete stammen aus dem Bereich des Online-Community-Support und der gesellschaftlichen Navigationsunterstützung. Reputations- und Vertrauenssysteme sind ebenso gut vorstellbar.

Das onefC-System befindet sich derzeit in einem prototypischen Stadium. Trotzdem kann man die systemarchitektonischen Grundzüge erkennen. Die Datenhaltung bei onefC ist dezentral: Identitätsdaten können sowohl beim Anwender als auch bei einer *Trusted Third Party* gespeichert werden. Zusätzlich ist vorgesehen, dass die Identitätsdaten beim Kommunikationspartner zwischengespeichert werden können, damit sie nicht mehrfach übertragen werden müssen. Dies hat zur Folge, dass Persönlichkeitsbilder entstehen, die gepflegt und aktualisiert werden müssen. Die Datenschemata (Teil-Ontologien) sind frei wählbar und können dynamisch sein. Durch die Auftrennung der Gesamt-Ontologie in Teil-Ontologien wurde eine Modularisierung erreicht, die Erweiterungen während der Laufzeit erlaubt.

Aufgrund der Offenheit des onefC-Systems kann es sowohl anbieterorientierte als auch anwenderorientierte Anwendungen unterstützen, wobei der Schwer-

punkt auf letzteren liegt. Die Anwendungen können Client/Server-basiert oder nach dem Peer-to-Peer Modell aufgebaut sein, also mit gleichberechtigten Kommunikationspartnern. Letzteres ist zum Beispiel in einem Chat- oder Instant-Messenger-System denkbar. Allerdings muss auf allen Seiten der Kommunikation (Client, Server, Peer) eine Änderung des Systems und der Software vorgenommen werden: Es muss sowohl die Sitzungsschicht auf die Transportschicht gesetzt werden, als auch jede Anwendung, die das System nutzen soll, an die Sitzungsschicht angepasst werden. Das System ist also stark invasiv.

Das onefC-System ist nicht direkt interoperabel mit anderen Systemen, denn diese müssten dafür auf die onefC-Sitzungsinfrastruktur aufsetzen. Eine Interoperabilität auf Anwendungsebene ließe sich durch Verwendung von onefC als Datenspeicher nutzen – die freie Gestaltung der Datenschemata erlaubt beliebige Anwendungskontexte.

Das onefC-System zeichnet sich also als vorwiegend an der Selbstdarstellung orientiertes Identitätsmanagement-System aus. Tabelle 7.1 kann damit um eine Spalte für onefC erweitert werden (siehe Tabelle 8.1).

Tabelle 8.1.: Einordnung von onefC in die bestehende Kategorisierung

	.NET Passport	Liberty Alliance	Cookie-Cooker	ATUS	DRIM	onefC
Architektur	Client/-Server	Client/-Server	Peer-to-Peer	lokal	Client/-Server	frei
Datenhaltung	zentral	dezentral (Server)	lokal	lokal	dezentral (lokal / TTP)	dezentral (lokal / TTP)
invasiv	wenig (Server)	je nach Anwendung	sehr wenig	wenig	wenig	stark
Datenschema	fest	serverseitig	–	fest	dynamisch	dynamisch
Interoperabilität	keine	möglich	keine	möglich (aufwändig)	möglich	möglich
Orientierung	Anbieter	Anbieter	Anwender	Anwender	Anwender	beides möglich
Intention	–	–	Privacy-Driven	Privacy-Driven	Privacy-Driven	Self-Portrayal-Driven

9. Anwendungsentwicklung mit dem onefC-System

Nachdem im vorigen Kapitel die Konstruktion des prototypischen Identitätsmanagement-Systems onefC beschrieben worden ist, soll nun gezeigt werden, wie diese infrastrukturelle Komponente für Anwendungen genutzt werden kann. Dazu wird zunächst an einem sehr einfachen Beispiel der Einsatz der onefC-Session-API gezeigt: Es wird ein einfacher Chat programmiert, bei dem eine Serverkomponente auf eingehende Verbindungsgesuche wartet, und ein Client eine solche Verbindung anfordert. Sobald die Sitzung aufgebaut ist, wird ein Transportkanal erstellt, über den Textnachrichten ausgetauscht werden können. Über den Sitzungskanal können mit Hilfe des Identitäts- und Sitzungsmanagement Clients „mozidm“ Identitätsnachrichten ausgetauscht werden. Durch die Wahl des sehr einfachen Anwendungsszenarios soll der Unterschied des Einsatzes des onefC-Systems zum Einsatz der Java-Socket-API hervorgehoben werden.

Anschließend wird gezeigt, wie die Invasivität von onefC für einen Einsatz mit dem bestehenden und etablierten HTTP-Sitzungskonzept in Einklang gebracht werden kann, ohne dass alle Client- und Server-Anwendungen neu programmiert werden müssten. Eine prototypische Umsetzung dieses Konzepts wird vorgestellt; Der Sourcecode dieses Prototyps sowie eine lauffähige Demonstration findet sich unter `http://www.onefc.org`.

9.1. Ein einfaches Anwendungsbeispiel für onefC: „Chat“

Um den grundsätzlichen Umgang mit der onefC-Session-API und den Unterschied zur herkömmlichen Programmierung mit TCP-Sockets an einem Beispiel zu zeigen, soll in diesem Abschnitt die Implementierung eines Chat-Systems gezeigt werden. Dabei wird das Beispiel zunächst mit TCP-Sockets implementiert, danach mit der onefC-Java-Session-API (siehe Unterabschnitt 8.4.1). Beide Lösungen sind in Java implementiert. Mit der onefC-Lösung können die Chat-Teilnehmer neben dem eigentlichen Chat persönliche Daten austauschen. Als Identitätsmanagement-Client kommt „mozidm“ auf beiden Seiten der Kommunikation zum Einsatz (siehe Unterabschnitt 8.4.2). Das bedeutet, dass kein automatischer Identitätsdatenaustausch stattfindet – alle Anfragen und Antworten werden explizit

von den Anwendern getätigt. Da es sich hierbei um einen Zweibenutzer-Chat handelt und nicht um ein System für beliebig viele Teilnehmer, wird in Anlehnung an ein Standard-Unix-Programm der Name „Talk“ gewählt.

9.1.1. Aufbau des Chat-Systems

Das Chat-System ist grundsätzlich in die Teile „TalkServer“ und „TalkClient“ unterteilt. Beides sind Java Kommandozeilen-Programme ohne grafische Bedienschnittstelle. Der TalkServer wird auf einem Rechner gestartet und wartet auf Verbindungsgesuche. Der TalkClient wird auf einem weiteren Rechner gestartet und sendet ein Verbindungsgesuch an einen per Kommandozeilenparameter übergebenen Server.

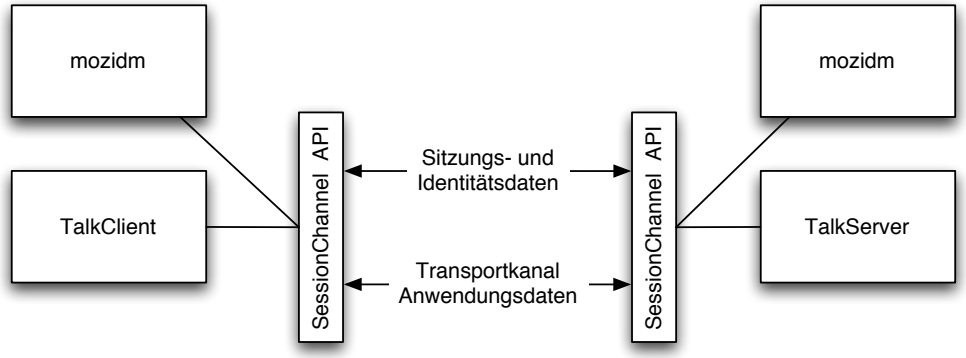

Abbildung 9.1.: Aufbau des Talk-Beispiels mit onefC Sitzungen

Der mozidm Identitäts- und Sitzungsmanagement-Client wird so konfiguriert, dass er neue Sitzungen des Talk-Systems erkennen und verwalten kann. Dies geschieht über den `DefaultSessionMasterAdapter`. Bevor die Talk-Verbindung also aufgebaut wird, muss mozidm auf beiden Seiten gestartet werden, damit auch Identitätsdaten ausgetauscht werden.

9.1.2. Implementierung des Chat-Systems

Die Implementierung von TalkServer und TalkClient ist sich sehr ähnlich. Beide benötigen zwei Threads: einen zum Lesen von Daten von der Kommandozeile, die der Gegenseite geschickt werden sollen, einen weiteren zum Empfangen der Daten aus dem Transportkanal. Die empfangenen Daten werden einfach auf der Kommandozeile ausgegeben. Für diese Aufgaben wurden zwei Thread-Klassen entwickelt: `TalkWriter` zum Schreiben der von der Kommandozeile gelesenen Daten in den Transportkanal, und `TalkReader` zum Lesen der Daten aus dem Transportkanal. `TalkWriter` benötigt im Konstruktor einen `java.io.DataOutputStream` als

Parameter, in den geschrieben werden soll, `TalkReader` einen `java.io.DataInputStream`, aus dem gelesen werden kann. Mit diesen Hilfsklassen ist die Implementierung der Talk-Komponenten sehr einfach. Es wird zunächst die Implementierung mit der Java-Socket-API gezeigt, woraufhin die Änderungen für den Einsatz der onefC-Session-API aufgeführt werden.

Chat über Java Sockets

Die Programmierung des Talk-Beispiels anhand der Java-Socket-API ist denkbar einfach. Hier werden die Hauptmethoden der Komponenten gezeigt. Zunächst öffnet der `TalkServer` einen `ServerSocket` an einem zuvor definierten Port, damit Verbindungsgesuche akzeptiert werden können. Aus einem per `accept()` entstandenen `Socket` können dann die Streams erfahren werden:

Listing 9.1: Hauptmethode des TalkServers für Java Sockets

```
public void start() {
  ServerSocket s;
  try {
    s = new ServerSocket(port);
    Socket socket = s.accept();

    // Thread zum Lesen aus dem Socket
    dis = new DataInputStream(socket.getInputStream());
    TalkReader listener = new TalkReader(dis);
    listener.start();

    // Thread zum Schreiben in den Socket
    dos = new DataOutputStream(socket.getOutputStream());
    TalkWriter writer = new TalkWriter(dos);
    writer.start();

  } catch (IOException e) {
    // ein Eingabe- oder Ausgabefehler ist aufgetreten
    e.printStackTrace();
  }
}
```

Auf der Seite des Clients wird eine Server-Adresse benötigt. Diese wird aus den Kommandozeilenparametern in der `main`-Methode gelesen und in der Klassenvariablen `server` gespeichert. Der Aufbau der Verbindung (respektive das Erstellen des Sockets) geschieht dann wie folgt:

Listing 9.2: Aufbau der Socket-Verbindung beim TalkClient

```
public void start() {
  try {
    socket = new Socket(server, port);
```

Der restliche Code ist identisch zum `TalkServer`. Die zusammenhängenden Sourcecodes – auch die der `TalkReader` und `TalkWriter` Klassen – finden sich in Anhang C.

Chat über eine onefC-Session

Die auf onefC aufsetzende Talk-Implementierung unterscheidet sich zu demjenigen, welches Sockets benutzt, lediglich im Verbindungsaufbau. Der Transportkanal, der nach dem Sitzungsaufbau verhandelt wird, bietet genau wie die API `java.io.Socket` die Klassen `java.io.DataInputStream` respektive `DataOutputStream` für den Datenaustausch der Applikationsprotokolle an[1].

Auf der Seite des Servers wird ein `SessionChannelServer` initialisiert und gestartet, der beim Akzeptieren einer Verbindung einen `SessionChannel` zurückgibt. Für diesen wird ein `ServerSocketTransport` gestartet, der letztendlich den Transportkanal erzeugt.

Listing 9.3: TalkServer mit onefC-Sitzung

```
public void start() {

  // SessionChannelServer wird initialisiert
  SessionChannelServer sessionServer =  new SessionChannelServer();
  try {
    while(true) {
      SessionChannel channel = null;
      ServerSocketTransport sT = null;
      SocketTransport socket = null;
      try {
        // warten auf Verbindungsanfrage
        sessionServer.listen(new URI(server));

        // Verbindung wird akzeptiert
        channel = sessionServer.accept();

        // Vorbereitung: warten auf Transportkanal
        sT = new ServerSocketTransport(channel,new URI(server));

        // Sitzung wird verhandelt
        while(!channel.negotiateBlocking(-1)) {
          try{Thread.yield(); }
          catch (Exception e) {}
        }

        // solange der Sitzungskanal besteht, soll
        // kommuniziert werden öknnen
        while(channel.isEstablished()) {

          // Transportkanal aufbauen
          socket = sT.acceptSocketTransport();
```

Für den Transportkanal wurde `socket` als Name gewählt: Dadurch ist der weitere Code fast identisch mit der zuvor vorgestellten Socket-Lösung. Lediglich die Fehlerbehandlung muss auf die Sitzungs-API angepasst werden. Der SocketTransport „socket“ bietet genau wie ein `java.io.Socket` die Methoden `getInputStream` und `getOutputStream` an.

[1] Das Applikationsprotokoll besteht in diesem Fall lediglich aus der Angabe, dass Textzeilen im UTF-8 Format ausgetauscht werden.

Der Client geht ähnlich vor wie der Socket-Client. Allerdings muss auch hier erst die onefC Sitzung aufgebaut werden.

Listing 9.4: TalkClient mit onefC Sitzung

```
public void start() {
  SessionChannel channel = null;
  SocketTransport socket = null;
  try {
    // ein SessionChannel Objekt wird erzeugt
    channel =
        SessionChannel.getChannel
          (new URI(server),
           SessionChannel.FCLIENT);

    // die physikalische Verbindung wird aufgebaut
    channel.openBlocking(-1);

    // nun wird die Sitzung aufgebaut
    channel.negotiateBlocking(-1);

    // der Transportkanal wird vorbereitet
    socket = new SocketTransport();

    // jetzt wird der Transportkanal sitzungsassoziiert aufgebaut
    socket.connect(new URI(server),channel);
```

Diese Talk-Komponenten können „identitätsbewusst“ gemacht werden, indem sie das Interface `org.onefc.ise.api.IdentityMessageListener` mit der Methode `identityMessage` implementieren. Damit könnte jede ein- oder ausgehende Identitätsnachricht mitgehört und verarbeitet werden. Es ist aber ebenso einfach, nun selbst Identitätsnachrichten zu senden, beispielsweise den gewünschten Spitznamen im Chat. Zu diesem Zwecke wurde der `TalkWriter` um die privaten Methoden `sendIDStatement` und `sendIDRequest` erweitert (siehe Listing C.7 im Anhang). Der `TalkWriter` untersucht jede zu sendende Nachricht auf Schlüsselwörter (`/set` und `/request`) und generiert daraus die Anfragen und Aussagen.

9.1.3. Mehrwert der onefC-Variante

Im vorigen Abschnitt wurde gezeigt, dass der Programmieraufwand für die Umsetzung einer Chat-Anwendung bei Verwendung der onefC-API gegenüber der Java-Socket-API nur geringfügig steigt. Dieser Aufwand wird jedoch dadurch belohnt, dass die reine Textkommunikation durch Übermittlung von Identitätsattributen angereichert werden kann. Die Kommunizierenden können sich gegenseitig Anfragen schicken und diese beantworten. Darüber hinaus können Teil-Identitäten angegeben werden, die in einer zukünftigen Sitzung wiederverwendet werden. Dadurch ist ein eindeutiges Wiedererkennen und damit auch die Wiederverwendung der bereits in Erfahrung gebrachten Identitätsattribute möglich.

Diese Informationen sind nicht nur vom Anwender nutzbar, sondern auch von der Anwendung selbst. Im vorliegenden Chat-Beispiel ließe sich dadurch der vom

Kommunikationspartner automatisch übertragene Spitzname vor seine Textnachrichten stellen, um sie eindeutig zu kennzeichnen. Diese in Chat-System äußerst gebräuchliche Funktion kann jedoch um beliebige, den Teil-Identitäten anhaftende Informationen erweitert werden.

9.2. Integration von HTTP-Sitzungen in onefC

Der Einsatz von onefC erfordert das Aufsetzen aller Kommunikationsprogramme sowohl bei Clients als auch bei Servern auf der Sitzungskanal-API. Durch diese starke Invasivität sind also spezielle Programme erforderlich, die beispielsweise mit den herkömmlichen HTTP-Clients und -Servern nicht kompatibel sind. Mit onefC-fähigen Programmen kann man nur andere onefC-fähige Programme ansprechen, alle anderen können mit dem hier verwendeten Konzept und Mechanismus der Session nicht arbeiten.

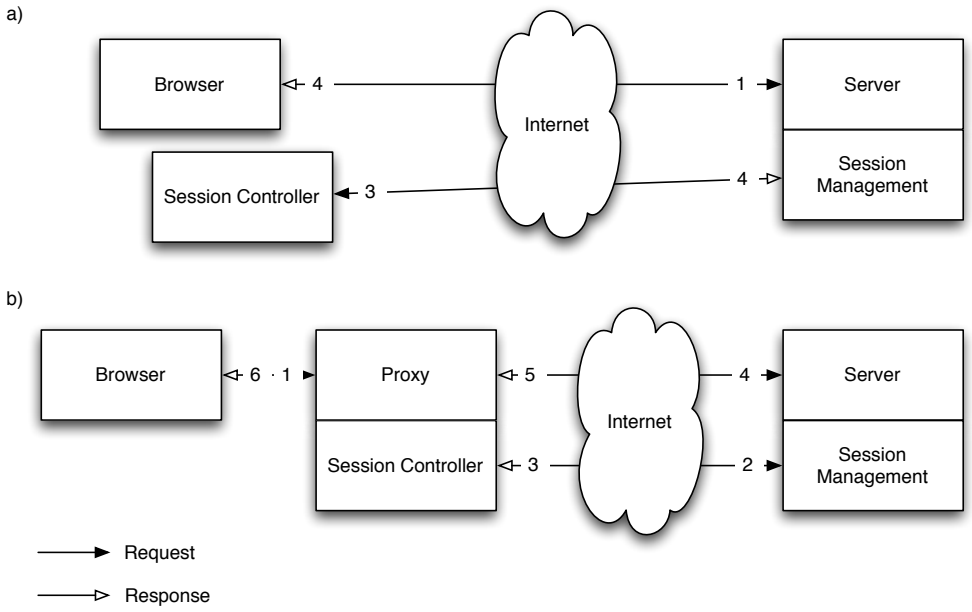

Abbildung 9.2.: Zwei verschiedene Wege, HTTP Browser und Server mit onefC zu verbinden.

Die Integration bisheriger Anwendungsprogramme ist daher nur über einen Umweg möglich. Dabei gibt es zwei verschiedene Szenarien (siehe Abbildung 9.2). Im Szenario a) hat der Client einen Session-Controller, der von einer Session-Management-Komponente auf dem Server zwecks Aufbau einer onefC-Sitzung kontaktiert wird. Über diese Sitzung werden dann die Identitätsdaten ausgetauscht, ohne dass der Browser des Anwenders damit in Berührung kommt –

wohl aber der Dienst auf dem Server. Damit weitere Anfragen des Clients mit dieser Sitzung assoziiert werden können, wird die Sitzungskennung der onefC-Sitzung entweder als Cookie auf dem Client-Browser gespeichert oder als Attribut zur HTTP-Sitzung hinzugefügt. Letzteres verspricht eine elegante Anbindung des onefC-Sitzungskonzepts an das bestehende HTTP-Sitzungskonstrukt; auf diesem Wege können Lebenszyklus und weitere Attribute der onefC-Sitzung mit der HTTP-Sitzung abgeglichen werden.

In der Variante b) spricht der Browser über einen Proxy mit dem Internet, welcher bei Anfragen zunächst testet, ob auf dem Server eine Session-Management-Komponente verfügbar ist und gegebenenfalls eine Sitzung aufbaut. Die Sitzungskennung wird dann der eigentlichen HTTP-Anfrage als weiterer HTTP Header hinzugefügt, damit der Server die passende Sitzung in der Session-Management-Komponente finden und nach Attributen des Anwenders fragen kann. Der Proxy fügt bei allen folgenden Anfragen an den selben Server automatisch den HTTP Header mit der onefC-Sitzungskennung in die Anfrage ein, wodurch die Sitzung auch längerfristig nutzbar ist.

Ein Vorteil der ersten Lösung ist, dass der Anwender seinen Browser nicht auf einen Proxy umkonfigurieren muss, sondern lediglich einen Session Controller starten muss. Der Nachteil dieser Lösung ist allerdings, dass die onefC-Sitzung vom Server aufgebaut wird, wodurch es notwendig wird, dass der Client-Rechner von außen erreichbar sein muss. Bei Rechnern mit einer Firewall oder hinter einem NAT (*network address translation*, automatische Umsetzung einer öffentlichen IP-Adresse auf verschiedene lokale Adressen) Router erfordert dies eine Konfigurationsänderung, damit die Anfragen des Servers nach einer onefC-Sitzung nicht blockiert werden. Dies birgt also einerseits eine Sicherheitslücke und andererseits Konfigurationsaufwand, der vermieden werden sollte.

In Variante b) dagegen muss der Anwender den Browser zur Nutzung eines Proxys umkonfigurieren. Dafür wird die onefC-Sitzung in diesem Szenario vom Client selbst aufgebaut, was die zuvor genannten Konfigurations- und Sicherheitsprobleme von Variante a) umgeht.

9.2.1. Prototypische Implementierung der Integration

Beide Varianten sind prototypisch umgesetzt worden, unter `http://www.onefc.org` verfügbar und können ebendort mit einer Beispielanwendung auch ausprobiert werden[2]. Dabei wird besonderer Augenmerk auf Variante b) gelegt. Der Proxy ist mit dem System Scone[3] entwickelt worden, das an der Universität Hamburg zum schnellen Entwickeln von Web-Prototypen entstanden ist (siehe

[2] Diese Beispielanwendung fragt allerdings lediglich per onefC Identitätsdaten-Anfrage nach dem Namen des Anwenders und begrüßt ihn dann auf der Web-Seite – ein klassisches, aber personalisiertes „Hello World“ Anfangsbeispiel.

[3] `http://www.scone.de`

Weinreich u. a. 2003). Der *Session Controller* auf der Seite des Clients ist ein Scone Plugin, das ausgehende HTTP Requests auf den Zielrechner untersucht, diesen nach einem onefC-Session-Management-Adapter befragt und im Falle einer positiven Antwort eine Sitzung erstellt.

An der Existenz einer Sitzungskennung im HTTP Request Header erkennt der Server, dass eine onefC-Sitzung besteht. Mit dieser Information kann das Servlet den prototypischen `OnefCSessionService`, der einen `SessionMasterAdapter` (siehe Unterabschnitt 8.4.1) und somit einen eigenen Identitätsmanager implementiert, mittels der Methode `askProperty` nach Attributen des Clients befragen. Die Methode `askProperty` bekommt den `HttpServletRequest`, aus dem die Sitzungskennung extrahiert wird, sowie den Namen des zu erfragenden Attributs samt Ontologienamen als Parameter.

9.2.2. Identitätsangereicherte Web-Anwendungen

Von der Anbindung HTTP-basierter Anwendungen an onefC Sitzungs- und Identitätsmanagement profitieren beide Seiten: Der onefC-Infrastruktur wird ein großer Bereich bestehender und beliebter Anwendungen erschlossen, wodurch die Nutzerbasis und damit die Effizienz des Systems erheblich gesteigert wird. Die Web-Anwendungen dagegen profitieren von der einfachen Integration des Identitätsmanagements, welche zur Personalisierung, zum automatischen Ausfüllen von – dann eventuell nicht mehr direkt sichtbaren – Formularen oder zur Integration von Community-Funktionen genutzt werden kann. In beiden Fällen profitiert der Anwender von einfacherer und einheitlicherer Benutzung sowie besseren Diensten.

9.3. Einordnung und Bewertung

Anhand der beiden Beispiele in diesem Kapitel wurde der praktische Einsatz des onefC-System veranschaulicht. Während das erste Beispiel – der Chat – nur einen sehr geringen eigenen Funktionsumfang bietet, können dennoch über den angebundenen onefC-Identitätsmanager beliebige Daten übermittelt werden. Das zweite Beispiel – die HTTP-Integration – ist sehr praxisnah, wird prototypisch eingesetzt und kann ausprobiert werden. Dabei ist allerdings keine Vielfalt an übermittelbaren Attributen gegeben: Um mehr als den Namen des Anwenders zu übertragen, müsste der Beispielcode angepasst werden. Anhand der hier beschriebenen Schnittstellen sollte dies jedoch ohne großen Aufwand möglich sein.

Die Beispiele zeigen, dass das Ziel der leichten Integration des onefC-Systems erreicht wurde. Es sind nur wenige Änderungen nötig, um eine Anwendung, die auf Java-Sockets aufsetzt, für den Einsatz mit onefC-Sessions anzupassen. Vorausgesetzt ist dabei die ordentlich gekapselte Nutzung der Kommunikationsschnittstelle im Originalsystem. Schwieriger als die programmiertechnische Integration

des Systems wird jedoch vermutlich die Entwicklung eines geeigneten Datenschemas, also einer Teil-Ontologie. Es gibt jedoch Hilfsmittel[4], die bei der Entwicklung von RDFS-Modellen, wie sie für onefC-Teil-Ontologien gebraucht werden, Unterstützung bieten. Die Möglichkeit zur dynamischen Übersetzung zwischen verschiedenen Teil-Ontologien bietet auch die Übersetzung zwischen verschiedenen Versionen einer bestimmten Teil-Ontologie, wodurch eine initiale Festlegung auf ein dauerhaftes Datenschema nicht notwendig ist.

Die beim Benutzer liegende Kontrolle über die eigenen Daten wird in beiden Beispielen deutlich. In keinem Fall wird eine zentrale Instanz zur Identitätsdatenverwaltung eingesetzt. Im Falle der serverseitigen Session- und Identitätsmanagement-Komponente im HTTP-Beispiel handelt es sich lediglich um einen lokalen Cache, der mehrfaches Übermitteln schon bekannter Identitätsattribute verhindert. Die Kontrolle über diese „fremden“ Identitätsdaten liegt in der Tat beim Web-Server. Genauso kann jedoch auch in privater Kommunikation ein Adressbuch über Kontakte geführt werden – der eigentliche Identitätsdatenhalter wird dadurch in seiner Kontrolle über die ursprünglichen Daten nicht eingeschränkt. Es ist sogar ein wesentlicher Aspekt des persönlichen Identitätsmanagement, dass Informationen über bekannte, fremde Teil-Identitäten in einem Cache gehalten werden können, ohne selbst Kontrolle darüber zu haben.

[4] zum Beispiel das Programm Protégé, `http://protege.stanford.edu`

10. Fazit

Identitätsmanagement ist eine anspruchsvolle Aufgabe für jeden kommunizierenden Menschen. Dies betrifft sowohl die reale als auch die digitale Welt. Während Identitätsmanagement in der realen Welt den meisten Menschen ohne Probleme gelingt, ist dieselbe Aufgabe in der digitalen Welt des Internets ungleich schwieriger. Um den Benutzer bei dieser Aufgabe zu unterstützen, werden digitale Identitätsmanagement-Systeme benötigt, die unterschiedliche Schwerpunkte setzen können.

10.1. Zusammenfassung der Arbeit

Diese Arbeit verdeutlicht, dass Identitätsmanagement in vielen Variationen entwickelt und in vielen Anwendungsbereichen gebraucht und auch schon eingesetzt wird. Neben der neu eingeführten, groben Einteilung der Varianten in organisatorisches, föderiertes und persönliches Identitätsmanagement (siehe Abschnitt 1.2) wurde in Kapitel 6 eine detaillierte Klassifizierung der Systeme entwickelt. Mittels dieser Klassifizierung sind bestehende und in der Forschung befindliche Systeme kategorisiert worden. Die Berücksichtigung der Intention – also der Absicht, mit der das System entwickelt und eingesetzt wird – stellt dabei die funktionalen Unterschiede in ein neues Licht. Es wurde gezeigt, dass der Bereich des persönlichen Identitätsmanagement bisher im Wesentlichen vom Datenschutz motiviert ist und die daraus entstandenen Systeme den Aspekt der Selbstdarstellung in digitalen Netzen vernachlässigen. Der Schwerpunkt dieser Arbeit liegt daher auf dem auf Selbstdarstellung ausgerichteten Identitätsmanagement. Für diesen Bereich des Identitätsmanagements wurde ein Konzept sowie eine prototypische Konstruktion hergeleitet und dargestellt. Es unterstützt den Anwender bei der Gestaltung, der Auswahl und dem Einsatz seiner digitalen Teil-Identitäten, ohne die direkte Kontrolle an eine zentrale Instanz abgeben zu müssen. Die dezentrale Kontrolle und freie Gestaltung der Datenschemata stellen dabei Herausforderungen an die unterstützenden Systeme, die im dritten Teil der Arbeit ausführlich aufgezeigt und adressiert worden sind. Das dabei entworfene, prototypische System zeigt eine dezentrale, komponenten- und dienstorienterte Architektur und demonstriert die Umsetzung mit modernen Mitteln. Es bietet einen Ausgangspunkt für Weiterentwicklungen, die weitere Aspekte der in Kapitel 4 genannten Elemente von Identitätsmanagement-Systemen umfassen können.

Die schnelle Entwicklung und Verbreitung von Identitätsmanagement-Systemen in Unternehmen und die Tendenz neuer Systeme, föderiertes Identitätsmanagement umzusetzen, zeigt das allgemeine Bestreben nach umfassenden Lösungen für das Thema. Dabei adressieren Industrie und Forschung teilweise gemeinsam die schwierigen Aufgaben der Konzeption und Konstruktion von Architekturen, Protokollen und Systemen (wie im Falle von Liberty Alliance und PRIME, siehe Abschnitt 7.2 und Abschnitt 7.5). Die Praxisrelevanz des Themas lässt sich auch aus der gesteigerten Medienpräsenz erkennen. Neben Fachmagazinen beschäftigen sich auch immer mehr allgemeine Magazine mit Themen des Identitätsmanagements[1], wodurch bei privaten Internet-Anwendern eine hohe Aufmerksamkeit erreicht wird. Dadurch wird auch die Notwendigkeit größer, entsprechende Systeme anzubieten, die an den Bedürfnissen der Anwender ausgerichtet sind. Diese Bedürfnisse umfassen eben nur in zweiter Linie den Datenschutz, denn ohne digitale Selbstdarstellung wäre dieser gar nicht notwendig.

10.2. Ausblick

Diese Arbeit bietet ein Konzept für ein auf Selbstdarstellung ausgerichtetes Identitätsmanagement-System und stellt einen funktionsfähigen, aber noch prototypischen System zur Demonstration der Herangehensweise an die Umsetzung des Konzeptes vor. Für eine weitere Verbreitung des Systems und den realen Einsatz in sicherheitskritischen Umgebungen ist jedoch die weitere Umsetzung bestimmter Elemente notwendig. Vor allem handelt es sich bei dem vorgestellten System um eine systemtechnische Architekturkomponente und nicht um ein Anwendungssystem.

Anwendungen

Die aufgeführten praktischen Anwendungsbeispiele (Kapitel 9) stellen keine realistischen Kommunikationsanwendungen dar, sondern dienen lediglich der Demonstration des Umgangs mit dem Prototypen. Es fehlen also praxistaugliche Anwendungen, anhand derer einerseits die Verbreitung des Systems und damit die Möglichkeit für die Anwender, verschiedene Teil-Identitäten und Teil-Ontologien zu entwickeln, geboten und andererseits eine Evaluation des Systems in realistischen Szenarien möglich wird. Dabei sind sowohl Anwendungen aus dem privaten Bereich (zum Beispiel Instant Messaging oder ein Foren-System) als auch personalisierte Dienste (beispielsweise Online-Einkauf, Nachrichtenbeschaffung oder Suchsysteme) zu berücksichtigen. Eine möglichst große Streuung der Anwendungen über verschiedene Anwendungsdomänen bewirkt eine größere Ver-

[1] beispielsweise durch das Interview des Autors im ZDF-Magazin „heute nacht“ vom 13.10.2004 mit dem Thema „Digitaler Ausweis“

breitung des Systems und damit einen günstigen Effekt für alle Arten von Online-Communities (siehe Unterabschnitt 6.2.1).

Umfassende Teil-Ontologien

Eine wichtige Entwicklung für ein praxistaugliches System wäre die Erstellung beziehungsweise dynamische Gestaltung von domänenspezifischen Teil-Ontologien. Je mehr umfassende Teil-Ontologien zu Beginn der Verbreitung einer Identitätsmanagement-Infrastruktur verfügbar sind und möglichweise zusammen mit dem System ausgeliefert werden, desto kleiner ist die Wahrscheinlichkeit, dass konkurrierende Teil-Ontologien entstehen. Fällt der Einstiegsaufwand der Erstellung einer Teil-Ontologie weg, könnte das System sowohl von Diensten als auch von Endanwendern besser akzeptiert werden. Die Bildung von organisationsübergreifenden Komissionen zu diesem Zweck könnte die Entwicklung beschleunigen und so Interoperabilität zwischen verschiedenen Diensten begünstigen.

Integration von persönlichem und föderiertem Identitätsmanagement

Eine Möglichkeit der Weiterentwicklung im gesamten Identitätsmanagement-Bereich wäre die Kombination der Ansätze aus dem persönlichen und dem föderierten Identitätsmanagement. Während persönliches Identitätsmanagement den Benutzer als oberste Kontrollinstanz seiner Daten und die freie Gestaltung der Datenschemata in den Vordergrund stellt, sieht das föderierte Identitätsmanagement die Austauschbarkeit der Identitätsdaten unter Dienstanbietern als wichtigsten Faktor an. Dies schließt sich nicht gegenseitig aus. Es stellt sich allerdings die Frage, wie der Benutzer Kontrolle über Daten behalten kann, die unter dritten geteilt werden. Fraglich ist, ob Unternehmen Interesse daran haben, Kundeninformationen, die möglicherweise unter großem Aufwand erstellt worden sind, tatsächlich unter die Kontrolle der Kunden selbst zu stellen, anstatt sie direkt an Partner in der Föderation weiterzuleiten. Für nicht-kommerzielle Organisationen und themenorientierte Online-Communities wäre diese Möglichkeit der nutzerkontrollierten Identitätsföderation eine sinnvolle Kombination der Konzepte.

Weitere Integration von Datenschutzmechanismen

Für die prototypische Umsetzung des in dieser Arbeit vorgestellten Konzeptes wurde nur am Rande auf Datenschutz- und Datensicherungsmechanismen geachtet. Ein System für den realen Einsatz mit sicherheitskritischen Daten müsste daher mehr Wert auf diesen Aspekt legen. Zum einen wäre eine Sicherung der Daten im Speicher des Identitätsmanagement-Systems ratsam, beispielsweise durch die Verschlüsselung der Datei oder einer Zugriffskontrolle der Datenbank. Im vorliegenden Prototypen ist der Zugriff lediglich durch den Zugriff auf das Compu-

tersystem beschränkt: Besteht kein Passwortschutz für den User-Account, unter dem die Daten gespeichert sind, so kann jeder, der das Computersystem bedient, alle Identitätsdaten einsehen und sogar einsetzen. Die Integration einer Zugriffskontrolle oder sogar die Auslagerung der Daten auf ein externes Speichermedium ist jedoch mit wenig Aufwand möglich.

Desweiteren ist die Identitätsdatenkommunikation derzeit nur verschlüsselt, nicht jedoch versteckt. Ohne die Integration eines Anonymisierungsdienstes können Dritte die Kommunikationsvorgänge beobachten. Eine Entschlüsselung der Datenströme ist zwar derzeit technisch nur unter unverhältnismäßig großem Aufwand möglich, im Sinne der Privatsphäre und des unbeobachteten Kommunizierens sollte jedoch ein MIX-System oder ähnliches integriert werden. Auch dies ist mit geringem Aufwand möglich, indem der Sitzungskanal zum Beispiel an den in Java vorliegenden MIX-Proxy JAP[2] angebunden wird (vgl. Unterabschnitt 3.4.2).

Untersuchung der Implikation von Identitätsverschmelzungen

In bestimmten Situation kann es wünschenswert oder notwendig sein, Teile eines Attributbaumes einer Teil-Identität in einer anderen, getrennten Teil-Identität mitzubenutzen, zum Beispiel um Wiederholungen und damit die Gefahr von Inkonsistenzen zu vermeiden. Die Implikation dieses Vorgangs auf die Zugriffsbeschränkung der Attribute sowie die mögliche Identifizierung des Eigentümers beider Teil-Identitäten sollten weitergehend untersucht werden. Durch die mögliche Komplexität des Datenmodells beinhaltet dieser Fragenkomplex viele unentscheidbare Probleme.

Evaluation

Eine Evaluation der Auswirkung einer Identitätsmanagement-Infrastruktur auf verschiedene Kommunikationssysteme im größeren Umfang stellt eine sehr große Herausforderung dar. Um Ergebnisse zu erhalten, die nicht von einem synthetischen Testaufbau in einer isolierten Testgruppe stammen und daher realitätsfern sind, müsste ein langfristiger Test mit Probanden in möglichst vielen verschiedenen Gesellschaftsgruppen (die möglicherweise sogar global verteilt sind) unter Berücksichtigung der Effekte mehrerer identitäts-angereicherter Kommunikationsanwendungen durchgeführt werden, während parallel eine ebenso aufgebaute Testgruppe ähnliche Anwendungen einsetzt, welche eben nicht auf das Identitätsmanagement-System aufsetzt. Der äußert aufwändige und komplexe Aufbau dieses Evaluations-Szenarios lässt begründeten Zweifel zu, ob eine solche Evaluation überhaupt möglich ist und verlässliche Ergebnisse liefern kann. Insofern beschränkt sich die Möglichkeit zur Evaluation eventuell auf verschiedene Bedienschnittstellen oder einzelne Kommunikationsanwendungen.

[2] `http://anon.inf.tu-dresden.de`

10.3. Schlusswort

Das Fernziel der umfassenden Integration von Identitätsmanagements in das digitale Kommunikationsleben ist auch durch diese Arbeit noch nicht erreicht. Die im vorigen Abschnitt erwähnten weiteren Forschungsmöglichkeiten zeigen erste Schritte auf dem weiteren Weg zu diesem Ziel auf. Dennoch bietet der derzeitige Stand der Forschung und Entwicklung einen guten Ausgangspunkt. Mit dieser Arbeit wurde die Situation zusammengefasst, der Blick auf einen bisher wenig beachteten Bereich geschärft und damit auch die Möglichkeit geschaffen, den Weg zur allgegenwärtig verfügbaren Identitätsinfrastruktur auf allgemeinerer Basis weiterzugehen. Der in dieser Arbeit vorgestellte Prototyp eines selbstdarstellungsorientierten Identitätsmanagement-Systems zeigt, wie dezentrale Strukturen die praktische Weiterentwicklung der Systeme fördern können. Letztendlich werden die Anwendungen und Dienste, die auf Identitätsmanagement-Infrastrukturen aufbauen, den größten Einfluss darauf haben, in welche Richtung die Entwicklung getrieben werden wird. Daher ist der Entwurf und die Konzeption identitätsbasierter Dienste der entscheidende nächste Schritt in der Entwicklung des digitalen Identitätsmanagements.

Anhänge

Anhang A.

Ausgewählte Ontologien des onefC-Systems

In diesem Teil des Anhangs werden zwei Teil-Ontologien wiedergegeben, die für das onefC-System von Bedeutung sind oder als Beispiel für weitere Teil-Ontologien dienen können.

A.1. Die Kern-Ontologie des Identitätsmanagements in onefC

Es folgt eine Darstellung der Kern-Ontologie des onefC-Systems im RDFS/XML Format. Ein Teil daraus ist in Abbildung 8.2 als grafisches RDF gezeigt. Sie repräsentiert die Grundkonzepte des Aufbaus einer digitalen Identität und legt fest, welcher Art die ihr anhaftenden Attribute sein können. Neben den systeminhärenten Attributen, die auch gleich definiert werden, beschreibt diese Teil-Ontologie den Unterschied zwischen privaten, eingeschränkt zugreifbaren und öffentlichen Attributen. Während private Attribute nie und öffentliche Attribute immer herausgegeben werden können, sind eingeschränkt zugreifbare Attribute mit einer Einschränkung (*Restriction*) versehen, mittels derer der Anwender bei unbefugtem Zugriff gewarnt werden kann.

Listing A.1: Kern-Ontologie des onefC-Systems

```
<?xml version="1.0"?>

<!DOCTYPE RDF:RDF [
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
  <!ENTITY idm "http://www.jwsdot.com/rdf/ISE/IDM#">
]>

<RDF:RDF xmlns:RDF="&rdf;"
   xmlns:RDFS="&rdfs;"
   xmlns:IDM="&idm;">

  <RDFS:Class rdf:about="&idm;IdentityClass">
   <RDFS:comment>The Super Class for all Identity Data</RDFS:comment>
   <RDFS:label>Identity Data Class</RDFS:label>
```

```
  <RDFS:subClassOf RDF:resource="&rdfs;Class">
    Identity Data Class
  </RDFS:subClassOf>
 </RDFS:Class>
 <RDFS:Class rdf:about="&idm;PropertyClass">
  <RDFS:comment>The Super Class for all Properties</RDFS:comment>
  <RDFS:label>Property</RDFS:label>
 </RDFS:Class>
 <RDFS:Class rdf:about="&idm;SystemPropertyClass">
  <RDFS:comment>The Super Class for all System Properties</RDFS:comment>
  <RDFS:label>System Property</RDFS:label>
  <RDFS:subClassOf RDF:resource="&idm;PropertyClass"/>
 </RDFS:Class>
 <RDFS:Class rdf:about="&idm;UserPropertyClass">
  <RDFS:comment>The Super Class for all User Properties</RDFS:comment>
  <RDFS:label>User Property</RDFS:label>
  <RDFS:subClassOf RDF:resource="&idm;PropertyClass"/>
 </RDFS:Class>

 <RDFS:Class rdf:about="&idm;RestrictedPropertyClass">
  <RDFS:comment>
   The Super Class for all Restricted Properties.
   Restricted Properties are only sent to a peer after previous
   negotiation.
  </RDFS:comment>
  <RDFS:label>Restricted Property</RDFS:label>
  <RDFS:subClassOf RDF:resource="&idm;UserPropertyClass"/>
  <RDFS:subClassOf RDF:resource="&idm;Restriction"/>
 </RDFS:Class>

 <RDFS:Class rdf:about="&idm;RestrictionClass">
  <RDFS:comment>The Super Class for all Restrictions.</RDFS:comment>
  <RDFS:label>Restriction</RDFS:label>
 </RDFS:Class>

 <RDFS:Class rdf:about="&idm;PolicyClass">
  <RDFS:comment>The Super Class for all Policies.</RDFS:comment>
  <RDFS:label>Policy</RDFS:label>
 </RDFS:Class>

 <RDFS:Class rdf:about="&idm;PrivatePropertyClass">
  <RDFS:comment>
   The Super Class for all Private Properties. Private Properties must
   not be sent to any peer in any circumstance
  </RDFS:comment>
  <RDFS:label>Private Property</RDFS:label>
  <RDFS:subClassOf RDF:resource="&idm;UserPropertyClass"/>
 </RDFS:Class>


 <RDF:Property rdf:about="&idm;property">
  <RDFS:comment>
   The Super Property for all Identity Data Properties
  </RDFS:comment>
  <RDFS:label>IDM Property</RDFS:label>
  <RDFS:subPropertyOf RDF:resource="&rdfs;member">
   Identity Data Class
  </RDFS:subPropertyOf>
 </RDF:Property>
```

```
<IDM:IdentityClass rdf:about="&idm;DigitalIdentity">
 <RDFS:comment>A general Role Identity</RDFS:comment>
 <RDFS:label>Digital Identity</RDFS:label>
</IDM:IdentityClass>

<IDM:IdentityClass rdf:about="&idm;MeIdentity">
 <RDFS:comment>A Me Identity</RDFS:comment>
 <RDFS:label>Me Identity (yourself)</RDFS:label>
 <RDFS:subClassOf RDF:resource="&idm;DigitalIdentity"/>
</IDM:IdentityClass>


<IDM:IdentityClass rdf:about="&idm;SessionIdentity">
 <RDFS:comment>A Session Identity</RDFS:comment>
 <RDFS:label>Session Identity</RDFS:label>
 <RDFS:subClassOf RDF:resource="&idm;DigitalIdentity"/>
 <RDFS:subClassOf RDF:resource="&idm;Policy"/>
</IDM:IdentityClass>


<IDM:SystemPropertyClass rdf:about="&idm;Nick">
 <RDFS:comment>The Nick Name of a digital Identity</RDFS:comment>
 <RDFS:label>Nick</RDFS:label>
 <IDM:propertytype>shorttext</IDM:propertytype>
</IDM:SystemPropertyClass>


<IDM:SystemPropertyClass rdf:about="&idm;PubKey">
 <RDFS:comment>
  The Public Key of this digital Identity - the final unique identifier
  </RDFS:comment>
 <RDFS:label>Public Key</RDFS:label>
 <IDM:propertytype>longtext</IDM:propertytype>
</IDM:SystemPropertyClass>


<IDM:SystemPropertyClass rdf:about="&idm;identifiedBy">
 <RDFS:comment>identified by</RDFS:comment>
 <RDFS:label>IdentifiedBy</RDFS:label>
 <IDM:propertytype>text</IDM:propertytype>
</IDM:SystemPropertyClass>


<IDM:SystemPropertyClass rdf:about="&idm;PrivateKey">
 <RDFS:comment>A Private Property: PrivateKey</RDFS:comment>
 <RDFS:label>PrivateKey</RDFS:label>
 <IDM:propertytype>longtext</IDM:propertytype>
</IDM:SystemPropertyClass>

<IDM:SystemPropertyClass rdf:about="&idm;Policy">
 <RDFS:comment>The Policy of a Session Identity</RDFS:comment>
 <RDFS:label>Policy</RDFS:label>
 <IDM:propertytype>longtext</IDM:propertytype>
</IDM:SystemPropertyClass>

<IDM:SystemPropertyClass rdf:about="&idm;Restriction">
 <RDFS:comment>The Restriction of an Attribute</RDFS:comment>
```

```
  <RDFS:label>Restriction</RDFS:label>
  <IDM:propertytype>longtext</IDM:propertytype>
 </IDM:SystemPropertyClass>

 <IDM:SystemPropertyClass rdf:about="&idm;EvaluationResult">
  <RDFS:comment>The result of a policy evaluation</RDFS:comment>
  <RDFS:label>EvalResult</RDFS:label>
  <IDM:propertytype>longtext</IDM:propertytype>
 </IDM:SystemPropertyClass>


</RDF:RDF>
```

A.2. Einfache Beispielontologie für Namen

Eine sehr kurze Teil-Ontologie für Namen und Kreditkarteninformationen ist die Names-Ontology. An diesem Beispiel kann man erkennen, wie Teil-Ontologien auf die Kern-Ontologie aufsetzen. Man beachte, wie die Namensräume der Kern-Ontologie und der RDF Systeme eingebunden sind. Die Names-Ontologie ist eine rein zu Demonstrationszwecken entwickelte Teil-Ontologie und hat wenig Praxistauglichkeit. Sie wird im HTTP-Integrationsbeispiel zur Übertragung des Namens verwendet (vergleiche Abschnitt 9.2).

Listing A.2: Teil-Ontologie mit Elementen für Namen und Kreditkartennummern

```
<?xml version="1.0"?>

<!DOCTYPE RDF:RDF [
 <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
 <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
 <!ENTITY idm "http://www.jwsdot.com/rdf/ISE/IDM#">
 <!ENTITY idmnames "http://www.jwsdot.com/rdf/ISE/IDMNames#">
]>

<RDF:RDF
         xmlns:RDF="&rdf;"
  xmlns:RDFS="&rdfs;"
  xmlns:IDM="&idm;"
  xmlns:IDMNames="&idmnames;">


 <IDM:RestrictedPropertyClass rdf:about="&idmnames;Name">
  <RDFS:comment>The Name of the digital Identity</RDFS:comment>
  <RDFS:label>Name</RDFS:label>
  <IDM:propertytype>shorttext</IDM:propertytype>
  <IDM:P3PCategory>physical</IDM:P3PCategory>
 </IDM:RestrictedPropertyClass>

 <IDM:RestrictedPropertyClass rdf:about="&idmnames;Nick">
  <RDFS:comment>The Nick Name of a digital Identity</RDFS:comment>
  <RDFS:label>Nick</RDFS:label>
  <IDM:propertytype>shorttext</IDM:propertytype>
```

```
  <IDM:P3PCategory>online</IDM:P3PCategory>
 </IDM:RestrictedPropertyClass>

 <IDM:RestrictedPropertyClass rdf:about="&idmnames;Surname">
  <RDFS:comment>The Surname of a digital Identity</RDFS:comment>
  <RDFS:label>Surname</RDFS:label>
  <IDM:propertytype>shorttext</IDM:propertytype>
  <IDM:P3PCategory>physical</IDM:P3PCategory>
 </IDM:RestrictedPropertyClass>

 <IDM:RestrictedPropertyClass rdf:about="&idmnames;Gender">
  <RDFS:comment>The Gender of a digital Identity</RDFS:comment>
  <RDFS:label>Gender</RDFS:label>
  <IDM:propertytype>shorttext</IDM:propertytype>
  <IDM:P3PCategory>demographic</IDM:P3PCategory>
 </IDM:RestrictedPropertyClass>

 <IDM:RestrictedPropertyClass rdf:about="&idmnames;CreditCard">
  <RDFS:comment>
   The Credit Card Information of this digital Identity
  </RDFS:comment>
  <RDFS:label>Credit Card</RDFS:label>
  <IDM:propertytype>shorttext</IDM:propertytype>
  <IDM:P3PCategory>financial</IDM:P3PCategory>
  <IDM:P3PCategory>purchase</IDM:P3PCategory>
 </IDM:RestrictedPropertyClass>

    <IDM:RestrictedPropertyClass rdf:about="&idmnames;HomeAddress">
            <RDFS:comment>The Home Address</RDFS:comment>
            <RDFS:label>HomeAddress</RDFS:label>
   <IDM:P3PCategory>physical</IDM:P3PCategory>
    </IDM:RestrictedPropertyClass>

</RDF:RDF>
```

Anhang B.

Schnittstellenbeschreibung

In diesem Teil des Anhangs wird als Beispiel für die Schnittstellenbeschreibung im WSDL Format die Schnittstelle des Datenschutzdienstes wiedergegeben. Sie umfasst die Beschreibung der gültigen Methoden (*wsdl:operation*) sowie der dazugehörigen Parameter. Ebenso wird das *binding* an die SOAP Schnittstelle definiert. Der hier angeführte Code wurde mit Hilfe des Axis-Systems aus der Java-Schnittstellenbeschreibung generiert.

Listing B.1: Schnittstelle des Datenschutzdienstes in WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  targetNamespace="urn:onefC.appel"
  xmlns:impl="urn:onefC.appel"
  xmlns:intf="urn:onefC.appel"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns2="urn:onefC.appelevaluator"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
 <wsdl:types>
  <schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:onefC.appelevaluator">
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
   <complexType name="Error">
    <sequence>
     <element name="code" type="xsd:int"/>
     <element name="message" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <complexType name="EvalReturn">
    <sequence>
     <element name="behaviour" nillable="true" type="xsd:string"/>
     <element name="description" nillable="true" type="xsd:string"/>
     <element name="finalPolicy" nillable="true"
       type="apachesoap:Document"/>
     <element name="personaIdentifier" nillable="true" type="xsd:string"/>
     <element name="promptMessage" nillable="true" type="xsd:string"/>
     <element name="returnError" nillable="true" type="tns2:Error"/>
     <element name="ruleNumber" type="xsd:int"/>
     <element name="showPrompt" type="xsd:boolean"/>
    </sequence>
   </complexType>
```

```
  </schema>
  <schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:onefC.appel">
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
   <complexType name="ArrayOf_xsd_string">
    <complexContent>
     <restriction base="soapenc:Array">
      <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]"/>
     </restriction>
    </complexContent>
   </complexType>
  </schema>
 </wsdl:types>
   <wsdl:message name="evaluateStringsRequest">
      <wsdl:part name="in0" type="xsd:string"/>
      <wsdl:part name="in1" type="xsd:string"/>
      <wsdl:part name="in2" type="xsd:string"/>
      <wsdl:part name="in3" type="xsd:string"/>
   </wsdl:message>
   <wsdl:message name="evaluateAsStringRequest">
      <wsdl:part name="in0" type="xsd:string"/>
      <wsdl:part name="in1" type="xsd:string"/>
      <wsdl:part name="in2" type="xsd:string"/>
      <wsdl:part name="in3" type="xsd:string"/>
   </wsdl:message>
   <wsdl:message name="evaluateRequest">
      <wsdl:part name="in0" type="apachesoap:Document"/>
      <wsdl:part name="in1" type="apachesoap:Document"/>
      <wsdl:part name="in2" type="apachesoap:Document"/>
      <wsdl:part name="in3" type="xsd:string"/>
   </wsdl:message>
   <wsdl:message name="evaluateResponse">
      <wsdl:part name="evaluateReturn" type="tns2:EvalReturn"/>
   </wsdl:message>
   <wsdl:message name="evaluateAsStringResponse">
      <wsdl:part name="evaluateAsStringReturn"
        type="impl:ArrayOf_xsd_string"/>
   </wsdl:message>
   <wsdl:message name="evaluateStringsResponse">
      <wsdl:part name="evaluateStringsReturn" type="tns2:EvalReturn"/>
   </wsdl:message>
   <wsdl:portType name="IAppelWS">
      <wsdl:operation name="evaluateStrings"
        parameterOrder="in0 in1 in2 in3">
         <wsdl:input name="evaluateStringsRequest"
           message="impl:evaluateStringsRequest"/>
         <wsdl:output name="evaluateStringsResponse"
           message="impl:evaluateStringsResponse"/>
      </wsdl:operation>
      <wsdl:operation name="evaluateAsString"
        parameterOrder="in0 in1 in2 in3">
         <wsdl:input name="evaluateAsStringRequest"
           message="impl:evaluateAsStringRequest"/>
         <wsdl:output name="evaluateAsStringResponse"
           message="impl:evaluateAsStringResponse"/>
      </wsdl:operation>
      <wsdl:operation name="evaluate"
        parameterOrder="in0 in1 in2 in3">
         <wsdl:input name="evaluateRequest"
```

```
            message="impl:evaluateRequest"/>
         <wsdl:output name="evaluateResponse"
            message="impl:evaluateResponse"/>
      </wsdl:operation>
   </wsdl:portType>
   <wsdl:binding name="AppelEvalServiceSoapBinding" type="impl:IAppelWS">
      <wsdlsoap:binding style="rpc"
        transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="evaluateStrings">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="evaluateStringsRequest">
            <wsdlsoap:body
              use="encoded"
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
              namespace="urn:onefC.appel"/>
         </wsdl:input>
         <wsdl:output name="evaluateStringsResponse">
            <wsdlsoap:body
              use="encoded"
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
              namespace="urn:onefC.appel"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="evaluateAsString">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="evaluateAsStringRequest">
            <wsdlsoap:body
              use="encoded"
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
              namespace="urn:onefC.appel"/>
         </wsdl:input>
         <wsdl:output name="evaluateAsStringResponse">
            <wsdlsoap:body
              use="encoded"
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
              namespace="urn:onefC.appel"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="evaluate">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="evaluateRequest">
            <wsdlsoap:body
              use="encoded"
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
              namespace="urn:onefC.appel"/>
         </wsdl:input>
         <wsdl:output name="evaluateResponse">
            <wsdlsoap:body
              use="encoded"
              encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
              namespace="urn:onefC.appel"/>
         </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="IAppelWSService">
      <wsdl:port name="AppelEvalService"
        binding="impl:AppelEvalServiceSoapBinding">
         <wsdlsoap:address
          location="http://localhost:8080/axis/services/AppelEvalService"/>
```

```
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

Anhang C.

Sourcecodes von onefC-Beispielanwendungen

Die angeführten Sourcecode-Beispiele gehören zum onefC Prototypen und sind somit im CVS Repository des Moduls „ise-java“ im SourceForge-Projekt „onefC“[1] enthalten. Sie stehen unter der Lizenz GPL; Der Lizenztext ist aus Gründen der Übersichtlichkeit hier ausgelassen. Desweiteren wurden javadoc-Kommentare ausgelassen. Die ausgelassenen Teile können im WWW oder per CVS nachgelesen werden.

C.1. Anwendungsbeispiel: Chat mit der Java-Socket-API

Listing C.1: Sourcecode des TalkServers mit Socket API

```
package org.onefc.ise.example.socket;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;

public class TalkServer {

  private int port;
  private DataInputStream dis;
  private DataOutputStream dos;

  public TalkServer() {
    this.port = 6789;
    start();
  }

  public void start() {
    ServerSocket s;
    try {
      s = new ServerSocket(port);
```

[1] http://sourceforge.net/projects/onefc

```
      Socket socket = s.accept();

      // start reader thread
      dis = new DataInputStream
        (socket.getInputStream());
      TalkReader listener =
        new TalkReader(dis);
      listener.start();

      // start writer thread
      dos = new DataOutputStream
        (socket.getOutputStream());
      TalkWriter writer =
        new TalkWriter(dos);
      writer.start();

    } catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }

  }

  public static void main(String[] args) {
    TalkServer t = new TalkServer();
  }
}
```

Listing C.2: Sourcecode des TalkClients mit Socket API

```
package org.onefc.ise.example.socket;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.net.UnknownHostException;

public class TalkClient {

  private int port;
  private String server;
  private Socket socket;
  private DataInputStream dis;
  private DataOutputStream dos;

  public TalkClient(String server) {
    this.server = server;
    this.port=6789;
    start();
  }

  public void start() {
    try {
      socket = new Socket(server, port);

      // start the reader thread
      dis = new DataInputStream
        (socket.getInputStream());
```

```
      TalkReader t =
        new TalkReader(dis);
      t.start();

      // start the writer thread
      dos = new DataOutputStream
        (socket.getOutputStream());
      TalkWriter writer =
        new TalkWriter(dos);
      writer.start();
    } catch (UnknownHostException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    } catch (IOException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }

  }

  public static void main(String[] args) {
    if (args.length!=1)
      throw new RuntimeException
        ("Syntax: java TalkClient <server>");

    TalkClient t = new TalkClient(args[0]);
  }
}
```

Listing C.3: Sourcecode des TalkReaders

```
package org.onefc.ise.example.socket;

import java.io.DataInputStream;
import java.io.IOException;

/**
 * @author baier
 *
 * TODO To change the template for this generated type comment go to
 * Window - Preferences - Java - Code Style - Code Templates
 */
public class TalkReader extends Thread {
  private DataInputStream dis;

  public TalkReader(DataInputStream dis) {
    this.dis = dis;
  }

  public void run() {
    try {
      while(true) {
        System.out.println(dis.readUTF());
      }
    }
    catch (IOException e) {
      System.err.println(e);
    }
  }
```

```
}
```

Listing C.4: Sourcecode des TalkWriters

```
package org.onefc.ise.example.socket;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class TalkWriter extends Thread {
  private DataOutputStream dos;

  public TalkWriter(DataOutputStream dos) {
    this.dos = dos;
  }

  public void run() {
    while (true) {
      BufferedReader br =
        new BufferedReader(new InputStreamReader(System.in));
      try {
        dos.writeUTF(br.readLine());
      } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
      }
    }
  }

}
```

C.2. Das Chat-Beispiel mit onefC Session API

Die Änderungen, die zum Umstellen des Chat-Programms auf die onefC-Session-API notwendig sind, sehen auf den ersten Blick größer aus, als sie tatsächlich sind: Der gestiegene Umfang des Quelltextes ist nicht proportional zum gestiegenen Aufwand. Die `import`-Statements werden von modernen Entwicklungswerkzeugen (im onefC-Projekt wurde die Open-Source-IDE Eclipse[2] eingesetzt) automatisch hinzugefügt. Die eigentlichen Änderungen sind mit Quelltextkommentaren dokumentiert.

Listing C.5: Sourcecode des TalkServers mit onefC Session API

```
package org.onefc.ise.example.session;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.URI;
```

[2] `http://www.eclipse.org`

```
import java.net.URISyntaxException;
import java.nio.channels.ServerSocketChannel;

import org.onefc.ise.api.IdentityMessageListener;
import org.onefc.ise.api.RemoteVerificationFailedException;
import org.onefc.ise.api.ServerSocketTransport;
import org.onefc.ise.api.SessionChannel;
import org.onefc.ise.api.SessionChannelNotAvailableException;
import org.onefc.ise.api.SessionChannelServer;
import org.onefc.ise.api.SessionEncryptionException;
import org.onefc.ise.api.SessionNotYetNegotiatedException;
import org.onefc.ise.api.SocketTransport;
import org.onefc.ise.scep.SCEPConsoleInterruptedException;
import org.onefc.ise.scep.spi.TransportConnectFailedException;
import org.onefc.ise.transport.TransportSlot;

import com.hp.hpl.jena.rdf.model.Model;

public class TalkServer {

  private DataInputStream dis;
  private DataOutputStream dos;


  // server = "itcp://localhost:4736";
  private String server = ServerSocketTransport.DEFAULT_URI.toString();

  public TalkServer(String server) {
    if(server!=null){
      this.server = server;
    }
    start();
  }

  public void start() {

    // 1st: initialize a SessionChannelServer
    // to accept new session channels
    SessionChannelServer sessionServer =
      new SessionChannelServer();
    try {
      while(true) {
        SessionChannel channel = null;
        SocketTransport transport = null;
        ServerSocketTransport sT = null;
        TalkReader listener=  null;
        TalkWriter writer = null;
        try {
          // start listening on for session channels
          // duplicate listen is no problem, since it
          // gets multiplexed anyway
          sessionServer.listen(new URI(server));

          // accept a new session
          channel = sessionServer.accept();

          // create a new ServerSocketTransport for example
          // best before negotiation, so no request may arrive
          // whose referred session does not yet exist
```

```
          sT = new ServerSocketTransport(channel,new URI(server));

          // finally negotiate the session.
          while(!channel.negotiateBlocking(-1)) {
            try{ Thread.yield(); }
            catch (Exception e) {}
          }
          channel.registerIdentityMsgListener(this);

          // as long as the channel is esablished
          while(channel.isEstablished()) {

            // accept new transports for this session
            transport = sT.acceptSocketTransport();

            // launch worker thread - reader
            dis = new DataInputStream
              (transport.getInputStream());
            listener = new TalkReader(dis);
            listener.start();

            // launch worker thread - writer
            dos = new DataOutputStream
              (transport.getOutputStream());
            writer = new TalkWriter(channel, dos);
            writer.start();
          }
        } catch (InterruptedException e1) {
          e1.printStackTrace();
        } catch (IOException e) {
          e.printStackTrace();
        } catch (SessionNotYetNegotiatedException e) {
          e.printStackTrace();
        } catch (RemoteVerificationFailedException e) {
          e.printStackTrace();
        } catch (SessionEncryptionException e) {
          e.printStackTrace();
        } catch (SessionChannelNotAvailableException e) {
          e.printStackTrace();
        }finally {

          try { listener.join(); }
          catch(Exception e) {}
          System.out.println("thread stopped: "+listener);

          try { transport.close(); }
          catch(Exception e) { e.printStackTrace(); }
          try { channel.close(); }
          catch(Exception e) {e.printStackTrace(); }
          try { sT.close(); }
          catch(Exception e) {e.printStackTrace(); }
        }
      }
    } catch (URISyntaxException e) {
      e.printStackTrace();
    } finally {
      try {
        sessionServer.close();
      } catch (IOException e1) {
```

```
        e1.printStackTrace();
      }
    }

  }

  public static void main(String[] args) {
    if (args.length!=1) {
      TalkServer t = new TalkServer(null);
    }else{
      TalkServer t = new TalkServer(args[0]);
    }
  }

}
```

Listing C.6: Sourcecode des TalkClients mit onefC Session API

```
package org.onefc.ise.example.session;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.nio.channels.ServerSocketChannel;

import org.onefc.ise.api.IdentityMessageListener;
import org.onefc.ise.api.RemoteVerificationFailedException;
import org.onefc.ise.api.ServerSocketTransport;
import org.onefc.ise.api.SessionChannel;
import org.onefc.ise.api.SessionChannelNotAvailableException;
import org.onefc.ise.api.SessionChannelServer;
import org.onefc.ise.api.SessionEncryptionException;
import org.onefc.ise.api.SessionNotYetNegotiatedException;
import org.onefc.ise.api.SocketTransport;
import org.onefc.ise.scep.SCEPConsoleInterruptedException;
import org.onefc.ise.scep.spi.TransportConnectFailedException;
import org.onefc.ise.transport.TransportSlot;

import com.hp.hpl.jena.rdf.model.Model;

public class TalkServer {

  private DataInputStream dis;
  private DataOutputStream dos;


  // server = "itcp://localhost:4736";
  private String server = ServerSocketTransport.DEFAULT_URI.toString();

  public TalkServer(String server) {
    if(server!=null){
      this.server = server;
    }
    start();
  }

  public void start() {
```

```
    // 1st: initialize a SessionChannelServer
    // to accept new session channels
    SessionChannelServer sessionServer =
      new SessionChannelServer();
    try {
      while(true) {
        SessionChannel channel = null;
        SocketTransport transport = null;
        ServerSocketTransport sT = null;
        TalkReader listener=  null;
        TalkWriter writer = null;
        try {
          // start listening on for session channels
          // duplicate listen is no problem, since it
          // gets multiplexed anyway
          sessionServer.listen(new URI(server));

          // accept a new session
          channel = sessionServer.accept();

          // create a new ServerSocketTransport for example
          // best before negotiation, so no request may arrive
          // whose referred session does not yet exist
          sT = new ServerSocketTransport(channel,new URI(server));

          // finally negotiate the session.
          while(!channel.negotiateBlocking(-1)) {
            try{ Thread.yield(); }
            catch (Exception e) {}
          }
          channel.registerIdentityMsgListener(this);

          // as long as the channel is esablished
          while(channel.isEstablished()) {

            // accept new transports for this session
            transport = sT.acceptSocketTransport();

            // launch worker thread - reader
            dis = new DataInputStream
              (transport.getInputStream());
            listener = new TalkReader(dis);
            listener.start();

            // launch worker thread - writer
            dos = new DataOutputStream
              (transport.getOutputStream());
            writer = new TalkWriter(channel, dos);
            writer.start();
          }
        } catch (InterruptedException e1) {
          e1.printStackTrace();
        } catch (IOException e) {
          e.printStackTrace();
        } catch (SessionNotYetNegotiatedException e) {
          e.printStackTrace();
        } catch (RemoteVerificationFailedException e) {
          e.printStackTrace();
```

```
        } catch (SessionEncryptionException e) {
          e.printStackTrace();
        } catch (SessionChannelNotAvailableException e) {
          e.printStackTrace();
        }finally {

          try { listener.join(); }
          catch(Exception e) {}
          System.out.println("thread stopped: "+listener);

          try { transport.close(); }
          catch(Exception e) { e.printStackTrace(); }
          try { channel.close(); }
          catch(Exception e) {e.printStackTrace(); }
          try { sT.close(); }
          catch(Exception e) {e.printStackTrace(); }
        }
      }
    } catch (URISyntaxException e) {
      e.printStackTrace();
    } finally {
      try {
        sessionServer.close();
      } catch (IOException e1) {
        e1.printStackTrace();
      }
    }

  }

  public static void main(String[] args) {
    if (args.length!=1) {
      TalkServer t = new TalkServer(null);
    }else{
      TalkServer t = new TalkServer(args[0]);
    }
  }

}
```

Während der `TalkReader` im onefC-Beispielsystem unverändert vom Socket-API-System übernommen worden ist, wurden dem `TalkWriter` private Methoden zum Identitätsdatenaustausch hinzugefügt. Eingegebene Textzeilen werden vor der Übermittlung auf die Schlüsselwörter „request“ (für Identitätsdatenanfragen), „set“ (für Identitätsdatenaussagen) und „setns“ (zum Ändern der aktiven Teil-Ontologie) untersucht, woraufhin die jeweilige Aktion ausgeführt wird.

Listing C.7: Sourcecode des angepassten TalkWriters für identitätsbewusste Talk-Komponenten

```
package org.onefc.ise.example.session;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
```

```
import java.io.InputStreamReader;

import org.onefc.ise.api.SessionChannel;
import org.onefc.ise.api.SessionChannelNotAvailableException;
import org.onefc.ise.util.UUIDGen;

import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.vocabulary.*;

public class TalkWriter extends Thread {
  private DataOutputStream dos;
  private boolean stopped = false;
  private SessionChannel channel;
  private static final boolean DEBUG = false;
  private String namespace;

  public TalkWriter(SessionChannel channel, DataOutputStream dos) {
    this.dos = dos;
    this.channel = channel;
    setName("TalkWriter");
    // the default namespace for identity statements
    this.namespace = "http://www.jwsdot.com/rdf/ISE/IDMNames";
  }

  public void run() {
    try {
      while (true) {
        BufferedReader br =
          new BufferedReader(new InputStreamReader(System.in));
        String line = br.readLine();
        if (line == null) return;
        //ATTENTION: return statement here!

        if(line != null && line.length() > 8 && line.startsWith("/request"))
        {
          int i1 = line.indexOf("␣");
          int i2 = line.indexOf("␣",i1+1);
          String subject = line.substring(i1+1,i2);
          String propType = line.substring(i2+1);

          sendIDRequest(channel, subject, namespace + "#" + propType);
          continue;
        }

        if(line != null && line.length() > 4 && line.startsWith("/set"))
        {
          int i1 = line.indexOf("␣");
          int i2 = line.indexOf("␣",i1+1);
          String type = line.substring(i1+1,i2);
          String content = line.substring(i2+1);

          sendIDStatement(channel, namespace + "#" + type, content);
          continue;
        }

        if(line != null && line.length() > 6 && line.startsWith("/setns"))
        {
          int i1 = line.indexOf("␣");
          String ns = line.substring(i1+1);
```

```
          this.namespace = ns;
          continue;
        }

        dos.writeUTF(line);
      }
    } catch (IOException e) {
      e.printStackTrace();
      stopped = true;
      try { dos.close(); } catch(Exception e1) {}
    } catch (SessionChannelNotAvailableException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
    }
  }

  private void sendIDRequest(
      SessionChannel channel2,
      String subject,
      String propType) throws SessionChannelNotAvailableException {
    Model model = ModelFactory.createDefaultModel();
    String ms="http://www.jwsdot.com/ISE/rdf/IDM/request/IdentityMessage";
    String req="onefc:GETProperty-"+UUIDGen.gen()+"/request";

    if (subject.equals("local"))
      subject = channel.getSessionContext().getLocalSessionNick();
    if (subject.equals("remote"))
      subject = channel.getSessionContext().getRemoteSessionNick();

    model.add(
      model.createResource(req),
      model.createProperty(
       "http://www.w3.org/1999/02/22-rdf-syntax-ns#","type"),
      model.createResource(ms));

    model.add(
      model.createResource(req),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","type"),
      model.createLiteral("GetProperty"));

    model.add(
      model.createResource(req),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","session"),
      model.createResource(channel2.getSessionHandle().toString()));

    model.add(
      model.createResource(req),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","hasSubject"),
      model.createResource(subject));

    model.add(
      model.createResource(req),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","hasSrc"),
      model.createResource(SessionChannel.APP_NICK)
          );
```

```
    model.add(
      model.createResource(req),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","hasDest"),
      model.createResource(
        channel.getSessionContext().getLocalSessionNick())
          );

    model.add(
      model.createResource(req),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","ptype"),
      model.createResource(propType)
          );


    if(DEBUG)
      model.write(System.err);

    channel.sendIdMsg(model);
  }

  private void sendIDStatement(
      SessionChannel channel2,
      String propType,
      String content) throws SessionChannelNotAvailableException {
    Model model = ModelFactory.createDefaultModel();
    String ms="http://www.jwsdot.com/ISE/rdf/IDM/request/IdentityStatement";
    String stmt = "onefc:sendIDStatement-"+UUIDGen.gen()+"/statement";

    int i1 = propType.indexOf("#");
    String proptypeshort = propType.substring(i1+1);
    String contenturi = "onefc:"+UUIDGen.gen()+"#"+proptypeshort;

    model.add(
      model.createResource(stmt),
      model.createProperty(
       "http://www.w3.org/1999/02/22-rdf-syntax-ns#","type"),
      model.createResource(ms));

    model.add(
      model.createResource(stmt),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","type"),
      model.createLiteral("PropertyStatement"));

    model.add(
      model.createResource(stmt),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","session"),
      model.createResource(channel2.getSessionHandle().toString()));

    model.add(
      model.createResource(stmt),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","hasSubject"),
      model.createResource(
        channel.getSessionContext().getLocalSessionNick()));
```

```
    model.add(
      model.createResource(stmt),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","object"),
      model.createResource(contenturi));

    model.add(
      model.createResource(stmt),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","hasSrc"),
      model.createResource(SessionChannel.APP_NICK)
          );

    model.add(
      model.createResource(stmt),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","hasDest"),
      model.createResource(
        channel.getSessionContext().getLocalSessionNick())
          );

    model.add(
      model.createResource(stmt),
      model.createProperty(
       "http://www.jwsdot.com/ISE/rdf/IDM/requestproperties#","ptype"),
      model.createResource(propType)
          );

    // the content
    Resource nick =
      model.createResource(contenturi).addProperty(RDFS.label,content);

    if(DEBUG)
      model.write(System.err);

    channel.sendIdMsg(model);
  }

  public void shutdown() throws IOException {
    this.stopped = true;
    dos.close();
  }
}
```

Literaturverzeichnis

Die Literaturangaben sind alphabetisch nach den Namen der Autoren sortiert und entsprechen der DIN 1505. Bei mehreren Autoren wird nach dem ersten Autor sortiert. Mehrere Veröffentlichungen eines Autors sind chronologisch sortiert. Die Nummerierung der Angaben dient lediglich zur besseren Orientierung – Zitate im Text werden im Autor-Jahr-Format angegeben.

1 Abts 2003 Abts, Dietmar: *Aufbaukurs Java: Client/Server-Programmierung mit JDBC, Sockets, XML-RPC und RMI.* Braunschweig : Vieweg Verlag, 2003

2 Adkinson u. a. 2002 Adkinson, W. F. ; Eisenach, J. A. ; Lenard, T. M.: Privacy Online: A Report on the Information Practices and Policies of Commercial Web Sites / Progress and Freedom Foundation. Washington DC, 3 2002. – Forschungsbericht

3 Alonso u. a. 2004 Alonso, G. ; Casati, F. ; Kuno, H. u. a.: *Web Services - Concepts, Architectures and Applications.* Springer, 2004

4 Altheim u. a. 2001 Altheim, Murray ; Boumphrey, Frank ; Dooley, Sam ; McCarron, Shane ; Schnitzenbaumer, Sebastian ; Wugofski, Ted: Modularization of XHTML / World Wide Web Consortium. 2001. – W3C Recommendation

5 Baier und Kunze 2004a Baier, Tobias ; Kunze, Christian P.: Identity-Enriched Session Management. In: Lamersdorf, W. (Hrsg.) ; Tschammer, V. (Hrsg.) ; Amarger, Stéphane (Hrsg.): *Building the E-Service Society: E-Commerce, E-Business, and E-Government* IFIP (Veranst.), Kluwer Academic Publishers, 8 2004, S. 329–342

6 Baier und Kunze 2004b Baier, Tobias ; Kunze, Christian P.: Identity Management for Self-Portrayal. In: Deswarte, Yves (Hrsg.) ; Cuppens, Frédéric (Hrsg.) ; Jajodia, Sushil (Hrsg.) ; Wang, Lingyu (Hrsg.): *Information Security Management, Education and Privacy* IFIP (Veranst.), Kluwer Academic Publishers, 8 2004, S. 231–244

7 Baier u. a. 2004 Baier, Tobias ; Weinreich, Harald ; Wollenweber, Frank: Verbesserung von Social Navigation durch Identitätsmanagement. In: R. Keil-Slawik, G. S. (Hrsg.): *Mensch und Computer 2004: Allgegenwärtige Interaktion* Gesellschaft für Informatik (Veranst.), Oldenbourg Verlag, 2004, S. 189–198

8 Baier u. a. 2003 Baier, Tobias ; Zirpins, Christian ; Lamersdorf, Winfried: Digital Identity: How To Be Someone On The Net. In: Reis, Pedro I. Antonio palma dos (Hrsg.): *e-Society 2003* Bd. II. Lisbon, Portugal : IADIS Press, Juni 2003, S. 815–820

9 Bechhofer u. a. 2004 Bechhofer, Sean ; Harmelen, Frank van ; Hendler, Jim ; Horrocks, Ian ; McGuinness, Deborah L. ; Patel-Schneider, Peter F. ; Stein, Lynn A.: OWL Web Ontology Language Reference / World Wide Web Consortium (W3C). URL `http://www.w3.org/TR/2004/REC-owl-ref-20040210/`, 10 2004. – W3C Recommendation

10 Berners-Lee u. a. 1996 Berners-Lee, T. ; Fielding, R. ; Frystyk, H.: *RFC 1945: Hypertext Transfer Protocol — HTTP/1.0.* 1996. – URL `ftp://ftp.internic.net/rfc/rfc1945.txt`

11 Berners-Lee u. a. 2000 Berners-Lee, Tim ; Harmelen, Frank v. ; Horrocks, Ian ; Brickley, Dan ; Dean, Mike ; Decker, Stefan ; Fikes, Richard ; Hayes, Pat ; Heflin, Jeff ; McDermott, Drew ; Swick, Ralph R.: *DAML-ONT Initial Release.* 10 2000. – URL `http://www.daml.org/2000/10/daml-ont.html`

12 Berthold und Köhntopp 2001 Berthold, Oliver ; Köhntopp, Marit: Identity management based on P3P. In: *International workshop on Designing privacy enhancing technologies*, Springer-Verlag New York, Inc., 2001, S. 141–160. – ISBN 3-540-41724-9

13 Beutelspacher u. a. 2004 Beutelspacher, Albrecht ; Schwenk, Jörg ; Wolfenstetter, Klaus-Dieter: *Moderne Verfahren der Kryptographie.* 5. Wiesbaden : Friedr. Vieweg und Sohn Verlag, 1 2004. – ISBN 3-528-46590-5

14 Boger 1999 Boger, Marko: *Java in verteilten Systemen - Nebenläufigkeit, Verteilung, Persistenz.* dpunkt.verlag Heidelberg, 9 1999

15 Boswell u. a. 2002 Boswell, David ; King, Brian ; Oeschger, Ian ; Collins, Pete ; Murphy, Eric: *Creating Applications with Mozilla.* O'Reilly & Associates, Inc., 2002. – ISBN 0-596-00052-9

16 Brink und Tauber 2004 Brink, Linda van d. ; Tauber, James: *XML software - xslt engines.* 13 2004. – URL `http://www.xmlsoftware.com/xslt.html`

17 Brockhaus 1989 Brockhaus: *BROCKHAUS ENZYKLOPÄDIE in vierundzwanzig Bänden: Zwölfter Band Kir–LAG*. F.A. Brockhaus, 1989

18 Brose u. a. 2001 Brose, Gerald ; Vogel, Andreas ; Duddy, Keith: *Java programming with CORBA : advanced techniques for building distributed applications*. New York, USA : Wiley, 2001

19 Burkert 1997 Burkert, Herbert: *Technology and privacy: the new landscape*. Kap. Privacy-enhancing technologies: typology, critique, vision, S. 125–142, MIT Press, 1997. – ISBN 0-262-01162-X

20 Chadwick 1994 Chadwick, David: *Understanding X.500 the Directory*. Chapman & Hall, Ltd., 1994

21 Chaum 1981 Chaum, David: Untraceable electronic mail, return addresses, and digital pseudonyms. In: *Communications of the ACM* 24 (1981), Nr. 2, S. 84–90. – ISSN 0001-0782

22 Chaum 1985 Chaum, David: Security without identification: transaction systems to make big brother obsolete. In: *Communications of the ACM* 28 (1985), Nr. 10, S. 1030–1044. – ISSN 0001-0782

23 Clark 1999 Clark, James: XSL Transformations (XSLT) / W3C. URL `http://www.w3.org/TR/xslt`, 1999. – W3C Recommendation

24 Clarke 1994 Clarke, Roger: The Digital Persona and Its Application to Data Surveillance. In: *The Information Society* 10 (1994), 6, Nr. 2, S. 77–92. – URL `http://www.anu.edu.au/people/Roger.Clarke/DV/DigPersona.html`

25 Clauß und Köhntopp 2001 Clauss, Sebastian ; Köhntopp, Marit: Identity management and its support of multilateral security. In: *Computer Networks* 37 (2001), Nr. 2, S. 205–219. – ISSN 1389-1286

26 Conklin 1987 Conklin, Jeff: Hypertext: An Introduction and Survey. In: *IEEE Computer* 20 (1987), September, Nr. 9, S. 17–40

27 Corbate 1962 Corbate, F.J.: An experimental time-sharing system. In: *AFIPS*, 1962

28 Cranor u. a. 2002 Cranor, Lorrie ; Langheinrich, Marc ; Marchiori, Massima ; Presler-Marshall, Martin ; Reagle, Joseph: The Platform for Privacy Preferences 1.0 (P3P1.0) Specification / World Wide Web Consortium. URL `http://www.w3.org/TR/P3P/`, 2002. – W3C Specification

29 Cranor 2002 Cranor, Lorrie F.: *Web Privacy with P3P*. O'Reilly & Associates, Inc., 9 2002. – 344 S. – ISBN Add data for field: ISBN

30 Cranor und LaMacchia 1998 Cranor, Lorrie F. ; LaMacchia, Brian A.: Spam! In: *Communications of the ACM* 41 (1998), Nr. 8, S. 74–83. – ISSN 0001-0782

31 Dasgupta 1988 Dasgupta, Partha: *Trust: Making and Breaking Cooperative Relations*. Kap. Trust as a Commodity, S. 49–72, Department of Sociology, University Oxford, 1988. – URL `http://www.sociology.ox.ac.uk/papers/dasgupta49-72.pdf`

32 Díaz und Serjantov 2003 Díaz, Claudia ; Serjantov, Andrei: Generalising Mixes. In: Dingledine, Roger (Hrsg.): *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, Springer-Verlag, LNCS 2760, March 2003

33 Dierks und Allen 1999 Dierks, T. ; Allen, C.: *RFC 2246 – The TLS Protocol Version 1.0*. 1999

34 Diffie und Hellman 1976 Diffie, Whitfield ; Hellman, Martin E.: New Directions in Cryptography. In: *IEEE Transactions on Information Theory* IT-22 (1976), Nr. 6, S. 644–654. – URL `citeseer.ist.psu.edu/diffie76new.html`

35 Donath 1997 Donath, J.: Identity and Deception in the Virtual Community. In: Kollock, P. (Hrsg.) ; Smith, M. (Hrsg.): *Communities in Cyberspace: Perspectives on New Forms of Social Organization*. Berkeley : University of California Press, 1997. – URL `citeseer.nj.nec.com/article/donath97identity.html`

36 Döring 2003 Döring, Nicola: *Sozialpsychologie des Internet: die Bedeutung des Internet für Kommunikationsprozesse, Identitäten, soziale Beziehungen und Gruppen*. 2. vollständig überarbeitete Auflage. Hogrefe, 2003. – ISBN 3-8017-1466-7

37 Dourish und Chalmers 1994 Dourish, Paul ; Chalmers, Matthew: Running Out of Space: Models of Information Navigation. In: *Proceedings of Human Computer Interaction (HCI'94)*. Glasgow,UK, August 1994

38 Duden 1996 *Duden, Rechtschreibung der deutschen Sprache*. Bd. 21. Dudenredaktion, 1996. – ISBN 3-411-04013-0

39 Erikson 1956/1966 Erikson, E.H.: *Identität und Lebenszyklus*. Suhrkamp, Frankfurt/Main, 1956/1966

40 Fahrenholtz und Bartelt 2001 Fahrenholtz, Dietrich ; Bartelt, Andreas: *Towards a Sociological View of Trust in Computer Science*. Eighth Research Symposium on Emerging Electronic Markets (RSEEM). 10 2001

41 Fensel u. a. 1999 Fensel, Dieter ; Harmelen, Frank v. ; Horrocks, Ian: OIL: A Standard Proposal for the Semantic Web / Vrije Universiteit Amsterdam. URL `http://www.ontoknowledge.org/oil/downl/otk.del02.pdf`, 29 1999 (IST-1999-10132). – Forschungsbericht

42 Fielding u. a. 1999 Fielding, R. ; Gettys, J. ; Mogul, J. ; Frystyk, H. ; Masinter, L. ; Leach, P. ; Berners-Lee, T.: *RFC 2616 – Hypertext transfer protocol (HTTP).* 1999

43 Gamma u. a. 1995 Gamma, Erich ; Helm, Richard ; Johnson, Ralph: *Design Patterns. Elements of Reusable Object-Oriented Software*. Reading, Massachusetts : Addison-Wesley, 1995 (Addison-Wesley Professional Computing Series). – ISBN 0-201-63361-2

44 Gburzynski und Maitan 2004 Gburzynski, Pawel ; Maitan, Jacek: Fighting the spam wars: A remailer approach with restrictive aliasing. In: *ACM Trans. Inter. Tech.* 4 (2004), Nr. 1, S. 1–30. – ISSN 1533-5399

45 Gerd tom Markotten 2002 Gerd tom Markotten, Daniela: User-Centered Security Engineering. In: *Proceedings of the fourth EurOpen/USENIX Conference - NordU2002*, February 2002

46 Gerhards und Mende 2003 Gerhards, Maria ; Mende, Annette: ARD/ZDF-Offline-Studie 2003 – Offliner 2003: Stabile Vorbehalte gegenüber dem Internet. In: *Media-Perspektiven* (2003), August

47 Griffel 1998 Griffel, Frank: *Componentware.* dpunkt.verlag Heidelberg, 1998

48 Gruber 1993 Gruber, Thomas R.: A Translation Approach to portable Ontology Specifications / Stanford University. URL `ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-92-71.ps.gz`, 1993 (KSL 92-71). – Forschungsbericht

49 Hansen u. a. 2003 Hansen, Marit ; Krasemann, Henry ; Rost, Martin ; Genghini, Riccardo: Datenschutzaspekte von Identitätsmanagementsystemen. In: *Datenschutz und Daetnsicherheit* 27 (2003), S. 551–555

50 Henrisch 1976 Henrisch, Dieter: Identität und Objektivität: eine Untersuchung über Kants transzendentale Deduktion. In: *Sitzungsberichte der Heidelberger Akademie der Wissenschaften – Philosophisch-Historische Klasse* Bd. 1. Winter, 1976, S. 54ff

51 Hodges und Wason 2003 Hodges, Jeff ; Wason, Tom: Liberty Architecture Overview / Liberty Alliance Project. 2003 (1.1). – Forschungsbericht

52 Howard 1998 Howard, L.: RFC 2307 – An Approach for Using LDAP as a Network Information Service / Network Working Group. URL `http://www.faqs.org/rfcs/rfc2307.html`, 3 1998. – Forschungsbericht

53 Iannella 2001 Iannella, Renato: Representing vCard Objects in RDF/XML / World Wide Web Consortium. URL `http://www.w3.org/TR/vcard-rdf`, 2001. – W3C Recommendation

54 IMS-Study 2003 *Identity Management Systems (IMS): Identification and Comparison Study.* European Union, 2003. – URL `http://www.datenschutzzentrum.de/projekte/idmanage/study.htm`. – Independent Centre for Privacy Protection Schleswig Holstein and Studio Notarile Genghini

55 Jendricke 2002 Jendricke, Uwe: *Sichere Kommunikation zum Schutz der Privatsphäre durch Identitätsmanagement*, Albert-Ludwigs-Universität Freiburg i.Br., Wirtschafts- und Verhaltenswissenschaftliche Fakultät, Dissertation, 2002

56 Jendricke und Gerd tom Markotten 2000 Jendricke, Uwe ; Gerd tom Markotten, Daniela: Usability meets Security - The Identity-Manager as your Personal Security Assistant for the Internet. In: *Proceedings of the 16th Annual Computer Security Applications Conference*, URL `http://www.acsac.org/2000/papers/90.pdf`, 2000

57 Jendricke u. a. 2002 Jendricke, Uwe ; Kreutzer, Michael ; Zugenmaier, Alf: Mobile Identity Management / Institut für Informatik, Universität Freiburg. URL `ftp://ftp.informatik.uni-freiburg.de/documents/reports/report178/report00178.ps.gz`, 2002 (178). – Forschungsbericht. – 8 S. Workshop on Security in Ubiquitous Computing, UBICOMP 2002

58 Jensen und Potts 2004 Jensen, Carlos ; Potts, Colin: Privacy policies as decision-making tools: an evaluation of online privacy notices. In: *Proceedings of the 2004 conference on Human factors in computing systems*, ACM Press, 2004, S. 471–478. – ISBN 1-58113-702-8

59 Jones und Pittman 1982 Jones, E. E. ; Pittman, T. S.: Toward a general theory of strategic self-presentation. In: Suls, S. (Hrsg.): *Psychological perspectives on the self* Bd. 1. Lawrence Erlbaum, Hillsdale, NJ, 1982, S. 231–263

60 Kaulbarsch 2005 Kaulbarsch, Gordian: *Identitäten und ihre Schnittstellen auf Basis von Ontologien in einer dezentralen Umgebung*, Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Verteilte Systeme und Informationssysteme (VSIS), Diplomarbeit, 2 2005

61 Kerner 1995 Kerner, Helmut: *Rechnernetze nach OSI*. Addison-Wesley (Deutschland), 1995. – ISBN 3-89319-801-6

62 Kienzle und Elder 2003 Kienzle, Darrell M. ; Elder, Matthew C.: Recent worms: a survey and trends. In: *WORM'03: Proceedings of the 2003 ACM workshop on Rapid Malcode*, ACM Press, 2003, S. 1–10. – ISBN 1-58113-785-0

63 Koch 2001 Koch, Michael: Community-Support-Systeme. In: Schwabe, Gerhard (Hrsg.) ; Streitz, Norbert (Hrsg.) ; Unland, Rainer (Hrsg.): *CSCW-Kompendium*. Berlin : Springer Verlag, 2001, S. 286–296

64 Koch 2002 Koch, Michael: Global Identity Management to Boost Personalization. In: Schubert, Petra (Hrsg.) ; Uwe Leimstoll, Petra und (Hrsg.): *Proceedings of the Ninth Research Symposium on Emerging Electronic Markets*. Basel, Switzerland, September 2002, S. 137–147. – URL `citeseer.nj.nec.com/545201.html`

65 Koch 2003 Koch, Michael: Community Support in Universities–, The Drehscheibe Project. In: Huysman, M. (Hrsg.) ; Wenger, E. (Hrsg.) ; Wulf, V. (Hrsg.): *Proc. Intl. Conf. on Communities and Technologies (CT2003)*. Amsterdam, Niederlande : Kluwer Academic Publishers, September 2003, S. 445–464

66 Koch 2004 Koch, Michael: Building Community Mirrors with Public Shared Displays. In: *Proc. eChallenges e-2004 Conference*. Vienna, Austria, 2004

67 Koch und Wörndl 2001 Koch, Michael ; Wörndl, Wolfgang: Community Support and Identity Management. In: *Proc. Europ. Conference on Computer-Supported Cooperative Work (ECSCW2001)*. Bonn, Deutschland, September 2001, S. 319–338. – URL `citeseer.nj.nec.com/koch01community.html`

68 Kohl und Neuman 1993 Kohl, J. ; Neuman, C.: RFC 1510 – The Kerberos Network Authentication Service (V5) / Network Working Group. RFC Editor, 1993. – Forschungsbericht

69 Köhntopp und Pfitzmann 2001 Köhntopp, Marit ; Pfitzmann, Andreas: Informationelle Selbstbestimmung durch Identitätsmanagement. In: *it+ti Informationstechnik und Technische Informatik*. IT-Sicherheit. Elsevier North-Holland, Inc., 9 2001, S. 227–235

70 Kormann und Rubin 2000 Kormann, David P. ; Rubin, Aviel D.: Risks of the passport single signon protocol. In: *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, North-Holland Publishing Co., 2000, S. 51–58

71 Kunze 2003 Kunze, Christian P.: *Digitale Identität und Identitäts-Management*. Hamburg, Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Verteilte Systeme und Informationssysteme (VSIS), Diplomarbeit, 2003

72 Kuri 2004 Kuri, Jürgen: *Novell: Linux und Identity Management*. September 2004. – URL `http://www.heise.de/newsticker/meldung/50993`

73 Li u. a. 2003 Li, Zichen ; Zhang, Junmei ; Kou, Weidong: The Unlinkability of Randomization-Enhanced Chaum's Blind Signature Scheme. In: *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, IEEE Computer Society, 2003, S. 244.2. – ISBN 0-7695-1926-1

74 Liberty Alliance 2003 Identity Systems and Liberty Specification Version 1.1 Interoperability / Liberty Alliance Project. URL `http://www.projectliberty.org`, 2003. – Forschungsbericht

75 Loshin und McCarthy 2000 Loshin, Pete ; McCarthy, Bill: *Big Book of Lightweight Directory Access Protocol (Ldap) Rfcs*. Academic Press, Inc., 2000. – ISBN 0124558437

76 Luft und Ingham 1955 Luft, Joseph ; Ingham, Harry: The Johari Window: a graphic model for interpersonal relations / UCLA, Western Training Lab. 1955. – Forschungsbericht

77 Luhmann 1989 Luhmann, Niklas: *Vertrauen. Ein Mechanismus zur Reduktion sozialer Komplexität*. 3. Stuttgart : Ferdinand Enke Verlag, 1989

78 Lupu und Sloman 1997 Lupu, Emil C. ; Sloman, Morris: Towards A Role-Based Framework for Distributed Systems Management. In: *Journal of Network and Systems Management* 5 (1997), Nr. 1, S. 5–30. – ISSN 1064-7570

79 Machilek u. a. 2004 Machilek, Franz ; Schütz, Astrid ; Marcus, Bernd: Selbstdarsteller oder Menschen wie du und ich? In: *Zeitschrift für Medienpsychologie* (2004)

80 Madsen 2004 Madsen, Paul: *Executive Overview of the Security Assertions Markup Language (SAML) v2.0*. 11 2004. – URL `www.oasis-open.org`

81 Manola und Miller 2004 Manola, Frank ; Miller, Eric: RDF Primer / World Wide Web Consortium. 2004. – W3C Recommendation

82 Marshall und Shipman 2003 Marshall, Catherine C. ; Shipman, Frank M.: Which semantic web? In: *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, ACM Press, 2003, S. 57–66. – ISBN 1-58113-704-4

83 McBride u. a. 2002 McBride, Brian ; Wenning, Rigo ; Cranor, Lorrie: An RDF Schema for P3P / World Wide Web Consortium. URL `http://www.w3.org/TR/p3p-rdfschema/`, 2002. – W3C Note

84 McLuhan 1964 McLuhan, Marshall: *Understanding Media - The Extensions of Man*. Cambridge, Massachusetts, United States : MIT Press, 1964

85 Mead 1932 Mead, George H.: *Mind, Self and Society from the Standpoint of a Social Behaviorist*. Kap. The 'I' and the 'me', S. 173–178, University of Chicago, 1932

86 Mintert 2002 Mintert, Stefan (Hrsg.): *XML & Co.* ADDISON-WESLEY, 2002

87 Mitra 2004 Mitra, Nilo: SOAP Version 1.2, W3C Recommendation / World Wide Web Consortium. URL `http://www.w3.org/TR/2002/CR-soap12-part0-20021219/`, 2004. – Forschungsbericht

88 Mittelstraß 1984 Mittelstrass, Jürgen: *Enzyklopädie Philosophie und Wissenschaftstheorie 2*. Wissenschaftsverlag, 1984

89 Mont u. a. 2003 Mont, Marco C. ; Bramhall, Pete ; Pato, Joe: On Adaptive Identity Management: The Next Generation of Identity Management Technologies / Trusted Systems Laboratory, HP Laboratories Bristol. 2003. – Forschungsbericht

90 Nejdl u. a. 2000 Nejdl, Wolfgang ; Wolpers, Martin ; Capelle, Christian: The RDF Schema Specification Revisited. In: *Proceedings of Modellierung*, URL `http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2000/modeling2000/wolpers.pdf`, 7 2000

91 Norlin und Durand 2002 Norlin, Eric ; Durand, Andre: PingID: Federated Identity Management / PingID Network. 2002. – Whitepaper

92 Oaks und Gong 2002 Oaks, Scott ; Gong, Li: *JXTA in a Nutshell*. O'Reilly & Associates, Inc., 2002. – ISBN 059600236X

93 Papazoglou und Georgakopoulos 2003 Papazoglou, M. P. ; Georgakopoulos, D.: Service-oriented computing: Introduction. In: *Communications of the ACM* 46 (2003), Nr. 10, S. 24–28

94 Pfitzmann 2001 Pfitzmann, Andreas: Multilateral Security: Enabling Technologies and Their Evaluation. In: *Informatics - 10 Years Back. 10 Years Ahead.*, Springer-Verlag, 2001, S. 50–62. – ISBN 3-540-41635-8

95 Pfitzmann und Hansen 2004 Pfitzmann, Andreas ; Hansen, Marit: *Anonymity, Unobservability, Pseudonymity, and Identity Management – A Proposal for Terminaology v0.18.* 2004

96 Pfitzmann und Waidner 2002a Pfitzmann, Birgit ; Waidner, Michael: Privacy in browser-based attribute exchange. In: *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, ACM Press, 2002, S. 52–62. – ISBN 1-58113-633-1

97 Pfitzmann und Waidner 2002b Pfitzmann, Birgit ; Waidner, Michael: Token-based Web Single Signon with Enabled Clients / IBM Research. Rüschlikon, Schweiz, 2002. – Forschungsbericht

98 Preece 2000 Preece, Jenny: *Online Communities: Designing Usability and Supporting Socialbilty.* Chichester, England : John Wiley & Sons, Inc., 2000. – ISBN 0-471-805998

99 Raggett u. a. 1998 Raggett, D. ; Hors, A. L. ; Jacobs, I.: *HTML 4.0 specification.* 1998

100 Rennhard und Plattner 2003 Rennhard, Marc ; Plattner, Bernhard: Practical Anonymity for the Masses with Mix-Networks. In: *Proceedings of the IEEE 8th Intl. Workshop on Enterprise Security (WET ICE 2003).* Linz, Austria, June 2003

101 Rickert 2004 Rickert, Thoralf: *Integration von Datenschutzmechanismen in Identitäteninfrastrukturen*, Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Verteilte Systeme und Informationssysteme (VSIS), Diplomarbeit, 2004

102 Riedl 2004 Riedl, Reinhard: Rethinking Trust and Confidence in European E-Government. In: Lamersdorf, Winfried (Hrsg.) ; Tschammer, Volker (Hrsg.) ; Amarger, Stéphane (Hrsg.): *Building the E-Service Society: E-Commerce, E-Business, and E-Government* IFIP (Veranst.), Kluwer Academic Publishers, 2004, S. 89–108

103 Rosenberg u. a. 2002 Rosenberg, J. ; Schulzrinne, H. ; Camarillo, G. ; Johnston, A. ; Peterson, J. ; Sparks, R. ; Handley, M. ; Schooler, E.: RFC 3261 – SIP: Session Initiation Protocol / Network Working Group. 2002. – Forschungsbericht

104 Sack 2005 Sack, Alexander: *Sitzungsmanagement zur Unterstützung von Identitätsinfrastrukturen*, Universität Hamburg, Fachbereich Informatik, Verteilte Systeme und Informationssysteme, Diplomarbeit, 1 2005

105 Schlichter u. a. 1998 Schlichter, Johann H. ; Koch, Michael ; Xu, Chengmao: Awareness - The Common Link Between Groupware and Community Support Systems. In: Ishida, Toru (Hrsg.): *Community Computing and Support Systems*, Springer Verlag, Juni 1998, S. 77–93. – URL `citeseer.nj.nec.com/schlichter98awareness.html`

106 Schollmeier und Hermann 2003 Schollmeier, Rüdiger ; Hermann, Felix: Topology-Analysis of Pure Peer-to-Peer Networks. In: Irmscher, K. (Hrsg.) ; Fähnrich, K.-P. (Hrsg.): *Kommunikation in Verteilten Systemen (KiVS)* Gesellschaft für Informatik (Veranst.), Springer-Verlag Berlin, 2003, S. 359–370

107 Stoica u. a. 2004 Stoica, Ion ; Adkins, Daniel ; Zhuang, Shelley ; Shenker, Scott ; Surana, Sonesh: Internet indirection infrastructure. In: *IEEE/ACM Trans. Netw.* 12 (2004), Nr. 2, S. 205–218. – ISSN 1063-6692

108 Swick 1999 Swick, Ralph R.: Putting it together: RDF: weaving the web of discovery. In: *netWorker* 3 (1999), Nr. 2, S. 21–25. – ISSN 1091-3556

109 Tamma und Bench-Capon 2002 Tamma, Valentina ; Bench-Capon, Trevor: An ontology model to facilitate knowledge-sharing in multi-agent systems. In: *Knowl. Eng. Rev.* 17 (2002), Nr. 1, S. 41–60. – ISSN 0269-8889

110 Tanenbaum 1996 Tanenbaum, Andrew: *Computer Networks*. 3. New Jersey : Prentice-Hall, Inc., 1996. – ISBN 0-13-394248

111 Tanenbaum und van Steen 2003 Tanenbaum, Andrew ; Steen, Marten van: *Veteilte Systeme: Grundlagen und Paradigmen*. Pearson Studium, 2003. – ISBN 3-8273-7057-4

112 Tavani und Moor 2001 Tavani, Herman T. ; Moor, James H.: Privacy protection, control of information, and privacy-enhancing technologies. In: *SIGCAS Comput. Soc.* 31 (2001), Nr. 1, S. 6–11. – ISSN 0095-2737

113 Web Services Architecture 2004 Booth, David (Hrsg.) ; Haas, Hugo (Hrsg.) ; McCabe, Francis (Hrsg.) ; Newcomer, Eric (Hrsg.) ; Champion, Michael (Hrsg.) ; Ferris, Chris (Hrsg.) ; Orchard, David (Hrsg.): *Web Services Architecture*. 2 2004. – URL `http://www.w3.org/TR/ws-arch/`

114 Weinreich u. a. 2003 Weinreich, Harald ; Buchmann, Volkert ; Lamersdorf, Winfried: Scone: Ein Framework zur evaluativen Realisierung von Erweiterungen des Webs. In: K. Irmscher, K.-P. F. (Hrsg.): *Tagungsband Kommunikation in Verteilten Systemen - KiVS 2003*, Springer-Verlag Heidelberg, 2 2003, S. 31–42

115 Weiser 1993a WEISER, Mark: Some computer science issues in ubiquitous computing. In: *Communications of the ACM* 36 (1993), Nr. 7, S. 75–84. – ISSN 0001-0782

116 Weiser 1993b WEISER, Mark: Ubiquitous Computing. In: *IEEE Computer Hot Topics* (1993). – URL `http://sandbox.parc.com/hypertext/weiser/UbiCompHotTopics.html`

117 Wellman und Gulia 1999 WELLMAN, Barry ; GULIA, Milena: Net surfers don't ride alone. In: WELLMAN, Barry (Hrsg.): *Networks in the Global Village*. Boulder, CO : Westview Press, 1999, S. 331–366

118 Wikipedia 2004 *Wikipedia: Identische Abbildung*. 2004. – URL `http://de.wikipedia.org/wiki/Identische_Abbildung`. – Version vom 22. Aug 2004

119 Williams 1998 WILLIAMS, Tad: *Otherland, City of Golden Shadows*. Bd. 1. Orbit, 1998. – ISBN 1857236041

120 Wollenweber 2004 WOLLENWEBER, Frank: *Kollaborative Nutzung des World Wide Webs*, Universität Hamburg, Fachbereich Informatik, Arbeitsbereich Verteilte Systeme und Informationssysteme (VSIS), Diplomarbeit, 2004

121 Woodward u. a. 2002 WOODWARD, John D. ; ORLANS, Nicholas M. ; HIGGINS, Peter T.: *Biometrics and Strong Authentication*. McGraw-Hill, Inc., 10 2002. – ISBN 0-07-222227-1

Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Hamburg, den 23. Juni 2005

Tobias Baier