Fakultät für Informatik
der Technischen Universität München

Privatheit bei dezentraler Verwaltung von Benutzerprofilen

Wolfgang Wörndl

Fakultät für Informatik
der Technischen Universität München

Privatheit bei dezentraler Verwaltung von Benutzerprofilen

Wolfgang Wörndl

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. Florian Matthes
Prüfer der Dissertation:	1.	Univ.-Prof. Dr. Johann Schlichter
	2.	Univ.-Prof. Dr. Gunnar Teege, Universität der Bundeswehr München
	3.	Univ.-Prof. Dr. Martin Bichler

Die Dissertation wurde am 16.04.2003 bei der Technischen Universtität München eingereicht und durch die Fakultät für Informatik am 17.07.2003 angenommen.

Kurzfassung

Eine dezentrale Verwaltung von Benutzerprofilen ermöglicht die Wiederverwendung von persönlichen Daten für verschiedene Personalisierungs-Dienste, erzeugt aber auch Probleme hinsichtlich der Privatheit der Informationen. Benutzer brauchen Mechanismen, um den Zugriff auf ihre Daten kontrollieren und ein Identitätsmanagement durchführen zu können. Zunächst werden dazu in dieser Arbeit bestehende Ansätze auf die Eignung in dem betrachteten Szenario untersucht. Dazu werden Zugriffskontrolle, Privacy Enhancing Technologies und bestehende Anwendungen für Identitätsmanagement im Internet betrachtet, die jedoch nur Teilaspekte der betrachteten Problemstellung lösen können.

Darauf aufbauend wird daher ein Mechanismus für Autorisation von Benutzerprofilzugriffen erarbeitet, dessen Lösungsidee aus zwei Phasen besteht:

1. Aushandlung von Zugriffsrechten und Generierung eines „Access Tickets“
2. Zugriff auf die Benutzerdaten mit dem Access Ticket

Die Aushandlung der Zugriffsrechte erfolgt mittels vom Benutzer festgelegten Datenschutz-Präferenzen (Regeln) und Integration von Benutzerinteraktion. Die Zugriffsentscheidung kann z.B. abhängig davon gemacht werden, ob die Identität des Benutzers offenbart wird oder der Zugriff anonymisiert erfolgt. Das Ergebnis der Aushandlung ist ein so genanntes Access Ticket, das eine digital signierte Formalisierung von Zugriffsrechten in XML für Benutzerprofile darstellt. Das Access Ticket realisiert eine Abbildung der für den Zugriff auf Benutzerprofile wichtigen Aspekte wie z.B. dem Zweck eines Profilzugriffs. Dabei wird eine Erweiterung von Zugriffskontrolle um Datenschutz-Vokabular vorgenommen.

Der Mechanismus ermöglicht somit eine Verbindung von Techniken zur Verbesserung von Privatheit mit (XML-basierter) Zugriffskontrolle, zusammen mit neuen Aspekten. Zur Evaluierung des Ansatzes wird u.a. ein Systementwurf und Teil-Implementierung in einem bestehenden Projektumfeld zur Benutzerprofilverwaltung beschrieben, sowie die Umsetzbarkeit in Benutzerschnittstellen und eine Integrierbarkeit in bestehende Standards erläutert.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Angestellter des Lehrstuhls für Angewandte Informatik – Kooperative Systeme (Prof. Schlichter) der Technischen Universität München.

An erster Stelle möchte ich mich bei Herrn Prof. Dr. Johann Schlichter bedanken, der mir diese Arbeit ermöglicht und durch viele hilfreiche Anregungen zu deren Qualität beigetragen hat. Er hat mir genügend Freiraum bei der Bearbeitung gelassen, stand aber immer für konstruktive Unterstützung und Gespräche bereit. Herzlichen Dank gebührt auch den beiden weiteren Gutachtern, Herrn Prof. Dr. Gunnar Teege (Universität der Bundeswehr München) und Herrn Prof. Dr. Martin Bichler (Technische Universität München) für wertvolle Hinweise zu dieser Arbeit.

Des Weiteren möchte ich mich bei meinen Kollegen am Lehrstuhl Schlichter für die fruchtbare Zusammenarbeit und das hervorvorragende Arbeitsklima bedanken. Insbesondere gilt mein Dank Herrn Dr. Michael Koch, der durch viele kritische Anmerkungen und der Durchsicht einer Vorversion maßgeblich zum Gelingen der Arbeit beigetragen hat.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

„A free and democratic society requires respect for the autonomy of individuals, and limits the power of both state and private organizations to intrude on that autonomy ...“
Aus: Preamble To Australian Privacy Charter, 1994

1.1 Personalisierung, Benutzerprofile und Community Unterstützung

Die weltweite Vernetzung von Informations-Quellen führt u.a. dazu, dass Benutzer[1] Probleme haben, in der Fülle von Information, die für sie wichtige und relevante herauszufinden. Ein Erfolg versprechendes Konzept ist die Personalisierung von Information, wozu möglichst gute Benutzerprofile nötig sind. Ein Benutzerprofil enthält neben allgemeinen und demographischen Angaben über den Benutzer z.B. Interessensgebiete, Qualifikationen oder getätigte Transaktionen.

Es gibt mittlerweile im Internet auch viele Systeme, die Profile von Benutzern anlegen und versuchen, daraus personalisierte Web-Seiten oder Empfehlungen abzuleiten. Üblicherweise interagiert ein Benutzer dabei mit verschiedenen Diensten, die jeweils Informationen über ihn sammeln und speichern (siehe Abb. 1.1 nach [Dys02a]). Diese Verwaltung von Benutzerprofilen bei den nutzenden Diensten bringt allerdings einige Probleme mit sich [KLW01a, KLW01b]:

- Benutzer müssen sich explizit bei verschiedenen Diensten registrieren und sich die oftmals unterschiedlichen Anmeldedaten (i.d.R. Benutzername und Passwort) verschiedener Dienste merken.
- Benutzer müssen ihre Profilinformationen wie z.B. demographische Informationen und Interessen immer wieder angeben. Es gibt kaum Möglichkeiten, neue Information automatisch an verschiedene Dienste zu verteilen oder eine Synchronisation der Daten zu erreichen.
- Wenn Benutzerprofile an unterschiedlichen Stellen erfasst und verwaltet werden, haben Benutzer wenig Kontrolle und Übersicht darüber, welche Daten wo, wie und von wem verwaltet werden.

Eine mögliche Lösung ist es, Benutzerprofile in dezentralen Benutzerprofilagenten oder „ID Repositories“ [KoWö01] unter der Kontrolle des Benutzers zu speichern, so dass die gleiche Benutzerinformation für verschiedene Personalisierungsdienste wieder verwendet werden kann. Dieser Ansatz wird auch im Projekt *IMC/Cobricks* (Information Management for Communities / Bricks for

[1] Der Einfachheit halber werden in diesem Text nur die männlichen Formulierungen verwendet.

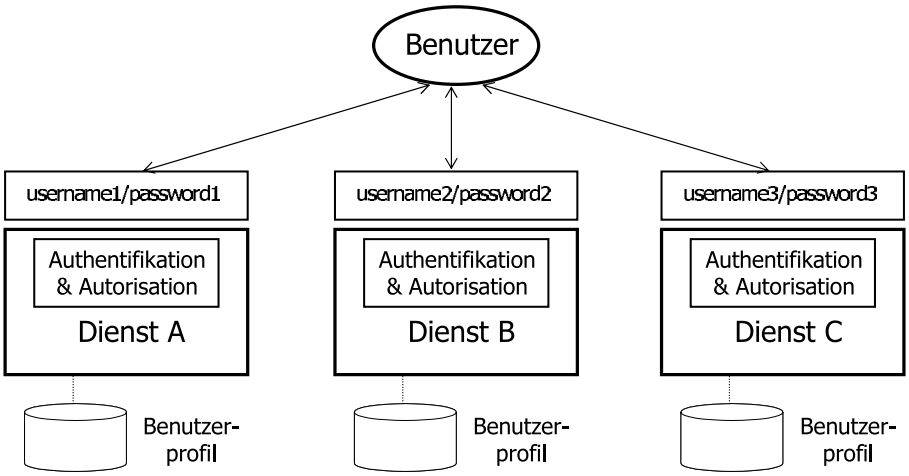

ABBILDUNG 1.1: Verwaltung von Profilen bei den nutzenden Diensten

supporting communities (siehe www11.in.tum.de/proj/imc/, [BKL+01, Koch02, KoWö01]) zur Unterstützung von Communities verfolgt. Communities sind lose gekoppelte Gruppen von Personen, die eine Gemeinsamkeit, z.B. ähnliche Interessen, haben. Dabei werden die in Benutzerprofilagenten verwalteten Informationen von Community Agenten verwendet. Ein Community Agent verwaltet Informationen einer Community, wie z.B. Beiträge von Mitgliedern und stellt Dienste wie Empfehlungsgenierierung bereit.

Community-Unterstützungssysteme sind eine wichtige Klasse von Personalisierungs-Anwendungen, weil damit das Finden von Wissensträgern und die direkte Interaktion von Mensch zu Mensch unterstützt werden kann. Dies kann einen Transfer auch von Wissen, welches nur schwer externalisierbar ist, ermöglichen. Die Systeme bieten allgemein folgende grundlegenden Funktionalitäten [KLW01a, KLW01b]:

- Bereitstellung eines Mediums für die direkte Kommunikation und den Austausch von Informationen und Kommentaren innerhalb des Kontextes der Community

- Aufdeckung und Visualisierung von Beziehungen (Mitgliedschaft in derselben Community, Existenz gemeinsamer Interessen). Dies kann Personen helfen, potentielle Kooperationspartner für direkte Interaktion zu entdecken (Awareness, Matchmaking, Expertensuche)

- Nutzung des Wissens über Beziehungen um (halb-)automatische Filterung und Personalisierung von Informationen durchzuführen

Das Ziel des Projektes Cobricks ist es, eine allgemeine Architektur mit erweiterbaren Schnittstellen und wiederverwendbaren Komponenten für Community-Unterstützungssysteme zu erstellen. Ein Teil davon ist die dezentrale Speicherung von Benutzerprofilen unter der Kontrolle des Benutzers. Diese Konzepte können auch auf andere Szenarien übertragen werden, z.B. (Agenten-basierter) E-Commerce [GMM99] und andere Dienste, die Benutzerprofile oder andere sensitive Daten nutzen.

1.2 Identitätsmanagement

Die Entwicklung des Internets basierte ursprünglich auf der fundamentalen Annahme, dass der Benutzer anonym bleibt [Abe02]. Diese Anonymität geht bei einer Personalisierung zumindest zum Teil verloren, da Informationen über den Benutzer benötigt werden. Wenn neben den Profildaten verschiedene Rollen eines Benutzers – wie z.B. „beruflich“ oder „privat“ – verwaltet werden, spricht man auch von digitalen Identitäten des Benutzers. Dies entspricht einer virtuellen Repräsentation eines Benutzers, die in elektronischer Interaktion eingesetzt werden kann. Ein Identitätsmanagementsystem soll es dem Benutzer ermöglichen, verschiedene Identitäten zu definieren und zu wählen, in welcher Rolle er gegenüber dem Kommunikationspartner auftritt und welche Informationen er über sich offenbart [Köh00].

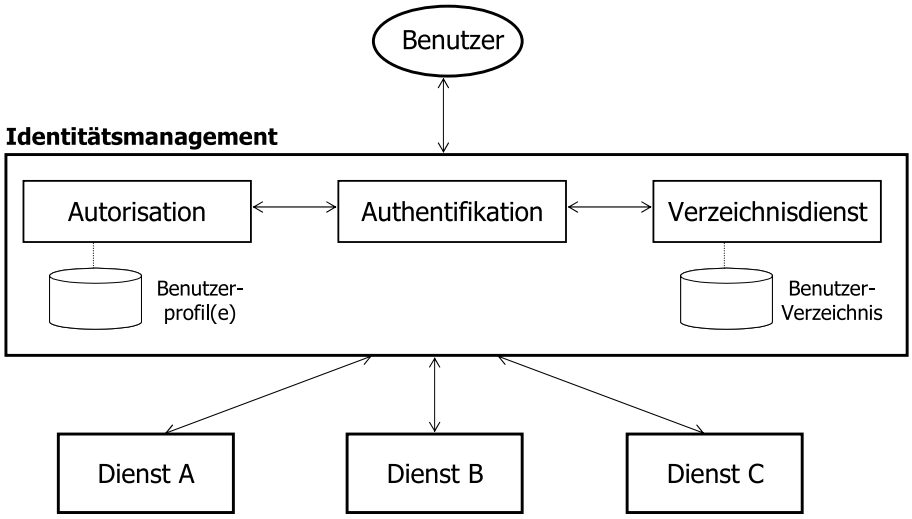

ABBILDUNG 1.2: Identitätsmanagement

Abb. 1.2 (nach [Dys02a]) zeigt die Situation einer dezentralen Verwaltung von Benutzerprofilen und Identitäten. Dabei werden die Profile und Informationen zur Authentifikation nicht mehr von den nutzenden Diensten, sondern getrennt davon in Identitätsmanagementsystemen gespeichert. Diese bilden somit eine separate Kontroll-Schicht zwischen dem Benutzer und den Diensten [Dys02a].

Kommerzielle Identitätsmanagementsysteme wie Microsoft .NET Passport oder das Open Source Liberty Alliance Projekt sind bereits in Einsatz oder werden entwickelt. Diese System fokussieren auf der Dienst-übergreifenden Authentifikation von Benutzern (Single Sign On) und bieten Dienste wie z.B. automatisches Ausfüllen von Web-Formularen oder die Abwicklung von Zahlungen im Internet an.

1.3 Motivation der Aufgabenstellung

Die bestehenden Systeme konzentrieren sich bisher auf die Komponente „Authentifikation“ in Abbildung 1.2, die z.B. von der Liberty Alliance Spezifikation gut abgedeckt wird. Allerdings gibt es noch kaum Ansätze für „Autorisation“, also der (verteilten) Zugriffskontrolle auf die Benutzerprofile. Manche Communities oder andere Dienste sind vertrauenswürdiger als andere für einen Benutzer und Benutzer wollen nicht alle Informationen im Profil für jede Community zur Verfügung stellen. Ein

Identitätsmanagement sollte den Benutzer in die Lage versetzen, die *Privatheit* (engl. privacy) ihrer persönlichen Daten sicherstellen und ihre Präferenzen bezüglich Datenschutz durchsetzen zu können. Privatheit kann in diesem Zusammenhang kurz definiert werden, als die Möglichkeit von Personen die Sammlung, Nutzung und Verbreitung von personenbezogenen Daten zu kontrollieren [MaAd02].

Der Entwurf eines Mechanismus zur Zugriffskontrolle auf dezentral verwaltete Benutzerprofile, also die Ausgestaltung der Komponente „Autorisation“ in Abb. 1.2, ist der Hauptbeitrag dieser Arbeit zur Verbesserung der Privatheit.

Die Präferenzen des Benutzers bezüglich der Herausgabe von Daten für Dienste können von verschiedenen Kontext-Attributen abhängig sein, insbesondere:

- Auf welchen Teil des Benutzerprofils wird zugegriffen? Sensitive Information wie Kreditkartennummern erfordern restriktivere Zugriffsbeschränkungen als öffentlich zugängliche Informationen wie eine Telefonnummer
- Welcher Dienst greift auf persönliche Daten zu? Benutzer möchten i.d.R. bekannten und ihnen vertrauenswürdigen Diensten mehr Zugriffrechte einräumen
- Der Zweck eines Zugriffs: Zum Beispiel ist es notwendig, einem Dienst zur Auslieferung einer Bestellung des Benutzers den Zugriff auf die Postanschrift zu gewähren, nicht jedoch zur Subskription eines Email-Newsletters
- Wie werden die Daten verwendet? Werden Daten z.B. öffentlich auf einem Web-Server verfügbar gemacht oder an andere Dienste weitergegeben?
- Der Kontext der Datenübertragung, z.B. ob die Daten offen über das Internet übertragen werden oder dies über eine gesicherte Verbindung erfolgt

Mit der fortschreitenden Vernetzung wird der Aspekt der Privatheit immer wichtiger [Cur02]. Dazu trägt auch eine Verbesserung der Technologie und des Verständnisses beim Nachverfolgen der Aktionen eines Benutzers und beim Auswerten der vielfach gesammelten persönlichen Daten bei. Umfragen zeigen, dass Benutzer über ihre Privatheit im Internet besorgt sind, was auch als Hindernis für E-Commerce angesehen wird [MaAd02]. Ein zuverlässiges Zugriffsschutzsystem könnte dazu führen, dass Benutzer mehr bereit sind, persönliche Informationen herauszugeben und sich dadurch die Qualität der Benutzerprofile verbessert.

In vielen Ländern sind zwar Datenschutzgesetze vorhanden oder werden entwickelt. Die rechtlichen Möglichkeiten stehen dabei aber erst am Anfang, insbesondere die (internationale) Durchsetzbarkeit von Datenschutzansprüchen ist noch problematisch. Das bedeutet, dass legislative und juristische Mittel alleine auf keinen Fall ausreichend sind, auch die technischen Möglichkeiten für den Benutzer müssen verbessert werden, wozu diese Arbeit beitragen soll.

1.4 Methodik und Aufbau der Arbeit

In dieser Arbeit wird also ein Zugriffsschutzmechanismus erarbeitet, der eine Autorisation bei dezentraler Verwaltung von Benutzerprofilen ermöglicht und somit die Privatheit der Benutzer verbessert. Zur Lösung der beschriebenen Ausgabenstellung wird ein konstruktivistischer Ansatz verfolgt. Beim Konstruktivismus werden Sachverhalte und Behauptungen aus vorgegebenen (oder bereits konstruierten) Elementen schrittweise erzeugt [Lor87, LuKö94]. Zunächst wird dazu in dieser Arbeit das zu lösende Problem identifiziert und gezeigt, dass bestehende Systeme dies bisher nicht lösen. Dann

erfolgt eine inkrementale Entwicklung einer Lösung der betrachteten Problemstellung, unter Berücksichtigung existierender Ansätze. Der erarbeitete Mechanismus wird begründet und abschließend evaluiert.

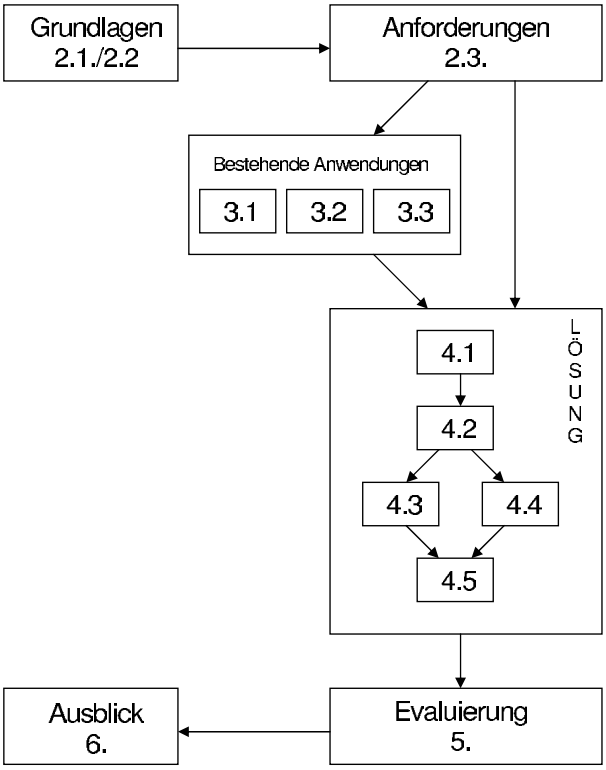

ABBILDUNG 1.3: Aufbau der Arbeit

In Abb. 1.3 ist der Aufbau der Arbeit anhand der wichtigsten Kapitel gezeigt.[2]

Zunächst werden in Kapitel 2 die nötigen Grundlagen geschaffen, insbesondere wird dargestellt, was Benutzerprofile sind, wie man sie verwalten kann und was Privatheit in diesem Zusammenhang bedeutet. Außerdem werden die genaueren Anforderungen der Aufgabenstellung in Abschnitt 2.3 erarbeitet. Im folgenden Kapitel 3 werden bestehende Systeme hinsichtlich der Aufgabenstellung evaluiert und bewertet. Dies gliedert sich in drei Teile, nämlich Modelle und Systeme zur Zugriffskontrolle (3.1), Privacy Enhancing Technologies (3.2) und Anwendungen für Identitätsmanagement im Internet (3.3).

Darauf aufbauend wird in Kapitel 4 ein eigener Mechanismus für das betrachtete Szenario erarbeitet. Dazu wird zunächst ein Überblick über den Mechanismus gegeben (4.1). Dies betrifft insbesondere eine Aufteilung der Zugriffskontrolle in zwei Phasen. Dann werden in 4.2 Access Request und Access Ticket, die den Kern der Arbeit ausmachen, erläutert. Anschließend werden die beiden Phasen des Ansatzes dargestellt: Aushandlung der Zugriffsrechte (4.3) und Datenzugriff (4.4). Es folgt ein Abschnitt über die Verwaltung verschiedener Identitäten eines Benutzers (4.5) und eine abschließende Bewertung.

Zur Evaluierung und Beurteilung der Lösung wird in Kapitel 5 zunächst die Umsetzbarkeit in

[2]Die Abbildung dient dabei nur zum groben Überblick über die Vorgehensweise und den Aufbau der Arbeit. Dies ist nicht ganz so streng linear, wie die Grafik suggerieren könnte.

Benutzerschnittstellen gezeigt. Anschließend wird eine Integration in einen aktuellen und relevanten Standard zur Benutzerprofilverwaltung und dem konkreten Projektumfeld gezeigt. Letzteres geschieht anhand eines Systementwurfes und der Beschreibung einer Teil-Implementierung. Des Weiteren wird eine mögliche Übertragbarkeit des Mechanismus dieser Arbeit auf andere Anwendungsbereiche untersucht. Die Arbeit schließt mit einer kurzen Zusammenfassung und einem Ausblick, der einige Punkte für zukünftige Forschungsbereiche, u.a. in Hinblick auf Benutzerschnittstellen, aufzeigt.

Kapitel 2

Grundlagen

„There was of course no way of knowing whether you were being watched at any giving moment. How often, or on what system, the Thought Police plugged in on any individual wire was guesswork. It was even conceivable that they watched everybody all the time. But at any rate they could plug in your wire whenever they wanted to.“
George Orwell, „1984“

In diesem Kapitel werden die Grundlagen dieser Arbeit beschrieben und die nötigen Begriffe geklärt. Dies betrifft insbesondere die genauere Erläuterung von Benutzerprofilverwaltung und Privatheit. Aus der Diskussion von Aspekten der Privatheit in dem betrachteten Zusammenhang ergeben sich auch die Anforderungen einer Autorisation, die die Basis der Diskussion in den folgenden Kapitel darstellen.

2.1 Benutzerprofile und deren Verwendung und Verwaltung

2.1.1 Benutzerprofile

Nachdem in dieser Arbeit Autorisation und Privatheit bei Verwaltung von Benutzerprofilen betrachtet werden, soll in diesem Abschnitt erläutert werden, was Benutzerprofile sind und wie sie verwendet werden können. Auch werden Möglichkeiten zur Speicherung und Verwaltung von Profilen insbesondere unter dem Gesichtspunkt der Privatheit diskutiert. Zunächst werden einige grundlegende Begriffe geklärt.

2.1.1.1 Identität und Benutzerprofile

Eine *Identität* ist aus soziologischer Sicht das „dauernde innere Sich-Selbst-Gleichsein, die Kontinuität des Selbsterlebens eines Individuums, die im wesentlichen durch die dauerhafte Übernahme bestimmter sozialer Rollen und Gruppenmitgliedschaften sowie durch die gesellschaftliche Anerkennung als jemand, der die betreffenden Rollen innehat bzw. zu der betreffenden Gruppe gehört, hergestellt wurde“ [Köh00].

Eine Identität beinhaltet zunächst mal die Identifizierung einer Person, z.B. durch den Namen, die Nummer des Personalausweises und/oder der Wohnanschrift eines Individuums [GGPS97]. Darüber hinaus enthält eine Identität auch weitere zum Teil mehr oder weniger dynamische Eigenschaften einer Person, wie z.B. politische Anschauungen, Interessen oder Funktionen. Diese einen Benutzer

charakterisierenden Eigenschaften nennt man – insbesondere im Umfeld einer elektronischen Speicherung und Verarbeitung – *Benutzerprofil*. Welche Eigenschaften oder Attribute ein Benutzerprofil enthalten kann, wird im folgenden Abschnitt 2.1.1.3 erläutert. Eine Identität in diesem Sinne wird in der Literatur z.T. auch als *Persona* bezeichnet [SoCr98, DGLP97, P3P02].

Eine Person kann unterschiedliche Rollen übernehmen (z.B. beruflich und privat) und somit mehrere Identitäten annehmen. Bei einer eher temporären oder willkürlich wählbaren Identität spricht man auch von einer *Pseudoidentität* oder einer *virtuellen Identität*. *Identitätsmanagement* impliziert die Möglichkeit einer Person, seine Identität oder Rolle, in der man gegenüber einem Kommunikationspartner auftritt, zu wählen [FeBe00]. Dazu gehört auch die Entscheidung, welche Teile des Benutzerprofils dem Partner offenbart werden. Der (oft unbewusste) Wechsel der Identität lässt sich auch in der realen Welt beobachten: Beispielsweise kann eine Person tagsüber eine Rolle als Angestellter einer Firma ausüben, danach anonym einige Einkäufe tätigen und schließlich am Abend die private Identität in der Familie einnehmen.

Verschiedene Identitäten können dadurch modelliert werden, indem das Profil eines Benutzers in einzelne Profile für jede Identität aufgeteilt wird, oder indem man die Attribute des Profils folgendermaßen gruppiert. Attribute eines Profils gelten dann entweder

- für alle Identitäten eines Benutzers (z.B. Größe und Gewicht einer Person)
- nur für eine oder mehrere Identitäten (z.B. Nummer des Firmenausweises)
- oder haben bei verschiedenen Identitäten unterschiedliche Ausprägungen (z.B. berufliche und private Interessen)

2.1.1.2 Pseudonymität

Ein *Pseudonym* ist ein Bezeichner für eine (virtuelle) Identität. Es soll die Zuordnung und Verkettung bestimmter Handlungen zu einer Person ermöglichen, ohne den bürgerlichen Namen des Benutzers aufzudecken, oder eine Zuordnung zu einem anderen Pseudonym der gleichen Person erlauben zu müssen. Ein Pseudonym ist also ein „Spitzname“ eines Benutzers in einem bestimmten Kontext und wird z.B. oft in Internet-Chats oder Diskussionsforen verwendet. Bei der Betrachtung von Pseudonymen sind insbesondere folgende drei Eigenschaften interessant (nach [Köh99a]):

- Zuordnung: Wie wird ein Pseudonym einer Person zugeordnet? Kann das Pseudonym frei gewählt werden? Ist es auf eine andere Person übertragbar?
- Verkettbarkeit: Wie können Pseudonyme verkettet werden? Das bedeutet, wie ist es ersichtlicht, dass mehrere Transaktion von dem gleichen Benutzer getätigt wurden. Wenn gar keine Verkettung möglich ist, spricht man von Anonymität.
- Aufdeckbarkeit: Wer kann wie die Zuordnung eines Pseudonyms zu einer Person aufdecken?

Diese Eigenschaften spielen eine Rolle, wenn man verschiedene Ausprägungen von Pseudonymität betrachtet. Man kann sich dabei eine Reihe von *Identitätsstufen*, einer Einteilung nach dem Grad der Anonymität, vorstellen [FiHü01, FeBe00, PfKö01]:

- Preisgabe der/einer Identität
- Persönliches Pseudonym, z.B. ein Spitzname

- Rollen-Pseudonym, z.B. anhand einer Rolle oder Aufgabe in einer Firma
- Ein Pseudonym pro Kommunikationspartner
- Völlige Anonymität, d.h. ein neues Pseudonym pro Transaktion

2.1.1.3 Modellierung von Benutzerprofilen

Es gibt verschiedene Ansätze, Benutzerprofile zu modellieren. Typischerweise werden die Attribute in einem Profil inhaltlich in einzelne Abschnitte gegliedert und hierarchisch aufgebaut. Ein Profil kann u.a. folgende Informationen enthalten:

- Identifikator(en) (z.B. ein X.500 Verzeichnisname)
- digitale(s) Zertifikat(e) des Benutzers
- demographische Informationen (z.B. Email-Adresse oder Postanschrift)
- Zahlungsinformation (z.B. Daten einer Kreditkarte)
- Beziehungen (mit anderen Benutzern)
- Bewertungen, Interessen, Qualifikationen, persönliche Präferenzen
- Transaktions-Historie (z.B. gekaufte Produkte oder besuchte Web Seiten)

Es gibt verschiedene Ansätze zur Modellierung von Benutzerprofilen. vCARD [HSD98] ist ein Standard, der eine elektronische Visitenkarte abbilden soll und enthält daher Informationen wie z.B. die private und berufliche Postanschrift. Das in Abschnitt 3.2.5 noch näher behandelte Platform for Privacy Preferences (P3P) Project [P3P02] enthält auch ein Datenformat für Benutzerinformationen. Ein Modell für E-Commerce ist z.B. CPExchange (www.cpexchange.org).

Der genaue Aufbau eines Benutzerprofils soll in dieser Arbeit nicht diskutiert werden. Für die Repräsentation eines Profils bietet sich eine Darstellung in der Extensible Markup Language (XML) [XML00] an, da damit insbesondere auch eine einfache und übersichtliche Strukturierung von Daten erreicht werden kann. Abbildung 2.1 zeigt ein Beispiel für ein Benutzerprofil in XML.

Ein Identifikator ist ein (in der Regel eindeutiger) Bezeichner für eine Identität, z.B. ein X.500 Verzeichnisname. Das hier gezeigte Profil ist inhaltlich in einzelne Kategorien gegliedert. In den XML-Auszeichnungselementen stehen die Bezeichnungen der Attribute bzw. der Kategorien von Profil-Attributen. Als Elemente sind die Ausprägungen der Attribute vorhanden. Die Attribute können z.B. auch Konfigurationseinstellungen einzelner Anwendungen sein.

Bestehende Ansätze, auch mit unterschiedlicher Ausrichtung oder Anwendungs-Domäne, lassen sich in ein entsprechendes XML-Dokument, wie im obigen Beispiel gezeigt, überführen [Ian01]. Die XML-Form ist nicht unbedingt geeignet für eine effiziente Speicherung des Profils[1], insbesondere bei sehr umfangreichen und/oder dynamischen Daten wie z.B. Web Zugriffslogs. Sie kann aber sehr gut zur Veranschaulichung des Konzepts eines Benutzerprofils dienen.

[1] Für eine effizientere Verarbeitung kann das Profil z.B. in einer relationalen Datenbank abgelegt werden.

```
<PROFILE>
  <IDENTIFICATION>
    <IDENTIFICATOR TYPE="X.500">
       @c=DE@o=TU-MUENCHEN@cn=WOERNDL</IDENTIFICATOR>
  </IDENTIFICATION>
  <DEMOGRAPHIC>
     <EMAIL>woerndl@in.tum.de</EMAIL>
     <POSTAL> ... </POSTAL>
  </DEMOGRAPHIC>
  <INTERESTS>
    <INTEREST>web applications</INTEREST>
    <INTEREST>travelling</INTEREST>
    <INTEREST>baseball</INTEREST>
  </INTERESTS>
  <RATINGS>
    <MOVIE>
      <NAME>Dances With Wolves</NAME>
      <RATING TYPE="percentage">95</RATING>
    </MOVIE>
    <BOOK>
      <NAME>Lonely Planet Australia Guide</NAME>
      <ISBN>123456789</ISBN>
      <RATING TYPE="textual">Very Good</TEXT>
    </BOOK>
  </RATINGS>
  <MISC>
    <BOOKMARK>http://www.traveller-world.com</BOOKMARK>
    <BOOKMARK>http://www.rosenheim89ers.de</BOOKMARK>
    <CONFIGURATION APP="http://drehscheibe.in.tum.de">
      <BACKGROUNDCOLOR>grey</BACKGROUNDCOLOR>
      <STARTPAGE>courses</STARTPAGE>
    </CONFIGURATION>
  </MISC>
  <TRANSACTIONS>
     <LOG AREA="web">
        <SITE>http://www11.in.tum.de</SITE>
        <DATE>04-07-2001 10:43</DATE>
     </LOG>
  </TRANSACTIONS>
</PROFILE>
```

ABBILDUNG 2.1: Beispiel Benutzerprofil

2.1.1.4 Abbildung mehrerer Identitäten

Wie in Abschnitt 2.1.1.1 erläutert wurde, kann ein Benutzer mehrere Identitäten annehmen, dies muss auch im Benutzerprofil abgebildet werden können. Dazu können in dem hier verwendeten XML-Format im <IDENTIFICATION> Abschnitt mehrere Identifikatoren angegeben werden (Abb. 2.2). Auch ist es möglich, für Identitäten Pseudonyme zu definieren.

```
<PROFILE>
  <IDENTIFICATION>
    <IDENTIFICATOR TYPE="X.500" ID="work">
       @c=DE@o=TU-MUENCHEN@cn=WOERNDL</IDENTIFICATOR>
    <IDENTIFICATOR TYPE="PassportNumber" ID="private">
       123456789</IDENTIFICATOR>
    <PSEUDONYM ID="private">
       nickname</PSEUDONYM>
  </IDENTIFICATION>
  <DEMOGRAPHIC>
     <EMAIL ID="work">woerndl@in.tum.de</EMAIL>
     <EMAIL ID="private">mail@wolfgang-woerndl.de</EMAIL>
     <HEIGHT ID="work,private" UNIT="cm">180</HEIGHT>
  </DEMOGRAPHIC>
  <INTERESTS ID="private">
    <INTEREST ID="work">web applications</INTEREST>
    <INTEREST>travelling</INTEREST>
    <INTEREST>baseball</INTEREST>
  </INTERESTS>
</PROFILE>
```

ABBILDUNG 2.2: Benutzerprofil mit verschiedenen Identitäten

Eine Zuordnung von Identitäten zu Profil-Attributen geschieht über ein XML-Attribut „ID“[2] (z.B. „work“) im Tag der Elemente. Dadurch werden alle Teile des Profils, die mit dem gleichen „ID“ Attribut gekennzeichnet sind, der gleichen Identität des Benutzers zugeordnet. Bei verschiedenen Identitäten in einem Element und einer übergeordneten Kategorie, gilt die speziellere Ausprägung. Es ist auch möglich ein Element mehreren Identitäten zuzuordnen. Wenn kein „ID“ vorhanden ist, gilt der Profileintrag für alle Identitäten.

2.1.2 Verwaltung von Benutzerprofilen

Es gibt verschiedene prinzipielle Möglichkeiten, Benutzerprofile zu verwalten, die im folgenden insbesondere auch hinsichtlich Privatheit diskutiert werden.

2.1.2.1 Erfassung und Speicherung der Profile bei den nutzenden Diensten

Im World Wide Web (WWW) werden heutzutage auf vielfache Weise Profile von (Web-)Benutzern erstellt und verwaltet. Zum Beispiel werten Internet-Shops wie amazon.com Aktionen von Benutzern aus, um auf dieser Grundlage Empfehlungen zu generieren. Aktionen können dabei z.B. die Web-Zugriffe eines Benutzers, der Kauf von Artikeln oder die explizite Bewertung von Produkten sein. Auch versuchen viele Web-Sites ihre Seiten oder die Bannerwerbung auf den Seiten gemäß den Präferenzen der Benutzer zu personalisieren, zum Teil geschieht dies bei Bannerwerbung auch übergreifend über mehrere Sites. Einzelne Web-Zugriffe können dabei u.a. mit Hilfe von *Cookies*[3]

[2] Dabei handelt es sich nicht um den XML-Attributtyp „ID“, sondern um ein XML-Attribut mit dieser Bezeichnung

[3] Ein Cookie ist ein kurzer Text, der vom Browser gespeichert und bei einem Besuch an den Server, der das Cookie gesetzt hat, zurückgeschickt wird. Damit kann ein Web-Server einen Benutzer wieder erkennen.

einem Benutzer – genauer ausgedrückt, einem Benutzer eines bestimmten Web-Browsers auf einem bestimmten Rechner – zugeordnet werden.

Benutzer werden auch aufgefordert, persönliche Daten explizit in Web-Formulare einzugeben, um z.B. Personalisierungs-Funktionen zu nutzen oder an Gewinnspielen teilzunehmen. Damit kann dann auch eine Zuordnung von beobachteten Daten anonymer Benutzer, wie z.B. Eingaben in Suchmaschinen im WWW, zu personenbezogenen Daten hergestellt werden. Es ist daher anzunehmen, dass für die meisten WWW-Benutzer ein mehr oder weniger detailliertes Profil bei verschiedenen Institutionen oder Firmen mit oder ohne Wissen bzw. Einverständnis des Benutzers vorhanden ist. Zum Teil sind diese Profile anonymisiert, zum Teil enthalten sie aber sicherlich auch den bürgerlichen Namen und andere personenbezogene Daten eines Benutzers [Les01].

Allerdings hat diese Server-seitige Speicherung bei den Diensten, die sie nutzen, inhärente Probleme, wie schon in der Einleitung dargestellt wurde:

- Profilinformationen können nur für denjenigen Dienst verwendet werden, der diese Daten gesammelt hat. Die Information über bei Barnes&Noble gekaufte Bücher kann nicht für Empfehlungen auch bei Amazon genutzt werden. Auch müssen Benutzer immer wieder neu die gleichen Informationen wie eine Email-Adresse eingeben und wenn sich diese ändert, kann sie nicht in einem Schritt allen betreffenden Diensten bekannt gemacht werden.

- Eine Server-seitige Speicherung von personenbezogenen Daten verursacht Probleme in Bezug auf Privatheit. Benutzer haben keine Kontrolle darüber, welche Informationen über sie von wem und warum gespeichert werden.

2.1.2.2 Server-seitige Profilverwaltung

Es gibt einige Ansätze, Benutzerprofile Server-seitig zu speichern und für mehrere Dienste wiederzuverwenden, z.B. Microsoft .NET Passport (www.passport.com) oder digitalme (www.digitalme.com) von Novell. Dabei werden Dienste wie Verwaltung von Benutzername/Passwort für verschiedene Web-Server oder Übermittlung von Adress-, Zahlungs- und anderen Informationen an E-Commerce Systeme angeboten.

Der Schwerpunkt liegt dabei auf einer Dienst-übergreifenden Authentifikation, einer Abwicklung von Zahlungen oder der Verwaltung von elektronischen Visitenkarten. Es wird kein komplettes Benutzerprofil modelliert und auch eine Erweiterung oder Anpassung von Profilfeldern ist nicht vorgesehen. Eine wichtige Funktion ist bei diesen Anwendungen das *„Single Sign On“ (SSO)*. Benutzer müssen sich nur einmal authentifizieren, z.B. bei einem Passport-Server und können dann verschiedene Passport-fähige Dienste nutzen, ohne sich jedes Mal neu anzumelden. Diese Systeme werden in Abschnitt 3.3 noch genauer betrachtet, insbesondere unter dem Gesichtspunkt der Privatheit für den Benutzer.

Das grundsätzliche Problem bei allen vorhanden, kommerziellen Systemen ist es, dass sie zu sehr auf die Vermarktung der Benutzerdaten und nicht auf den Schutz der Privatheit ausgerichtet sind. Ferner könnten Daten aus unterschiedlichen Quellen ohne Einverständnis des Benutzers zusammengeführt werden. Außerdem wird dem Benutzer die Möglichkeit genommen, Informationen nur teilweise herauszugeben, wenn er glaubt, ein Dienst ist nicht vertrauenswürdig. Selbst bei Zusicherung einer Speicherung nur zu einem vereinbarten Zweck kann es Probleme geben, wenn z.B. die Firma, die Profile verwaltet, Konkurs anmelden muss und vorher noch seine Kundendatei verkauft.

Ein Vorteil einer zentralisierten Speicherung auf einem Server wäre, dass ein Zugriffskontrollsystem und andere Sicherheitsmechanismen eventuell leichter zu realisieren wären. Allerdings stellt

dies auf der anderen Seite auch einen „Single Point of Attack“ dar und ist dadurch z.B. durch Denial of Service Angriffe leichter lahm zu legen als ein stärker verteiltes System.

2.1.2.3 Client-seitige Speicherung

Eine Möglichkeit, Benutzerprofile für verschiedene Dienste wieder zu verwenden, besteht darin, diese Client-seitig, also auf dem Rechner des Benutzers, abzulegen.

Dazu gibt es Werkzeuge wie z.B. Jotter oder Infospace (www.infospace.com). Infospace bietet neben einer (Server-seitigen) personalisierten Portal-Seite auch ein Client-seitigen „Personal Desktop Portal“ (vgl. Abb. 2.3). Dies ist eine adaptive Symbolleiste mit dessen Hilfe ein Benutzer sein Profil pflegen kann und Funktionalitäten wie Dokumenten-Verwaltung, einen Kalender, Integration mit Nachrichten-Diensten oder eine Initiierung von personalisierten Suchvorgängen nutzen kann.

ABBILDUNG 2.3: Infospace Personal Desktop

Diese Client-seitige Speicherung kann das Vertrauen des Benutzers in die Verwaltung seines Profils verbessern, da die Informationen auf seinem eigenen Rechner gespeichert sind, es gibt dabei aber auch einige Probleme. Die Profile sind nicht (auf einfache Weise) portabel: Informationen, die auf einem Rechner abgelegt sind, können nicht (ohne weiteres) auf einem anderen Rechner verwendet werden [MuSc00]. Auch ist trotz der Client-seitigen Speicherung nicht unbedingt absolute Kontrolle für den Benutzer gegeben, weil die Weitergabe und Verbreitung seiner Profilinformationen durch das Werkzeug im Einzelnen kaum überwacht werden kann.

2.1.2.4 Infomediaries

Eine Verwaltung von Benutzerprofilen durch eine dritte Partei im Auftrag des Benutzers wird durch Anwendungen realisiert, die man als *Infomediary* [HaSi99, Cra99] bezeichnet. Der Begriff stammt von Hagel/Singer:

> „In order for customers to strike the best bargain with vendors, they’ll need a trusted third party – a kind of personal agent, information intermediary, or infomediary – to aggregate their information with that of other consumers and to use the combined market power to negotiate with vendors on their behalf.“ ([HaSi99], S.19)

Ein Infomediary kann sowohl Server- als auch Client-seitig realisiert werden. Weniger entscheidend aus Sicht der Privatheit ist dabei der physikalische Ort der Speicherung, sondern die Frage, wer die Kontrolle über die Benutzerprofile ausübt. Also z.B. die Festlegung, welche Daten überhaupt gesammelt werden, wer die Zugriffsrechte vergibt und die Pflege und Löschung von Daten vornimmt. Dies sollte von Benutzer selber oder einer vertrauenswürdigen dritten Partei erfolgen. In [KoWö01, WöKo02] wird dazu der Ansatz von „ID-Repositories“ vorgestellt. Dabei werden die Benutzerprofile in verteilten, von den nutzenden Diensten unabhängigen, ID-Repositories verwaltet, was auch eine gute Skalierbarkeit und Ausfallsicherheit der Architektur ermöglicht.

Eine dezentrale Speicherung von Benutzerprofilen bietet aber noch keine Verbesserung der Privatheit der Benutzerinformationen per se, ist aber die Grundlage und Voraussetzung für ein leistungsfähiges Zugriffsschutzsystem unter der Kontrolle des Benutzers. Unabhängig von einer Server- oder Client-seitigen Speicherung braucht man ein Zugriffsschutzsystem, dessen Realisierungsmöglichkeiten in diesem Beitrag diskutiert werden.

2.1.3 Szenario: Agenten-basierter E-Commerce und Community Unterstützung

Nachdem jetzt erläutert wurde, was Benutzerprofile sind, und wie man sie speichern kann, soll jetzt die Verwendung der Profile angesprochen werden. Dazu wird zunächst kurz auf die Möglichkeiten der Nutzung von Benutzerprofilen für Personalisierung eingegangen, dann die Verwaltung von Profilen mit Software-Agenten besprochen und darauf aufbauend die beiden Anwendungsbereiche E-Commerce und Community-Unterstützungssteme vorgestellt. Schließlich wird das betrachtete Anwendungsszenario in diesem Kontext erläutert.

2.1.3.1 Personalisierung von Informationen

Ein typisches Anwendungsgebiet für Personalisierung von Informationen ist eine adaptive Web-Site. Adaptive Web-Sites versuchen ihre Seiten anhand von Präferenzen des Benutzers zu personalisieren. Dies kann entweder durch Beobachtung der Aktionen eines Benutzers oder durch explizite Angabe von Präferenzen geschehen. In beiden Fällen wird ein Benutzerprofil aufgebaut, das dann ausgewertet werden kann. Bei einer Personalisierung werden oftmals Empfehlungssysteme (engl. recommender systems) verwendet, die meist auf einer der folgenden Techniken basieren:

- Inhaltsbasiertes Filtern: Inhalte, wie z.B. Dokumente, werden mit Schlüsselwörtern versehen, die mit – explizit gemachten oder implizit abgeleiteten – Interessen eines Benutzers verglichen werden
- Kollaboratives Filtern [Koch01a]: Es wird versucht, Benutzer mit ähnlichen Interessen zu finden und abzugleichen. Dies wird z.B. bei Online-Buchhändlern wie amazon.com verwendet
- Regelbasiertes Filtern: Die Generierung von Empfehlungen geschieht auf Basis von Benutzerspezifischen Regeln

In allen Fällen werden verschiedene Informationen über den Benutzer aus dessen Profil benötigt.

2.1.3.2 Benutzerprofile und Software-Agenten

Wie in Abschnitt 2.1.2 erläutert, ist es vorteilhaft, Benutzerprofile dezentral und unabhängig von den Diensten, die sie verwenden, zu verwalten. Zur Realisierung möglichst unabhängiger Komponenten bietet sich die Verwendung von *Software-Agenten* an, da diese (mehr oder weniger) autonom – also unabhängig von anderen Komponenten oder Interaktion mit dem Benutzer – agieren.

(Software-)Agenten sind weiterhin durch folgende Eigenschaften charakterisiert, die sie von anderen, konventionellen Programmen unterscheiden:

- Proaktivität: Agenten können von sich aus Aktionen initiieren
- Kooperation: Agenten kooperieren oft mit anderen Agenten um ein (gemeinsames oder komplementäres) Ziel zu erreichen

- Adaption: Agenten können sich an veränderte Situation anpassen oder aus Erfahrungen lernen
- Kommunikation durch Austausch von Nachrichten

Diese Eigenschaften bedingen auch Anforderungen hinsichtlich Sicherheit und Privatheit. Das Agentenparadigma ermöglicht außerdem eine Modularisierung von Diensten in einer offenen Architektur und die lose Kopplung unabhängiger Komponenten [BBB+97, KLW01a]. Für die Kommunikation zwischen den Agenten wird eine Agent Communication Language (ACL) verwendet. Eine ACL wie FIPA ACL [FIPA99] oder KQML [FLM97] definiert ein Schema zum Austausch von Nachrichten zwischen Agenten hinsichtlich Syntax, Semantik und Pragmatik und basiert auf der Sprechakt-Theorie. Dies erlaubt die Verwendung einer Sprache zur Kommunikation zwischen unabhängig voneinander entwickelten Software-Agenten.

Es ist hier im Kontext dieses Artikels nicht entscheidend, dass die Speicherung und Verarbeitung von Benutzerprofilen durch Software-Agenten erfolgt, sondern nur durch autonome, dezentrale Komponenten, um eine Unabhängigkeit der Systemkomponenten zu garantieren. Eine zentralisierte Verwaltung sensibler Daten hat eventuell ganz andere Anforderungen, auf die hier nicht näher eingegangen wird.

Im Zusammenhang der Verwaltung von Benutzerprofilen sind insbesondere zwei Gruppen von Agenten-basierten Systemen interessant, die daher im folgenden etwas genauer betrachtet werden sollen:

- Agenten-basierter E-Commerce und
- Agenten-basierte Community-Unterstützungssysteme

Im dieser Arbeit wird das Agenten-Paradigma insoweit berücksichtigt, dass von einer dezentralen Verwaltung von Benutzerprofilen in von den nutzenden Diensten unabhängigen Komponenten ausgegangen wird.

2.1.3.3 Software-Agenten und Electronic Commerce

Software-Agenten können das Kaufen und Verkaufen von Produkten und Dienstleistungen im Internet unterstützen, wobei die Agenten in der Regel als Vermittler zwischen einem Käufer und Verkäufer auftreten, daher spricht man dann auch von *Agent-Mediated E-Commerce* [MGM99].

Agenten können verschiedene Phasen des elektronischen Handels unterstützen, z.B. Produktauswahl, Preisbestimmung oder auch die Festlegung von Liefermodalitäten. Ein wichtiger Aspekt ist dabei die Verhandlung. Agenten agieren autonom, verfolgen verschiedene Ziele und versuchen unabhängig voneinander ein möglichst günstiges Ergebnis zu erreichen. Durch eine Automatisierung von Verhandlung durch Agenten ist neben einer Optimierung von Marktabläufen auch Kostenreduzierung im E-Business möglich.

Ein Beispiel unter vielen Projekten zum Einsatz von „Agenten“ im E-Commerce ist *COGITO* („E-Commerce with Guiding Agents based on Personalized Interaction Tools“) [ThSt00]. Das Ziel des Projektes ist eine verstärkte Bindung von Kunden an E-Commerce-Anbieter durch Agentenbasierte Technologien. Ein Teil des Projektes ermöglicht eine Interaktion des Benutzers mit dem E-Commerce System in natürlicher, geschriebener Sprache mit Hilfe sogenannter „chatterbots“. Ein anderer Bereich von COGITO behandelt die Auswertung von Benutzerprofilen zur Generierung von Empfehlungen, wobei inhaltsbasiertes und kollaboratives Filtern kombiniert werden. Der Ansatz basiert auf „intelligent personalized agents“, die virtuelle Assistenten oder Berater für Benutzer darstellen. COGITO unterstützt damit in erster Linie die Produktauswahl.

Bei einem sinnvollen Einsatz von Agenten für E-Commerce sind oftmals Information über den Benutzer nötig, in dessen Auftrag er handelt, z.B. Präferenzen und Zahlungsinformationen, oder auch dessen Reputation. Dabei stellt sich insbesondere auch das Problem, die Privatheit der Profilinformationen sicherzustellen, da der Benutzer einen Teil der Kontrolle über sein Profil einem autonom agierenden Agenten anvertraut.

2.1.3.4 Community-Unterstützungssysteme

(Virtuelle) Communities bezeichnen Gruppen von Personen, die eine Gemeinsamkeit, z.B. ähnliche Interessen, haben. Eine genauere Definition findet sich bei Mynatt et.al.:

> „[A community is a social grouping which exhibit in varying degrees: shared spatial relations, social conventions, a sense of membership and boundaries, and an ongoing rhythm of social interaction.“ (aus: [MAIO97], S. 211)

Im Gegensatz zu einer „Gruppe“ oder einem „Team“ ist eine Community nur eine lose gekoppelte Menge von Menschen. In der Regel fehlt bei Communities ein gemeinsames Ziel und ein Gruppenbewusstsein. Communities können aber eine gute Quelle zur Beschaffung von Informationen sein, weil Wissen oft nur schwer externalisierbar ist und daher die direkte Interaktion mit Experten eine wichtige Rolle im Wissensmanagement spielen kann [KLW01a, KLW01b]. Dies soll durch *Community-Unterstützungssysteme* [Koch01b] realisiert werden, welche meist einen Teil der folgenden Funktionalitäten anbieten:

- Bereitstellung eines Mediums für direkte Interaktion zwischen Benutzern, z.B. durch ein Chat-System
- Verwaltung von Community-Informationen, z.B. Anmerkungen zu Publikationen in einer Forschergruppe
- Aufdecken und Visualisieren von Beziehungen zwischen Community-Mitgliedern, z.B. Finden eines Benutzers mit den gleichen Interessen
- Filterung und Personalisierung von Informationen, z.B. Generierung einer Liste von Produkten, die Benutzer mit ähnlichen Interessen für gut befunden haben

Im Projekt IMC/Cobricks (Information Management for Communities / Bricks for supporting communities [BKL+01, Koch02, KoWö01] werden Agenten-basierte Systeme zur Unterstützung von Communities untersucht. Dabei werden die in dezentralen Benutzerprofilagenten gespeicherten Informationen über Benutzer von *Community Agenten* verwendet. Ein Community Agent verwaltet dabei die Informationen einer Community, z.B. Beiträge von Mitgliedern, und stellt Dienste wie Empfehlungsgenerierung oder den Abgleich von Benutzern, die an einer vergleichbaren Aufgabe arbeiten, bereit.

Ein Beispiel einer Anwendung in dieser Architektur ist das *CommunityItemsTool* [KLW01a]. Es ermöglicht einen Austausch von Community-Informationen wie z.B. Bookmarks oder bibliographische Referenzen in einer Forscher-Community. Benutzer können u.a. die Referenzen in einer persönlichen Ordnerstruktur ablegen [LWKB00] oder Bewertungen abgegeben, wobei diese Benutzerinformation in einem Benutzerprofilagenten gespeichert werden.

2.1.3.5 Anwendungsszenario

Die erläuterten Anwendungen für Agenten-basierte Verwaltung von Benutzerprofilen lassen sich zu folgendem Szenario (Abb. 2.4) zusammenfassen.

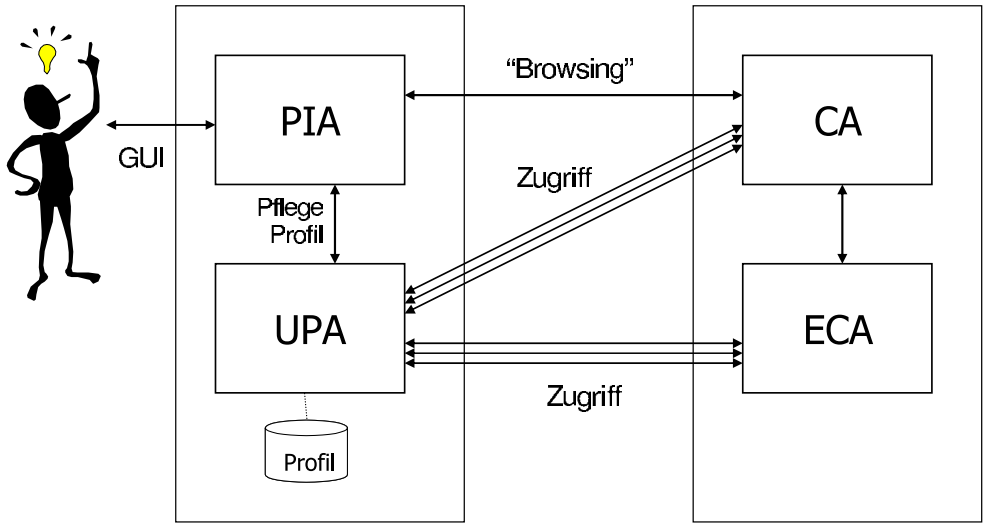

ABBILDUNG 2.4: Anwendungsszenario

Der Benutzer greift über ein „Personal Identity Assistant“ Werkzeug (PIA) auf die Komponenten zu. Der PIA kann ein erweiterter Web-Browser, ein Client-seitiges Infomediary (vgl. Abschnitt 2.1.2.4) oder ein ähnliches Werkzeug sein. Ein Community Agent (CA) oder E-Commerce Agent (ECA) stellt Personalisierungs- und andere Dienste für den Benutzer bereit. Dazu benötigen sie Informationen aus dem Profil des Benutzers, wozu sie mit dem Benutzerprofilagenten (User Profile Agent, UPA) kommunizieren. Die Dienstagenten (CA, ECA) können gegebenenfalls auch untereinander Informationen austauschen. Der UPA verwaltet das Profil des Benutzers, welcher seine Daten über eine Schnittstelle pflegt, die z.B. der PA zur Verfügung stellt. Für eine Kommunikation zwischen den Agenten ist nicht unbedingt eine Initiierung durch den Benutzer erforderlich, sondern es ist auch möglich, dass ein Dienstagent von sich aus personenbezogene Informationen anfordert, da es sich um autonome Komponenten handelt. Auch kann es sein, dass der UPA (vom Benutzer oder anderen, dazu autorisierten, Komponenten oder Agenten) geänderte Profilinformationen an die betreffenden Dienstagenten ohne Abfrage derer weitergibt. Wichtig bei dem Szenario ist, wie schon erwähnt, eine Trennung der Benutzerprofilverwaltung von den Diensten, die sie nutzen.

Das Szenario deckt die oben beschriebenen Anwendungen in den Bereichen Community Support und E-Commerce ab. Der Fokus dieser Arbeit ist es, dabei die Privatheit und insbesondere die Autorisierung von Benutzerprofilzugriffen zu untersuchen. Dies ist hier besonders wichtig, da durch den Einsatz von – zumindest konzeptionell – autonomen Agenten zur Verwaltung von personenbezogenen Informationen auch ein Teil der Kontrolle über sein Profil für den Benutzer verloren gehen kann. Bevor in Kapitel 3 mögliche Lösungen dafür betrachtet werden, soll im nächsten Abschnitt zunächst geklärt werden, was „Privatheit“ eigentlich ist und welche Anforderungen sich daraus ergeben.

2.2 Privatheit, Sicherheit und Datenschutz

Ausgehend von dem recht allgemeinen Konzept der „Privatheit“ werden im Folgenden auch rechtliche Rahmenbedingungen und Charakteristika von Schutzzielen für mehrseitige Sicherheit in Hinblick auf Privatheit untersucht. Das Ziel dieses Kapitels ist es, Anforderungen an ein technisches System für Privatheit und Zugriffskontrolle in dem erläuterten Umfeld zu erarbeiten.

2.2.1 Privatheit

Überlegungen zu *Privatheit* (engl. privacy) reichen schon sehr lange zurück. Bereits 1890 schrieben Samuel D. Warren und Louis D. Brandeis:

> „... The intensity and complexity of life, attendant upon advancing civilization, have rendered necessary some retreat from the world ... so that solitude and privacy have become more essential to the individual; but modern enterprise and invention have, through invasion upon his privacy, subjected him to mental pain and distress far greater than could be inflicted by mere bodily injury.“ (aus: [WaBr1890])

Weiterhin definieren sie Privatheit als Recht, alleine gelassen zu werden („to be let alone“). Der Grund der Publikation bestand darin, dass durch technologische Fortschritte die Privatheit bedroht schien, z.B. auf dem Gebiet der Fotografie und der zunehmenden Veröffentlichung von Fotos in Boulevard-Zeitungen [FiHü01]. Obwohl obiger Text schon mehr als 110 Jahre zurückliegt, ist die prinzipielle Aussage, dass Privatheit immer wichtiger für ein Individuum wird, noch gültig und wieder sehr aktuell.

Die wohl am häufigsten verwendete Definition von Privatheit ist von Alan Westin: „Privacy is the claim of individuals, groups or institutions to determine for themselves, when, how and to what extent information about them is communicated to others.“ (aus: [Wes67])

Essentiell ist dabei der Aspekt der Kontrolle. Benutzerinformation sollen schon verwaltet werden können, schließlich kann der Benutzer dadurch personalisierte Dienste nutzen. Wichtig ist aber, dass der Benutzer jederzeit überwachen und bestimmen kann, welche persönlichen Daten wie verwendet werden, und keine Informationen ohne sein Einverständnis weitergegeben werden.

Es gibt verschiedene Aspekte oder Dimensionen von Privatheit [FiHü01, Lau00], u.a. die Privatheit der Person, die z.B. einen Schutz vor physischer Annäherung impliziert. Mit der Verbreitung des Internets und einer stark zunehmenden Speicherung personenbezogener Daten wird die Dimension der Privatheit in Bezug auf persönliche Daten immer wichtiger und ist hier in dem Umfeld einer Verwaltung von Benutzerprofilen besonders interessant. Dies wird auch als *„Information Privacy“* bezeichnet:

> „Information Privacy refers to the claims of individuals that information about themselves should generally not be available to other individuals or organizations, and that, where data is possessed by another party, the individual must be able to exercise a substantial degree of control over that data and its use.“ (aus: [Cla99])

Der Anspruch auf Privatheit ist dabei kein absolutes oder unabdingbares Recht, ergibt sich aber auch aus entsprechenden gesetzlichen Vorgaben, die in Abschnitt 2.2.2 besprochen werden. Westin teilt die Menschen in drei Kategorien ein [Les01]: Menschen, die sehr beunruhigt sind bezüglich ihrer Privatheit (25% laut [Les01]) und starke Einschränkungen in Kauf nehmen, um ihre Privatheit

zu schützen. 12% der Personen sind überhaupt nicht besorgt und geben persönliche Daten in beliebiger Weise heraus. Die Mehrzahl der Menschen (63%) fällt in eine dazwischenliegende Kategorie: Bedenken bezüglich der Gefahren, aber auch Interesse an möglichen Vorteilen einer Teilaufgabe von Privatheit und z.B. der Verwaltung von Benutzerprofilen zur Personalisierung von Diensten. Benutzer haben dabei insbesondere folgende Bedenken in Bezug auf die Privatheit ihrer persönlichen Daten [Cra99, MaLa01]:

- Gewährleistung einer sicheren Speicherung und Übertragung sensibler Daten
- Unwissenheit darüber, welche Benutzerinformationen überhaupt von wem gespeichert sind
- Befürchtung einer unbefugten Preisgabe oder Verwendung personenbezogener Daten
- Uneinheitliche oder unklare gesetzliche Situation (siehe dazu auch die Ausführungen über rechtliche Rahmenbedingungen in Abschnitt 2.2.2)
- Verbesserte technologische Möglichkeiten, große Mengen an personenbezogenen Daten mit relativ geringen Aufwand zu sammeln und auszuwerten (z.B. mit Methoden des *Data Mings*[4])
- Unsicherheit über Möglichkeiten einer nachträglichen Korrektur oder Löschung von Daten

Daher ist es wichtig, Mechanismen zur Verfügung zu stellen, die es erlauben, die Privatheit personenbezogener Daten sicherzustellen. Benutzer müssen über die Speicherung und Verwendung personenbezogener Daten informiert und in die Lage versetzt werden, Entscheidungen bezüglich der Verwaltung ihrer Daten zu treffen. Dies wird in der Literatur auch als „informed consent“ [LiLo98], oder „notice & choice“ [Cra99] bezeichnet.

Man kann nicht davon ausgehen, dass die geforderte Kontrolle, Informiertheit und Entscheidungsmöglichkeit für Internet-Benutzer gegeben ist:

> „People are not in control of the technology that surrounds them. We have important data and personal information scattered in hundreds of places across the technology landscape, locked away in applications, product registration databases, cookies, and Web site user tracking databases.“ (aus: [Mic01])

Ein weiterer wichtiger Punkt ist in diesem Zusammenhang auch das *Vertrauen* der Interaktionspartner, was im Internet oftmals nicht gegeben ist. Vertrauen kann in diesem Zusammenhang definiert werden als „Gewissheit (d.h. die innere Repräsentanz des Eintretens) einer erwünschten Zukunft. Es beruht

- auf der Kontinuität des regelhaften und erwünschten Verhaltens der Umgebung
- oder auf der Hilfe vertrauter Menschen (auch in unwägbarer Lage)
- oder auf der eigenen Kenntnis und Beherrschung der Lage (einschließlich ihrer Unwägbarkeiten)“ (aus: [Gri01], S.69)

Ein Fehlen dieser Gewissheit in dem betrachteten Szenario erfordert Mechanismen, um Vertrauen zwischen den Akteuren aufzubauen, z.B. durch eine Verbesserung der „eigenen Kenntnis und Beherrschung der Lage“.

Privatheit ist eine deutsche Bezeichnung für „privacy“. Manchmal wird synonym dafür auch der Begriff „Datenschutz“ verwendet, obwohl mit Datenschutz hauptsächlich die rechtlichen Rahmenbedingungen gemeint sind, was nur einen Teil von „Privatheit“ ausmacht.

[4] Beim Data Mining wird versucht, mit Hilfe statistischer Methoden komplexe Zusammenhänge und Trends in Massendaten herauszufinden.

2.2.2 Gesetzliche Rahmenbedingungen für Datenschutz

In diesem Beitrag sollen Möglichkeiten der Informatik zur Verbesserung der Privatheit bei der Verwaltung personenbezogener Daten diskutiert werden. Eine technische Lösung hat jedoch zum einen relativ wenig Sinn, wenn keine gesetzlichen Mittel vorhanden sind, um dies gegebenenfalls auch rechtlich durch zu setzen[5]. Zum anderen ergeben sich aus den gesetzlichen Rahmenbedingungen Grundsätze und Anforderungen auch für das Zugriffskontrollsystem auf technischer Ebene (vgl. dazu Abschnitt 2.3). Daher soll hier ein kurzer Überblick über relevante juristische Aspekte von Datenschutz und Privatheit gegeben werden. Es werden die Richtlinien der OECD, das deutsche Bundesdatenschutzgesetz (BDSG) und Teledienstedatenschutzgesetz (TDDSG), sowie die Situation in den USA betrachtet.

2.2.2.1 OECD Richtlinien

Die Organisation für wirtschaftliche Zusammenarbeit und Entwicklung (OECD) hat 1980 in einer „Empfehlung des Rates über Leitlinien für den Schutz des Persönlichkeitsbereichs und den grenzüberschreitenden Verkehr personenbezogener Daten“ sieben Grundsätze zum Schutz personenbezogener Daten aufgestellt [OECD80, Cla99].

Es sollte eine Beschränkung der Beschaffung personenbezogener Daten geben („collection limitation“). Daten sollen im Hinblick auf ihren Verwendungszweck erheblich und, soweit es der Verwendungszweck erfordert, sachlich richtig, vollständig und auf den neuesten Stand gebracht sein („data quality“). Die Zwecke, für die personenbezogene Daten beschafft werden, sollen im einzelnen angegeben werden („purpose specification“). Die Verwendung der Daten soll beschränkt sein („use limitation“). Personenbezogene Daten sollen durch angemessene Sicherungsmaßnahmen gegen Gefahren wie Verlust, unbefugten Zugang sowie unbefugte Zerstörung, Verwendung, Änderung oder Preisgabe geschützt werden („security safeguards“). Es soll allgemein gewährleistet werden, dass Entwicklung, Praxis und Politik hinsichtlich personenbezogener Daten durchschaubar sind (Transparenz, „openness“). Der Betroffenen soll ein Recht auf Auskunft über die Datenerfassung und Korrektur, Löschung, Vervollständigung und Änderung haben („individual participation“). Ein Verantwortlicher für eine Datensammlung soll für die Beachtung der Maßnahmen verantwortlich sein, welche die oben genannten Grundsätze verwirklichen („accountability“).

Diese Richtlinien sind zwar rechtlich nicht verbindlich, sind aber in viele nationale Gesetze eingegangen.

2.2.2.2 Bundesdatenschutzgesetz und Teledienstedatenschutzgesetz

Das Bundesdatenschutzgesetz (BDSG) enthält einige allgemeine Grundsätze über den Umgang mit personenbezogenen Daten. Ein Datum gilt nach dem BDSG als *personenbezogen*, wenn es Angaben über persönliche oder sachliche Verhältnisse einer bestimmbaren natürlichen Person enthält. Entscheidend ist also die Verknüpfung von Daten mit der Identität einer Person. Das BDSG legt fest, dass Betroffene ein Recht auf Auskunft haben, welche Daten warum gespeichert werden, sowie die Möglichkeit, eine Korrektur, Löschung und Sperrung seiner Daten zu verlangen.

Spezieller auf die Anforderungen des Datenschutzes in der Informationsverarbeitung geht das Teledienstedatenschutzgesetz (TDDSG) ein, das ein Teil des Informations- und Kommunikationsdienste-Gesetzes (IuKDG) ist. Das TDDSG legt fest, welche personenbezogenen Daten ein Anbieter speichern

[5] Allerdings gibt es z.Zt. kaum weltweit gültige Rechtsvorschriften und ist die Anwendung nationaler Gesetze im grenzüberschreitenden Internet sehr problematisch [Bäu00].

darf, wie er damit umgehen muss und wie der Benutzer Kontrollmöglichkeiten ausüben kann. Zwei Prinzipien lassen sich insbesondere daraus ableiten: Datensparsamkeit und Zweckbindung [ScEn00, FHO98].

Ein Grundsatz ist die *Datensparsamkeit* bzw. *Datenvermeidung*, dies wird auch als *Erforderlichkeit* der Datenerfassung bezeichnet. Dabei muss ein Diensteanbieter sicherstellen, nur die für die Erbringung des vom Benutzer erwünschten Dienstes notwendigen personenbezogenen Daten zu erheben und zu verarbeiten. Darunter fällt auch die Möglichkeit, einen Dienst anonym oder unter einem Pseudonym anzubieten, soweit dies technisch machbar ist. Verweigert der Benutzer eine Herausgabe personenbezogener Daten, darf er nicht vom Dienst ausgeschlossen werden.

Das zweite wichtige Prinzip ist die *Zweckbindung* bei der Speicherung personenbezogener Daten. Der Zweck einer Datenerfassung muss sich entweder aus den gesetzlichen Regelungen ergeben oder der Benutzer hat für die Erhebung und Nutzung zu einem spezifizierten Zweck seine ausdrückliche Einwilligung erteilt. Daten dürfen dann nur für diesen Zweck verwendet werden.

2.2.2.3 Situation in den USA

Die Situation in den USA bezüglich rechtlicher Rahmenbedingungen ist auch im Hinblick auf Privatheit von einem etwas anderen Rechtsverständnis als in Europa geprägt. Es gibt weniger gesetzliche Regelungen, sondern man geht von einer Selbstregulierung des Marktes aus. Verstöße gegen Privatheitsansprüche werden daher eher als Bruch einer Vereinbarung zwischen einem Unternehmen und einem Kunden bzw. als Betrug gewertet, und nicht der Missachtung eines Gesetzes.

Um die aus EU-Sicht nicht ausreichenden gesetzlichen Vorschriften auszugleichen und für EU-Bürger ein „angemessenes Schutzniveau“ [Tät00] gegenüber Drittstaaten wie den USA zu gewährleisten, wurde die so genannte „safe harbour“ Vereinbarung getroffen. Sie sieht im Grundsatz vor, dass US-amerikanische Firmenzusammenschlüsse sich gemeinschaftlich verpflichten, für die von Europa zu ihnen exportierten Daten ein Datenschutzniveau einzuhalten, das europäischen Maßstäben entspricht. Dabei sollen die folgenden Prinzipien gelten [Tät00]:

- „notice“: Informationspflichten über die Art der Datenerhebung und -verarbeitung sowie über ihren Zweck, die Empfänger und die Wahlmöglichkeiten hinsichtlich der Begrenzung und der Nutzung und Übermittlung
- „choice“: Wahlrecht hinsichtlich der Nutzung der Daten
- „onward transfer“: bei der Weitergabe der Daten an Dritte wird sichergestellt, dass dort das Datenschutzniveau nicht abfällt
- „security“: technische und organisatorische Maßnahmen zur Sicherheit der Datenverarbeitung
- „data integrity“: Sicherstellung der Integrität der Daten, also von Richtigkeit, Vollständigkeit, Aktualität und Erforderlichkeit im Einzelfall
- „access“: das Recht der Betroffenen auf Auskunft über die zu ihrer Person gespeicherten Daten
- „enforcement“: eine effektive Durchsetzung der Prinzipien

Es fehlen dabei zwar einige Punkte aus den deutschen Datenschutzgesetzen wie Datensparsamkeit oder das Verbot des Ausschlusses von Benutzern bei Verweigerung der Zustimmung einer Speicherung personenbezogener Daten. Die „safe harbour“ Grundsätze können aber, zusammen mit den

anderen, oben erläuterten Prinzipien gesetzlicher Regelungen, als Grundlage für die Speicherung und Nutzung personenbezogener Daten aus rechtlicher Sicht mit angesehen werden (vgl. Anforderungen in Abschnitt 2.3).

2.2.3 Privatheit und E-Commerce

Privatheit hat einem hohen Stellenwert bei Electronic Commerce, was auch eine Betrachtung von Privatheit wichtig macht. Umfragen unter Internet Benutzern zeigen, dass diese Bedenken in Bezug auf ihre Privatheit haben. Zum Beispiel äußerten 87% der Befragten in einer Studie von Ackermann et.al. Besorgnis bezüglich ihrer Privatheit im Internet [ACR99]. Benutzer sind außerdem weniger gewillt, Informationen herauszugeben, wenn es sich um personenbezogene Daten handelt:

> „In a scenario involving a banking Web site, 58% of respondents said that they would provide information about their income, investments, and investment goals in order to receive customized investment advice. However only 35% said they would also supply their name and address so that they could receive an investment guide booklet be mail.“ (aus: [ACR99], S. 5)

Die Bereitschaft, Daten bereitzustellen, sinkt also deutlich, wenn dies nicht mehr anonym erfolgt. Diensteanbieter können somit auch von einer Verbesserung der Mechanismen zum Schutz der Privatheit profitieren, da Benutzer dann eher bereit sind, mehr und bessere persönliche Informationen herauszugeben, wenn sie sicher sein können, dass diese Daten nicht in unbeabsichtigter Weise verwendet werden [BeKö00].

Forrester Research argumentiert in einer Studie „Surviving The Privacy Revolution“ vom Februar 2001 [For01], dass Privatheit einer der wichtigsten Gesichtspunkte beim Erfolg von E-Commerce ist, und dass Unternehmen, die keine Maßnahmen zum Schutz der Privatheit ihrer Kunden treffen, Nachteile erleiden könnten. Insbesondere gilt dies auch für Aspekte mobiler Kommunikation („M-Commerce“), wobei z.B. Dienste, die den aktuellen Standort des Benutzers auswerten, eine wichtige Rolle spielt. Die Privatheit in mobilen Diensten und Interaktion hat noch andere Anforderungen, als das hier betrachtete Szenario, wobei darauf in diesem Beitrag nicht näher eingegangen wird.

Bei E-Commerce ist insbesondere im Privatkundengeschäft Vertrauen sehr wichtig, was im Internet u.a. durch den Verlust eines persönlichen Kontaktes beim Abschluss eines Geschäftes oftmals nicht gegeben ist. Die Befürchtung vor einem „gläsernen Internet-Kunden“ hält außerdem viele Menschen davon ab, im Internet aufzutreten [Gri01]. Dies könnte durch verbesserte technologische Unterstützung ausgeglichen werden.

2.2.4 Schutzziele mehrseitiger Sicherheit und Privatheit

Um konkrete Anforderungen für Privatheit in diesem Szenario zu entwickeln, können die folgenden Schutzziele mehrseitiger Sicherheit als Ausgangsbasis dienen.

2.2.4.1 Mehrseitige Sicherheit

In dem hier betrachteten Szenario interagieren verschiedene, über ein offenes Netzwerk verbundene Teilnehmer miteinander, die z.T. einander nicht kennen oder vertrauen. Jeder Kommunikationspartner verfolgt verschiedene Interessen und hat unterschiedliche Anforderungen in Bezug auf Privatheit und Sicherheit. Es muss deshalb ein Abgleich konkurrierender Interessen erfolgen. Zum Beispiel könnte bei einer E-Commerce Transaktion ein Händler möglichst viel über seinen Kunden wissen wollen,

während ein Benutzer möglichst wenig von sich preisgeben will oder anonym auftreten will, z.B. um unerwünschte Werbung zu verhindern. Auch ist es möglich, dass unbefugte Dritte durch Abhören von Kommunikationsbeziehung an sensible Daten herankommen. *Mehrseitige Sicherheit* ([PSWW00], [WoPf00]) bedeutet die Berücksichtigung der Sicherheitsanforderungen aller beteiligten Parteien.

Sowohl a priori fehlendes Vertrauen der Kommunikationspartner, als auch potentielle Angriffe von Dritten sollen dabei durch Schutzmechanismen wie z.B. Verschlüsselung oder Anonymität bei der Kommunikation ausgeglichen werden. Dies ist insbesondere wichtig bei der Verwaltung von sensiblen Daten, wie z.B. Benutzerprofilinformationen.

In der Begriffswelt der mehrseitigen Sicherheit wurden Schutzziele entwickelt, die für Verwaltung von Benutzerprofilen sehr relevant sind und die daher im Folgenden näher untersucht werden.

2.2.4.2 Schutzziele mehrseitiger Sicherheit

Tabelle 2.1 zeigt die Schutzziele mehrseitiger Sicherheit und deren (beispielhafte) Bedeutung bei der Verwaltung von Benutzerprofilen ([PSWW00, WoPf00, CC98]).

Diese Ziele müssen sowohl gegenüber dem Kommunikationspartner, als auch gegenüber potentiellen Dritten, betrachtet werden. Die Schutzziele sind nicht unabhängig voneinander, sondern haben Wechselwirkungen. Zum Beispiel wird Vertraulichkeit durch Unbeobachtbarkeit impliziert, Verdecktheit verstärkt Anonymität und Anonymität ist komplementär zu Zurechenbarkeit.

2.2.4.3 Aspekte einer E-Privacy

Wie schon angeführt, haben verschiedene Schutzziele unterschiedliche Bedeutung in dem hier betrachteten Szenario des Zugriffs auf dezentrale Benutzerprofile. Die Schutzziele lassen sich in Kategorien einordnen, die die Hauptaspekte einer *„E-Privacy“* [Bäu00] ausmachen. Anonymität und Pseudonymität sind verwandt und lassen sich daher eine Kategorie „Identitätsziele“ einordnen. Weiterhin haben Vertraulichkeit, Verdecktheit und Unbeobachtbarkeit miteinander zu tun und können zu „Vertraulichkeitszielen“ zusammengefasst werden. Schließlich kann man Integrität, Verbindlichkeit und Zurechenbarkeit als eine Menge von „Absicherungszielen“ auffassen.

Noch nicht berücksichtigt ist dabei folgender, bei einer Verwaltung von personenbezogenen Daten wichtige Gesichtspunkt: Wie kann sich der Benutzer im Klaren sein, welcher Aspekt seiner Person zu irgendeinem Zeitpunkt überwacht wird, und unter welchen Umständen dies geschieht [MaLa01]? Dies kann man als „Transparenzziel“ bezeichnen und es ergibt sich auch aus den rechtlichen Rahmenbedingungen für Datenschutz. Auch betonen die bisher betrachteten Aspekte eher die Absicherung gegenüber unbefugten Dritte, weniger die Verwendung von Benutzerprofilen beim Nutzer der Informationen. Unter dem Gesichtspunkt Privatheit personenbezogener Daten ist letzteres aber auch sehr wichtig. Eine Berechtigung zum Zugriff auf Daten wird meist als Autorisierung bezeichnet und durch ein Zugriffskontrollsystem umgesetzt. Die einzelnen Punkte dieses „Autorisierungszieles“ werden im nächsten Abschnitt erarbeitet.

Als Kategorien einer E-Privacy lassen sich also zusammenfassend festhalten[6]:

- Identitätsziele: Anonymität, Pseudonymität
- Vertraulichkeitsziele: Vertraulichkeit, Verdecktheit, Unbeobachtbarkeit
- Absicherungsziele: Integrität, Verbindlichkeit, Zurechenbarkeit

[6] Die Einordnung der Schutzziele in Kategorien dient hier hauptsächlich dazu, die Anforderungen in Abschnitt 2.3 zu strukturieren, manche Ziele spielen in mehreren Kategorien eine Rolle.

Schutzziel	**Bedeutung bei Benutzerprofilverwaltung**
Vertraulichkeit: Sichert die Geheimhaltung von Daten während der Übertragung, kein unbefugter Dritter kann den Inhalt der Nachricht erkennen	Unbefugte (d.h. vom Benutzer nicht autorisierte) dürfen keinen Zugriff auf persönliche Daten erhalten. Dies betrifft sowohl die Speicherung und Übertragung der Daten, als auch die Verwendung der Benutzerprofile, also insbesondere auch die Weitergabe von Daten
Verdecktheit: Versteckt die Übertragung einer Nachricht, kein Dritter soll die Existenz einer Nachricht erkennen können	Die Übertragung (von Teilen) eines Benutzerprofils von Profilagenten zu einem Dienst soll verdeckt stattfinden
Unbeobachtbarkeit: Sichert, dass ein Benutzer Dienste oder Ressourcen nutzen kann, ohne dass andere beobachten können, dass der Dienst oder die Ressource genutzt wird	Die Kommunikation mit einem Dienst darf nicht ersichtlicht sein, da ein unbefugter Dritter z.B. aus der Nutzung einer Informationsquelle zu gesundheitlichen Fragen auf eine Krankheit einer Person schließen könnte
Anonymität: Kommunikationspartner und Dritte erfahren nicht die Identität eines Benutzers	Benutzer sollen anonymisiert mit einem Dienst kommunizieren können
Pseudonymität: Nutzung einer Ressource ist einem Benutzer zurechenbar, ohne dass dieser seine Identität offenbaren muss	Um personalisierte Dienstleistungen geben zu können, ist eine vollständige Anonymität nicht immer wünschenswert, sondern eine Interaktion mit einem Pseudonym nötig
Zurechenbarkeit: Sichert, dass das Senden (bzw. Empfangen) von Information gegenüber Dritten bewiesen werden kann	Z.B. ist bei einer Bestellung von Produkten durch einen E-Commerce Agenten im Auftrag eines Benutzers für den Händler ein Nachweis der Bestellung nötig
Integrität: Sichert, dass (unbefugte) Modifikationen einer Nachricht durch den Empfänger erkannt werden können	Benutzerprofilinhalte sollen bei der Kommunikation von Benutzerprofilagenten zum Community Agenten nicht verändert werden können, bzw. eine Manipulation muss erkannt werden können
Verbindlichkeit: Sichert, dass ein Nutzer belangt werden kann, um seine Zusagen innerhalb einer angemessenen Zeit zu erfüllen	Betrifft auch die Einhaltung von Zusagen der Verwendung von Attributen aus dem Benutzerprofil, z.B. keine Weitergabe an Dritte
Verfügbarkeit: Sichert die Nutzbarkeit von Diensten und Ressourcen für einen Benutzer	Keine besondere Relevanz in Bezug auf Privatheit beim Zugriff auf Benutzerprofile
Erreichbarkeit: Sichert, dass mit einer Ressource (z.B. hier auch ein anderer Nutzer) Kontakt aufgenommen werden kann, wenn gewünscht	Keine besondere Relevanz in Bezug auf Benutzerprofilverwaltung, schlechte Erreichbarkeit (oder Verfügbarkeit) kann tendenziell die Privatheit von Benutzern verbessern

TABELLE 2.1: Schutzziele und deren Bedeutung bei Benutzerprofilverwaltung

- Transparenzziele: Überwachungs- und Kontrollfunktionen für Benutzer
- Autorisierungsziele: Bereitstellung einer geeigneten Zugriffskontrolle

2.3 Anforderungen

Die erläuterten Vertraulichkeitsziele beziehen sich hier auf die Vertraulichkeit bei der Datenübertragung – in unserem Szenario – vom Benutzerprofilagent zu einem Dienstanbieter. Wie schon erwähnt wurde, muss auch festgelegt werden können, ob dieser Dienst überhaupt eine Zugriffserlaubnis auf die betreffenden Profilattribute haben soll. Ein wichtiger Punkt ist daher die Möglichkeit, Zugriffsrechte für einzelne Attribute des Benutzerprofils festlegen zu können. Die Anforderungen an ein technisches System zur Gewährleitung von Privatheit lassen sich daher aus drei Ausgangspunkten ableiten:

- Gesetzliche Rahmenbedingungen
- Schutzziele mehrseitiger Sicherheit
- Notwendigkeit eines Zugriffskontrollsystems

Dem Benutzer müssen also Mechanismen zur Verfügung gestellt werden, um zu bestimmen, welcher Dienstagent welche Zugriffsrechte auf das dezentral verwaltete Benutzerprofil ausüben darf. Dies ist eine Hauptfunktion des in Abschnitt 2.1.1.1 erläuterten Identitätsmanagements. Der wichtigste Aspekt dabei ist, dass der Benutzer die volle Kontrolle für sein Benutzerprofil ausübt. Folgende konkrete Anforderungen an die Zugriffskontrolle kann man dabei festhalten:

- Autorisierungsziele:
  - Wählbare Granularität (Rechte für einzelne Profilattribute, sowie auch für Kategorien möglich)
  - Kontrolle der Weitergabe von Daten
  - Flexibilität bei der Zugriffskontrolle, z.B. durch Aushandlung
  - Möglichkeit, Optionen wie „Zugriff nur bei gesicherter Übertragung“ zu realisieren
  - Integration von „Zweckbindung“
  - Möglichkeiten zur zeitlichen Begrenzung und Zurückziehbarkeit von Rechten
  - Möglichkeiten, Benutzerprofile über Dienstgrenzen hinweg zu synchronisieren, um den Überblick über die gespeicherten Daten für Benutzer zu verbessern
  - Möglichkeit, Zugriffsregeln sowohl für bestimmte Dienste oder auch unabhängig von einem konkreten Dienst zu formulieren
  - Geeignete Benutzerschnittstellen zur Administration von Rechten

In den weiteren Kategorien einer E-Privacy lassen sich folgende Anforderungen an einen Mechanismus zur Autorisation und Sicherstellung von Privatheit bei dezentraler Verwaltung von Benutzerprofilen formulieren:

- Identitätsziele:
  - Mechanismen für Identitätsmanagement
  - Zweifelsfreie Überprüfung der Identität der Kommunikationspartner
  - Möglichkeit für Benutzer, anonym zu kommunizieren
  - Möglichkeit der Verwendung eines Pseudonyms

  - Unverkettbarkeit von Pseudonymen (bzw. Verkettbarkeit nur unter der Kontrolle des Benutzers)
  - Möglichkeiten, verschiedene Stufen von Pseudonymität zu wählen (vgl. Abschnitt 2.1.1.2)

- Vertraulichkeitsziele:
  - Möglichkeit zur Festlegung von Zugriffsrechten für Benutzerprofil-Attribute (Zugriffskontrollsystem)
  - Flexibilität bei der Zugriffskontrolle, z.B. durch Regeln und Verhandlung, sowie bei der Granularität der Rechtevergabe
  - Realisierung von Unbeobachtbarkeit und Verdecktheit in den Kommunikationsbeziehungen
  - Möglichkeit, Daten über eine gesicherte Verbindung zu übertragen
  - Datensparsamkeit, Datenvermeidung bzw. Erforderlichkeit eines Zugriffs
  - Einhaltung von (technischen und organisatorischen) Richtlinien zur Sicherheit bei der Datenspeicherung und -verarbeitung

- Absicherungsziele:
  - Speicherung des Profils unter der Kontrolle des Benutzers
  - Zweckbindung von Profilzugriffen
  - Regelung der „Weitergabe von Daten“
  - Möglichkeit, Zugriffsrechte zeitlich zu beschränken und ggf. auch wieder zurückziehen zu können
  - Entscheidungsmöglichkeit beim Benutzer bezüglich der Herausgabe von Daten
  - Möglichkeit der Unterstützung von vertrauenswürdigen Organisationen (z.B. für die Prüfung der zugesagten Verwendung der Daten)
  - Möglichkeiten der Zurechenbarkeit und Unabstreitbarkeit (z.B. Nachweis einer Bestellung)
  - Einhaltung gesetzlicher Rahmenbedingungen
  - Verantwortlichkeit derjenigen Unternehmung oder Organisation, die personenbezogene Daten speichert
  - Möglichkeit der Durchsetzung der erläuterten Prinzipien gegenüber Unternehmungen, die personenbezogene Daten verarbeiten
  - Integritätsanforderung: Daten sollen gegen Verfälschung insbesondere während einer Übertragung in offenen Netzen geschützt werden können
  - Möglichkeit der Signierung eines Zugriffsrechts

- Transparenzziele:
  - Möglichkeit, vergebene Zugriffsrechte jederzeit überprüfen zu können, auch durch vertrauenswürdige Institutionen
  - Möglichkeit für Benutzer, Zugriffe überwachen zu können, Protokollierung aller Zugriffe
  - Sinnvolle Benutzerschnittstellen zur Festlegung von Rechten

– Vertrauenswürdiger Benutzeragent

Um diese Anforderungen umsetzen zu können, kommen verschiedene Mechanismen in Frage. Zum einen ein Zugriffskontrollsystem, das dem Benutzer eine Festlegung erlaubt, wer wie auf sein Benutzerprofil zugreifen darf. Dies alleine reicht aber nicht aus, da damit Gesichtspunkte wie zum Beispiel Identitätsmanagement oder Anonymisierung nicht abgedeckt werden können. Dafür gibt es zum anderen so genannte „Privacy Enhancing Technologies“, die diese Aspekte behandeln. Daher soll im folgenden Kapitel 3 dieser Arbeit untersucht werden, ob und welche Teile der Anforderungen mit den bestehenden Mechanismen und Verfahren erfüllt werden können.

Kapitel 3

Bestehende Systeme

„You have zero privacy anyway, get over it!“
Scott McNealy, CEO Sun Microsystems (25.01.1999)

Dieses pessimistische Zitat drückt eine vorhandene allgemeine Unsicherheit bezüglich der Privatheit im Internet aus, es gibt aber durchaus Systeme und Anwendungen, die eine Verbesserung erreichen. Im dem folgenden Kapitel sollen diese Anwendungen untersucht werden, insbesondere dahingehend, ob sie für eine Verbesserung von Privatheit und Autorisation bei dezentraler Verwaltung von Benutzerprofilen geeignet sind.

Die beiden wichtigsten bestehenden Ansatzpunkte sind Modelle und Verfahren zur *Zugriffskontrolle*, sowie Mechanismen, die versuchen die Privatheit im Internet zu verbessern (*Privacy Enhancing Technologies (PET)*). In diesem Abschnitt soll zunächst untersucht werden, inwieweit bestehende Modelle und Systeme zur Zugriffskontrolle geeignet sind, zur Lösung der Aufgabenstellung beizutragen. Die einzelnen Mechanismen und Verfahren werden hier nur sehr kurz und im Überblick behandelt. Auch werden einzelne Aspekte, die für Benutzerprofilverwaltung interessant erscheinen, herausgestellt, auch wenn diese nicht unbedingt die Kernpunkte eines vorgestellten Ansatzes sind. Anschließend werden verschiedene PET-Anwendungen besprochen.

Des Weiteren werden Anwendungen für Identitätsmanagement unter dem Gesichtspunkt der Privatheit diskutiert, da diese Systeme das hier betrachtete Szenario einer dezentralen Verwaltung von Benutzerprofilen in der Praxis umsetzen. Das Kapitel schließt mit einer zusammenfassenden Bewertung.

3.1 Zugriffskontrolle

3.1.1 Grundlagen, Zugriffsrechte und Zugriffskontrollmatrix

Zugriffskontrolle ist der Teil eines Sicherheitsmodells, der die Autorisierung von Zugriffsanforderungen an Ressourcen von (bereits authentifizierten) Benutzern oder Systemkomponenten behandelt. Auch ist die Administration von Zugriffsrechten wichtig, also die Entscheidung, wer auf welche Weise Rechte oder auch Zugriffsregeln o.ä. festlegt (siehe Abschnitt 3.1.4).

Benutzerprofile enthalten Daten, die vor unerlaubten oder unerwünschten Zugriff geschützt werden sollen, dies fällt also genau in die Domäne der Zugriffskontrolle. Daher sollen nach einem Überblick über die Grundlagen der Zugriffskontrolle verschiedene Modelle und Systeme betrachtet werden und auch deren Eignung in Hinblick auf dezentrale Benutzerprofilverwaltung untersucht werden.

Es werden bei der Zugriffskontrolle folgende Entitäten unterschieden:

- Objekte: (passive) Komponenten, die geschützt werden sollen (z.B. Dateien oder Attribute in einem Benutzerprofil)
- Subjekte: Systemkomponenten, die auf Objekte zugreifen (z.B. Benutzer, Programme oder Software-Agenten)
- Zugriffsrechte: Mögliche Zugriffsaktionen (z.B. Lesen, Schreiben, Löschen, Ausführen, ...)

Die Beziehung zwischen diesen Entitäten ist der Kernpunkt der Zugriffskontrolle. Der Zugriff der Subjekte auf die Objekte muss eingeschränkt und festgelegt werden können. Ein Zugriffskontrollsystem hat außerdem zu gewährleisten, dass alle Subjekte und Objekte eindeutig identifiziert werden und jedes Objekt von der Rechteverwaltung erfasst wird [Eck01].

Der Ausgangspunkt für die Realisierung einer Zugriffskontrolle ist die *Zugriffskontrollmatrix* (engl. access control matrix, ACM). Dabei stellen die Subjekte (in unserem Szenario die E-Commerce und Community Agenten) die Zeilen einer zweidimensionalen Matrix dar, die Objekte (hier die Attribute des Benutzerprofils) die Spalten der Matrix. In den Schnittpunkten der Matrix stehen die Rechte für den Zugriff des Subjekts auf das betreffende Objekt (vgl. Beispiel in Tabelle 3.1).

	Email-Adresse	**Kreditkartennummer**	**private Interessen**
E-Commerce Agent A		Lesen	
Community Agent B	Lesen		
Community Agent C			Lesen, Schreiben

TABELLE 3.1: Zugriffskontrollmatrix

Es können auch mehrere Attribute zusammengefasst werden und z.B. Rechte sowohl für alle „privaten Interessen“ als auch einzelne Interessen verwaltet werden, die Granularität der betrachteten Objekte ist also auch wichtig.

Die Implementierung erfolgt mit Hilfe der in den folgenden Abschnitten erläuterten Konzepte und Modelle. Eine direkte Implementierung der ACM ist in der Regel nicht sehr effizient, weil die Matrix oft nicht sehr dicht besetzt ist, also nur für relativ wenige Objekt/Subjekt Paare explizite Rechte festgelegt sind.

3.1.2 Konzepte zur Implementierung

Es gibt im Prinzip drei Alternativen, die Zugriffskontrollmatrix zu implementieren, nämlich eine spalten- oder zeilenweise Realisierung, sowie eine kombinierte Vorgehensweise.

3.1.2.1 Zugriffskontrollliste

Eine spaltenweise Implementierung der ACM ist die *Zugriffskontrollliste* (engl. access control list, ACL). Dies ist eine Liste pro zu schützendem Objekt, die die Zugriffsrechte von Subjekten definiert. Die Zugriffskontrollliste realisiert somit eine Objekt-bezogene Sichtweise. Der Vorteil dabei ist, dass es leicht ist, festzustellen, welche Subjekte auf ein Objekt Zugriff hat, also z.B. eine Frage der Art „wer hat Zugriff auf meine Kreditkartennummer?“ zu beantworten. Auch ist eine Rechterücknahme relativ einfach zu realisieren. Ein Nachteil einer ACL ist es, dass sie nicht (oder nur mit großem Aufwand) transparent ist in Bezug auf eine mögliche Fragestellung des Benutzers „welche Rechte hat Community Agent X beim Zugriff auf mein Profil?“.

Die Zugriffskontrollliste ist effizient bei vielen Objekten, bietet aber keine gute Skalierbarkeit bei sehr vielen Subjekten, was eventuell bei Benutzerprofilverwaltung der Fall sein könnte. Daher wurde eine Erweiterung des ACL-Konzeptes vorgeschlagen, bei dem nicht mehr die Rechte einzelner Subjekte festgelegt werden, sondern den Subjekten Rollen zugeordnet werden oder sie in Gruppen eingeteilt werden und die Zugriffsrechte nur noch für diese Rollen in der ACL festgelegt werden (siehe Rollen-basierte Zugriffskontrolle, Abschnitt 3.1.3.3).

3.1.2.2 Zugriffsausweise

Ein *Zugriffsausweis* (engl. capability) ist ein unverfälschbares Ticket, das den Inhaber zum Zugriff auf die im Ticket genannten Objekte berechtigt. Dies entspricht einer zeilenweisen bzw. Subjektbezogenen Implementierung der ACM.

Die Zugriffskontrolle kann dadurch im Vergleich zur ACL vereinfacht werden, da nur noch der Zugriffsausweis überprüft werden muss und nicht mehr möglicherweise sehr lange Listen durchsucht werden müssen. Auch ist ein Vorteil, dass man bei einer Benutzerprofilverwaltung in übersichtlicher Weise die Zugriffsrechte eines einzelnen Dienstes ausdrücken kann. Durch eine fälschungssichere Realisierung lässt sich zudem ein dezentrales Sicherheitsmanagement durchführen, da die Komponenten, die die Capabilities ausstellen von den Komponenten, welche die Zulässigkeit von Zugriffen prüfen, getrennt werden können.

Ein Zugriffsausweis ist üblicherweise unabhängig von Inhaber des Ausweises. Somit kann auch eine Delegation von Rechten erfolgen (Ausweisweitergabe), was allerdings nicht immer wünschenswert beim Zugriff auf Benutzerprofile erscheint. Bei Verwaltung von personenbezogenen Daten muss die Weitergabe von Zugriffsrechten beschränkt werden können. Ein weiteres Problem bei Capabilities ist es, dass eine Rechterücknahme problematisch ist, da sich die Frage stellt, wie ein Widerruf eines Ausweises bekannt gegeben und in einem verteilten System durchgesetzt werden kann. Mögliche Lösungen sind eine Rückforderung ausgegebener Zugriffsausweise, was nicht immer praktikabel ist [Eck01], oder das ungültig machen von Capabilities.

3.1.2.3 Kombinierter Ansatz

Um die Nachteile – insbesondere das Rechterücknahmeproblem – von Zugriffsausweisen zu umgehen, können die beiden vorgestellten Konzepte zur Realisierung der Zugriffskontrollmatrix kombiniert werden. Eine Lösung dieser Art wird als Schlüssel/Schlossverfahren bezeichnet [Eck01].

In der Grundform des Schlüssel/Schlossverfahrens wird jedem Subjekt s eine Capability-Liste zugeordnet, die Paare der Form (o,K) enthält. Dabei bezeichnet o ein Objekt, auf das s unter Anwendung des Schlüssels K zugreifen darf. Jedes Objekt o besitzt seinerseits eine Zugriffskontrollliste, die Einträge der Form (L,a) enthält, wobei L ein Schloss ist und a diejenigen Zugriffsrechte sind, die den Benutzer des zum Schloss L passenden Schlüssels K eingeräumt werden.

Möchte nun ein Subjekt s auf das Objekt o zugreifen, so legt es der Zugriffskontrolle von o seine Capability (o,K) vor. Ein Zugriff ist dann zulässig, wenn der Schlüssel K in ein Schloss L der Zugriffsliste von o passt, d.h. wenn es einen Eintrag (L,a) gibt, mit K=L. Dieser Test muss unter Berücksichtigung der angeforderten Zugriffsrechte erfolgen.

Das Problem der Rechterücknahme kann dann einfach und effizient dadurch realisiert werden, indem das Schloss L eines Objektes verändert wird, so dass in dem Zugriffsausweis enthaltene Schlüssel K nicht mehr passt.

Diese beschriebene Form ist jedoch für einen praktischen Einsatz zu aufwändig, daher wird die Kombination von ACL und Zugriffsausweisen meist in einer stark vereinfachten Form eingesetzt,

Windows NT verwendet z.B. ein „access token“ für jeden authentifizierten Benutzer [Eck01].

3.1.3 Strategien und Modelle

Die Konzepte zur Implementierung behandeln noch keine Modelle, wie man gewünschte Strategien umsetzen kann. Dies wird in diesem Abschnitt besprochen.

3.1.3.1 Diskrete Zugriffskontrolle

Die bisher vorgestellten Möglichkeiten einer Realisierung der Zugriffskontrollmatrix fallen in den Bereich der *diskreten Zugriffskontrolle* (engl. discretionary access control, DAC; auch als „benutzerbestimmte“ Zugriffskontrolle bezeichnet). DAC basiert auf dem Eigentümer-Prinzip auf Basis von authentifizierten Benutzern oder Gruppen von Benutzern. Das bedeutet, dass der Eigentümer (engl. owner) eines Objektes – oftmals ist dies der Erzeuger – entscheiden kann, inwieweit er Zugriffsrechte an andere Benutzer weitergibt. Die Rechtevergabe in einem UNIX-Dateisystem ist ein typisches Beispiel für diskrete Zugriffskontrolle.

Ein Benutzer mit Leserecht kann dabei auch durch Kopieren einer Information und Zuweisung entsprechender Zugriffsrechte an die Kopie, die Information auch anderen Benutzern zugänglich machen, obwohl dies vielleicht nicht vom ursprünglichen Erzeuger der Ressource so vorgesehen war. Aus diesem Grunde lassen sich unautorisierte Informationsflüsse in diesem Modell kaum vermeiden.

Das Problem bei DAC ist somit, dass die Umsetzung einer Sicherheitspolitik von der Diskretion der Subjekte abhängt, was besonders auch bei Verwaltung von Benutzerprofilen nicht erwünscht oder zweckmäßig ist. Auch ist z.B. keine Zuordnung von Aufgaben und einem Zweck zu einem Zugriff möglich. Daher ist diskrete Zugriffskontrolle in der Grundform ohne zusätzliche Mechanismen nicht geeignet, Zugriffskontrolle für Benutzerprofile zu verwirklichen.

Grundsätzlich ist jedoch der Ansatz einer benutzerbestimmten Zugriffskontrolle in der Beziehung bei einem Benutzerprofilzugriff interessant, dass dabei der Eigentümer der Daten die Festlegungen bezüglich der Verwendung der Profildaten trifft.

3.1.3.2 Mandatorische Zugriffskontrolle

Mandatorische Zugriffskontrolle (engl. mandatory access control, MAC; auch als „systembestimmte“ oder „regelbasierte“ Zugriffskontrolle bezeichnet) versucht den Mangel einer systemglobalen Zugriffsstrategie bei diskreter Zugriffskontrolle zu vermeiden, indem der Zugriff verweigert wird, wenn es eine systembestimmte Festlegung gibt, die den Zugriff verbietet, obwohl die benutzerbestimmten Rechte vorhanden wären. MAC setzt nicht bei der Kontrolle des Datenzugriffs an, sondern bei der Kontrolle der Informationsflüsse [Opp97].

Der bekannteste Vertreter von MAC ist das Modell von Bell-LaPadula [BeLa73]. Die Zugriffsrechte sind dabei beschränkt auf die Menge execute, read, append, write. Die Objekte und Subjekte werden dabei von vertrauenswürdigen Systemkomponenten in *Sicherheitsklassen (SC)* eingeteilt, die nicht überschritten werden dürfen. Die Sicherheitsklassen stellen eine Hierarchie dar, z.B. „Streng Geheim“ >= „Geheim“ >= „Vertraulich“ >= „Unklassifiziert“. Eine Dominanz-Relation „>=“ definiert einen Verband über den Systemklassen. Der Zugriff auf ein Objekt ist nur dann gestattet, wenn eine bestimmte Relation (abhängig vom geforderten Zugriffsrecht) zwischen den Sicherheitsklassen des Objekts und Subjekts erfüllt ist.

Insbesondere sollen die beiden folgenden Regeln gelten:

- „no-read-up“ (auch „simple security property“ genannt): Lesen oder Ausführen eines Objektes o durch Subjekt s ist nur dann gestattet, wenn die betreffenden Zugriffsrechte vorhanden sind und gilt: SC(s) >= SC(o). Dieser Grundsatz verhindert das Lesen von vertraulichen Daten von Subjekten niedrigerer Systemklassen

- „no-write-down“ (oder „*-Eigenschaft“): Ein Zugriff „append“, also das Anfügen an eine Information, ist nur dann erlaubt, wenn ein entsprechendes Zugriffsrecht existiert und gilt: SC(s) <= SC(o). Für einen schreibenden Zugriff muss gelten: SC(s) = SC(o). Damit soll ein zufälliges oder absichtliches Hinunterschreiben von sensiblen Daten auf Stufen niedrigerer Systemklassen vermieden werden

Bell-LaPadula ist ein recht restriktives, formalisiertes Modell mit beschränkter Ausdrucksfähigkeit, eine konkrete Anwendung für ein System zur Benutzerverwaltung erscheint nicht möglich. Ein Problem ist auch das „blinde Schreiben“, wobei Objekte, also Benutzerprofilinhalte in unserem Kontext, von Subjekten niedrigerer Systemklassen überschrieben werden können, auch wenn kein lesender Zugriff darauf erlaubt ist. MAC wurde für den militärischen Bereich entwickelt und wird hauptsächlich auch dort angewendet.

Ein weiterer Nachteil bei einem Einsatz von MAC in dem betrachteten Szenario ist, dass eine Einteilung von Objekten in Sicherheitsklassen nicht so ohne weiteres möglich ist. Auch soll der Zugriff auf personenbezogenen Daten nicht abhängig von einer globalen Einteilung in Sicherheitsklassen sein, sondern von den persönlichen Präferenzen des Benutzers unter Berücksichtigung des Kontextes des Zugriffs, insbesondere des Zwecks.

Allerdings braucht man in unserem Szenario systembestimmte Festlegungen zumindest zusätzlich zu diskreten Zugriffsrechten, sonst wäre z.B. die Verhindung der Weitergabe von Informationen oder Rechten nicht möglich.

3.1.3.3 Rollen-basierte Zugriffskontrolle

In der Praxis wird oft *Rollen-basierte Zugriffskontrolle* (engl. role based access control, RBAC) [FeKu92] verwendet. Dabei werden die Zugriffsrechte nicht an Subjekte vergeben, sondern Zugriffsrechte an zu definierende Rollen erteilt und den Subjekten (möglicherweise mehrere) Rollen zugewiesen. Die Rollen entsprechen an Subjekten zugeteilten Aufgaben. Oftmals werden die Rollen hierarchisch organisiert, wodurch es ermöglicht wird, Rechte zu vererben.

Der Unterschied von RBAC zu DAC ist, dass es Benutzern nicht erlaubt ist, die Zugriffsrechte im eigenen Ermessen an andere Benutzer weiterzugeben [FeKu92]. RBAC ist also eine Form mandatorischer Zugriffskontrolle, wobei es für praktische Anwendungen leichter zu realisieren ist und daher eine größere Bedeutung besitzt, als die erläuterte Grundform von MAC, da es nicht auf starren Sicherheitsklassen beruht.

Ein Vorteil von RBAC in Hinblick auf die Verwaltung von Benutzerprofilen besteht in der Möglichkeit, Zugriffsrechte mit Aufgaben zu verbinden. Man könnte z.B. eine Rolle „Lieferant“ definieren, lesenden Zugriff auf Profildaten wie Postanschrift oder Zahlungsinformationen für diese Rolle gestatten und E-Commerce Agenten zur Abwicklung einer Bestellung im Auftrag des Benutzers diese Rolle zuweisen. Allerdings erscheint es nicht praktikabel, eine sinnvolle Zuordnung von Subjekten zu Rollen und Zugriffsrechten zu Rollen explizit vom Benutzer für alle Profilattribute durchführen zu lassen. Ein Rollen- oder Gruppen-basierter Ansatz ist in erster Linie dann sinnvoll, wenn sich die Rollen aus der Anwendungsdomäne ergeben, z.B. kann man bei einer E-Learning Anwendung zwischen „Dozent“, „Kursbetreuer“, „Student“ oder „Gast“ unterscheiden, oder in einem Firmen-Intranet kann ein Administrator Rollen vergeben, die sich aus der Organistionsstruktur oder der Funktion der

Mitarbeiter ableiten lassen. Eine nahe liegende Rollendefinition und -zuordnung ergibt sich bei dezentraler Benutzerprofilverwaltung nicht, da hierbei der Benutzer die verschiedenen Dienste in Gruppen einteilen müsste, was wohl nicht immer in sinnvoller Weise möglich ist.

Auch ist die Rollenzuweisung keine vollständige Realisierung einer Zweckbindung des Zugriffs, wie es in den Anforderungen verlangt wurde. Des Weiteren lässt sich die Verwendung von Daten nicht vernünftig modellieren. Benutzer sollen z.B. den Zugriff auf Daten in Abhängigkeit davon, ob die Information weitergegeben wird, erlauben oder verbieten können.

Ein weiterer Vorteil eines Rollen-basierten Ansatzes könnte es eventuell sein, dass anonymisierte Zugriffe leichter realisiert werden könnten, nachdem die Rechte nicht mehr an Subjekte, sondern nur noch an Rollen gebunden sind. Ferner kann RBAC die Skalierungsprobleme einer ACL-basierten Implementierung verbessern, weil nicht mehr einzelne Rechte für alle Subjekte vergeben werden müssen.

Nachdem in den bisher betrachteten Modellen keine (explizite) Modellierung der Erforderlichkeit und Zweckbindung gemacht wurde, soll im folgenden Abschnitt ein Modell betrachtet werden, dessen Ziel eine Formalisierung von Zugriffskontrolle hinsichtlich Aspekten der Privatheit und des Datenschutzes ist.

3.1.3.4 Ein Modell für Privatheit in Zugriffskontrolle

Simone Fischer-Hübner hat ein formales[1] Modell ausgearbeitet, das u.a. die Modellierung des Zwecks eines Zugriffs beinhaltet und damit eine Datenschutz-konforme Zugriffskontrolle sicherstellen soll [FiHü01, FHO98]. Es werden dabei neben Subjekten, Objekten und Zugriffsrechten u.a. modelliert:

- Rollen („user role“): Den Subjekten werden wie bei RBAC Rollen zugewiesen, wobei die Rollen hier die Verantwortlichkeiten bezüglich der Administration regeln
- Objektklassen („object-classes“): Jedes Objekt ist genau einer Objektklasse zugeordnet
- Aufgaben („tasks“): Jedes Subjekt hat genau eine (aktuelle) Aufgabe aus einer Menge von Aufgaben
- Zwecke („purposes“): Jede Aufgabe hat (genau) einen Zweck, jeder Objektklasse ist mindestens ein Zweck zugeordnet
- Einwilligung („consent“): Der Betroffene eines Zugriffs auf ein Objekt kann in die Verarbeitung für bestimmte Zwecke einwilligen

Mit Hilfe dieser Konzeptualisierung können nun in Form von Invarianten Anforderungen an ein Zugriffsschutzsystem formuliert werden. Um einen Datenschutz-orientierten Zustand zu erreichen, muss (etwas vereinfacht) erfüllt sein: „ein Subjekt hat nur dann Zugriff, wenn es für seine aktuelle Aufgabe erforderlich ist“ (Erforderlichkeit), sowie: „ein Subjekt hat nur dann Zugriff, wenn der Zweck seiner aktuellen Aufgabe in den Zwecken der Objektklasse des Objektes enthalten ist oder eine Einwilligung des Betroffenen für den Zweck der Aufgabe und das Objekt vorliegt“ (Zweckbindung).

Weitere Eigenschaften können in Form von Einschränkungen („constraints“) z.B. bezüglich der Erfassung bzw. Erzeugung von Daten festgelegt werden. Die Formalisierung ist ein „task-based privacy model“, wobei alle Aktionen durch Zustandsübergangsfunktionen beschrieben werden und wodurch eine Einhaltung der Datenschutz-Invarianten bewiesen werden können soll.

[1] Der Ansatz wird im folgenden aber nur informell und zusammenfassend beschrieben.

Das Modell bietet eine Integration von Datenschutz-relevanten Aspekten in die Zugriffkontrolle, insbesondere erscheint eine Formalisierung des Zwecks und der Erforderlichkeit eines Datenzugriffs sinnvoll und notwendig. Allerdings stellt sich in diesem sehr komplexen Schema die Frage, wie der Benutzer damit umgehen soll und wie z.B. die Menge der Aufgaben und Zweck in einem konkreten Szenario gewählt werden können. Auch ist eine Implementierung bisher nur im Betriebssystemumfeld verwirklicht, die Realisierung bei einer dezentralen Verwaltung von Benutzerprofilen erscheint unklar.

3.1.4 Administration von Zugriffsrechten

In Hinblick auf Administration bei der Zugriffskontrolle muss festgelegt werden können, wer autorisiert ist, Zugriffsrechte für Objekte zu vergeben und in welcher Art und Weise dies geschehen soll. Diese Aufgabe stellt sich bei allen vorgestellten Verfahren.

Bei mandatorischer Zugriffskontrolle ist eine Festlegung von Sicherheitsklassen und Einordnung von Subjekten und Objekten in diese Klassen erforderlich. Dies wird von einem Administrator systemweit festgelegt. Bei einer Verwaltung von Benutzerprofilen ist eine systemweite Definition jedoch nicht unbedingt sinnvoll, da jedem Benutzer die Möglichkeit gegeben werden sollte, selbst die Zugriffsrechte auf sein Benutzerprofil geeignet festlegen zu können.

Bei diskreter Zugriffskontrolle gibt es verschiedene Möglichkeiten [SaSa94]: zentralisierte (von einem Administrator festgelegt), hierarchische (anhand einer Organisationsstruktur), kooperative (Kooperation mehrerer Administratoren nötig), dezentrale (erlaubt Delegation von Rechten) oder Eigentümer-bestimmte (Besitzer eines Objekts bestimmt Rechte) Administration. RBAC bietet ein ähnliches Spektrum von Möglichkeiten, dabei können Rollen definiert werden, denen entsprechende Rechte zur Manipulation von Zugriffsrechten zugewiesen werden können.

Bei Verwaltung von Benutzerprofilen ist der Benutzer Eigentümer seines Profils und sollte die volle Kontrolle darüber haben, also insbesondere auch die Zugriffsrechte von Diensten auf seine Profil-Attribute bestimmen können. Daher erscheint eine Eigentümer-bestimmte Administration am zweckmäßigsten. Dabei stellen sich allerdings in der Praxis die folgenden Probleme:

- Benutzer brauchen detailliertes Wissen über das Zugriffsschutzsystem: wie werden Rechte festgelegt, was hat das für Auswirkungen etc. Die Rechte-Administration muss leicht zu erlernen und zu benutzen sein.
- Eine Kenntnis über die Subjekte ist erforderlich, Benutzer müssen die Dienste, die auf die Profilinformationen zugreifen wollen, in Hinblick auf die Vertrauenswürdigkeit einschätzen können.
- Rechte müssen nicht nur für einzelne Profil-Attribute, sondern auch für Kategorien oder Gruppen von Attributen definierbar sein; die gewünschte Granularität ist nicht immer vorgesehen.
- Eine einfache Zurückziehbarkeit vergebener Rechte ist oft nicht möglich.
- Die Festlegung von Rechten ist mit einem gewissen Aufwand verbunden, es ist wohl nicht realistisch, dies von jedem Benutzer zu erwarten.

Daher ist bei der Administration von Zugriffsrechten ein flexiblerer Ansatz notwendig, z.B. in Form einer Aushandlung von Rechten des Benutzerprofilagenten mit dem Dienstagenten, der einen Zugriff anfordert. Dies muss unter Berücksichtigung der Verwendung der Daten (Zweckbindung), unter dem Grundsatz der Datensparsamkeit und transparent für den Benutzer erfolgen. Mit den bestehenden Verfahren ohne zusätzliche Unterstützung ist dies nicht in sinnvoller Weise realisierbar.

3.1.5 XML-basierte Verfahren

Seit der Entwicklung von XML zu einem wichtigen Standard zum Datenaustausch in heterogenen Umgebungen sind einige XML-basierte Ansätze zu Zugriffskontrolle entstanden, die in diesem Abschnitt diskutiert werden. Es geht dabei darum, Zugriffsrechte in einem XML-Schema auszudrücken. Dies bietet sich insbesondere auch für die Verwaltung von Benutzerprofilen an, da XML-Dokumente sowohl Rechner-verarbeitbar als auch für menschliche Benutzer verständlich sind, und dies daher die Transparenz für Benutzer bei der Administration von Zugriffskontrolle verbessern kann.

3.1.5.1 Zugriffskontrolle für XML Dokumente

Das Ziel von XACL (XML Access Control Language, [KuHa00a, KuHa00b]) ist es, XML-Dokumente mit einem Zugriffsschutzmechanismus zu versehen, insbesondere auch um einen sicheres Aktualisieren von XML-Datenbeständen zu realisieren. Jedem XML-Dokument wird eine sogenannte „policy“ zugeordnet, die die Zugriffsrechte auf das Dokument in XACL in Form einzelner Regeln spezifiziert. Abb. 3.1 (nach [KuHa00a, KuHa00b]) zeigt ein Beispiel für eine solche Zugriffsregel.

```
<policy>
  <xacl>
    <object href="/profile/demographic/email"/>
    <rule>
      <acl>
        <subject>
          <uid>Amazon Agent</uid>
        </subject>
        <action name="read" permission="grant">
          <provisional_action name="log"/>
        </action>
        <action name="write" permission="deny"/>
      </acl>
    </rule>
  </xacl>
</policy>
```

ABBILDUNG 3.1: XACL Beispiel

Es gibt folgende Menge von Zugriffsrechten, hier „Aktionen“ genannt: read, write, create, delete. Es ist sowohl explizites Erlauben als auch Verbieten von Zugriffen für Teile von XML-Dokumenten vorgesehen. Die Zugriffe können mit einem Parameter versehen werden, der angibt, ob die Rechte nach oben bzw. unten im XML-Baum propagiert werden sollen. Es ist auch die Angabe einer Rolle als Subjekt möglich, damit können Zugriffsrechte vergleichbar zu RBAC spezifiziert werden.

XACL enthält als Erweiterung traditioneller Zugriffskontrolle so genannte „Provisional Actions“ [KuHa00b]. Zugriffe können dabei abhängig von einer „Provision“ gemacht werden, d.h. Zugriffsanfragen können mit einer zusätzlichen Bedingung verknüpft werden, die erfüllt sein muss, um den Zugriff zu erlauben. Bisher ist folgende Menge von Provisions vorgesehen: write, create, delete, transform, log, verify, encrypt, wobei die drei letzteren folgende Semantik haben:

- „log“: Protokollierung der Authorisierungsentscheidung und/oder Zugriffen
- „verify“: Überprüfung des Zugreifers mittels einer digitalen Signatur

- „encrypt“: Aktion ist nur erlaubt, wenn das zu lesende Objekt verschlüsselt wird

Die Menge der Provisions kann erweitert werden. Damit könnten sich z.B. Bedingungen wie „Leseoperation wird nur gestattet, wenn der Zugreifer einer Nutzungsbedingung zustimmt“ oder „Zugriff auf Kreditkartendaten werden nur erlaubt, wenn die Daten verschlüsselt übertragen werden“ realisieren lassen.

Zu XACL vergleichbare Systeme sind der Ansatz von Damiani et.al. [DVPS00], sowie Author-X [BCF01], wobei bei letzterem der Schwerpunkt mehr auf einer Integration von Benutzer-Authentifikation, Verschlüsselungsverfahren und dem Ablauf liegt, als der Spezifikation von Zugriffsrechten. XACL oder eines der anderen Verfahren könnte in unserem Szenario interessant sein, da die Benutzerprofile in Abschnitt 2.1.1.3 als XML-Dokument modelliert wurden. Aus der Sicht der Zugriffskontrolle bieten sie aber im Wesentlichen nur ein anderes Realisierungsschema bekannter Verfahren, mit der Ausnahme der Provisions bei XACL.

Alle angesprochenen Projekte benutzen zur Adressierung von Teilen von XML-Dokumenten den W3C Standard XPath [XPa99]. Dabei wird die hierarchische Struktur von XML abgebildet, indem einzelne Ebenen im XML-Baum durch ein „/“-Symbol getrennt werden. Ein Beispiel: Um die Email-Adresse im Profil in Abb. 1 anzusprechen, müsste das Objekt „/profile/demographic/email“ (vgl. Beispiel in Abb. 6) adressiert werden. Damit können beliebige Teile eines XML-Dokumentes adressiert werden und somit eine flexible Granularität von Objekten, die auch bei der Verwaltung von XML-Benutzerprofilen notwendig ist, erreicht werden.

Die erwähnten Bestrebungen fließen in einen Standard *Extensible Access Control Markup Language* (XACML) bei OASIS ein (vgl. xml.coverpages.org/xacml.html). Dies ist insbesondere eine Fortführung von XACL.

XACML legt – wie die bereits erwähnten Ansätze – ein XML-Vokabular fest, um Autorisierungs-Regeln für XML-Dokumente zu spezifizieren. Dabei können auch Bedingungen formuliert werden und Festlegungen zur Kombination und Auswertung von Regeln getroffen werden. Das Ergebnis einer Auswertung ist dabei immer entweder „allow“ oder „deny“. Es sind auch Bedingungen möglich wie „this data is only viewable if accessed over HTTPS“.

Allerdings sind die „Provisions“ von XACL nicht mehr explizit enthalten, sondern in einer allgemeinen Form, die eine bilaterale Vereinbarung zwischen zwei Komponenten die Einhaltung einer Policy notwendig macht.

3.1.5.2 Digital Rights Management

Vergleichbar mit den soeben angesprochenen Ansätzen zur Zugriffskontrolle auf XML-Dokumente sind Verfahren des *„Digital Rights Managements“*. Digital Rights Management (DRM) behandelt die Spezifikation von Rechten, Bedingungen und Konditionen für den Gebrauch digitaler Inhalte [Xrml00]. Das Ziel ist es, den Handel von digitalen Medien zu fördern und dabei die Urheber-Rechte der Rechteinhaber zu schützen. Es gibt mehrere (konkurrierende) Ansätze, darunter *Open Digital Rights Language (ODRL)* (www.odrl.net) und *eXtensible rights Markup Language (XrML)* (www.xrml.org, [Xrml00]).

Die Projekte behandeln dabei insbesondere auch die Prozesse und erforderlichen Software-Komponenten für eine sichere – und unabstreitbare – Durchführung eines Kaufes und der Auslieferung von multimedialen Inhalten, was in dem hier betrachteten Szenario eher nicht so wichtig ist. Es ist aber auch eine Methodik enthalten, um Zugriffsrechte in einer XML-Sprache ausdrücken zu können. Die Zugriffsrechte beinhalten bei DRM u.a. „play“, „display“ oder „print“, sind aber erweiterbar. Ein

wichtiger und interessanter Punkt dabei ist, dass Zugriffsrechte mit Einschränkungen („constraints“) versehen werden können. Damit lassen sich u.a. folgenden Arten von Restriktionen formulieren:

- nach Benutzern oder Benutzergruppen
- nach zeitlichen Einschränkungen (z.B. nur über einen gewissen Zeitraum)
- nach räumlichen Gesichtspunkten (z.B. nach Länder) oder auch nach IP-Adressen etc.

Dies ist auch bei Benutzerprofilverwaltung interessant. Damit lassen sich eventuell Einschränkungen wie „Lesen ist erlaubt, die Information darf aber nicht weitergegeben werden“ formulieren. Enthalten sind auch Modelle zur Abbildung von Rechteinhabern sowie der Administration von Rechten.

Die Ansätze für DRM sind zwar sehr auf multimediale Inhalte ausgerichtet, z.B. durch Integration von „Watermarking“-Mechanismen zur unsichtbaren Kennzeichnung von digitalen Medien, Ideen wie die Formulierung von Nutzungs-Restriktionen lassen sich aber auch für die Verwaltung von Rechten in Bezug auf Benutzerprofile einsetzen.

3.1.5.3 XML Tickets

Fujimura et.al. beschreiben in [FNS99, FuNa98] ein XML-Schema, mit dessen Hilfe man „elektronische Tickets“ modellieren kann (vgl. Beispiel in Abb. 3.2, aus [FNS99]). Ein *XML Ticket* ist dabei ein digitales Medium, das ein bestimmtes Recht des Inhabers des Tickets garantiert. Als Anwendungsgebiete werden u.a. Software Lizenzen, Zugriffsrechte für Ressourcen oder Eintrittskarten genannt.

```
<SignedDescription>
  <Ticket typeID="eventTicket" ticketID="001234">
    <IssuerID fingerprint="..."/>
    <OwnerID fingerprint="..."/>
    <Validity>
      <NumberOfTimes>ONCE</NumberOfTimes>
      <ValidPeriod>2001-10-03</ValidPeriod>
    </Validity>
    <View resource="http://ticket.ntt.co.jp/ticket1.gif"/>
    <Promise>
      <Place>Boston Symphony Hall</Place>
      <Seat>H24</Seat>
    </Promise>
  </Ticket>
  <Signature>...</Signature>
</SignedDescription>
```

ABBILDUNG 3.2: XML Ticket

Ein XML Ticket definiert eine Zusicherung des Ausstellers des Tickets, dass der Ticket-Inhaber ein spezifiziertes Versprechen oder Recht besitzt. Der Ansatz schlägt ein „general-purpose digital ticket framework“ vor, bei dem verschiedene Charakteristika von Tickets wie Anonymität, Übertragbarkeit, Verfahren bei der Einlösung des Tickets und einmalige oder mehrfache Gültigkeit abgebildet werden können. Das Ticket wird vom Aussteller digital signiert. Insbesondere diese Signierung der XML Tickets ist für eine Benutzerprofilverwaltung interessant, da damit fälschungssichere Rechte, z.B. zum Zugriff auf bestimmte Profilattribute, ausgegeben werden könnten.

3.1.5.4 FIRM

Einen Ansatz, der zwar nicht XML verwendet, aber von der Zielsetzung und Ausrichtung her vergleichbar zu den XML-basierten Verfahren zu DRM ist, bietet das Projekt *FIRM (Framework for Interoperable Rights Management)* [RW97a, RW97a, Rös97]. Dabei wurde ein Modell für Sicherheit und Zugriffskontrolle in offenen Umgebungen mit Hilfe von elektronischen Verträgen zwischen Kommunikationspartnern entwickelt. FIRM geht nicht von einem Informationszugriffs-Paradigma aus, sondern von einer Verwaltung von Beziehungen („relationsship management“) und der Anwendung eines Modells für (elektronische) Verträge darauf. Die zu schützenden Objekte werden von den Zugriffsrechten darauf getrennt, wobei letztere in Form von Beziehungsobjekten („relationsships objects“, auch „commpacts“ in dem Modell genannt) verwaltet werden, die Verweise auf „echte“ Verträge darstellen.

Das Framework besteht aus zwei Teilen:

- Domänen-unabhängige Sprache bzw. Objektmodell zur Repräsentation von generischen Prinzipien von Verträgen
- Format zur Spezifikation von (Domänen-abhängigen) Zugriffsrechten

FIRM ist damit selbst keine Sprache, um Zugriffsrechte auszudrücken, sondern stellt den Rahmen dafür bereit.

Die Möglichkeit einer konkreten Anwendbarkeit von FIRM in dem betrachteten Szenario ist nicht so einfach zu beurteilen, da das Modell sehr abstrakt gehalten ist, aber eine Idee, weniger von Zugriffen und Zugriffsrechten, sondern mehr von der Konzeptualisierung der Beziehungen zwischen den Interaktions-Partnern auszugehen, ist vielleicht auch im Bereich Benutzerprofilverwaltung sinnvoll.

3.1.6 Bewertung

Wie bei der Herleitung der Anforderungen in Abschnitt 2.3 begründet, braucht man in dem betreffenden Szenario ein Zugriffskontrollsystem, um grundsätzlich die Fragestellung zu entscheiden, welcher Dienst auf welche Teile des Benutzerprofils wie zugreifen darf. Die verschiedenen Modelle und Strategien der Zugriffskontrolle haben dabei unterschiedliche Eigenschaften, auf die jeweils in den einzelnen Teilabschnitten hingewiesen wurden.

Ein Hauptproblem beim Einsatz der bestehenden Verfahren zur Zugriffskontrolle für dezentrale Benutzerprofile ist, dass voraus gesetzt wird, dass der Eigentümer der Daten – also der Benutzer – Zugriffsrechte für einzelne Subjekte, hier den Dienstagenten, vergibt. Alternativ werden wie bei RBAC Subjekte in Rollen oder Gruppen eingeteilt. Es stellt sich dabei jedoch die Frage, wie der Benutzer damit umgehen soll, da es nicht praktikabel erscheint, für alle Attribute und zugreifende Subjekte einzelne Zugriffsrechte oder Rollenzuordnungen festzulegen. Auch sind dem Administrator nicht unbedingt die Dienste bzw. deren Vorgehensweise bei der Verarbeitung persönlicher Daten, bekannt, so dass eine Zugriffskontrolle z.B. in Abhängigkeit der Verwendung der Daten mit den bestehenden Systemen nicht zu realisieren ist. In Abschnitt 3.2.5 wird das „Platform for Privacy Preferences Project“ vorgestellt, dass versucht, Datenschutzpraktiken von Firmen und Institutionen zu formalisieren.

Fehlende Aspekte im Detail bei den meisten Systemen sind insbesondere (vgl. dazu die Anforderungen in Abschnitt 2.3):

- Bei einem Zugriff auf personenbezogenen Daten ist der Zweck des Zugriffs sehr wichtig und muss abgebildet werden können
- Keine Modellierung des Rechts „Weitergabe von Daten“

- Keine Aushandlung oder Verhandlung von Rechten
- Keine Möglichkeiten, Zugriffe als „verpflichtend“ oder „optional“ zu kennzeichnen, wie dies z.B. bei Web-Formularen üblich ist
- Fehlende Optionen bei den Zugriffsrechten wie „wenn notwendig“ oder „wenn möglich“
- Alternativen oder Abhängigkeiten in Zugriffsanfragen sind nicht vorgesehen, z.B. für Zahlungsinformationen: „Zugriff auf Daten einer Bankverbindung (für Lastschrift) oder einer Kreditkarte“
- Keine (einfach zu realisierende) inhaltliche Gruppierung von Attributen (z.B. Strasse und Hausnummer)
- Keine Möglichkeiten, eine notwendige Übertragung über eine gesicherte Verbindung abzubilden
- Keine Berücksichtigung von Aspekten des Identitätsmanagements, wie z.B. Anonymisierung oder Verwendung von Pseudonymen

Die betrachteten Zugriffsverfahren auf Basis von XML erscheinen prinzipiell gut geeignet, Zugriffsrechte für Benutzerprofile abzubilden, da Benutzerprofile in XML modelliert werden können und damit eine Zugriffskontrolle für XML-Dokumente anwendbar ist. Allerdings stellen sich dabei genauso wie bei den anderen Verfahren die angesprochenen Probleme. Eine Modellierung in einer XML-Sprache kann zu Absicherungs- und Transparenzzielen beitragen, z.B. durch eine digitale Signierung von in Form von XML Tickets ausgegeben Zugriffsrechten.

Eine Umsetzung von gesetzlichen Rahmenbedingungen ist zum Teil durch Zugriffskontrolle gegeben. Datensparsamkeit kann z.B. bei den meisten Verfahren durch eine Grundeinstellung „kein Zugriff“ und Gewährung expliziter Zugriffsrechte für vertrauenswürdige Subjekte realisiert werden. Allerdings ist eine Modellierung von Zweckbindung nur ansatzweise bei einigen Verfahren vorhanden.

Es fehlt auch an einer Integration in eine Umgebung mit autonom interagierenden Komponenten. Dabei spielen auch die Prozesse und Abläufe beim Datenzugriff eine stärkere Rolle als bei statisch festzulegenden Zugriffsrechten vorgesehen. Bei einer Aushandlung zwischen Agenten kann sich der Kontext eines Zugriffs ändern, was von einem Administrator von Zugriffsrechten kaum vollständig berücksichtigt werden kann. Gerade im Agenten-Bereich wurden Fragen von Zugriffskontrolle und Privatheit bisher wenig untersucht.

Interessante Aspekte bei den vorgestellten Verfahren hinsichtlich einer dezentralen Verwaltung von Benutzerprofilen unter Beachtung von Privatheit sind insbesondere:

- Integration eines formalen Modells für Zweckbindung und andere bei Betrachtung von Privatheit relevanter Aspekte in dem Ansatz von S. Fischer-Hübner
- Zugriffkontrolle in Abhängigkeit von Bedingungen, wie z.B. der Protokollierung von Zugriffen in Form von „Provisional Actions“ bei XACL
- Signatur von Rechten bei XML Tickets

Ein Zugriffskontrollsystem kann nur wenig zu Identitätsmanagement oder Anonymisierung beitragen. Dazu wurden Anwendungen entwickelt, die im folgenden Kapitel behandelt werden.

3.2 Privacy Enhancing Technologies (PET)

3.2.1 Kategorisierung

Privacy Enhancing Technologies (PET)[2] sind Anwendungen, die die Privatheit von Benutzern und deren Daten im Internet verbessern sollen. Üblicherweise werden diese Systeme in folgende Gruppen aufgeteilt [Cra00a]:

- Verschlüsselungs- und Filtersysteme
- Verfahren zur Anonymisierung und Pseudonymisierung
- „Policy Tools“: Anwendungen, die es Diensteanbeitern und Benutzern erlauben, Praktiken im Bezug auf Privatheit von Diensten offen zu legen und auszuwerten

Diese Anwendungen erscheinen geeignet, zumindest einen Teil der Anforderungen aus Kapitel 2.3 zu erfüllen. Anonymisierung, Pseudonymisierung und Verschlüsselung sind wichtig für Vertraulichkeit und die Erfüllung der Identitätsziele. Policy Tools können zu allen Anforderungen, insbesondere auch zu Absicherung- und Transparenzzielen beitragen.

Im Folgenden soll daher genauer untersucht werden, inwieweit diese Anwendungen geeignet sind, die Problemstellung der Gewährleistung der Privatheit bei dezentraler Verwaltung von Benutzerprofilen zu lösen.

3.2.2 Verschlüsselungs- und Filtersoftware

3.2.2.1 Verschlüsselung

Mit Hilfe von *Kryptographie* (Verschlüsselungsverfahren) ist man in der Lage, Daten in eine Form zu transformieren, die ein Abhören durch Angreifer bei der Kommunikation über offene Netzwerke unmöglich, oder zumindest unpraktikabel aufwändig macht [Kyus98]. Kryptographie soll also eine gesicherte Übertragung über unsichere Netze ermöglichen. In unserem Szenario betrifft dies z.B. die Kommunikation der Benutzerprofil- und Dienstagenten oder auch den Datentransfer vom Browser des Benutzers zu einem (Server-seitigen) Dienst zur Profilverwaltung. Eine Verschlüsselung bei der Kommunikation macht die anderen besprochenen Mechanismen, z.B. zur Zugriffskontrolle, nicht überflüssig, sondern wäre ergänzend dazu.

In der Kryptographie gibt es zwei verschiedene Methoden zur Verschlüsselung:

- *asymmetrische* Verschlüsselung *(Public-Key-Verfahren)*: eine Nachricht wird mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und somit ist (nur) der intendierte Empfänger in der Lage ist, die Nachricht zu dekodieren (mit seinem privaten Schlüssel)
- *symmetrische* Verschlüsselung *(Private-Key-Verfahren)*: es wird nur ein einziger Schlüssel für Ver- und Entschlüsselung verwendet wird

Symmetrische Verfahren haben das Problem des Schlüsselaustausches: Wie kommen alle Kommunikationspartner in den Besitz des notwendigen Schlüssels? Für eine Anwendung im Internet werden daher oft asymmetrische Verfahren verwendet, dessen Nachteil ein größerer Aufwand bei der Kodierung bzw. Dekodieren im Vergleich zu symmetrischen Verfahren ist.

[2] Ein deutscher Begriff für PET, wie etwa „datenschutzfördernde Techniken“, hat sich (noch) nicht durchgesetzt, daher wird in dieser Arbeit der englische Begriff beibehalten.

Bekannte und recht verbreitete Anwendungen von Verschlüsselung sind *Pretty Good Privacy (PGP)* [Gar94] für die Verschlüsselung von Emails, sowie „Secure Socket Layer“. *Secure Socket Layer (SSL)* [FKK96] ist ein von Netscape entwickeltes Verfahren zur gesicherten Übertragung zwischen Web-Browser und -Server, das asymmetrische und symmetrische Kryptographie kombiniert. Mit Hilfe eines Public-Key-Verfahrens wird dabei ein Sitzungsschlüssel generiert, der dann für die (symmetrische) Verschlüsselung der eigentlichen Datenübertragung verwendet wird. Insbesondere Kreditkartendaten werden im WWW oft über SSL übertragen. Benutzer haben (zu Recht) mehr Vertrauen in SSL im Vergleich zu ungesicherten Verbindungen.

Eine Verschlüsselung durch SSL (bzw. das etwas allgemeinere Transport Layer Security (TLS) oder vergleichbarer Verfahren) kann eine vertrauliche Übertragung gegenüber unbefugten Dritten von Daten, z.B. vom Speicherort der Benutzerprofile zum Dienst, der die Daten nutzt, sicherstellen. Dies löst somit einen Aspekt der Vertraulichkeitsanforderungen. Allerdings muss auf der anderen Seite durch die benötigten Zertifikate und digitale Signaturen die Identität des Benutzers aufgedeckt werden, so dass die Privatheit in Hinblick einer anonymen Kommunikation abnimmt.

3.2.2.2 Filtersoftware

Eine weitere Anwendungsklasse von Privacy Enhancing Technologies ist Filtersoftware, die versucht, die technischen Möglichkeiten zur Verfolgung von Benutzeraktionen z.B. über Cookies oder „Web Bugs“[3] zu verhindern. So genannte „Cookie Cutters“ erlauben es, die Verwendung von Cookies zu verhindern bzw. zu kontrollieren und können damit die Privatheit von Benutzern etwas verbessern. Allerdings gibt es auch andere Möglichkeiten als mit Hilfe von Cookies, das Benutzerverhalten nach zu verfolgen und aufzuzeichnen, so dass diese Werkzeuge nur bedingt geeignet sind und keinesfalls vollständig für den Schutz der Privatheit dienen können.

In diesen Bereich von Werkzeugen fallen auch Anwendungen zum Filtern von Internet-Inhalten, in erster Linie zum Jugendschutz (Child Protection Software), was aber in dem hier betrachteten Szenario der Privatheit bei dezentraler Verwaltung von Benutzerprofilen nicht relevant ist.

3.2.3 Identifikation und Authentifikation

Die in diesem Abschnitt erläuterten Konzepte Identifikation und Authentifikation gehören zwar nicht direkt zu Privacy Enhancing Technologies, werden aber hier etwas näher betrachtet, weil sie auch bei der Verwaltung von Benutzerprofilen eine wichtige Rolle spielen.

Oftmals werden die folgenden Begriffe missverständlich verwendet, so dass zunächst deren genaue Definition geklärt werden soll (aus: [Gar02], S.120):

- „Identification: associating an identity with a subject“
- „Authentification: establishing the validity of something, such as an identity“
- „Authorization: associating right or capabilities with a subject“

Diese Konzepte haben miteinander zu tun, sind aber klar voneinander trennbar. Ein Beispiel für den Unterschied zwischen Identifikation und Authentifikation: Um alkoholische Getränke zu kaufen muss man sich nicht identifizieren, sondern sich authentifizieren, beispielsweise als über 18-jährig.

In dieser Arbeit geht es in erster Linie um Autorisation, die anderen Aspekte werden am Rande betrachtet.

[3] Web Bugs sind kleine, für den Benutzer nicht sichtbare, Bilder auf Web-Seiten, deren Zweck es ist, den Zugriff auf Web-Seiten von Benutzern verfolgen zu können.

3.2.3.1 Identifikation, Biometrie und Privatheit

Ein sehr wichtiger Aspekt bei der Verwaltung von Benutzerprofilen ist die Identifikation des Benutzers, z.B. gegenüber seinem Benutzerprofilagenten. Es muss eine eindeutige und korrekte Zuordnung von Benutzern zu Benutzerprofilen und deren Rechte sichergestellt werden. Dabei werden *biometrische Verfahren* zur Identifikation von Personen in Zukunft vermutlich eine große Rolle spielen. Diese Verfahren beinhalten z.B. Iris- oder Netzhaut-Scanner, Fingerabdruckleser, Gesichts-, Sprech-, oder Schrifterkennung [Wir99]. Bei einer Verwaltung von Benutzerprofilen ist es wichtig, eine eindeutige und fälschungssichere Identifikation von Benutzern zu gewährleisten, um einen Missbrauch von Profilen und personenbezogenen Daten auszuschließen. Unter dem Gesichtspunkt der Privatheit und des Datenschutzes stellen diese Systeme sowohl ein Möglichkeit dar, den Grad der Datensicherheit zu erhöhen, als auch ein Risiko für das informelle Selbstbestimmungsrecht des Einzelnen. Man unterscheidet bei Identifizierungsverfahren die folgenden Bereiche [Köh99b]:

- „Wissen“: Identifikation durch Eingabe eines Passworts o.ä.
- „Besitz“: z.B. Verwendung einer Smart- oder Chipkarte
- „Sein“: basiert auf physiologischen oder verhaltenstypischen Charakteristika des Benutzers (Biometrie)

Die letztgenannten, biometrischen Verfahren sind einerseits besonders gut für eine Identifikation des Benutzers bei der Verwaltung seines Profils geeignet, weil sie sich nur schwer von Unbefugten fälschen oder kopieren lassen. Allerdings ist eine Identifikation auf Basis von Biometrie andererseits auch bedenklich hinsichtlich Privatheit, da immer eine eindeutige Zuordnung zur natürlichen Person gegeben ist und dies auch Möglichkeiten von Missbrauch bietet.

Ein Hauptproblem der biometrischen Verfahren ist es derzeit noch, dass sie technisch noch nicht ausgereift sind und eine hohe Fehlerquote aufweisen können. Man unterscheidet dabei den Anteil der fälschlich zurückgewiesenen Benutzer (False Rejection Rate, FRR) und den Anteil der falsch vom System zugelassenen Benutzer (False Acceptance Rate, FAR). FRR und FAR sind i.d.R. abhängig voneinander, d.h. eine Einstellung des Systems zur Minimierung von FRR hat oft eine Erhöhung von FAR zur Folge. Für einen sicheren Einsatz eines biometrischen Verfahrens ist es notwendig, dass die Fehlerraten möglichst nahe an den Idealwert Null herankommen, was die bestehenden technischen Systeme mit Fehlerraten bis zu 20% [Sur02] noch nicht erreichen. Daher könnte der Einsatz von Biometrie auch eine falsches Gefühl für Sicherheit beim Benutzer erzeugen, welches (noch) nicht gegeben ist.

Biometrische Merkmale unterliegen als personenbezogenen Daten den Bestimmungen der Datenschutzgesetze (vgl. Abschnitt 2.2.2) und sind daher auch dementsprechend zu behandeln. Besonders wichtig ist es auch, vertrauenswürdige Endgeräte bereitzustellen [PPSW99].

Für die Zukunft ist zu erwarten, dass die technischen Möglichkeiten weiter verbessert werden und der Einsatz von Biometrie zur Identifikation bei der Verwaltung von Benutzerprofilen eine größere Rolle spielen könnte.

3.2.3.2 Authentifizierung, Zertifikate und digitale Signaturen

Bei allen Public-Key-Verfahren stellt sich folgendes, fundamentale Problem: Woher weiß man, dass der öffentliche Schlüssel wirklich demjenigen gehört, der behauptet dessen Eigentümer zu sein? Eine Lösung dafür ist die *Zertifizierung* von Schlüsseln. Ein *digitales Zertifikat* ist dabei eine Bescheinung

einer unabhängigen und vertrauenswürdigen dritten Partei, der *Zertifizierungsstelle* (certification authority, CA), die die Korrektheit der im Zertifikat enthaltenen Daten, also insbesondere der Name und andere Daten des Inhabers des Zertifikats und die Zuordnung des öffentlichen Schlüssels zu diesem Zertifikat, korrekt sind. Diese Bescheinigung erfolgt mit Hilfe einer *digitalen Signatur*.

Dazu wird eine Nachricht vom Absender mit seinem privaten Schlüssel verschlüsselt, so dass der Empfänger durch Anwendung des dazugehörigen öffentlichen Schlüssels prüfen kann, ob die Nachricht wirklich von dem Inhaber des privaten Schlüssels – bzw. der betreffenden digitalen Signatur – stammt. Es gibt auch im XML-Umfeld dazu einen entsprechend Standard, *XML Signature* [XSig02], um XML Dokumenten mit einer digitalen Signatur zu versehen. Dies könnte in unserem Szenario verwendet werden, um das (mit XML modellierte) Benutzerprofil bzw. oder Teile davon zu signieren.

Die digitale Signatur ist seit 2001 in Deutschland rechtsgültig, d.h. rechtlich ist eine digitale Signatur unter bestimmten Voraussetzungen einer eigenhändigen Unterschrift gleichgestellt. Die Anforderungen, die dabei gestellt werden, sind im Signaturgesetz, sowie der Signaturverordnung geregelt. In welchen Fällen die Gleichstellung zutreffen soll, ist im Rahmen des „Gesetzes zur Anpassung der Formvorschriften des Privatrechts und anderer Vorschriften an den modernen Rechtsgeschäfteverkehrs“ vom 1.8.2001 festgelegt worden. Es ist daher zu erwarten, dass die digitale Signatur an Bedeutung und Verbreitung gewinnt, was auch die Voraussetzung wäre, dass digitale Signaturen für eine Authentifizierung der (End-)Benutzer in unserem Szenario eingesetzt werden können.

Dazu ist der Aufbau einer *Public Key Infrastructure (PKI)* nötig, was ein allgemeiner Begriff für eine technische (und organisatorische) Infrastruktur für das Ausstellen, Verteilen und Verwalten von (öffentlichen und privaten) Schlüsseln für asymmetrische Kryptographie und den dazugehörigen Zertifikaten ist.

Mit Hilfe einer digitalen Signatur kann auch die Zurechenbarkeit gewährleistet werden, was zu den Absicherungszielen gehört. Z.B. könnte so ein E-Commerce Agent die Bestellung eines Benutzers durch den Nachweis einer entsprechend digital signierten Nachricht belegen. Darüber hinaus bieten digitale Signaturen auch eine Möglichkeit, die Integrität von Nachrichten zu gewährleisten.

Als weiteres, konkretes Beispiel für ein Authentifikations-System sei das auf symmetrischer Verschlüsselung basierende *Kerberos* [SNS98] erwähnt. Es dient der Authentifikation von Subjekten für den Zugriff auf verschiedene Netzwerkdiensten, sowie der Generierung von (zeitlich beschränkten) Sitzungsschlüsseln. Ein Benutzer oder eine System-Komponente – bei Kerberos „Principal“ genannt – authentifiziert sich dazu bei einem zentralen Kerberos-Server und erhält dabei ein „Ticket“, das nur für einen Dienst (z.B. Datei-Server) gültig ist. Der Kerberos-Server muss dazu entsprechend abgeschirmt und geschützt werden. Im Gegensatz zu einem Zugriffsausweis der Zugriffskontrolle dient ein Kerberos-Ticket nur der authentifizierten Nutzung eines Dienstes [Eck01]. Nachteilig bei dem Versuch der Anwendung von Kerberos auf das betrachtete Szenario ist die Abhängigkeit von einem zentralen Server.

Es gibt auch einen Ansatz, Authentifizierungs-Dienste mit Hilfe von XML zu beschreiben, nämlich die *Security Assertion Markup Language (SAML)* (www.oasis-open.org/committees/security/). Das Ziel dabei ist es, die Interoperabilität von E-Commerce Anwendungen durch Bereitstellung eines XML-Schemas zum Austausch von Authentifizierungs- und Authorisierungsinformationen zu verbessern. Insbesondere geht es dabei auch um die Spezifikation der Delegation von Rechten, um Benutzer zu authentifizieren [Wag01]. Neben der Unterstützung der oben erläuterten Methoden zur Verschlüsselung durch digitale Zertifikate, braucht man unabhängig davon Verfahren zur Authentifizierung von Benutzern und Agenten in unserem Szenario. Eine E-Commerce Anwendung muss z.B. zweifelsfrei überprüfen können, welche Person eine Bestellung abgegeben hat. Auf der anderen Seite braucht der Benutzerprofilagent eine Möglichkeit zur Kontrolle, dass Profildaten wirklich an denjenigen Dienst übermittelt werden, für den sie bestimmt sind.

3.2.4 Anwendungen zur Anonymisierung

Als nächstens sollen in diesem Abschnitt Möglichkeiten zur Anonymisierung untersucht werden, die Aspekte der Identitäts- und Vertraulichkeitsanforderungen betreffen.

3.2.4.1 Anonymisierungs-Proxies

Eine relativ einfache Möglichkeit, anonym auf Web-Seiten zuzugreifen, wird durch *Anonymisierungs-Proxies* wie z.B. www.anonymizer.com realisiert. Der Benutzer kann dabei über ein Web-Formular eine Web-Adresse eingeben, die dann vom Proxy abgerufen wird. Damit greift der Benutzer (nur) auf den Proxy, statt direkt auf den Dienst zu und kann somit „Datenspuren“ (z.B. seine IP-Adresse in den Log-Dateien des Web-Servers) vermeiden. Zusätzlich entfernt der Proxy weitere potentiell personenbezogenen Daten (wie z.B. Cookies) in den Headern der Web-Anfrage. Der Anonymisierungs-Proxy muss dabei vertrauenswürdig sein.

Mit Hilfe von Anonymisierungs-Proxies kann das Identitätsziel einer anonymen Kommunikation und eine Verbesserung der Vertraulichkeit erreicht werden. Dies könnte für eine anonymisierte Übermittlung von Profildaten genutzt werden, so dass z.B. Interessen ausgewertet werden können, ohne dass eine Zuordnung zu Web-Zugriffen oder anderen Aktionen eines Benutzers gemacht werden kann.

3.2.4.2 Crowds

Ein etwas anderer Ansatz zur Anonymisierung ist *Crowds* [ReRu97]. Dabei werden die Web-Anfragen einzelner Benutzer in einer Menge von Crowds-Benutzern „versteckt“. Eine Web-Anfrage wird nicht an den betreffenden Web-Server direkt, sondern an einen zufällig ausgewählten, anderen Crowds-Benutzer – bzw. dessen Crowds-Client, „Jondo“ genannt – geschickt. Die Anfrage durchläuft damit zunächst mehrere Jondos anderer Teilnehmer und ein Diensteanbieter kann Anfragen nicht mehr einem bestimmten Benutzer zuordnen.

Ein Vorteil von Crowds ist es, dass keine zentrale Komponente wie etwa ein Anonymisierungs-Proxy erforderlich ist. Auch könnten in unserem Szenario die Benutzerprofilagenten die Rollen der Jondos einnehmen. Im Vergleich zu den anschließend auf dem Mix-Konzept basierenden Verfahren bietet Crowds eine bessere Performance. Allerdings kann ein Angreifer, der den Kommunikationsverkehr abhört, durch eine zeitliche Verkettung von Nachrichten eine Verfolgung von Nachrichten erreichen. Crowds bietet also aus kryptographischer Sicht keine perfekte Anonymisierung. Auch erzielt Crowds nur eine Sender- und keine Empfängeranonymität. Ein Nachteil ist auch, dass eine genügend große Menge von (aktiven) Benutzern vorhanden sein, um eine Anonymität zu gewährleisten.

3.2.4.3 Mix-Konzept

Viele Systeme in der Literatur basieren auf dem *Mix-Konzept* [PPW88, Pfi90], das auf David Chaum zurückgeht [Cha81]. Dabei wird die Kommunikationsbeziehung zwischen Sender und Empfänger einer Nachricht durch Knoten im Netzwerk, so genannte *Mixe* verborgen, die eine Verkettung von Nachrichten verhindern. Ein Mix speichert dabei genügend viele Nachrichten von verschiedenen Absendern, kodiert diese um, so dass keine direkte Zuordnung mehr zu den ursprünglichen Nachrichten mehr möglich ist, und verändert die Reihenfolge der ausgehenden Nachrichten. Dabei entstehen Verzögerungszeiten in den Mixen, was auch notwendig ist, um zu verhindern, dass Angreifer, die die Kommunikationsbeziehungen abhören, aus der zeitlichen Abfolge von (eingehenden und ausgehenden) Nachrichten Rückschlüsse ziehen können.

Das Umkodieren einer Nachricht in einem Mix geschieht mit Hilfe asymmetrischer Verschlüsselung, wobei jeder Mix im Netzwerk die Nachricht – bzw. dessen äußere Schicht – mit seinem privaten Schlüssel entschlüsselt und an den nächsten Mix weiterschickt („layered“ Public-Key Verschlüsselung). Dazu muss der Sender die zu mixende Nachricht entsprechend vorbereiten, d.h. mit den öffentlichen Schlüsseln der nachfolgenden Mixe verschlüsseln.

Falls nicht genügend Nachrichten vorhanden sind, um eine Zuordnung von eingehenden und ausgehenden Nachrichten an einem Mix zu verhindern, müssen „dummy“ Nachrichten erzeugt werden, um die Verzögerungszeit in den Mixen bzw. die Laufzeit einer Nachricht zu minimieren, da man ansonsten warten müsste, bis eine für die Anonymisierung ausreichenden Menge von Nachrichten vorhanden ist. Es sollte auch sichergestellt werden, dass alle versendeten Nachrichten gleich groß sein, damit Außenstehende durch die Größe einer Nachricht nicht auf die Entfernung zum Empfänger schließen oder eine Zuordnung von ein- und ausgehenden Nachrichten vornehmen könnten.

Die Mix-Systeme können eine Unbeobachtbarkeit und Verdecktheit bei der Kommunikation erzielen. Der Nachteil davon ist, dass die mehrfache asymmetrische Verschlüsselung jeder Nachricht sehr aufwändig ist. Es gibt einige Ausprägungen dieser Systeme mit unterschiedlichen Eigenschaften [Kes00], so dass eine Abwägung zwischen Performance und kryptographischer Sicherheit erfolgen kann.

Das Mix-Konzept bietet einen allgemeinen und gut untersuchten Ansatz von Unbeobachtbarkeit und Verdecktheit beim Nachrichtenaustausch in offenen Umgebungen. Es könnte damit die Kommunikation zwischen Benutzerprofil- und Dienstagenten anonymisiert werden. Neben der schlechten Performance, stellt sich auch das Problem, dass einzelnen Nachrichten in den Mix-Stationen verzögert werden, was bei synchroner Kommunikation nicht immer zumutbar ist.

Zu bemerken ist auch, dass anonymisierte Kommunikation konträr zu Verschlüsselungsverfahren mit digitalen Zertifikaten ist. So hat z.B. ein E-Commerce Agent bei einer Bestellung durch eine anonymisierte Nachricht keinen Nachweis über den Auftraggeber der Bestellung mehr. Auch stellt sich die Frage, wie der Benutzer mit Anwendungen dieser Art umgehen soll, da eine grundsätzliche anonyme Kommunikation wohl nicht sinnvoll ist und somit der Benutzer auswählen muss, wie mit welchem Dienst kommuniziert wird. Dies kann z.B. nicht von der Verwendung der Daten und dem Zweck des Zugriffs abhängig gemacht werden.

3.2.4.4 Realisierung von Mixen

Ein Nachteil der Verfahren, die auf dem beschriebenen, „klassischen“ Mix-Konzept beruhen, ist, dass sie für Email- bzw. asynchrone Kommunikation entwickelt wurden. Daher sind sie wegen der Verzögerungszeiten in den Mixen nicht für Echtzeit-Bedingungen, wie es z.B. für Community-Unterstützungssysteme erforderlich ist, geeignet. Die meisten Realisierungen des Mix-Prinzips nutzen daher ein leicht modifiziertes Konzept [FeMa98], als Beispiele seien im folgenden „Onion Routing“ und in Abschnitt 3.2.4.6 das Produkt „Freedom“ der kanadischen Firma Zero-Knowledge System vorgestellt. Eine weitere Implementierung ist der „Java Anon Proxy“ der TU Dresden (http://anon.inf.tu-dresden.de/).

Beim *Onion-Routing* (www.onion-router.net) [GRS99] wird zunächst eine „Onion“ gesendet, die zu einem Aufbau eines anonymen Kanals zwischen Sender und Empfänger dient. Eine Onion ist dabei eine Nachricht, die mit Schichten asymmetrischer Verschlüsselung versehen ist, wie in Abschnitt 3.2.4.3 erläutert wurde. Die Onion enthält außerdem die Adresse des nächsten Onion-Routers, der hier die Funktion einer Mix-Station einnimmt, sowie Schlüsselmaterial, das für eine nachfolgende Etablierung eines anonymen Kanals verwendet wird.

Bei dem Lauf durch das Netz wird nun die Onion Schritt für Schritt abgebaut, d.h. im jeweiligen

Onion-Router entschlüsselt, und gleichzeitig der anonyme Kanal aufgebaut. Hierzu merkt sich jeder Onion-Router, woher er eine Onion erhalten hat und wohin er die verbleibende Onion geschickt hat und zusätzlich ein Kennzeichen, eine so genannte „Pfad-ID“. Empfängt ein Onion-Router Daten für eine bestimmte ID, so verschlüsselt er die erhaltenen Daten mit einem symmetrischen Kryptosystem, dessen Schlüssel er aus dem Schlüsselmaterial der Onion gewonnen hat [FeMa98].

Eine Zurechenbarkeit von Nachrichten zu Sender oder Empfänger ist nicht möglich, da (sehr) viele Nachrichten die Menge der Onion-Router passieren und damit keine Zuordnung von eingehenden zu ausgehenden Nachrichten vorgenommen werden kann.

Mit Hilfe eines der vorgestellten Anonymisierungsverfahren kann die Kommunikation sowohl gegenüber unbefugten Dritten als auch gegenüber nicht vertrauenswürdigen Diensten anonymisiert werden. Somit kann – in Kombination mit anderen Verfahren – eine Erfüllung des Identitätsziels einer anonymen Kommunikation sowie von Vertraulichkeitszielen erreicht werden.

3.2.4.5 Pseudonymität

Oftmals ist eine vollständige Anonymität nicht wünschenswert, da es z.B. für Personalisierungsdienste erforderlich sein kann, dass sich Benutzer gegenüber einem Dienst identifizieren, damit einzelne Aktionen, z.B. Webzugriffe, einem Benutzer zuordnen zu können. Um diese Funktionalität zu ermöglichen, werden Pseudonyme verwendet, vergleiche dazu auch die Ausführungen in Abschnitt 2.1.1.2. Eine Anwendung dazu ist *Lucent Personal Web Assistant (LPWA)* [GGK+99].

Benutzer identifizieren sich an einem LPWA Proxy Server. Dieser generiert automatisch und für den Benutzer transparent einen Benutzernamen und Passwort für jede Site, die der Benutzer besucht. Damit können Web-Sites Personalisierungs-Funktionen wahrnehmen und z.B. auch Benutzerinteressen auswerten, es ist aber nicht möglich, zwischen den Pseudonymen für verschiedene Server eine Beziehung oder einen Rückschluss auf die Identität des Benutzers herzustellen. Der LPWA Proxy stellt auch einen Anonymisierungs-Dienst dar, so dass Betreiber Web-Site auch nicht aus der IP-Adresse des Clients auf die Identität des Benutzers schließen können. Die Anonymisierungs-Funktionalität ist vergleichbar zu den eher einfachen Systemen der Anonymisierungs-Proxies (vgl. Abschnitt 3.2.4.1) und keine Realisierung eines Mix-Konzeptes.

LPWA oder vergleichbare Pseudonymisierungs-Dienste können Beschränkungen von Anonymisierungstools aufheben, ohne eine Preisgabe der Identität des Benutzers nötig zu machen. Es wird eine Implementierung des Persona-Konzepts bereitgestellt, mit Hilfe dessen Benutzer ein Identitätsmanagement durchführen können. Es wird das Problem gelöst, dass sich Benutzer nicht selber für Dienste virtuelle Identitäten erzeugen müssen, weil bei sehr vielen Benutzerkennungen der Überblick verloren gehen kann. Als weitere Besonderheit kann durch Erzeugen einer Pseudo-Email Adresse eine Herausgabe einer (wahren) Email-Adresse vermieden werden. Eine Email-Adresse ist oft für Dienste erforderlich, erlaubt aber meist auch (eigentlich unerwünschte) Rückschlüsse auf die Identität des Benutzers bzw. birgt die Gefahr unerwünschter Werbe-Emails.

LPWA ist auch deshalb interessant, da nicht nur eine Anonymisierung auf Kommunikations-Ebene erfolgt, sondern auch Dienst-spezifische Benutzernamen und Email-Adressen erzeugt werden können

3.2.4.6 Freedom

Eine interessante Anwendung ist das Produkt *Freedom* von Zero-Knowledge Systems (ZKS) [BSG00, SaHa00]. Dabei wird versucht, einen Mix-basierten Anonymisierungsansatz mit einem etwas weitergehenden Konzept für Pseudonyme als bei LPWA, zu verbinden. Jeder Freedom-Benutzer erhält da-

bei eine Menge von Pseudonymen, hier „Nyms“ genannt, die zur Identifizierung bei verschiedenen Diensten genutzt werden können. Solange ein Benutzer Nachrichten unter dem gleichen Pseudonym durch das Freedom-Netz schickt, sind diese verkettbar, die „wahre“ Identität des Absenders ist aber geschützt [FeBe00]. Das Haupteinsatzgebiet der Nyms dürfte das pseudonyme Senden und Empfangen von E-Mail sein [FeBe00], das Grundprinzip lässt sich allerdings auch auf andere Aspekt des Identitätsmanagement anwenden. Bei der Erzeugung von Nyms sind auch Funktionen zur Abrechnung für den Anbieter des Freedom Anonymisierungs-Dienstes mit integriert. Der Ablauf dabei ist [SaHa00]:

- Vertraulicher Kauf einer Seriennummer, die zum Erwerb eines so genannten „Nym-Tokens“ berechtigt
- Übermittlung der Seriennummer an einen Token-Server einer vertrauenswürdigen dritten Partei, der (nach erfolgreicher Überprüfung) ein oder mehrere Nym-Tokens an den Benutzer aushändigt
- Erstellung eines Nyms auf einem Nym-Server mit Hilfe des Nym-Tokens

Bei diesem Nym-Erzeugungsprozess sind mehrere Parteien beteiligt, damit für den Anbieter des Dienstes eine Zuordnung eines Nyms zu einer Identität (z.B. über das verwendete Zahlungsverfahren) ausgeschlossen ist, und der Benutzer nicht einer einzigen Partei vertrauen muss.

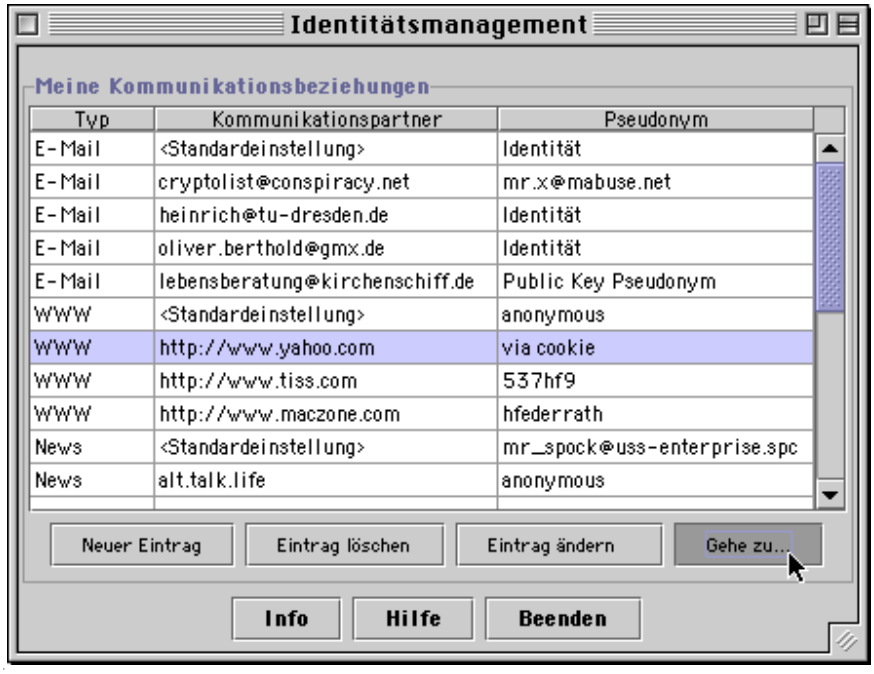

ABBILDUNG 3.3: Beispiel einer Oberfläche eines Identitätsmanagers

Eine Kombination von Anonymisierungsverfahren mit einem Konzept für Pseudonyme ist ein Baustein zur Realisierung von Identitäts- und Vertraulichkeitszielen. Auch ist das Verfahren zur Abwicklung des „Kaufes eines Pseudonyms“ interessant.

Ein grundsätzliches Problem ist die Handhabung der Pseudonyme, die bei Freedom, LPWA und vergleichbaren Ansätzen von den Benutzern selbst verwaltet werden müssen. Abb. 3.3 aus [FeBe00] zeigt dazu ein Beispiel, wie sich die Verwaltung der Pseudonyme für einen Benutzer präsentieren könnte. Es stellt sich dabei die Frage, ob eine Oberfläche dieser Art praktikabel für den (End-) Benutzer ist. Kommerzielle Anwendungen für Identitätsmanagement im Internet werden in Abschnitt 3.3 betrachtet.

Außerdem wollen Benutzer nicht unbedingt manuell auswählen, sondern die Verwendung eines Pseudonyms oder den Zugriff auf ihre Daten soll abhängig von der Verwendungen der Daten oder den Datenschutzpraktiken eines Dienstes sein, wozu das nachfolgend behandelte „Platform for Privacy Preferences Project“ beitragen kann.

3.2.5 Platform for Privacy Preferences Project (P3P)

„Policy Tools“ sollen Benutzer informieren, welche personenbezogenen Daten über sie gesammelt und wie diese Informationen verwendet werden und somit insbesondere auch die Transparenzziele der Anforderungen erfüllen. Als wichtigster und interessantester Vertreter dieser Art von Privacy Enhancing Technology wurde der Standard *Platform for Privacy Preferences (P3P) Project* [P3P02] vom World Wide Web Consortium (W3C) unter Beteiligung von Firmen wie AOL, IBM und Microsoft entworfen.

3.2.5.1 Motivation

Viele Web-Sites haben Informationen über ihre Praktiken in Bezug auf Datenschutz und Privatheit in einer *Datenschutzerklärung* (engl. privacy policy) veröffentlicht. Diese Datenschutzerklärungen sind jedoch oft umfangreich und für einen Laien schwer verständlich und durchschaubar. P3P soll es Betreibern von Web-Sites ermöglichen, ihre Erklärungen in einem standardisierten Schema in Maschinen-lesbarer Form auf ihrem Server abzulegen. Die Formulierung der P3P Erklärung soll durch entsprechende Werkzeuge unterstützt werden. Der Benutzer greift mit einem Client-seitigen Benutzeragenten, z.B. einem P3P-fähigen Web-Browser, auf eine Web-Site zu. Der Server antwortet mit seiner P3P Erklärung, die der Benutzeragent dann auswerten kann. Benutzer können damit ihre eigenen Präferenzen hinsichtlich Privatheit und Datenschutz mit den Erklärungen von Web-Sites abzugleichen, um zu entscheiden, ob die Site besucht werden soll oder nicht. Dazu werden vom Benutzer festgelegte oder ausgewählte Regeln ausgewertet.

P3P besteht aus folgenden Komponenten:

- Protokoll zum Austausch der P3P Erklärung
- Schema zur Beschreibung der Datenschutzerklärung
- Datenformat und Vokabular
- Regeln zur Auswertung der Erklärungen

Für den Austausch der P3P Erklärung gibt es zwei Möglichkeiten. Entweder es wird eine Erweiterung im HTTP Header verwendet oder der Dienstanbieter legt eine „Policy Reference“ Datei an der URL „/p3p.xml“ des Web-Servers ab. Diese Policy Reference Datei enthält dann Informationen darüber, welche P3P Datei für welchen Teil des Servers gültig ist. Eine Möglichkeit zur detaillierten Aushandlung der Datenschutzerklärung zwischen Client und Server war zwar in den ersten P3P Entwürfen angedacht, wurde aber letztendlich nicht im Standard aufgenommen, um diesen zunächst möglichst schlank zu halten.

3.2.5.2 Schema der Datenschutzerklärung

Eine P3P Datenschutzerklärung kann in einer XML-Form dargestellt werden, was im diesem Teilabschnitt anhand des folgenden Beispiels (nach [P3P02]) näher erläutert wird:

```
<POLICY xmlns="http://www.w3.org/2000/12/P3Pv1"
    discuri="http://www.catalog.example.com/Privacy/PrivPractShop.html"
    opturi="http://catalog.example.com/preferences.html">
 <ENTITY>
 <DATA-GROUP>
   <DATA ref="#business.name">CatalogExample</DATA>
   <DATA ref="#business.contact-info.postal.street">40 Lincoln Ave.</DATA>
   <DATA ref="#business.contact-info.postal.city">Birmingham</DATA>
   <DATA ref="#business.contact-info.postal.stateprov">MI</DATA>
   <DATA ref="#business.contact-info.postal.postalcode">48009</DATA>
   <DATA ref="#business.contact-info.postal.country">USA</DATA>
   <DATA ref="#business.contact-info.online.email">cat@example.com</DATA>
  </DATA-GROUP>
 </ENTITY>
 <STATEMENT>
  <CONSEQUENCE>
    We tailor our site based on your past visits.
  </CONSEQUENCE>
  <PURPOSE><tailoring/><develop/></PURPOSE>
  <RECIPIENT><ours/></RECIPIENT>
  <RETENTION><stated-purpose/></RETENTION>
  <DATA-GROUP>
   <DATA ref="#dynamic.cookies">
    <CATEGORIES><state/></CATEGORIES>
   </DATA>
   <DATA ref="#dynamic.miscdata">
    <CATEGORIES><preference/></CATEGORIES>
   </DATA>
  </DATA-GROUP>
 </STATEMENT>
<STATEMENT>
  <CONSEQUENCE>
    We use this information when you make a purchase.
  </CONSEQUENCE>
  <PURPOSE><current/></PURPOSE>
  <RECIPIENT><ours/></RECIPIENT>
  <RETENTION><stated-purpose/></RETENTION>
  <DATA-GROUP>
   <DATA ref="#user.name"/>
   <DATA ref="#user.home-info.postal"/>
   <DATA ref="#user.home-info.telecom.telephone"/>
   <DATA ref="#user.home-info.online.email"/>
   <DATA ref="#dynamic.miscdata">
    <CATEGORIES><purchase/></CATEGORIES>
   </DATA>
  </DATA-GROUP>
 </STATEMENT>
 <STATEMENT>
```

```
  <CONSEQUENCE>
    At your request, we will send you carefully selected marketing
    solicitations that we think you will be interested in.
  </CONSEQUENCE>
  <PURPOSE>
   <contact required="opt-in"/>
   <customization required="opt-in"/>
   <tailoring required="opt-in"/>
  </PURPOSE>
  <RECIPIENT required="opt-in"><ours/><same/></RECIPIENT>
  <RETENTION><stated-purpose/></RETENTION>
  <DATA-GROUP>
   <DATA ref="#user.name" optional="yes"/>
   <DATA ref="#user.business-info.postal" optional="yes"/>
   <DATA ref="#user.business-info.telecom.telephone" optional="yes"/>
  </DATA-GROUP>
 </STATEMENT>
</POLICY>
```

Jede P3P Erklärung wird mit einem <POLICY> Auszeichnungselement eingeleitet. Dabei können eine natürlichsprachliche Variante (Attribut „discuri“) der Erklärung, sowie eine URL einer Beschreibung für die Vorgehensweise für den Benutzer, um einer Datenerfassung zuzustimmen oder abzulehnen („opturi“), angegeben werden. Als nächstes wird die Entität beschrieben, die die personenbezogenen Informationen sammeln und verarbeiten möchte, wozu das hierarchisch aufgebaute Datenformat von P3P zur Bezeichnung der Felder verwendet wird.

Die P3P Erklärung besteht dann aus einem oder mehreren „Statements“, die jeweils folgende Teilbereiche enthalten:

- <CONSEQUENCE>: Natürlichsprachlicher Text, der dem Benutzer angezeigt werden kann
- <PURPOSE>: Zweck der Erfassung von Benutzerprofildaten, sehr wichtig aus Sicht der Privatheit, <current/> bedeutet dabei z.B. die aktuell durchzuführende Aufgabe, die in der Regel vom Benutzer initiiert wurde, z.B. eine Suchanfrage
- <RECIPIENT>: Empfänger der Daten, z.B. auch eine dritte Partei
- <RETENTION>: Verfahrensweise bei der Einbehaltung der personenbezogenen Daten, z.B. <stated-purpose/> (nur im Sinne des angegebenen Zwecks) oder <legal-requirement/>
- <DATA-GROUP>: Bezeichnung der Profilattribute, die vom Dienst benötigt werden
- <CATEGORIES>: Hinweise für den Benutzer (bzw. dessen Agenten), wie die Daten verwendet werden

Während <PURPOSE> den grundsätzlichen, recht allgemeinen, Zweck der Datenerfassung spezifiziert, ermöglicht <CATEGORIES> eine genauere Angabe zur Unterscheidung inhaltlicher Aspekte in den Regeln. Das obige Beispiel enthält drei Statements. Der erste sagt aus, dass die Web-Site versucht, mit Hilfe von Cookies die Seiten zu personalisieren. Das zweite Statement spezifiziert die benötigten Informationen bei einem Kauf. Das abschließende Statement regelt die Weitergabe einiger Daten, wobei mit „required = ...“ angegeben werden kann, ob der Benutzer die Möglichkeit hat, diesem Zweck mittels „opt-in“ oder „opt-out“ zustimmen. „opt-in“ bedeutet dabei, dass der Benutzer

explizit einer Datenerfassung zustimmen muss, bei „opt-out“ muss der Benutzer einer Verarbeitung von Daten explizit widersprechen. Diese Unterscheidung ist bei Verwaltung personenbezogener Daten grundsätzlich sehr wichtig. Aus Sicht der Privatheit wird in der Regel eine vorgegebene Einstellung im Sinne von „opt-in“ gefordert.

Das P3P Datenformat bildet die wichtigsten Elemente eines Benutzerprofils ab, es stellt keine vollständige Struktur eines Benutzerprofils dar, kann aber erweitert werden, um Attribute abbilden zu können, die nicht im P3P Standard vorgesehen sind.

3.2.5.3 Regeln zur Auswertung

Diese Datenschutzerklärung kann von Benutzeragenten anhand von Regeln ausgewertet werden. Zur Definition der Regeln ist eine Sprache *A P3P Preferences Exchange Language (APPEL)* [APPEL02] vom W3C vorgesehen ist. In Abb. 3.4 (aus [APPEL02]) ist eine Beispielregel abgebildet.

```
<appel:RULE behavior="block" description="Service collects
              personal data for 3rd parties">
 <p3p:POLICY>
  <p3p:STATEMENT>
    <p3p:DATA-GROUP>
     <p3p:DATA>
      <p3p:CATEGORIES appel:connective="or">
       <p3p:physical/>
       <p3p:demographic/>
       <p3p:uniqueid/>
      </p3p:CATEGORIES>
     </p3p:DATA>
   </p3p:DATA-GROUP>
   <p3p:RECIPIENT appel:connective="or">
    <p3p:same/>
    <p3p:other-recipient/>
    <p3p:public/>
    <p3p:delivery/>
    <p3p:unrelated/>
   </p3p:RECIPIENT>
  </p3p:STATEMENT>
 </p3p:POLICY>
</appel:RULE>
```

ABBILDUNG 3.4: Beispiel APPEL Regel

Der Wert des XML-Attributs „behavior“ legt fest, wie sich der Agent verhält, wenn die Regel zutrifft. Dies ist dann der Fall, wenn die angegebenen, mit boolschen Operatoren verknüpfbaren, Bedingungen in der auszuwertenden P3P Erklärung erfüllt sind. Es gibt in APPEL 1.0 die folgenden drei „Behaviors“, es ist dabei keine eigene Erweiterungsmöglichkeit vorgesehen:

- „request“: Die Erklärung ist in Ordnung in Bezug auf die Regel, mit der Empfehlung (für den Benutzer bzw. Benutzeragenten), dass auf die betreffende URL zugegriffen werden kann
- „block“: Die Erklärung ist nicht konform zu den Regeln, die Ressource soll nicht genutzt werden

- „limited“: Die Erklärung ist teilweise annehmbar, der Benutzeragent könnte beispielsweise mit einer Warnung für den Benutzer reagieren oder beim Zugriff möglichst wenig Header Daten in der HTTP-Anfrage übertragen

Außerdem kann mit einem optionalen Attribut „prompt“ angegeben werden, ob der Benutzer über die Auswirkung der Regel informiert werden soll. Das Attribut „prompt“ kann dabei die Werte „yes“ oder „no“ annehmen. Dies ist eine Vorgabe für den Benutzeragenten, einen entsprechenden Text an den Benutzer auszugeben. Mit Hilfe eines zusätzlichen (optionalen) Attributes „promptmsg“ kann der Text spezifiziert werden, der dem Benutzer angezeigt werden soll.

Mit Hilfe einer Menge solcher Regeln können die Präferenzen hinsichtlich Privatheit des Benutzers umgesetzt werden. Die Zielsetzung ist es, sinnvolle Regelmengen von vertrauenswürdigen Organisationen bereitstellen zu lassen, da die Entwicklung eigener Regeln wohl für den durchschnittlichen Endbenutzer nicht unbedingt praktikabel erscheint.

3.2.5.4 Einhaltung der Datenschutzerklärungen

Eine wichtige Fragestellung bei P3P ist es, die Einhaltung der Datenschutzerklärungen zu kontrollieren, um die Anforderungen der Absicherung (vgl. Abschnitt2.3) erfüllen zu können.

Ein Ansatz dazu ist es, die Praktiken von Unternehmen hinsichtlich Privatheit und Datenschutz von unabhängigen Institutionen untersuchen zu lassen. Im Internet gibt es Organisationen wie TRUSTe (www.truste.org), CPA WebTrust (www.webtrust.org) oder BBBOnLine (www.bbbonline.org), die dieses als Dienst anbieten. Diese Organisationen überprüfen dazu die Datenschutzpraktiken von Unternehmen und stellen bei Konformität der Erklärung bzw. Einhaltung gewisser Mindestanforderungen ein *Gütesiegel* (engl. (privacy) seal) aus. Teilnehmende Unternehmen dürfen dann durch eine entsprechende Grafik in ihren Web-Seiten auf das Gütesiegel hinweisen. Das Gütesiegel, bzw. ein Verweis darauf, kann in P3P integriert und somit Teil der Datenschutzerklärung werden (siehe Abb. 3.5).

```
<DISPUTES-GROUP>
  <DISPUTES resolution-type="independent"
    service="http://www.PrivacySeal.org"
    short-description="PrivacySeal.org">
   <IMG src="http://www.PrivacySeal.org/logo.gif">
   <REMEDIES><correct/></REMEDIES>
  </DISPUTES>
 </DISPUTES-GROUP>
```

ABBILDUNG 3.5: Gütesiegel in P3P

Die Eignung von Gütesiegeln zur Überprüfung der Einhaltung von Datenschutzerklärungen wird später genauer besprochen (siehe Abschnitt 4.5.5.2).

Das Element <correct/> im Auszeichnungselement <REMEDY> bedeutet dabei, dass bei Verstößen gegen die Erklärung Rechtsmittel durch die angegebene Organisation eingelegt werden. Auch unabhängig von Gütesiegeln kann eine Verletzung von Datenschutzerklärungen zu rechtlichen Folgen führen. Eine Integration der Prüfung von Datenschutzerklärungen ist insbesondere auch bei der Verwaltung von Benutzerprofilen sinnvoll und erforderlich.

3.2.5.5 P3P und Identitätsmanagement

Wie könnte jetzt P3P für das betrachtete Szenario verwendet werden? Wichtige Beiträge von P3P für eine Verwaltung von Benutzerprofilen sind unter anderem:

- Modellierung von Zweck der Datenerfassung
- Neben der Erfassung expliziter Profildaten wie Name, Email usw. werden auch dynamische Daten wie Einträge in Log-Dateien oder Cookies als Teil eines Benutzerprofils berücksichtigt
- Weitergabe von Daten an eine dritte Partei, dabei kann berücksichtigt werden, ob die dritte Partei vergleichbare Datenschutzpraktiken hat oder nicht
- Möglichkeit von „opt-in“ und „opt-out“ Optionen
- Möglichkeit der Verbesserung der Transparenz und Übersicht für den Benutzer durch die Auswertbarkeit von P3P Datenschutzerklärungen durch Benutzeragenten

Durch P3P ist zumindest ein Vokabular für den Austausch von Datenschutz Erklärungen und Präferenzen gegeben. Auch ist die Möglichkeit der Spezifikation von Regeln für die Formulierung von Zugriffspräferenzen viel versprechend, so dass nicht wie bei den meisten Verfahren zur Zugriffskontrolle explizite Zugriffsrechte für einzelne Dienste und Profilattribute festgelegt werden müssen.

Zur Unterstützung eines Identitätsmanagements bietet P3P weiterhin die Möglichkeit der Verwaltung verschiedener Identitäten über ein Persona-Konzept. Ein Persona wird dabei definiert als eindeutiger Identifikator für eine Menge von Werten der Datenelemente [APPEL02]. Benutzeragenten können dann verschiedene Ausprägungen von Profil-Attributen speichern und es Benutzern ermöglichen, zwischen den verschiedenen Personas bzw. Identitäten zu wechseln. Dazu kann in den APPEL-Regeln als Attribut „persona = ...“ angegeben werden. Auf diese Weise können unterschiedliche Präferenzen beim Zugriff auf z.B. berufliche oder private Email-Adressen spezifiziert werden.

Interessant ist die Erweiterungsfähigkeit von P3P. Neben der schon erwähnten Möglichkeit, das Datenschema zu erweitern, können auch explizit eigene Elemente mit Hilfe eines <EXTENSION> Auszeichnungselements in eine P3P Erklärung eingefügt werden. Damit würden sich eventuell für Identitätsmanagement notwendige Erweiterungen ohne Verletzung der P3P Syntax realisieren lassen. Auch wäre eine Einbindung der P3P Erklärungen in einen Agenten-basierten Ansatz denkbar, da das P3P Schema getrennt und unabhängig vom Protokoll ist, so dass letzteres durch ein für Agenten-Kommunikation besser geeignetes Aushandlungs- und Koordinationsprotokoll ersetzt werden kann.

Ein P3P Benutzeragent muss nicht als selbstständige Anwendung realisiert werden. Es bietet sich insbesondere die Integration in Web-Browser an. Auch ist eine Zusammenführung eines P3P Clients mit den in Abschnitt 2.1.2.4 beschriebenen Infomediaries nahe liegend. Ein um zusätzliche Funktionen erweiterter P3P-Client könnte in Zukunft die Rolle eines allgegenwärtigen Werkzeugs zum Identitätsmanagement einnehmen [BeKö00], insbesondere auch im Kontext mobiler Kommunikation.

Allerdings besteht dabei auch die Gefahr einer zu starken Abhängigkeit von einem Werkzeug zum Identitätsmanagement [BeKö00]. Zum Beispiel könnten durch fehlerhafte Realisierung von Sicherheitsfunktionen Angreifer die Identität des Benutzers annehmen oder „stehlen“, was wohl bei einer anderen, weniger Technik-unterstützten, Verwaltung von Benutzerinformationen nicht so leicht der Fall sein könnte. Auch könnten Benutzer durch die Bereitstellung technischer Möglichkeiten zum

Schutz ihrer Privatheit verleitet werden, mehr Informationen herauszugeben als eigentlich erforderlich oder sinnvoll wäre. Dadurch würde u.a. das von den rechtlichen Rahmenbedingungen (vgl. Abschnitt 2.2.2) geforderte Prinzip der „Datensparsamkeit“ untergraben. Auf diese rechtlichen und sozialen Aspekte wird im Folgenden nächsten Abschnitt noch etwas genauer eingegangen.

3.2.5.6 P3P und Datenschutz

Die Eignung von P3P zur Verbesserung von Datenschutz – insbesondere im Verbindung mit europäischen Rechtsvorschriften – wird zum Teil recht kontrovers diskutiert. Die meisten Literaturstellen (z.B. [Lan00, Cra00b, GrRo00]) beurteilen P3P dabei grundsätzlich positiv als geeignetes technisches Hilfsmittel zur Umsetzung bestehender Datenschutzrichtlinien:

- P3P kann eine bessere Transparenz für Benutzer erreichen
- Es lassen sich die „notice&choice“ bzw. „informed consent“ Prinzipien umsetzen (vgl. Abschnitt 2.2.2.3), und dadurch eine bessere Kontrolle für die Benutzer verwirklichen
- P3P ermöglicht eventuell einen „Wettbewerb von Anbietern“ in Bezug auf den besten Datenschutz für Kunden, dadurch könnte eine gewisse Selbstregulierung erzielt werden [Lan00]
- Vertrauen in Online-Transaktionen insgesamt könnte wachsen, wenn Benutzern aussagekräftige Informationen und Wahlmöglichkeiten hinsichtlich des Datenschutzes von Web-Anbietern angeboten werden [Cra00b]

Allerdings wird bezweifelt, ob P3P alleine dem Nutzer ein ausreichendes Maß an Datenschutz garantieren kann [Sie01, Lan00, GrRo00]. P3P stellt dabei nur einen technischen Basisstandard zur Verfügung [Lan00]. Insbesondere werden folgende Punkte kritisiert [Epic01, Kuh01]:

- Benutzern könnte ein falsches Gefühl für Sicherheit suggeriert werden, im Endeffekt werden vielleicht mehr personenbezogene Daten herausgegeben, als es ohne P3P der Fall wäre [Epic01]
- Unsicherheit darüber was passiert, wenn Benutzer der Datenschutzerklärung nicht oder nur teilweise zustimmt [Kuh01].
- Probleme bei der Unterstützung der Kontrollrechte des Benutzers auf Auskunft, Berichtigung, Sperrung und Löschung von Daten [Kuh01], sowie der Durchsetzung der Policies [Epic01].
- Eine vermeintliche, aber nicht ausreichende Lösung könnte übergreifende gesetzliche Regelung in weite Ferne rücken lassen [Sie01].

Es erscheint klar, dass man noch ergänzende, wirksame Datenschutzkontrolle und präzise Rechtsnormen braucht [Lan00], und dass zusätzliche technische Unterstützung für die Benutzer neben P3P erforderlich ist. Der Internet Explorer Web-Browser von Microsoft hat P3P Funktionen integriert, daher erscheint es möglich, dass P3P in Zukunft noch eine größere Unterstützung durch Web-Sites erfahren könnte.

3.2.6 Bewertung

Zusammenfassend lässt sich festhalten, dass durch Verschlüsselungsmechanismen eine gesicherte Übertragung in offenen Systemen erzielt werden kann. Dies ist z.B. für den Transfer sensibler Daten zwischen einem Benutzer- und Dienstagenten erforderlich. Dazu ist auch eine Authentifizierung der Kommunikationspartner unerlässlich, was durch digitale Zertifikate realisiert werden kann. Andererseits dienen Anonymisierungsverfahren dazu, die Identität eines Benutzers zu verbergen und damit diesen für Privatheit wichtigen Aspekt umzusetzen. Verfahren zur Pseudonymisierung erlauben eine zur Erbringung bestimmter Dienste nötige Zuordnung einzelner Transaktionen zu einem Pseudonym ohne eine Aufdeckung weiterer Profilattribute des Benutzers. Damit können einzelne Privacy Enhancing Technologies bestimmte Aspekte der Anforderungen bei dezentraler Benutzerrprofilverwaltung erfüllen.

Ein interessanter und wichtiger Ansatz für die Modellierung von Datenschutzpraktiken und Präferenzen im Internet ist P3P, mögliche Beiträge von P3P zu einem Identitätsmanagement wurden schon im Abschnitt 3.2.5.5 („P3P und Identitätsmanagement“) erläutert. Auf bestehende Probleme und Unsicherheiten von P3P in Bezug auf Datenschutz wurde schon im vorausgegangenen Abschnitt 3.2.5.6 („P3P und Datenschutz“) eingegangen. Darüber hinaus fehlen im P3P Standard noch folgende Punkte:

- Keine Verbindung mit Anonymisierungs-Anwendungen und anderen Privacy Enhancing Technologies
- Das P3P Vokabular ist nicht ausreichend und muss erweitert werden, z.B. gibt es nur eine relativ kleine Menge pauschaler Zweckbestimmungen für die Festlegung des Zugriffszwecks [Kuh01]
- Keine vollständige Modellierung von Zugriffsrechte möglich, z.B. ist das „Schreiben in ein Benutzerprofil“ nicht abgedeckt
- Auch gibt es grundsätzlich keine Integration mit Zugriffskontrollsystemen für eine verfeinerte Festlegung von Zugriffsrechten

Ferner wurde P3P für das „Browsen von Web-Seiten“ entwickelt, d.h. für eine Benutzer-initiierte Anfrage an einen Dienst, der diese Anfrage beantwortet. Dabei wurde insbesondere auch auf die Eigenschaften von HTTP, z.B. die Berücksichtigung von Cookies, eingegangen. Es fehlt aber an Interaktions-Modellen für eine Kommunikation autonomer Komponenten, obwohl P3P schon als Basis einer Verhandlung von Diensten mit autonomen Benutzeragenten entwickelt wurde. Eine mehr auf Agenten bzw. autonome Komponenten bezogene Sichtweise ist auch bei den anderen betrachteten PET Ansätzen nötig.

Die Berücksichtigung von Verhandlung zwischen Diensteanbietern und Benutzeragenten wurde zwar nicht in den aktuellen Standard [P3P02] aufgenommen, könnte aber in zukünftigen Erweiterungen enthalten sein. Weitere interessante Aspekte, die für spätere P3P Versionen geplant sind, sind:

- Möglichkeit, eine Menge von P3P Erklärungen zur Auswahl anzubieten
- Unabstreitbarkeit von Vereinbarungen, z.B. über digitale Signaturen (bisher ist das Problem bei Personas, dass die Identitäten nicht bewiesen werden können)
- Automatischer Datentransfer

Grundsätzlich kann man zusammenfassen, dass P3P für den Bereich Identitätsmanagement und zur Unterstützung der Verwaltung von Benutzerprofilen sehr interessant ist, wobei einige Anpassungen und Erweiterungen wie beschrieben sinnvoll wären. Insbesondere ist dies dann der Fall, wenn sich dieser Standard für den Zugriff auf Web-Seiten durchsetzt und bewährt, da dabei gegebenenfalls Datenschutzerklärungen und -präferenzen übernommen werden können.

Ein wichtiger Punkt bei allen PET ist, dass die Ansätze eventuell zu kompliziert und aufwändig für durchschnittliche End-Benutzer sind. Ein Beispiel dazu ist die notwendige Vorgehensweise nur bei einer Erzeugung von Nyms im Freedom System (vgl. Abschnitt 3.2.4.6). Dazu kommt noch eine manuelle Verwaltung der Pseudonyme, die von einem Benutzer durchgeführt werden muss. Die Systeme müssen mehr auf die Bedürfnisse von Benutzern zugeschnitten werden, dazu gehört auch der Entwurf sinnvoller Benutzungsschnittstellen, z.B. die Untersuchung von geeigneten graphischen Benutzeroberfläche zur Unterstützung von Privatheit.

Des Weiteren muss bei den bestehenden Systemen der Benutzer die Wahl einer Identität bzw. Aktivierung einer anonymisierten Datenübertragung selbst vornehmen. Man braucht aber (auch) eine Anonymisierung der Art, dass bestimmte Profildaten automatisch anonymisiert übertragen werden. Auch ist eine technische Unterstützung von Privatheit in dem betrachteten Szenario eine Integration von Privacy Enhancing Technologies in Zugriffskontrollsysteme notwendig.

3.3 Anwendungen für Identitätsmanagement im Internet

Im Folgenden sollen Anwendungen für Identitätsmanagement im Internet aus zwei Gründen etwas genauer untersucht werden. Zum einen sind diese sehr nahe an dem hier betrachteten Szenario einer dezentralen Verwaltung von Benutzerprofilen und Wiederverwendung von Authentifizierungs- und Profilinformationen für verschiedene Dienste. Zum anderen haben diese Systeme aus Sicht der Privatheit einige interessante Eigenschaften.

3.3.1 Überblick

3.3.1.1 Architektur und Funktionalitäten

Bei den Grundlagen im Abschnitt 2.1.2 wurden schon einige Grundsätze bei der Verwaltung von Benutzerprofilen erläutert. Die Identitätsmanagementanwendungen setzen diese Aspekte in praktische Systeme um. Ein wichtiges Differenzierungsmerkmal von Identitätsmanagementanwendungen liegt im Grundkonzept, man unterscheidet dabei zwischen einer Server- und Client-basierten Architektur.

Identitätsmanagementsysteme stellen in der Regel einen Teil der folgenden Funktionalitäten zur Verfügung:

- Single Sign On (SSO)
- Verwaltung verschiedener Identitäten und Ermöglichung anonymer und pseudonymer Kommunikation
- Verwaltung von Benutzerprofilattributen und Zugriffsrechten darauf
- Automatisches Ausfüllen von Web-Formularen o.ä.
- evtl. Abwicklung von Zahlungen im Internet

Die grundlegendste Funktionalität des Identitätsmanagements ist der *Single Sign On (SSO)* Dienst (vgl. Abb. 3.6), wodurch wird eine Dienst-übergreifende Authentifikation realisiert wird, oftmals mit Hilfe von *HTTP-Redirect* oder einem vergleichbaren Mechanismus. Dabei wird nach einer Anfrage eines Benutzeragenten (Web-Browser) ohne Authentifizierungsinformation ein Redirect zum Server des Identitätsanbieters[4] gemacht. Dieser authentifiziert den Benutzer, z.B. mit Hilfe einer Eingabe von Benutzernamen und Passwort, und macht einen weiteren Redirect zurück zum ursprünglichen Dienst (Schritt 5) mit einem Authentifizierungstoken, das i.d.R. als Cookie realisiert ist. Der Identitätsanbieter wird dabei nur eingeschaltet, wenn der Benutzeragent nicht in Besitz des Authentifizierungstokens ist. Nach einer erfolgten Authentifizierung kann das Token für eine automatische Anmeldung bei weiteren Diensten genutzt werden (vgl. Schritt 6). (Die Antwort der Dienste auf die Benutzeranfrage wurde in der Grafik weggelassen.)

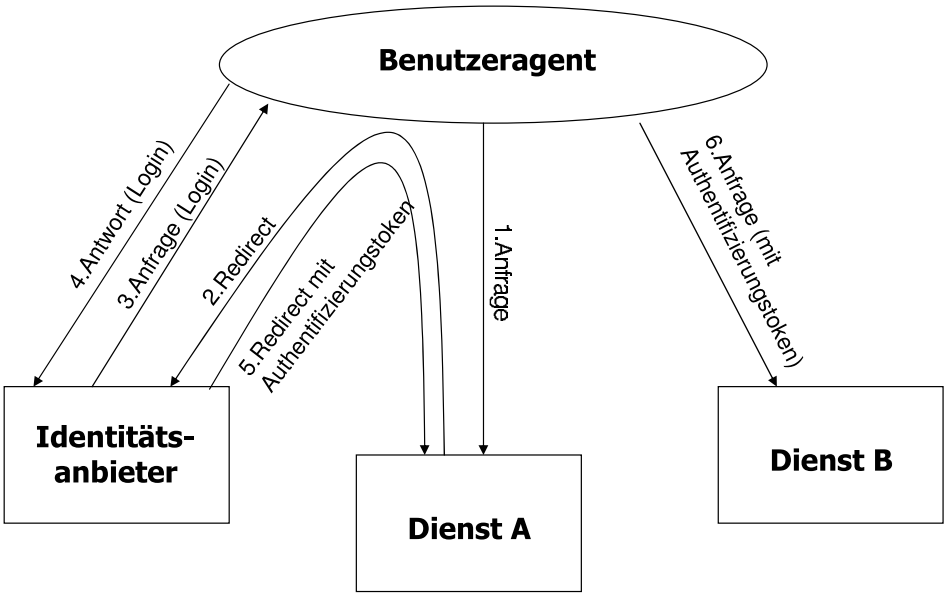

ABBILDUNG 3.6: Single Sign On Protokoll

Man muss dabei unterscheiden zwischen einer Identität des Benutzers (dies ist vereinfacht ein Teil seines Benutzerprofils und kann Authentifizierungsinformationen enthalten), einer Kennung bei einem Dienst (dies ist z.B. Benutzername und Passwort bei einem Web-basierten Dienst) und dem Authentifizierungstoken (dies soll eine lokale oder Dienst-übergreifende Authentifikation ermöglichen).

Der inhärente Schwachpunkt dieser Technik aus Sicht von Sicherheit und Privatheit liegt darin, dass alleine der Besitz des Authentifizierungstokens für eine Benutzeranmeldung ausreichend ist [Zeh02]. Eine bessere Lösung bietet z.B. ein Authentikator, wie er in Kerberos zum Einsatz kommt [SNS98]. Eine Authentifizierung mit HTTP-Redirect und Cookies birgt insbesondere zwei Sicherheitsrisiken:

- Abhören der übertragenen Daten, wenn dies nicht mit SSL o.ä. gesichert wird
- Gefahr der Verwendung von gestohlenen Authentifizierungscookies

[4] Der „Identitätsanbieter“ entspricht dem Benutzerprofilagenten in unserem Szenario, nur dass der Fokus mehr auf der Verwaltung der Identitäten und Authentifizierungsinformationen liegt.

3.3.1.2 Systeme und Einsatzgebiete

Dieser Abschnitt gibt einen Überblick über die wichtigsten Identitätsmanagementsysteme [Zeh02].

Microsoft .NET Passport (www.passport.com) ist ein Teil der .NET („dot net“) Infrastruktur. Es realisiert mit einem zentralen, Server-basierten Authentifizierungssystem einen SSO-Dienst basierend auf HTTP-Redirect wie oben gezeigt. Passport verwaltet eine sehr große Menge an Kennungen, da z.B. jeder Benutzer des kostenlosen Email-Dienstes Hotmail von Microsoft automatisch eine Passport-Kennung hat.

Ein zu .NET vergleichbares Framework ist *Sun ONE* (wwws.sun.com/software/sunone). Dabei sind auch Dienste und Komponenten zum Identitätsmanagement enthalten, u.a. wird ein *Sun Identity Server* (wwws.sun.com/software/products/identity_srvr/home_identity.html) angeboten.

Das *Liberty Alliance* Projekt (www.projectliberty.org) ist eine Vereinigung von ca. 120 Firmen wie AOL, Mastercard oder Sun Microsystems mit dem Ziel, einen offenen Standard als Grundlage zur Realisierung eines SSO-Dienstes und Möglichkeiten zum Austausch von Authentifizierungs-, Autorisierungs- und Benutzerprofilinformationen zu definieren. Der Ansatz sieht im Gegensatz zu .NET Passport eine verteilte, föderierte Benutzerprofilverwaltung vor.

Novell digitalme (www.digitalme.com) bietet einen Verzeichnisdienst, eine Verwaltung verschiedener Identitäten eines Benutzers mit zugehörigen Profilinformationen und Funktionalitäten wie einem Client-seitigen, automatischem Ausfüllen von Web-Formularen.

Extensible Name Service (XNS) (www.xns.org) stellt eine offene Plattform bereit, die eine Definition und Verwaltung von Identitäten und Beziehungen zwischen diesen erlaubt. Dies ist vergleichbar mit der Liberty Alliance Spezifikation. Darauf aufbauend wird ein SSO-Dienst und ein automatisches Ausfüllen von Web-Formularen realisiert.

Eine Client-basierte Lösung zum SSO bietet *PingID* (www.pingid.com). Es stellt ferner ein Open Source Framework für Entwickler bereit, mit dem eigene Anwendungen um Identitätsmanagement-Funktionalität erweitert werden können (vgl. www.pingid.org bzw. www.sourceid.org). PingID hat im Gegensatz zu den meisten anderen Anwendungen den Fokus stärker auf einer Benutzer-zu-Benutzer Beziehung, nur die Authentifikation wird über eine vertrauenswürdige dritte Partei abgewickelt. Dies ermöglicht eine Form von „persönlichem Identitätsmanagement“ [Dys02b].

Infospace (www.infospace.com) ist ein Client-basierter Ansatz (vgl. Abb. 2.3 im Grundlagenkapitel) und bietet dem Benutzer neben der Verwaltung von Profildaten, auch eine Dokumenten- und Geräteverwaltung und ermöglicht eine Synchronisation der Daten mit anderen Anwendungen wie dem Email-Programm Microsoft Outlook.

Neben einem SSO-Dienst stellt *Yodlee* (www.yodlee.com) eine Verwaltung von verschiedenen Online-Banking-Konten sowie kontenübergreifende zusammenfassende Informationen bereit. Yodlee hat den Schwerpunkt auf einer übersichtlichen Verwaltung von persönlichen Daten für den Benutzer, dazu kann die Menge der gespeicherten Attribute erweitert werden. Als interessante Besonderheit bietet Yodlee einen Versicherungsschutz auf finanziellen Verlust durch unautorisierten Transfer unter Verwendung der Yodlee-Konten mit einer Deckungssumme von US$100.000 je Benutzer und Schadensfall.

Weitere, noch wenig weit entwickelte Ansätze – zumindest in Bezug auf konkrete Anwendungen für Identitätsmanagement – sind *IDsec* (idsec.sourceforge.net) und die virtuellen Identitäten von *DotGNU* (www.dotgnu.org), einem Projekt mit Ziel, ein zu Microsoft .NET vergleichbares Open Source Framework zu realisieren. Die Architektur von IDsec ist in einem IETF (Internet Engineering Task Force) Draft Proposal spezifiziert (vgl. idsec.sourceforge.net/draft-zandbelt-idsec-01.txt).

Es gibt darüber hinaus einige Unternehmen, wie z.B. *Persona* (www.persona.com), die Produkte zur Personalisierung von Web-Seiten, zum Email Marketing oder der Unterstützung von Customer

Relationship Management (CRM), anbieten. Dabei werden auch SSO und andere Funktionalitäten eines Identitätsmanagements angeboten, z.T. auch Möglichkeiten für den Benutzer bzw. Kunden eine Verwendung von personenbezogenen Daten zu kontrollieren.

Open Privacy (www.openprivacy.org) realisiert in einem (Teil-)Projekt „User Content License“ u.a. ein sog. „Reversing the Privacy Policy Circle“. Dabei wird der Benutzeranfrage ein HTTP-Header angefügt, der Informationen über einen Copyright-Schutz von Benutzerdaten enthält. Damit sollen vor einer möglichen Datenübermittlung vom Client zum Server die Bedingungen eines Datenzugriffs festgelegt werden.

Ein ähnliches Vorgehen wird auch in einem Server-basierten P3P-Ansatz von IBM Research [AKSX03] verfolgt. Dabei soll nicht die Datenschutzerklärung des Dienstes am Client mit den Benutzerpräferenzen abgeglichen werden, sondern letztere werden zum Dienst gesendet. Aus Sicht der Privatheit ist dies allerdings eher bedenklich, da mit der Übertragung von Benutzerpräferenzen an einen Dienst bereits ein wesentlicher Verlust von Privatheit verbunden ist. Dies sollte daher bei der Verwaltung von Benutzerprofilen nicht verwendet werden.

Im Folgenden werden weitere Aspekte hinsichtlich Sicherheit und Privatheit bei dreien der Anwendungen dargestellt.

3.3.2 Sicherheit und Privatheit bei ausgewählten Systemen

Viele Untersuchungen beschränken sich bei der Betrachtung von Anwendungen für Identitätsmanagement auf die Funktionalitäten der Systeme. Für den Benutzer ist aber auch eine Untersuchung der Privatheit dieser Anwendungen sehr wichtig, da personenbezogene Daten verwendet werden. Dabei sind insbesondere eine Authentifikation und im Kontext dieser Arbeit auch Möglichkeiten zur Benutzerprofilverwaltung und Autorisation maßgeblich.

In diesem Abschnitt werden .NET Passport, das Liberty Alliance Projekt und digitalme von Novell näher betrachtet. .NET Passport hat die größte Verbreitung in der Praxis und war bereits einer Reihe von Angriffen ausgesetzt. Der offene, föderierte Ansatz der Liberty Alliance könnte in der Zukunft eine wichtige Rolle beim Identitätsmanagement im Internet spielen. digitalme ist deshalb interessant, weil es stärker als die beiden anderen Systeme darauf ausgelegt ist, Benutzerprofilinhalte und verschiedene Identitäten eines Benutzers zu verwalten.

3.3.2.1 Microsoft .NET Passport

.NET Passport bietet eine Dienst-übergreifende Verwaltung von Benutzer-Kennungen und einen SSO-Dienst. Ein Dienst kann sich bei Passport als „Partner-Site“ registrieren und muss dazu zwischen drei Sicherheitsstufen wählen:

- „Standard Single Sign-in“: keine speziellen Sicherheitsanforderungen
- „Secure Channel Sign-in“: vollständige Absicherung über SSL und gesicherte Cookie-Übertragung notwendig (dies behebt eine der in 3.3.1.1 genannten SSO-Sicherheitsrisiken)
- „Strong Credential Sign-in“: zusätzliche Verwendung einer vierstelligen PIN

Authentifikation Das Standard Single Sign-in Protokoll zur Authentifikation ist im Prinzip das oben erläuterte SSO-Protokoll mit HTTP-Redirect, als Authentifizierungstokens werden dabei Cookies benützt. Die verschlüsselten Tokens liegen in einem Microsoft-proprietären Format vor, daher ist eine direkte Interoperabilität mit anderen SSO-Diensten bei der Authentifikation nicht möglich.

Bei einer Abmeldung von Passport werden Skripte auf jeder Partner-Sites aufgerufen, die der Benutzer besucht hat, um den Benutzer auch dort abzumelden. Dies funktioniert allerdings in der Praxis nicht immer zuverlässig, so dass es sein kann, dass nach einer „Abmeldung“ trotzdem Authentifizierungscookies einzelner Dienste erhalten bleiben (vgl. Abb. 3.7 aus [Zeh02]). Dieses Abmeldekonzept ist daher unter Gesichtspunkten der Sicherheit und der Transparenz für den Benutzer mehr als fragwürdig.

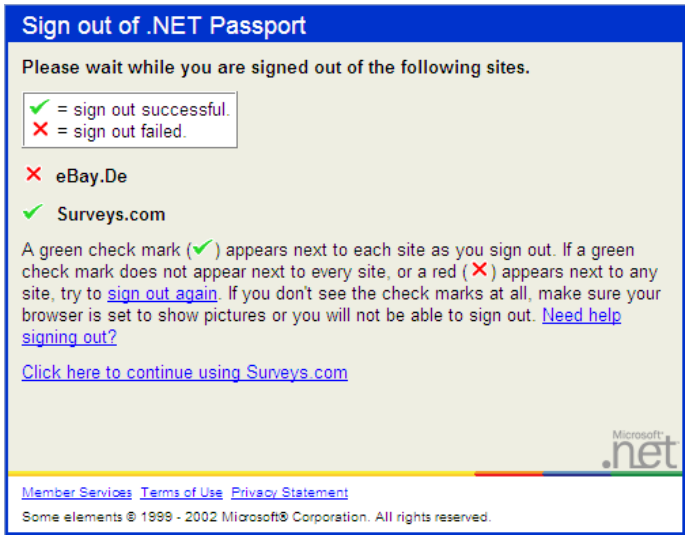

ABBILDUNG 3.7: Abmeldeproblem bei .NET Passport

Es bleibt weiterhin auf dem Rechner des Benutzers ein persistentes Cookie von .NET Passport gesetzt, welches die Email-Adresse des Benutzers speichert und diese bei einem weiteren Besuch in die Anmeldemaske einträgt. Dies ist nicht der Fall, wenn der Benutzer bei der Anmeldung eine – leicht zu übersehende – Option „Ich nutze einen frei zugänglichen Computer“ oder ähnliches explizit ausgewählt hat. Der genaue Text dieser Option in der Anmeldemaske ist u.a. abhängig vom verwendeten Web-Browser.

Die Authentifikation ist insgesamt also nicht gerade optimal bei Passport gelöst. Für 2003 ist die Unterstützung von Kerberos zur Authentifikation angekündigt, was einen Zugewinn an Sicherheit bringen könnte. Eine genauere Analyse der Abläufe bei An- und Abmeldung und den verwendeten Cookies findet sich in [Zeh02].

Autorisation Bei einer Neuanmeldung an .NET Passport kann man angeben, ob die Email Adresse oder weitere Registrierungsdaten (dies betrifft „Country/Region“, „State“, „ZIP Code“ und „Language“) mit anderen Diensten geteilt werden sollen, siehe den Ausschnitt aus der Anmeldemaske von Passport[5], Abb. 3.8. Eine Auswahl kann man später relativ leicht wieder rückgängig machen (mit Hilfe einer Funktion „Edit my .NET Passport profile“).

Eine Weitergabe von Benutzerdaten erfolgt an Passport Partner-Sites, eine Weitergabe an weitere Dienste wird in der Datenschutzerklärung ausgeschlossen. Eine genauere Unterteilung, an welche Dienste welche Informationen gegeben werden, ist nicht möglich. Auch wird der Benutzer nicht benachrichtigt oder in sonstiger Weise informiert, wenn neue Passport Partner hinzukommen und somit

[5]Die betreffende URL ist http://register.passport.net/reg.srf.

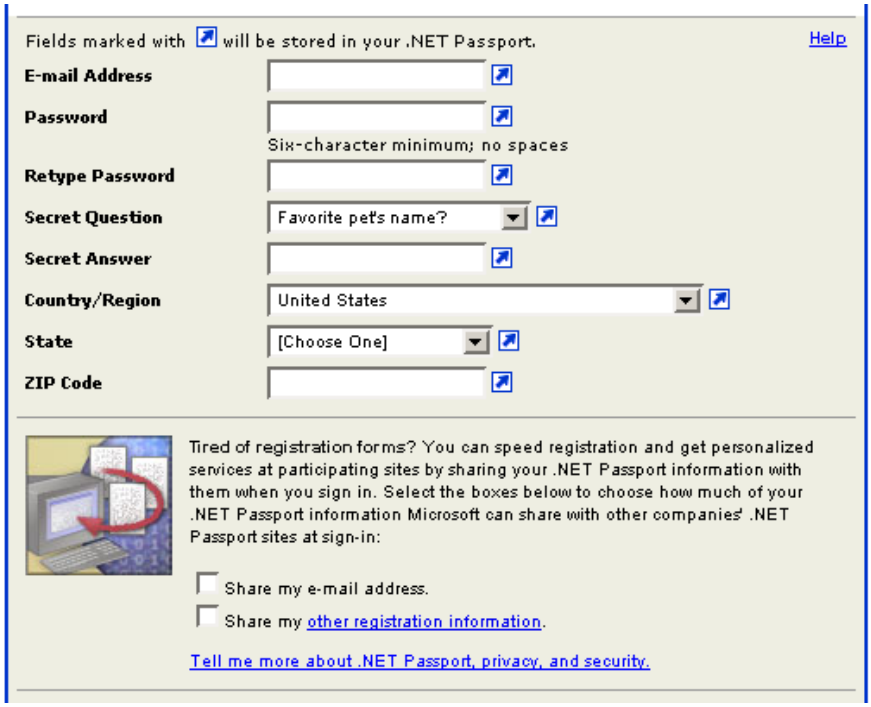

ABBILDUNG 3.8: Ausschnitt aus der Passport Anmeldemaske

Zugriff auf die Daten haben. Es ist auch nicht (direkt) ersichtlich, welche Sicherheitsstufe (siehe oben bei der Anbieterregistierung) eine Partner-Site verwendet, was ein wichtiges Kriterium zur Auswahl von Diensten für den Benutzer wäre.

Zusammenfassend sind die sehr spärlichen Mechanismen zur Autorisation bei Passport auf keinen Fall ausreichend und bieten dem Benutzer sehr wenig Kontrolle über die Verwendung seiner Daten. Es ist z.B. keine Differenzierung nach Dienst und Nutzungszweck möglich. Auch fehlt es an der Transparenz für den Benutzer, es wird kaum ersichtlich, an welche Dienste welche Informationen weitergegeben werden.

Sicherheitsprobleme Einige Sicherheitsprobleme von .NET Passport wurden von Kormann und Rubin in [KoRu00] aufgezeigt, auf die hier nicht im Detail eingegangen werden kann. Darunter fallen Probleme des Schlüssel-Managements, da zur Verschlüsselung aller Authentifizierungstokens nur ein Master-Key verwendet wird. Auch stellt der zentralisierte Ansatz von Passport ein potentielles Angriffsziel dar und ermöglicht Denial of Service Attacken, wobei letztere aus Sicht der Privatheit keine echte Gefährdung darstellen. Des weiteren könnten böswillige Sites eine Zugehörigkeit zu Passport (z.B. durch Wahl einer Domäne wie pasport.com) vortäuschen und dabei an Authentifizierungsdaten unaufmerksamer Benutzer herankommen.

Auch wenn einige der Sicherheitslücken mittlerweile von Microsoft behoben oder gelindert wurden, zeigt sich doch, dass das Konzept von Passport nur wenig auf Sicherheit und Privatheit ausgerichtet ist. Ein weiterer Bestandteil von .NET Passport, nämlich .NET Passport Wallet, der eine Abwicklung von Zahlungen bzw. eine Übertragung von Zahlungsdaten an Dienste unterstützt hatte („Express Purchase Service“), wird Anfang 2003 eingestellt (vgl. www.passport.net/Consumer/WalletLetter.asp). An dessen Stelle tritt ein „MSN Wallet Service“. Dies wird als ein Versuch angesehen, den schlechten

Ruf von Passport in Hinblick au Sicherheit und Privatheit abzulegen:

> „However, the move also may be aimed at helping Microsoft distance its Wallet system from the privacy questions that have dogged Passport. Microsoft recently settled charges with the Federal Trade Commission Latest News about Federal Trade Commission (FTC) that it had misrepresented the Passport option to some computer users.“ (aus: [Reg02])

Zur Beilegung des angesprochenen Rechtsstreits mit der FTC hat sich Microsoft für 20 Jahre verpflichtet, unabhängige Audit der Passport Identifikations- und Authentifizierungsverfahren durchführen zu lassen (siehe z.B. www.internetnews.com/ec-news/article.php/1455731).

3.3.2.2 Liberty Alliance

Im Gegensatz zu Microsoft .NET Passport ist Liberty Alliance kein Dienst, sondern eine Spezifikation. Direkt vergleichbar zu Passport wäre also eine konkrete Implementierung der Liberty Alliance Spezifikation. Derzeit unterstützt der Sun Identity Server 6.0 als einziges kommerzielles Produkt die Liberty Alliance Spezifikation. In der aktuell vorliegenden Version 1.1. der Spezifikation wird bisher nur ein Teilbereich der vorgesehenen Aspekte abgedeckt, nämlich den einer Dienst-übergreifenden Verwaltung von Benutzeridentitäten [Lib03].

Im Gegensatz zu Microsoft .NET Passport werden die Benutzerdaten innerhalb des Liberty Alliance Verbundes verteilt gespeichert. Ein Vorteil des offenen, föderierten Konzeptes ist es, dass es in Zukunft wohl eine Konkurrenz von unabhängigen Implementierungen geben wird, die miteinander operabel sind, keine zentrale Verwaltung unter der Kontrolle einer einzigen Firma wie bei Passport. Es werden des weiteren Standards wie SAML in der technischen Realisierung verwendet [Lib03], auf die im Rahmen der knappen Besprechung in dieser Arbeit nicht genauer eingegangen werden kann.

Der wichtigste Punkt ist eine Verwaltung von Identitäten („network identities“), eine Dienstübergreifenden Authentifikation und eine Verbindung von Identitäten. Der Benutzer kann dabei explizit auswählen, welche seiner verschiedenen Kennungen bei verschiedenen Dienstanbietern verbunden werden sollen. Dies ermöglicht ein vereinfachtes SSO für verbundene Identitäten. Bei der Abmeldung wird dem Benutzer außerdem eine Auswahlmöglichkeit gegeben, ob eine Abmeldung bei einem Dienst eine automatische Abmeldung bei allen Diensten mit verbundener Identität implizieren soll [Zeh02].

Die Liberty Alliance Architektur besteht aus [Lib03]:

- „Web Redirection“: SSO zur Authentifikation
- „Web Services“: Protokoll, das ein Kommunikation zwischen Liberty-fähigen Entitäten (sowohl Diensten auch Identitätsanbietern) erlaubt
- „Metadata and Schemas“: Gemeinsame Menge von Metadaten und Formaten

Bezüglich der Web Redirection sind zwei Varianten vorgesehen, zum einen ein HTTP-Redirect wie oben erläutert. Zum anderen gibt es einen „Form-POST-Based Redirection“, bei der HTTP-Redirect zur Authentifikation (also der Schritt 2 in Abb. 3.6) dadurch ersetzt, dass der Dienst zur Authentifizierung des Benutzer ein eigenes HTML-Formular bereitstellt, das nach Ausfüllen des Benutzers eine Anfrage an den Identitätsanbieter schickt. Dies erschwert einen Schein-Dienstanbieter Angriff, bei dem sich ein böswilliger Dritter bei der Abwicklung der SSO-Authentifikation zwischenschalten könnte.

Durch das ausgefeilte Schema von digitalen Identitäten und deren Verbindung, das aus Platzgründen hier nicht im Detail dargestellt werden konnte, bildet die Liberty Alliance Spezifikation eine

bessere Basis für sicheres und Datenschutz-konformes Identitätsmanagement als die meisten bestehenden Anwendungen. Damit kann z.B. die Anforderung einer Verkettbarkeit von Pseudonymen (nur) unter der Kontrolle des Benutzers erfüllt werden. Derzeit sind keine Mechanismen für eine Autorisation von Benutzerprofilzugriffen in der Liberty Alliance Spezifikation enthalten, dies ist aber für zukünftige Versionen vorgesehen.

3.3.2.3 Novell digitalme

Überblick Novell digitalme bietet eine Verwaltung von digitalen Identitäten und damit verbundener Profilinformationen mit Hilfe so genannter „meCards“, welcher den verschiedenen Identitäten eines Benutzers entsprechen. Dabei kann einer meCard eine Teilmenge aller Benutzerprofilattribute zugeordnet werden und eine Synchronisation zwischen verschiedenen Identitäten erreicht werden. Eine Möglichkeit, unterschiedliche Ausprägungen für einzelne Profilattribute zu realisieren oder die Menge der vorgegebenen Attribute zu erweitern besteht nicht, allerdings gibt es verschiedene Felder für z.B. „Home Email“ und „Work Email“.

Außerdem ermöglicht digitalme das automatische Ausfüllen von Web-Formularen, unter anderem zur Eingabe von Benutzernamen und Passwort Web-basierter Dienste. Dazu werden die Informationen auf einem digitalme-Server gespeichert und über ein Browser Plug-In in die Web-Seite eingetragen, wobei nicht alle Browser-Hersteller und -Versionen unterstützt werden. Es ist also kein Server-seitiger SSO wie oben beschrieben, sondern ein Client-basierter Ansatz.

Als positiv bei einer Beurteilung von digitalme läßt sich festhalten, dass die Kommunikation mit einem digitalme-Server komplett mit SSL verschlüsselt abläuft, was eine Vertraulichkeit gegenüber potentiellen Abhören von Informationen gewährleistet. Negativ ist, dass die im Rahmen des automatischen Ausfüllens von Web-Formularen gespeicherten Daten, also u.a. Kennungen und Passwörter für Web-Sites, im Klartext an die entsprechende Identität gebunden werden. Ein unerlaubter Zugriff auf den Verzeichnisdienst von digitalme hat also zur Folge, dass u.a. alle gespeicherten Authentifizierungsinformationen offen liegen [Zeh02].

Autorisation digitalme stellt Möglichkeiten bereit, eine Autorisation für den Zugriff auf die meCards zu definieren. Dies bezieht sich auf andere Benutzer, die „Contacts“ genannt werden. Dazu können andere Benutzer in Gruppen eingeteilt werden und für jede meCard soll spezifiziert werden können, an wen die betreffenden Informationen weitergeben werden darf bzw. wer diese einsehen darf. Bei einem Funktionstest war allerdings nur eine Einteilung nach „Public Access“ oder nicht öffentlich zugänglich möglich (vgl. Abb. 3.9).

Es gibt auch eine Funktion zum expliziten Verschicken von meCards an ausgewählte Contacts.

Der Autorisationsansatz bei digitalme ist allerdings nicht ausreichend:

- Die Zugriffskontrolle ist nur auf andere Benutzer bzw. Gruppen von Benutzern bezogen, nicht auf Dienste.
- Die Sichtbarkeit ist bei allen Attributen einer meCard identisch, d.h. man müsste für eine detaillierter Zugriffskontrolle viele einzelne meCards mit unterschiedlichen Rechten anlegen. Bei relativ vielen meCards und Benutzer(-gruppen) ist die Festlegung der Rechte aber zu umständlich und schränkt die Benutzbarkeit stark ein.
- Es erfolgt keine Berücksichtigung von weiteren Aspekten wie dem Zweck des Datenzugriffs oder eine mögliche Weitergabe von Informationen.

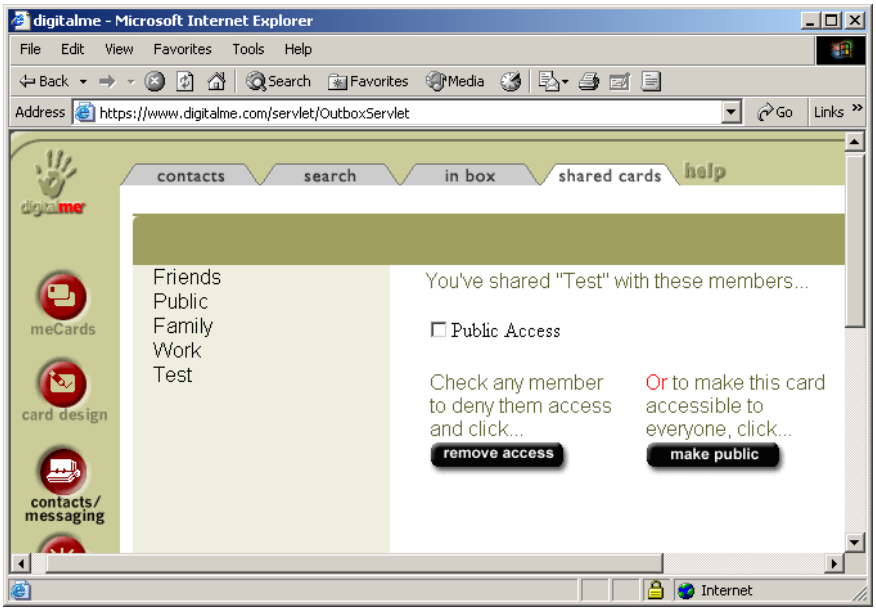

ABBILDUNG 3.9: Autorisation bei digitalme

3.3.3 Bewertung

Die Systeme zum Identitätsmanagement fokussieren auf einer Dienst-übergreifenden Authentifikation, oftmals basierend auf SSO mit HTTP-Redirect und bieten zusätzliche Funktionalitäten für den Benutzer. Single Sign On ist eine Erleichterung für den Benutzer, aber bietet wenig Vorteil hinsichtlich Sicherheit oder Privatheit, eher im Gegenteil, da Sicherheitsprobleme vorhanden sind und die Übersicht und Transparenz für den Benutzer leiden kann. Ein Vorteil ist, dass sich Benutzer nicht die Anmeldedaten verschiedener Dienst merken müssen.

Eine detaillierte Autorisation von Diensten zum Zugriff auf personenbezogenen Daten fehlt noch weitgehend. Wie dargestellt wurde, sind die bestehenden Mechanismen bei .NET Passport oder Novell digitalme auf keinen Fall ausreichend und fehlen im Liberty Alliance Projekt noch vollständig.

Die Liberty Alliance Spezifikation ist als gute Ausgangsbasis für eine sichere und benutzerzentriertes Authentifikation zu sehen. Es bleibt abzuwarten, inwieweit sich die Liberty Alliance Spezifikation gegen Microsoft .NET Passport behaupten kann. Derzeit ist eine Interoperabilität technisch wegen unterschiedlicher Details bei der Authentifikation nicht möglich.

3.4 Fazit

In diesem Teilabschnitt sollen nochmal die wichtigsten Aspekte der untersuchten Systeme in Bezug auf die Anforderungen in dem betrachteten Szenario (vgl. Abschnitt 2.3) zusammengefasst werden.

Durch eine Zugriffskontrolle können Teile der Autorisierungs- und Absicherungsziele realisiert werden. Allerdings fehlen einige wichtige Aspekte für die Verwaltung von Benutzerprofilen, wie in Abschnitt 3.1.6 erläutert wurde, insbesondere eine Integration mit Identitätsmanagement und Möglichkeiten zur Verbesserung der Vertraulichkeit in der Kommunikationsbeziehung.

Die Privacy Enhancing Technologies ermöglichen u.a. Anonymisierung in der Kommunikationsbeziehung (Vertraulichkeits- und Absicherungsziele) und eine technische Unterstützung von Daten-

schutz durch die maschinenlesbaren Datenschutzerklärungen von P3P, was die Transparenz für Benutzer bzgl. der Verwendung personenbezogener Daten verbessert. Allerdings werden keine Mechanismen zur Autorisation bereitgestellt oder in die bestehenden Verfahren integriert.

Anwendungen für Identitätsmanagement im Internet fokussieren auf eine Dienst-übergreifende Authentifikation von Benutzern (SSO) und anderer Funktionalitäten zur Verwaltung mehrerer Identitäten, wodurch Teile der Identitätsziele umgesetzt werden können. Allerdings ist der Schwerpunkt zu wenig auf der Sicherheit und Privatheit für den Benutzer, was sich u.a. in einem weitgehenden Fehlen von Mechanismen zur Autorisierung zeigt.

Abschließend lässt sich daher festhalten, dass die beschriebenen Mechanismen zur Zugriffskontrolle, die Privacy Enhancing Technologies und die Anwendungen für Identitätsmanagement alleine nicht ausreichend sind, um die Anforderungen der Gewährleistung von Privatheit bei dezentraler Verwaltung von Benutzerprofilen zu erfüllen. Es ist deshalb ein Entwurf eines Zugriffschutzmechanismus für die dezentrale Verwaltung von Benutzerprofilen notwendig.

Kapitel 4

Autorisation bei dezentraler Verwaltung von Benutzerprofilen

„Privacy considerations should be part of the initial design phase.
They should not be considered a property that can be added on later.“
Aus: [FFSS01]

Nachdem in den vorhergehenden Kapiteln u.a. dargestellt wurde, welche Anforderungen es an die Privatheit bei (dezentraler) Verwaltung von Benutzerprofilen gibt, soll nun ein eigener Mechanismus zur Autorisation bei dezentraler Verwaltung von Benutzerprofilen erarbeitet werden. Bestehende Ansätze und Systeme sind nur bedingt für die hier betrachtete Aufgabenstellung geeignet bzw. lösen nur Teilaspekte der Problematik, was in Kapitel 3 gezeigt wurde.

Wie im Abschnitt „Anwendungsszenario“ (2.1.3.5) erläutert, sollen Benutzerprofile dezentral in Benutzerprofilagenten (UPA) verwaltet werden. Dienstagenten (CA)[1] greifen bei Bedarf darauf zu, wozu entsprechende Zugriffsrechte erforderlich sind. Ein Dienst kann dabei z.B. ein Community-Unterstützungssystem, ein E-Commerce Agent oder eine personalisierende Web-Site sein. Dazu muss ein Dienst Rechte für den Zugriff auf das Profil haben.

4.1 Überblick

Zunächst werden in 4.1.1 einige Grundgedanken des Ansatzes in dieser Arbeit in Beispielen vorgestellt, dann wird eine Übersicht über den Mechanismus gegeben. Anschließend wird in den weiteren Abschnitten dieses Kapitels der Zugriffsschutzmechnismus und dessen Komponenten im Detail erläutert.

4.1.1 Beispiele

Im Folgenden wird zunächst anhand dreier Beispiele der grundsätzliche Ablauf des Zugriffsschutzmechanismus unabhängig der technischen Realisierung erläutert. Dies soll der Einleitung und Motivation dienen. In dem betrachteten Szenario interagiert ein Studierender über Client-Anwendungen (z.B. einem Web-Browser) mit verschiedenen Diensten, wobei ein Zugriff auf persönliche Daten des Studierenden erfolgt. Dabei soll dasselbe Benutzerprofil verwendet werden, mit unterschiedlichen

[1] CA (community agent) soll im Folgenden für alle Arten von Dienstagenten stehen, um die historisch belastete Abkürzung „SA“ für „service agent“ zu vermeiden.

Rechten je nach Dienst, Verwendungszweck usw. Das im diesem Kapitel zu erarbeitende Zugriffskontrollsystem setzt diese Beispiele in ein technisches System um. Die dabei betrachteten Aspekte sind u.a.:

- Gewährung/Ablehnung von Zugriffsanfragen
- Aushandlung von Zugriffsrechten
- Interaktion mit dem Benutzer
- Verwaltung von Pseudonymen
- Spezifikation eines Zugriffszwecks
- Berücksichtigung zusätzlicher Randbedingungen
- Transparenz für den Benutzer

In den Beispielen kommuniziert ein Benutzer mit mehreren Diensten. Man kann sich auch vorstellen, dass der Benutzer mit Hilfe mehrerer Identitäten mit nur einem Dienst interagiert, wobei der Dienst dabei nicht feststellen können soll – zumindest nicht ohne Erlaubnis des Benutzers – ob die „drei“ Benutzer derselben Person entsprechen.

4.1.1.1 Anonyme Personalisierung

Der Benutzer will eine (anonyme) Personalisierung von Informationen haben, z.B. eine Liste von Vorlesungen, die in seinem nächsten Semester für ihn in Frage kommen. Dies soll anhand seines Studiengangs, Semesterzahl, gewählten Schwerpunkten, bereits besuchten Vorlesungen etc. erfolgen. Dazu will der Dienst diese Informationen und zusätzlich „optional“ den Namen, die Postanschrift und die Email-Adresse des Benutzers vom Benutzerprofilagenten anfordern. Der Benutzerprofilagent wertet die Regeln des Benutzers aus und erlaubt den Zugriff auf einige Benutzerprofildaten. Der Zugriff auf Name und Postanschrift wird jedoch (automatisch) abgelehnt, da dies für den betreffenden Zweck („Personalisierung“) nicht explizit in den Regeln des Benutzers gestattet ist.

Der Benutzer wird je nach konfigurierter Benachrichtigungs-Funktion über den Zugriff informiert oder nicht. Alle Zugriffsanforderungen und Zugriffe werden protokolliert, so dass der Benutzer später jederzeit prüfen kann, auf welche Informationen Dienste Zugriffsrechte haben und – zusätzlich – welche Zugriffe tatsächlich erfolgt sind. Genauso werden abgelehnte Zugriffsversuche aufgezeichnet.

4.1.1.2 Community-Unterstützungssystem

Ein Community-Unterstützungssystem einer Hochschule stellt eine Möglichkeit für Studierende bereit, Kommentare zu Lehrveranstaltungen abzugeben, z.B. über ein Web-Forum. Der Benutzer schreibt mit einem Pseudonym darin Beiträge, so dass eine Zuordnung mehrerer Beiträge zu diesem Benutzer über das Pseudonym möglich sind, aber eine Zuordnung zu seiner richtigen Identität oder personenbezogenen Daten nicht möglich ist.

Das Community-Unterstützungssystem will u.a. Einstellungen im Benutzerprofil ablegen, um z.B. bei Kommentaren anderer Studenten zu einer bestimmten Vorlesung eine Nachricht senden zu können. Dazu wird dem Dienst ein Zugriff auf eine Email-Adresse des Benutzers gestattet und er erhält Rechte, um in einem bestimmten Teil des Profils Konfigurations-Parameter speichern zu können. Ein Zugriff auf diese Einstellungen ist für andere Dienste nicht möglich.

Als Bedingung für den Zugriff auf die Email-Adresse wird in den Zugriffrechten spezifiziert, dass diese Information nur für den explizit spezifizierten Zweck (im Beispiel: „Benachrichtigung bei neuen Nachrichten“) verwendet werden darf, und nicht z.B. an den Dozenten der betreffenden Vorlesung weitergegeben werden darf. Als zusätzliche Option wird vereinbart, dass der Dienst bei einer Änderung der Email-Adresse automatisch benachrichtigt wird. Dadurch kann der Benutzer sicherstellen, dass der Dienst immer eine aktuelle Email-Adresse verwendet und er geänderte Daten nicht manuell verschiedenen Diensten mitteilen muss.

Nach einer gewissen Zeit verliert der Benutzer das Interesse an dieser Community und will daher auch nicht mehr, dass diese Anwendung Zugriff auf persönliche Daten hat. Der Benutzer kann dazu jederzeit die ausgegebenen Zugriffsrechte zurückziehen. Der Dienst wird benachrichtigt und erhält bei einem weiteren Zugriffsversuch eine entsprechende Fehlermeldung.

4.1.1.3 E-Commerce

Der Benutzer bestellt ein Lehr-Buch mit seiner richtigen Identität. Der Dienst will dazu auf die Kreditkarten-Daten (alternativ Informationen für eine Lastschrift), die Adresse und (optional) auf das Geburtsdatum des Benutzers zugreifen. Der Benutzer hat eine Regel aktiviert, dass für einen Zugriff auf Kreditkarten- oder andere Zahlungsdaten immer eine explizite Zustimmung erforderlich ist.

Dem Benutzer wird daher ein Dialog präsentiert, in dem der Zugriffswunsch mit den nötigen Informationen (z.B. „Welcher Dienst?“, „Welche Bestellung?“ etc.) dargelegt wird. Die Identität des Dienstes wird mit einem digitalen Zertifikat nachgewiesen. Als zusätzliche Option wird eine „Übertragung über einen gesicherten Kanal“ angegeben. Der Benutzer gestattet den Zugriff und erlaubt diesem Dienst auch in Zukunft weitere Zugriffe auf Zahlungsdaten mit dem Zweck „Bestellung“, da er öfters bei dem ihm vertrauenswürdigen Dienst eine Bestellung machen will. Die Regelmenge des Benutzers wird daraufhin vom Benutzerprofilagenten um eine entsprechende Regel erweitert. Der Benutzer kann diese Regel jederzeit wieder aus seinem Regelsatz entfernen.

Ein Zugriff auf die Adresse wird bei dem betreffenden Zweck automatisch erlaubt. Für den Zugriff auf das Geburtsdatum kann der Benutzerprofilagent anhand der Regeln des Benutzers keine Erlaubnis ableiten, so dass dieser Zugriffswunsch abgelehnt wird.

4.1.2 Übersicht über den Ablauf

4.1.2.1 Zwei Phasen

Die Rechte für den Benutzerprofilzugriff sollen in einen Verhandlungsprozess zwischen Benutzerprofil- und Dienstagenten ausgehandelt werden. Dabei fordert ein Dienst Zugriffsrechte mit Hilfe eines sog. „Access Requests“ an. Das Ergebnis der Aushandlung ist ein „Access Ticket“, das die genehmigten Rechte enthält. Die Entscheidung, ob eine Zugriffsanfrage zulässig sein soll, trifft der Benutzerprofilagent auf Basis von Regeln, die vom Benutzer festgelegt werden. Diese Regeln entsprechen den Datenschutz-Präferenzen eines Benutzers. Mit den in Form des Access Tickets erhaltenen Zugriffsrechten kann der Dienst dann in einer zweiten Phase die Daten tatsächlich anfordern. Der in dieser Arbeit vorgeschlagene Mechanismus für Zugriffskontrolle auf Benutzerprofile besteht also aus zwei Phasen (Abb. 4.1):

1. Aushandlung von Zugriffsrechten, ggf. mit Benutzerinteraktion, und Generierung eines Access Tickets (Phase I, siehe 4.3)

2. (Effizienter) Datenzugriff mit dem ausgehandelten Access Ticket (Phase II, siehe 4.4)

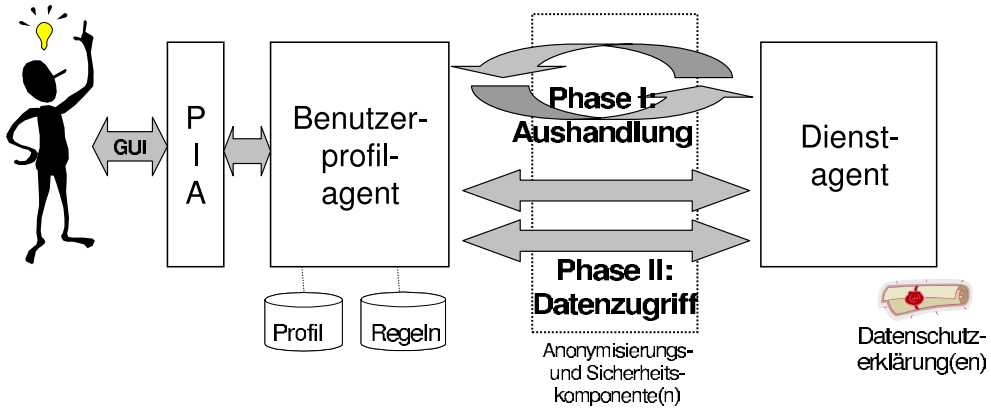

ABBILDUNG 4.1: Überblick über den Ablauf

In Phase I schickt ein Dienstagent (CA) einen Access Request, zusammen mit einer P3P-Datenschutzerklärung an den Benutzerprofilagenten (UPA). Dieser wertet die Anfrage anhand von Regeln des Benutzers aus und antwortet bei positiver Entscheidung mit einem Access Ticket.

Das Access Ticket ist der Kern dieses Ansatzes und wird im Folgenden ausführlich beschrieben. Das Access Ticket dient insbesondere auch der Transparenz für den Benutzer, um jederzeit feststellen zu können, welche Rechte zum Zugriff auf sein Profil an welche Dienste vergeben sind. Dies ist eine wichtige Anforderung für Zugriffskontrolle bei Benutzerprofilverwaltung. Die Motivation für die Aufteilung der Zugriffskontrolle in die beiden Phasen ist weiterhin:

- Ermöglichung der Formalisierung von Zugriffsrechten (Access Tickets) als Ergebnis von Phase I und Eingabe für Phase II
- Integration von Benutzerinteraktion in Phase I erforderlich: dies ist eventuell aufwändig und sollte vom eigentlichen Datenzugriff losgelöst werden
- Effizienter Datenzugriff in Phase II möglich: in Phase II sind die ausgehandelten Rechte schon vorhanden, so dass ein effizienter Datenzugriff erfolgen kann; dies ist wichtig, wenn man davon ausgeht, dass einzelnen Zugriffe häufig und wiederholt gemacht werden könnten
- Ermöglichung verschiedener Ansätze für Phase I (siehe Abschnitt „Herausgabe versus Aushandlung von Access Tickets“, 4.3.1)
- Möglichkeit der Herausgabe von sinnvollen Regelmengen für Phase I von vertrauenswürdigen Organisationen
- Weitergabe von Rechten möglich, da diese explizit festgehalten werden (sofern dies ausdrücklich erlaubt wird, siehe 4.2.3.3)
- Leichtere Administration von Rechten, da dies in einer allgemeineren Form von Regeln geschieht und der Benutzer das Ergebnis der Rechteaushandlung jederzeit überprüfen kann; damit können die in 3.1 beschriebenen Probleme von Zugriffskontrolle vermieden werden

Generell sind Benutzerprofil-Attribute erst dann für einen Dienst zugreifbar, wenn er explizit das Recht dafür erhalten hat, d.h. der Default-Wert bei der Zugriffskontrolle für Benutzerprofile ist stets „kein Zugriff“.

Auch wenn ein Dienst persönliche Daten einer Person lokal verwalten will (z.B. Präferenzen eines Benutzers), müssen entsprechende Rechte vorhanden sein, um personenbezogene Daten speichern zu dürfen. Es ist jedoch nicht unbedingt erforderlich, dass alle Daten immer im UPA aktualisiert werden. Ein Dienst kann also – allgemeiner ausgedrückt – auch Daten lokal verwalten, wenn die Rechte dazu vorhanden sind. Das „Caching“ von Daten ist somit unabhängig von dem hier vorgestellten Zugriffsmechanismus.

Die erläuterten Komponenten (z.B. Benutzerprofilagent) müssen nicht als Software-Agenten realisiert sein. Die Verwendung des Begriffes „Agent“ soll hier insbesondere die Unabhängigkeit der Komponenten verdeutlichen. Auch verfolgen die einzelnen „Agenten“ unterschiedliche, z.T. konträre, Ziele: der Benutzer will seine Privatheit wahren, während Personalisierungs-Diensten i.d.R. möglichst viel und gute Informationen über Benutzer (Kunden) haben wollen. Für eine konkrete Implementierung bietet sich z.B. die Verwendung von Web Services oder die Integration in das Liberty Alliance Projekt anstelle einer Implementierung mit (Software-)Agenten an. Dies wird in Kapitel 5 („Evaluierung und Systementwurf“) noch näher betrachtet.

In dem hier betrachteten Szenario sollen Benutzer nicht direkt auf Dienste zugreifen, sondern über ein (Client-seitiges) „Personal Identity Assistant“ (PIA) Werkzeug (s.a. Abb. 4.1). Das PIA dient u.a. als Schnittstelle des Benutzers zur Pflege seines Profils. Die genaue Funktion des PIA und weitere Aspekte der Benutzerinteraktion werden im Abschnitt 5.1 besprochen. Grundsätzlich ist es wichtig, eine geeignete Benutzerschnittstelle für die Verwaltung der Profile anzubieten. Als Alternative bietet sich z.B. die Integration von Identitätsmanagement-Funktionen in Web-Browser an. Der erarbeitete Ansatz ist jedoch unabhängig von der konkreten Ausgestaltung des Personal Identity Assistant Werkzeuges.

4.1.2.2 Access Request, Datenschutzerklärung und Access Ticket

Bevor in 4.2 und den folgenden Abschnitten Access Request und Ticket detailliert beschrieben werden, sollen zunächst die hier vorgesehenen Formalismen etwas genauer erklärt und differenziert werden.

Bei einer Zugriffsanfrage eines Dienstes schickt dieser einen Access Request (AR) zusammen mit einer P3P-Datenschutzerklärung (vgl. Abschnitt 3.2.5) an den Benutzerprofilagenten. Dieser antwortet nach Abgleich mit den Datenschutzpräferenzen und eventueller Benutzerinteraktion mit einem Access Ticket (AT), wenn ein Zugriff gestattet werden kann. Der Dienst kann dann mit dem Access Ticket konkret Daten anfordern.

Die P3P-Erklärung enthält dabei:

- Die grundlegenden Praktiken des Dienstes bzgl. Weitergabe von Informationen, Verwendung von Cookies usw.
- Allgemeine Informationen über einen Dienst, z.B. dessen Postanschrift (vgl. 3.2.5)
- Evtl. ein Gütesiegel einer entsprechenden Institution

Ein Access Request ist die (konkrete) Zugriffsanforderung eines Dienstes:

- Der AR enthält den konkreten Zweck und Kontext jedes Zugriffs, im Gegensatz zu P3P detailliert für jedes Benutzerprofil-Attribut. Dadurch wird auch den rechtlichen Anforderungen genüge getan, dass der Zweck eines Zugriffs bei jedem Zugriff dem Benutzer präsentiert werden muss.

- Ein AR kann im Gegensatz zu einem Access Ticket optionale Elemente, Alternativen o.ä. enthalten (siehe „Optionen bei der Zugriffsanfrage“, Kap. 4.2.3.2).

- Ein AR kann mehr Attribute enthalten, als der Dienst für einen einzelnen Zugriff benötigt, um die für den Benutzer und Dienst möglicherweise relativ aufwändige Aushandlung zusammenzufassen. Das bedeutet, dass ein Dienst für mehrere, unabhängige Datenzugriffe nur ein AR mit den einzelnen Zugriffsanforderungen an den UPA senden muss.

Ein Access Ticket enthält die genehmigten Rechte:

- Ein AT entspricht der Formalisierung von Rechten in XML-basierter Zugriffskontrolle und ähnlicher Ansätze (vgl. Abschnitt 3.1.5.1).

- Eventuell ist nur ein Teil der im Access Request angeforderten Rechte enthalten, wenn Benutzer bzw. dessen Regeln nur einen Zugriff auf einen Teil des Profils gestatten.

- Im Gegensatz zu einem AR ist bei einem AT Benutzerinteraktion nicht mehr erforderlich oder möglich (der Benutzer kann aber ausgegebene Tickets wieder zurückziehen).

Der Access Request bzw. Ticket ist eine Formalisierung von Zugriffskontrolle, während P3P eine Konzeptualisierung einer Datenschutzerklärung ist. Durch eine Kombination beider Konzepte kann eine Verbindung zwischen Zugriffskontrolle und Privacy Enhancing Technologies hergestellt werden, die es bisher in bestehenden Anwendungen und in der Literatur noch kaum gibt.

Wichtig hier ist, dass die P3P-Datenschutzerklärung Teil einer Vereinbarung zwischen Dienst und Benutzer bzw. dessen Benutzeragent ist. Daher ist ein Access Ticket immer nur zusammen mit der P3P-Erklärung gültig. Dies verbessert u.a. die rechtlichen Möglichkeiten bei einem Missbrauch von Daten.

Ein Rückgriff auf die P3P-Erklärung ermöglicht eine Kompabilität mit den bei vielen Web-Sites in P3P oder anderer Form bestehenden Datenschutzerklärungen. Eine Alternative wäre gewesen, ein eigenes Vokabular bzgl. Verwendung von Daten und anderer Terminologie bzgl. Privatheit in den Zugriffsformalismus mit aufzunehmen. Dies entspricht z.B. dem in Abschnitt 3.1.3.4 erläuterten Modell für Privatheit in Zugriffskontrolle von Simone Fischer-Hübner. Allerdings wäre dabei u.a. keine Kompabilität zu den bestehenden Datenschutzerklärungen und eventuell in APPEL formalisierten Benutzerpräferenzen mehr gegeben. So könnten zumindest teilweise nebeneinander P3P/APPEL für allgemeines Web-Surfen und P3P/APPEL mit den hier erläuterten Erweiterungen und zusätzlichen Mechanismen für Benutzerprofilzugriff angewendet werden.

4.1.3 Kapitelüberblick

Grundsätzlich wird in dieser Arbeit hauptsächlich Autorisierung in dem erläuterten Szenario genauer untersucht, also eine Zugriffskontrolle auf dezentral verwaltete Benutzerprofile. Andere Sicherheitsfunktionen, die auch wichtig sind, wie Identifikation, Authentifikation, Integrität etc. werden am Rande betrachtet.

Bevor jetzt die einzelnen Aspekte des Zugriffsschutzmechanismus’ ausführlich besprochen werden, soll ein kurzer Überblick über das Hauptkapitel dieser Arbeit gegeben werden. Zunächst werden Access Request und Access Ticket als zentraler Baustein in dem Mechanismus dargestellt. Dies beinhaltet auch einen Vergleich mit bestehenden Ansätzen am Ende von Kap. 4.2. In Anhang A und B findet sich dazu auch eine formale Beschreibung von Access Request und Ticket. Anschließend

wird in Abschnitt 4.3 die Phase I und in 4.4 die Phase II erarbeitet. Weitere Aspekte, die den Basis-Mechanismus nicht direkt betreffen, aber dennoch hier essentiell sind, folgen in Kapitel 4.5 („Identitätsmanagement“). Insbesondere wird dabei auf die Verwaltung mehrerer Identitäten eines Benutzers eingegangen. Des weitern werden in Abschnitt 4.5.5 („Vertrauensmanagement“) einige Punkte besprochen, die durch technische Mechanismen nicht (direkt) abgedeckt werden können.

In Kapitel 5 erfolgt eine Evaluierung des Ansatzes in dieser Arbeit.

4.2 Access Request und Access Ticket

Das Ziel von Access Request und Ticket ist die Formalisierung von Zugriffsrechten bei einer Verwaltung von Benutzerprofilen, wobei sich die einzelnen Elemente aus den Anforderungen in Abschnitt 2.3 ergeben. Der Entwurf von AR und AT ist an den z.T. in Kapitel 3 besprochenen XML-Schemas wie XRML für Digital Rights Management oder XACML orientiert. Die Darstellung erfolgt im Folgenden anhand des Aufbaus der Formate.

4.2.1 Access Request und Access Ticket

Ein *Access Request (AR)* ist die Formalisierung der Zugriffsanfrage eines Dienstes auf ein Benutzerprofil. Das AR wird in XML modelliert und ist wie in Abbildung 4.2 gezeigt aufgebaut. Das Element <ACCESS> kann mehrfach auftreten. Ein optionales Tag <STARTDATE> ist zusätzlich möglich, siehe Kapitel 4.2.3.6.

```
<ACCESSREQUEST>
  <USER> ... </USER>
  <SERVICE> ... </SERVICE>
  <VALIDITY> ... </VALIDITY>
  <POLICY> ... </POLICY>
  <ACCESS> ... </ACCESS>
</ACCESSREQUEST>
```

ABBILDUNG 4.2: Access Request

Ein *Access Ticket (AT)* manifestiert die Zugriffsrechte eines Dienstes auf das Profil eines Benutzers. Diese Rechte-Spezifikation ist bei einer dezentralen, föderierten Benutzerprofilverwaltung notwendig. Ein AT hat dieselben verpflichtenden Elemente wie ein AR, nur dass diese von einem <ACCESSTICKET> Tag umschlossen sind. Das AT ist das Ergebnis der Aushandlung zwischen Benutzerprofil- und Dienstagenten, also die vom Benutzer bzw. dessen Agenten genehmigten Zugriffsrechte, während ein AR die vom Dienst erwünschten Rechte enthält. Ein Benutzer bzw. dessen Profilagent kann Access Tickets konzeptionell auch ohne Anforderung eines Dienstes und Aushandlung herausgeben.

Das AT bedeutet nicht unbedingt, dass auch ein Datenzugriff erfolgen muss, sondern legt zunächst nur die Zugriffsrechte fest. D.h. ein Dienst könnte auch Rechte anfordern, wenn diese noch nicht erforderlich sind, damit bei Bedarf der Aushandlungsprozess nicht mehr nötig ist. Dies könnte z.B. der Fall sein, wenn sich ein Benutzer bei einem Dienst anmeldet oder diesen zum ersten Mal besucht, auch wenn vorerst noch keine Personalisierungsfunktionen o.ä. genutzt werden.

Das Access Ticket wird vom Benutzerprofilagent digital signiert, damit beim späteren Datenzugriff die Authentizität und Integrität geprüft werden kann. Dadurch wird sichergestellt, dass das AT

nicht verändert wurde und tatsächlich der Dienst zugreift, für den das AT ausgestellt wurde. Dies erlaubt eine verteilte Zugriffskontrolle.

4.2.2 Verpflichtende Komponenten

In diesem Abschnitt werden die einzelnen Komponenten eines AR/AT der Reihe nach erklärt. Bei den einzelnen Elementen sind jeweils kleine Beispiele angegeben, am Ende dieses Kapitels findet sich ein komplettes Beispiel eines AR. Die folgenden verpflichtenden Elemente müssen sowohl bei einem Access Request als auch bei einem Access Ticket enthalten sein.

4.2.2.1 Identifikation des Benutzers

```
<USER TYPE="X.500">@c=DE@o=TU-MUENCHEN@cn=WOERNDL</USER>
```

ABBILDUNG 4.3: Beispiel für <USER>

Zunächst wird ein Element <USER> benötigt, das den Benutzer – genauer gesagt, eine Identität des Benutzers – identifiziert. Als Attribut kann der Typ der Identifikation spezifiziert werden (Abb. 4.3):

- Keine Typangabe: Identifikation als Freitext, z.B. mit einem Pseudonym des Benutzers
- TYPE=”X.500”: Angabe eines X.500 Distinguished Name
- TYPE=”Cert”: Identifikation mit einem digitalen Zertifikat des Benutzers

Die Angabe bei <USER> entspricht dem Identifikator eines Benutzerprofils (vgl. Abschnitt 2.1.1.3 bei den Grundlagen). <USER> identifiziert also eine Identität des Benutzers, nicht etwa den Benutzerprofilagenten, auch bei anonymen Datenzugriff. Es kann aber eine (anonymisierte) Kennung eines Anonymisierungsdienstes als Identifikator des Benutzers angegeben werden. Optional kann eine „Identitätsstufe“ festgelegt werden, dies wird später in Abschnitt 4.2.3.1 behandelt. Die Authentifikation eines Benutzers kann auch durch die Angabe eines digitalen Zertifikats des Benutzers erfolgen. Dies ist dann nicht in einem AR enthalten, sondern wird vom UPA in das AT zur Identifikation des Benutzers eingefügt.

4.2.2.2 Identifikation des Dienstes

Für die Identifikation des Dienstes ist ein Element <SERVICE> vorgesehen (Abb. 4.4), es gibt zwei Möglichkeiten dafür, die durch Angabe eines XML-Attributs spezifiziert werden:

- Attribut TYPE=”X.500”: Angabe eines X.500 Distinguished Name
- Attribut TYPE=”Cert”: Angabe eines digitalen Zertifikats (analog zu <USER>)
- Attribut TYPE=”P3P”: Dies entspricht dem <ENTITY> Element einer P3P-Datenschutzerklärung (vgl. [P3P02], Abschnitt 3.2.4)

```
<SERVICE TYPE="P3P"><ENTITY>
  <DATA-GROUP>
    <DATA ref="#business.name">CatalogExample</DATA>
    <DATA ref="#business.contact-info.postal.street">
      4000 Lincoln Ave.</DATA>
    <DATA ref="#business.contact-info.postal.city">Birmingham</DATA>
    <DATA ref="#business.contact-info.postal.stateprov">MI</DATA>
    <DATA ref="#business.contact-info.postal.postalcode">48009</DATA>
    <DATA ref="#business.contact-info.postal.country">USA</DATA>
    <DATA ref="#business.contact-info.online.email">
      catalog@example.com</DATA>
    <DATA ref="#business.contact-info.telecom.telephone.intcode">
      1</DATA>
   <DATA ref="#business.contact-info.telecom.telephone.loccode">
      248</DATA>
   <DATA ref="#business.contact-info.telecom.telephone.number">
      3926753</DATA>
  </DATA-GROUP></ENTITY></SERVICE>
```

ABBILDUNG 4.4: Beispiel für <SERVICE>

Ein möglicher Nachweis, dass eine Angabe wie in Abb. 4.4 gezeigt auch wirklich mit dem Absender der Zugriffsanforderungen übereinstimmt, soll hier nicht näher diskutiert werden. Eine einfache, aber ineffiziente Lösung wäre es, bei jedem Zugriff ein digitales Zertifikat des Dienstes (Option „Cert“) zu fordern.

Ein AT ist zunächst immer nur für einen Dienst gültig, allerdings kann eine Weitergabe von Daten erlaubt sein (siehe Optionen bei Zugriffrechten, Abschnitt 4.2.3.3).

4.2.2.3 Gültigkeitszeitraum

Der Gültigkeitszeitraum – bzw. genauer: das Ablaufdatum – eines Tickets wird mit Hilfe des Elementes <VALIDITY> spezifiziert. Dabei wird ein Enddatum (und optional auch Zeit) festgelegt, bei einem späteren Datenzugriff ist das AT nicht mehr gültig und muss neu zwischen Benutzerprofil- und Dienstagent ausgehandelt werden. Für eine unbegrenzte Gültigkeit des AT kann <VALIDITY>infinite </VALIDITY> angegeben werden.

Ein AT kann unabhängig vom angegebenen Gültigkeitszeitraum jederzeit zurückgezogen werden, z.B. wenn der Benutzer seine Datenschutz Präferenzen ändert. Dies ist vergleichbar zum Zurückziehen eines digitalen Zertifikates. Das AT gilt in der Regel ab Ausgabe, ansonsten kann optional ein Element <STARTDATE> (siehe Kapitel 4.2.3.6) verwendet werden (Abb. 4.5.

```
<VALIDITY>31-12-2003 12:00</VALIDITY>
```

ABBILDUNG 4.5: Beispiel für ein Ablaufdatum

Im AR kann der Gültigkeitszeitraum auch leer gelassen werden. Die Formalisierung eines Ablaufdatums setzt die Anforderung einer zeitlichen Begrenzbarkeit von Zugriffsrechten um. Dies ist bestehenden XML-basierten Ansätzen nicht enthalten.

4.2.2.4 P3P-Datenschutzerklärung

Ein AT ist grundsätzlich nur in Verbindung mit einer P3P-Datenschutzerklärung gültig. Wenn eine Datenschutzerklärung nicht mehr gültig ist, dann verfällt unabhängig vom angegebenen Ablaufdatum auch das AT. Das bedeutet, dass beim nächsten Datenzugriff ein neues AT ausgehandelt werden muss. Ein Dienst kann mehrere P3P-Erklärungen verwalten, die z.B. hinsichtlich Datenschutz stärker oder weniger stark eingeschränkt sein können. Es ist Bestandteil des Aushandlungsprozesses, eine Datenschutzerklärung, die den Wünschen des Benutzers entspricht, zu vereinbaren. Im AR/AT spezifiziert das Element <POLICY> eine Referenz auf die für diese Zugriffsanfrage vorgesehene P3P-Datei (Abb. 4.6).

Die P3P-Erklärung enthält allgemeine Hinweise zu den Datenschutzpraktiken eines Dienstes. Dies kann z.B. ein Zertifikat einer Organisation sein, die die Einhaltung der Erklärung überprüft und bestätigt. Auch können allgemeine Angaben zum Zweck einer Abfrage von persönlichen Daten enthalten sein. Im Unterschied dazu enthält das Element <PURPOSE> im AR/AT (siehe Abschnitt 4.2.2.5) den genauen Grund bzw. Zweck des konkreten Datenzugriffs.

```
<POLICY>http://www.server.com/pathto/p3p.xml</POLICY>
```

ABBILDUNG 4.6: Beispiel für <POLICY>

4.2.2.5 Das Element <ACCESS>

<ACCESS> ist das Basiselement für die Zugriffrechte auf Teile des Benutzerprofils (Abb. 4.7). Dabei wird ein Ausschnitt des hierarchischen XML-Benutzerprofils adressiert. Es sind beliebig viele <ACCESS> Segmente in einem AR oder AT möglich. Ein <ACCESS> Element kann mehrere Rechte enthalten. Spezifiziert wird die benötigte Ressource als Attribut von <ACCESS>, sowie die (gewünschten oder erteilten) Rechte darauf und der Zweck des Zugriffs. Alle Elemente sind verpflichtend, insbesondere auch der Zugriffszweck. Die Verpflichtung zur Angabe eines Zwecks beim Zugriffs auf personenbezogenen Daten ergibt sich aus deutschen und internationalen Datenschutzbestimmungen (vgl. Abschnitt 2.2.2).

```
<ACCESS RESOURCE="/profile/payment/lastschrift/*">
    <READ/>
    <PURPOSE><p3p:tailoring/></PURPOSE>
</ACCESS>
```

ABBILDUNG 4.7: (Verpflichtende) Komponenten von <ACCESS>

Die einzelnen Komponenten von <ACCESS> werden in den folgenden Teilabschnitten erläutert.

Ressourcen Mit einem Attribut RESOURCE im Tag <ACCESS> kann der Teil des Benutzerprofils adressiert werden, für den das AT gelten soll. Die Adressierung erfolgt dabei mit Hilfe des W3C Standards XPath [XPa99]. Beispielsweise können damit folgende Teile eines Profils angesprochen werden:

- Alle Interessen:
  /profile/interests/*

- Die erste <INTEREST> Element in <INTERESTS>:
  /profile/interests/interest[1]
- Alle „privaten“ Interessen, d.h. Elemente <INTEREST> mit Attribut id=”private”:
  /profile/interests/interest[@id=’private’]
- Alle Teile des XML-Benutzerprofils mit id=”private”:
  //*[@id=’private’]
- Kreditkarten- oder Lastschrift-Informationen:
  /profile/payment/creditcard/* | /profile/payment/lastschrift/*

Für alle adressierbaren Teile eines Profils können verschiedene Rechte vergeben werden. Es kann sich dabei ergeben, dass für einen Teil des Profils mehrere <ACCESS> Abschnitte eines AT zutreffen können, z.B. unterschiedliche Rechte für /profile/payment/* und /profile/payment/creditcard/* festgelegt sind. Welcher <ACCESS> Abschnitt dann gelten soll, wird in Abschnitt 4.3.4.4 („Auswertestrategie“) diskutiert.

Die Adressierung mit XPath wird auch im XACML-Standard für Zugriffskontrolle auf XML-Dokumente verwendet. Eine Alternative wäre ein weniger ausdrucksstarker, dafür leichter implementierbarer Formalismus. Allerdings muss in einer Implementierung nicht der volle XPath-Standard umgesetzt werden, da wohl die meisten relevanten Zugriffe mit den einfacheren XPath-Elementen abgedeckt werden können.

Zugriffsrechte Wichtig bei der Zugriffskontrolle für Benutzerprofile ist eine Formalisierung von Zugriffsmodi, dies ist den bestehenden Privacy Enhancing Technologies, z.B. dem P3P-Standard, nicht enthalten. Es sind folgende Rechte vorgesehen:

- Lesen: <READ>
- Schreiben: <WRITE> und <APPEND>
- Löschen: <DELETE>
- Erzeugen: <CREATE>

Mit <WRITE> kann ein bestehendes Attribut überschrieben werden. Mit <APPEND> kann ein neues Element hinzugefügt werden, ohne bestehende Elemente zu verändern, z.B. das Ergänzen neuer Interessen. Mit Hilfe von <CREATE> wird nicht nur ein neues Element im Profil angelegt, sondern es wird ein neues Tag definiert. Es wird somit die DTD des Benutzerprofils erweitert, was auch durchaus erwünscht sein kann, um z.B. Applikations-spezifische Konfigurationseinstellungen in einem Profil aufnehmen zu können. Eine Unterscheidung zwischen „Schreiben“ und „Erzeugen“ erscheint wichtig, um den Benutzer auf Wunsch darauf hinweisen zu können, dass neue Abschnitte und Kategorien im Profil angelegt werden und nicht nur Daten ergänzt werden.

Es ist möglich, mehrere Rechte innerhalb eines <ACCESS> Elements anzugeben, z.B. Schreiben und Lesen. Ein Schreib-Recht alleine berechtigt ausdrücklich nicht zum Lesen. Ein Dienst soll z.B. Ergänzungen einer Transaktionshistorie machen dürfen und nicht unbedingt auch alle bestehenden Transaktionen sehen dürfen.

Ein Recht „Ausführen“ bzw. „Execute“ macht bei Zugriff auf Benutzerprofile weniger Sinn und wird daher auch nicht explizit betrachtet. Eine Erweiterung der Menge der Rechte wäre aber problemlos möglich. Dies wäre auch für eine Anwendung von Access Tickets auf andere Domänen notwendig

und sinnvoll. In Abschnitt 4.2.3.3 werden Optionen für die Zugriffsrechte diskutiert, z.B. eine Formalisierung von „Weitergabe von Daten“. Es wäre auch denkbar, diese Option als eigenständige Rechte zu formulieren (z.B. „<READ-DISTRIBUTE>“), dies wurde aber hier so nicht umgesetzt, weil man sich Optionen vorstellen kann, die für mehrere Rechte gelten.

In einem AT sind nur die explizit gestatteten, positiven Rechte aufgeführt, eine Festlegung von negativen Rechten ist nicht vorgesehen. Diese Eigenschaft verbessert die Übersichtlichkeit für den Benutzer, weil jedes Zugriffsrecht eine Entsprechung im AT hat und nicht etwa durch einen generell erlaubten Zugriff und Ausnahmen aus negative Rechte abgeleitet werden muss.

Die Formalisierung der Zugriffsrechte als eigene XML-Elemente orientiert sich an der Darstellung der Rechte in den Standards zum Digital Rights Management (vgl. Abschnitt 3.1.5.2).

Zugriffszweck Die Formalisierung des Zwecks eines Zugriffs wird im AR/AT durch das Element <PURPOSE> spezifiziert. Inhaltlich kann dazu z.B. das Vokabular von P3P verwendet werden (vgl. [P3P02], Abschnitt 3.3.4), u.a. „tailoring“ oder „develop“. Auf weitere Möglichkeiten, den Zweck zu modellieren, soll hier nicht genauer eingegangen werden. Es ist diesbezüglich eine gemeinsame Ontologie zwischen Benutzerprofil- und Dienstagenten nötig, wozu P3P als sinnvolle Ausgangsbasis dienen kann.

Der Zweck kann auch als Frei-Text angegeben werden. Dies kann dem Benutzer vom Benutzerprofilagenten angezeigt werden, damit ggf. der Benutzer eine informierte Entscheidung treffen kann.

Eine Integration des Zugriffszwecks ist eine wesentliche Anforderung einer Datenschutz-konformen Zugriffskontrolle. Eine allgemeine Formulierung von Bedingungen ist z.T. bei bestehenden Ansätzen möglich (z.B. „Provisions“ bei XACL), allerdings kann mit der expliziten Forderung einer Angabe des Zugriffszwecks eine gute Adaption an Erfordernisse der Verwaltung personenbezogener Daten erreicht werden.

4.2.3 Optionale Komponenten

In diesem Abschnitt werden optionale Komponenten von AR bzw. AT erläutert. Diese Elemente sind zum Teil nur bei Access Request oder Access Ticket sinnvoll, nicht bei beiden.

4.2.3.1 Identitätsstufe

Bei der Identifikation des Benutzers ist es nicht immer erwünscht oder sinnvoll, die „wahre Identität“ des Benutzers zu offenbaren. Pseudonymität oder auch völlig anonymer Datenzugriff muss in die Zugriffskontrolle für Benutzerprofile integriert werden. Es soll daher ein Grad der Anonymität bzw. eine Identitätsstufe spezifiziert werden können. Diese Angabe einer Identitätsstufe ist als Attribut bei <USER> möglich, es sind dabei folgende Möglichkeiten vorgesehen (s.a. Abb. 4.8):

- LEVEL=”veronymous”: Die Identität des Benutzers wird offenbart. Ggf. muss der Benutzer diese Identität mit einem Zertifikat belegen. Dies ist z.B. beim Zugriff auf das Profil aufgrund bei einer Bestellung eines Produktes sinnvoll. Dabei muss der Dienst prüfen können, welcher Person diese Bestellung zugeordnet werden kann.
- LEVEL=”pseudonymous”: Es wird nur ein Pseudonym angegeben. Damit ist für einen Dienst die Zuordnung mehrerer Transaktionen zu einem Benutzer möglich, ohne die wahre Identität des Benutzers offen legen zu müssen. Die Identifikation kann in diesem Fall z.B. durch die Angabe eines Spitznamens oder einer in einem Community-Unterstützungssystem verwendeten Kennung sein.

- LEVEL=”anonymous”: Der Zugriff erfolgt anonym, d.h. einzelne Aktionen können vom Dienst nicht zu einem Benutzer zugeordnet werden. Ein anonymer Zugriff auf Benutzerprofile ist teilweise sinnvoll, z.B. in Form einer Erstellung von Empfehlungen auf Basis von Interessen in einem anonymen Profil. Dazu kann auch auf personenbezogene Daten wie die Email-Adresse eines Benutzers zugegriffen werden, was bedeutet, dass ein Dienst trotz LEVEL=”anonymous” eventuell Rückschlüsse auf den Benutzer machen kann.

Die Angabe der Identitätsstufe beim AR ist optional. Wenn der Dienst bei der Zugriffsanfrage keine Angabe einer Identitätsstufe macht, bleibt es dem Benutzerprofilagenten überlassen, je nach Präferenzen des Benutzers eine geeignete Identitätsstufe auszuwählen. Dies könnte z.B. „anonymous“ bei unbekannten Diensten, „pseudonymous“ bei Diensten, die schon mal in Anspruch genommen wurden und „veronymous“ bei explizit als vertrauenswürdig gekennzeichneten Diensten sein. Der Benutzerprofilagent muss nicht der Anfrage entsprechen, wenn z.B. die Datenschutzerklärung des Dienstes nicht zufrieden stellend ist. Dies ist ein Bestandteil der Aushandlung. Bei einem „anonymen“ Benutzer kann das Element <USER> als Inhalt eine Kennung eines Anonymisierungsdienstes haben.

```
<USER TYPE="X.500" LEVEL="veronymous">@c=DE@o=TU-MUENCHEN@cn=WOERNDL
  </USER>
<USER LEVEL="pseudonymous">nickname123</USER>
<USER>user1234@anonymizer.com</USER>
```

ABBILDUNG 4.8: Beispiele für <USER> mit Identitätsstufen

Eine Integration von Identitätsstufen ist in bestehenden Ansätzen zur Zugriffskontrolle nicht enthalten, stellt aber einen wesentlichen Gesichtpunkt einer Autorisation für Benutzerprofile dar. Eine weitergehende Diskussion von Identitätsstufen und Identitäten in dem hier erarbeiteten Mechanismus findet sich später in Abschnitt 4.5 („Identitätsmanagement“).

4.2.3.2 Optionen bei Zugriffsanfrage

Mit Hilfe dieser Optionen sollen Wahlmöglichkeiten, wie sie z.B. bei Web-Formularen geläufig sind, nachgebildet werden können. Es gibt folgende Möglichkeiten:

- OPTION=”optional” oder OPTION=”mandatory”: Kennzeichnung von Zugriffsanforderungen als optional bzw. verpflichtend
- GROUP=”group-name”: Gruppierung von Profildaten („... und ...“)
- ALTERNATIVE=”alt-name”: Kennzeichnung von Alternativen („... oder ...“)

Abhängig von den Präferenzen eines Benutzers, wird der Zugriff auf optionale Elemente einer Anfrage erlaubt oder nicht erlaubt. Bei einer Gruppierung ist nur der Zugriff auf die gesamte Gruppe, d.h. alle Elemente mit dem gleichen „group-name“ sinnvoll. Bei Alternativen kann der Benutzeragent im Sinne der Präferenzen des Benutzers unter den Elemente mit dem gleichen „alt-name“ geeignet auswählen (z.B. Kreditkartendaten oder Informationen für eine Lastschrift). Diese Optionen sind nur bei einem AR sinnvoll. Nach erteilten Zugriffsrechten ist es egal, ob z.B. ein Teil einer Anfrage ursprünglich als optional deklariert wurde. In einem AT sind daher keine Alternativen oder Gruppen mehr enthalten. Die Optionen werden als Attribute bei <ACCESS> eingefügt, d.h. sie können eventuell auch zusammenfassend für mehrere Rechte gelten. Abbildung 4.9 zeigt einige Beispiele für Optionen.

```
<ACCESS RESOURCE="/profile/interests/*" OPTION="optional">
  <READ/><WRITE/><PURPOSE>...</PURPOSE></ACCESS>
<ACCESS RESOURCE="/profile/demographic/postal/locality" GROUP="postal">
  <READ/><PURPOSE>...</PURPOSE></ACCESS>
<ACCESS RESOURCE="/profile/demographic/postal/zip" GROUP="postal">
  <READ/><PURPOSE>...</PURPOSE></ACCESS>
<ACCESS RESOURCE="/profile/payment/creditcard/number"
    ALTERNATIVE="payment"><READ/><PURPOSE>...</PURPOSE></ACCESS>
<ACCESS RESOURCE="/profile/payment/lastschrift/*"
    ALTERNATIVE="payment"><READ/><PURPOSE>...</PURPOSE></ACCESS>
```

ABBILDUNG 4.9: Beispiele für Optionen

4.2.3.3 Optionen bei Zugriffsrechten

Bei den Zugriffsrechten ist es wichtig, zusätzliche Optionen wie einmaliger Datenzugriff oder die Weitergabe von Informationen zu modellieren. Dies kann für eine Entscheidung, ob ein Zugriff auf persönliche Daten erlaubt sein soll oder nicht, bedeutsam sein. Als Optionen sind für das Zugriffsrecht <READ> vorgesehen:

- ”once-only”: Die Information darf nur einmal gelesen werden. D.h. wenn die Daten mehrmals verwendet werden soll, muss sie auch mehrmals angefordert werden. Dies soll auch bedeuten, dass die betreffende Information beim Dienst nicht länger als nötig zwischengespeichert werden darf bzw. sofort nach Ende der Transaktion wieder gelöscht werden muss. Als Beispiel kann man sich einen einmaligen Zugriff auf Daten für eine Kreditkarten- oder Lastschriftbuchung für die Abwicklung einer Bestellung vorstellen. Nach Abschluss der Bestellung müssen die Informationen gelöscht und bei einer neuen Bestellung neu angefordert werden.

- ”distributable”: Die Informationen dürfen weitergegeben werden. An wen dies erlaubt ist, kann durch die (optionale) Spezifikation der möglichen Empfänger festgelegt werden:

  – Dazu muss ein zusätzliches Attribut RECIPIENT beim Zugriffsrecht angegeben werden. Dabei sind die im P3P-Standard für den entsprechenden Abschnitt einer P3P-Erklärung möglich [P3P02]: u.a. „ours“, „same“ (Dienste mit äquivalenten Datenschutzpraktiken), „other-recipient“ (Dienste mit abweichenden Datenschutzpraktiken), „unrelated“ oder „public“ (z.B. für öffentliche Diskussionsforen)

  – Wenn kein RECIPIENT explizit angegeben ist, wird „ours“ angenommen, d.h. die Informationen darf nur sehr eingeschränkt weitergegeben werden, z.B. an einen anderen Dienst der gleichen Firma.

- ”subscription”: Dies bedeutet, dass der im AT spezifizierte Dienst zusätzlich benachrichtigt wird, wenn sich der betreffende Teil des Benutzerprofils ändert. Aus Sicht der Privatheit erscheint dies zunächst nicht notwendig, aber der Benutzer erhält dadurch zumindest auch die Gewissheit, dass von ihm geänderte Daten auch weitergegeben werden. Dies kann zu einer Verbesserung der Transparenz für den Benutzer bei der Zugriffskontrolle beitragen.

Diese Optionen werden als Attribute bei den Rechten mit dem Attributnamen OPTION festgelegt. Dies ist sowohl bei AR als auch AT denkbar (Abb. 4.10).

Bei einer Veröffentlichung von personenbezogenen Daten, z.B. auf WWW oder öffentlich zugänglichen Community Support Systemen, müssen grundsätzlich die Optionen OPTION=„distributable“ und RECIPIENT=„public“ verwendet werden. In diesem Fall kann das AT auch weitergegeben werden und ist für einen anderen Dienst ohne Neuverhandlung gültig. D.h. ein weiterer Dienst kann auf Profildaten zugreifen, obwohl ein anderer Dienst im Element <SERVICE> des AT vermerkt ist.

Eine Kombination von „once-only“ mit „subscription“ ist nicht möglich. Bei den Rechten <WRITE>, <APPEND>, <CREATE> und <DELETE> sind „distributable“ und „subscription“ überhaupt nicht, „once-only“ nur sehr eingeschränkt sinnvoll. Daher werden die Optionen in diesem Teilabschnitt nur für <READ> vorgesehen.

```
<ACCESS RESOURCE="/profile/demographic/postal/locality">
  <READ OPTION="subscription"/><PURPOSE>...</PURPOSE></ACCESS>
<ACCESS RESOURCE="/profile/payment/creditcard/number">
  <READ OPTION="once-only"><PURPOSE>...</PURPOSE></ACCESS>
```

ABBILDUNG 4.10: Beispiele für Optionen bei Zugriffsrechten

Eine Weitergabe von Daten ist auch bei anonymisiertem Datenzugriff nicht gestattet, wenn es nicht explizit erlaubt ist. Durch die Möglichkeit einer Vermischung verschiedener Datenbestände können oftmals vermeintlich anonyme Daten tatsächlichen Benutzern zugeordnet werden, obwohl dies vom Benutzer nicht erwünscht ist. Daher muss eine Weitergabe von Daten grundsätzlich im AT gestattet sein.

Diese Optionen erfüllen u.a. die Anforderungen einer Kontrolle der Weitergabe von Daten und stellen ist eine Neuerung in der Zugriffskontrolle dar. Bisher sind entsprechende Elemente nur im P3P-Standard vorgesehen.

Beim Schreiben oder Anlegen von Benutzerprofilattributen von einem Dienst wäre eine Option „exclusive“ oder ähnliches denkbar, welche spezifizieren könnte, dass das Attribut im Profil nur für den betreffenden Dienst vorgesehen ist. Dies wurde jedoch nicht in den Entwurf des AR/AT aufgenommen, da es aus der Sicht der Privatheit des Benutzers keine Bedeutung hat und außerdem ein Dienst-proprietäres Profilfeld für einen anderen Dienst kaum verwendbar ist. Ggf. könnte ein Dienst ein Element verschlüsselt und mit einer digitalen Signatur in das Profil schreiben, um eine Vertraulichkeit gegenüber anderen Diensten zu gewährleisten.

4.2.3.4 Optionen bezüglich Datenübertragung

Des Weiteren sind Optionen bezüglich der Datenübertragung bei Verwaltung von Benutzerprofilen wichtig. Dies ergibt sich aus den Anforderungen und insbesondere den Schutzzielen Vertraulichkeit und Anonymität. Zum Beispiel sollen Kreditkartendaten nicht unverschlüsselt über das Internet übertragen werden. Es ist daher wichtig, dies bei der Zugriffskontrolle zu berücksichtigen. Es sind dazu folgende Optionen möglich, die bei den Zugriffsrechten eingefügt werden können (s.a. Abb. 4.11):

- <SIGNED>: Damit kann der Benutzeragent einzelne Teile des Access Tickets mit einer digitalen Signatur im Namen des Benutzers versehen. Dies könnte für eine Nachweisbarkeit bzw. Unabstreitbar des Zugriffsrechts für einen Dienstagenten von Interesse sein, z.B. für die explizite Erteilung eines Auftrags zum Lastschrifteneinzug beim Zugriff auf Zahlungsinformationen.

- <SECURE>: Die Daten werden über einen gesicherten Kanal übertragen. Dazu kann optional auch ein Attribut TYPE zur genaueren Spezifikation angegeben werden. Dies könnte z.B. die

Übertragung mittels Secure Socket Layer (SSL) sein.

- <ENCRYPTED>: Die Informationen werden verschlüsselt übertragen und mit dem Public Key eines Dienstes kodiert. Zur Dekodierung verwendet der Dienst seinen Private Key, um sicherstellen zu können, dass nur der angesprochene Dienst die Information entschlüsseln kann. Die Information kann dabei auch über verschiedene Zwischenknoten übertragen werden, während bei <SECURE> ausdrücklich ein sicherer Kanal zwischen Benutzerprofil- und Dienstagenten gefordert wird.

- <ANONYMIZED>: Die Daten sollen auf Kommunikationsebene anonymisiert übertragen werden. Dazu kann der Benutzerprofilagent die Möglichkeiten von Privacy Enhancing Technologies (vgl. Kapitel 3.2), wie etwa Anonymisierungs-Proxies, nutzen. Dies dient in erster Linie dazu, um gegenüber ungefugten Dritten einen Nachrichtenverkehr zwischen Benutzerprofil- und Dienstagent zu verbergen, während die Identitätsstufen bei <USER> eine gewisse Anonymität des Benutzers gegenüber dem Dienst gewährleisten sollen.

Diese Optionen können sowohl im AR als auch im AT vorhanden sein. <SIGNED> und <ENCRYPTED> können auch in Kombination verwendet werden, während dies mit <ANONYMIZED> nicht sinnvoll ist. Es sind verschiedene Optionen bei verschiedenen Rechten für dieselbe Ressource möglich, z.B. „Lesen“ anonymisiert und „Schreiben“ über einen sicheren Kanal.

```
<ACCESS RESOURCE="/profile/payment/creditcard/number">
  <READ><SECURE TYPE="SSL"/></READ><PURPOSE>...</PURPOSE>
</ACCESS>
<ACCESS RESOURCE="/profile/payment/lastschrift/*">
  <READ><SIGNED/></READ><PURPOSE>...</PURPOSE>
</ACCESS>
```

ABBILDUNG 4.11: Beispiele für <SECURE> und <SIGNED>

4.2.3.5 Zusätzliche Bedingungen

Ein Element <CONDITION> kann zur Spezifikation zusätzlicher Anforderungen, Restriktionen und Bedingungen angegeben werden (Abb. 4.12). Dies kann insbesondere auch der Absicherung und Transparenz für den Benutzer dienen. Möglicher Inhalt der Bedingung:

- Referenz auf anderen Access Request bzw. Access Ticket („AT nur gültig, wenn anderer AR auch genehmigt wird“)

- Frei-Text, z.B. als Referenz auf gesetzliche Bestimmungen oder als Hinweis auf eine Bestellung („erst bei/nach Auslieferung einer Bestellung ist der Zugriff auf Zahlungsinformationen erlaubt“)

- Inhalt des <REMEDIES> Elements des P3P-Standards [P3P02], zum Beispiel: <REMEDIES> <money/></REMEDIES>

- Technische oder andere Einschränkungen beim Zugriff, z.B. von einem Rechner mit einer bestimmten IP-Adresse aus

Mit Hilfe der Referenz auf einen anderen AR kann eine „Gegenleistung eines Partners“ spezifiziert werden. In dem hier betrachteten Modell greift immer ein „Dienst“ auf das Profil eines „Benutzers“ zu. Ein Benutzer kann jedoch auch die Rolle „Dienst“ übernehmen, z.B. bei einem gegenseitigen Austausch von Interessen zweier Benutzer, wozu zwei unabhängige AR/AT erforderlich sind. Daher ist es sinnvoll, die Gültigkeit eines AR von der Gültigkeit des anderen abhängig machen zu können. Der Unterschied zwischen <PURPOSE> und <CONDITION> ist, dass <PURPOSE> den Zweck des Datenzugriffs bezeichnet, der verpflichtend angegeben werden muss, während <CONDITION> (optionale) Bedingungen angibt, die zusätzlich erfüllt sein müssen, damit ein Zugriffsrecht gültig ist.

```
<ACCESS RESOURCE="/profile/payment/creditcard/number">
  <READ><SECURE></READ><PURPOSE>subscription</PURPOSE>
  <CONDITION TYPE="free-text">Die Daten müssen nach
  Abschluss der Transaktion gelöscht werden.</CONDITION>
</ACCESS>
```

ABBILDUNG 4.12: Beispiel für <CONDITION> als „free-text“

4.2.3.6 Startdatum

Wenn das AT nicht ab Ausgabe gelten soll, sondern erst später, kann als optionales Element im AR/AT auch ein Startdatum angegeben werden: <STARTDATE>

4.2.4 Komplettes Beispiel

Abbildung 4.13 zeigt ein komplettes Beispiel eines Access Requests.

4.2.5 Vergleich mit bestehenden Ansätzen

Der Kern dieser Arbeit, der Access Request und das Access Ticket, soll nun mit bestehenden Ansätzen verglichen werden und eine Bewertung erfolgen. Eine ausführliche Evaluation findet sich am Ende dieses Kapitel nach der Darstellung des gesamten Mechanismus.

Bestehende Systeme für Zugriffskontrolle wie RBAC (vgl. Abschnitt 3.1) sind tauglich innerhalb einer Community oder einer Firma, bei der Administratoren dedizierte Rechte für einzelne Benutzer oder Benutzergruppen festlegen. In dem hier betrachteten Szenario einer dezentralen Verwaltung von Benutzerprofilen ist es jedoch sehr schwierig für den Benutzer, z.B. geeignete Rollen festzulegen. Beim Zugriff auf personenbezogene Daten gibt es im Grunde nur zwei Rollen oder Gruppen, nämlich zum einen der Benutzer selbst, der vollständige Kontrolle über sein Profil haben sollte und zum anderen Dienste (oder andere Benutzer), die mehr oder weniger Zugriffsrechte auf Teile des Profils haben, nach Maßgabe des Profilinhabers. Daher ist ein Rollen- oder Gruppen-basiertes System hier wenig geeignet. Besser ist ein flexibler Ansatz basierend auf Datenschutzerklärungen und -präferenzen, zusammen mit einer Integration von Konzepten aus der Zugriffskontrolle, wie z.B. des Zugriffsmodus. Eventuell können aber bestehende Ansätze verwendet werden, um Zugriffsregeln zur Aushandlung von Access Tickets zu formulieren, dies wird später nochmal aufgegriffen (vgl. Abschnitt 4.3.3).

Der wichtigste Aspekt ist, dass der Ansatz hier speziell für Benutzerprofile entwickelt wurde. Dies ist gerade bei personenbezogenen Daten sehr wichtig. Beispielsweise ist im Unterschied zu anderen Anwendungsdomänen eine explizite Angabe des Zwecks essentiell und z.T. auch rechtlich

```
<ACCESSREQUEST>
  <USER LEVEL="pseudonymous">nickname123</USER>
  <SERVICE TYPE="X.500">@c=DE@o=AMAZON</SERVICE>
  <VALIDITY>31.12.2002</VALIDITY>
  <STARTDATE>01.04.2002 12:00</STARTDATE>
  <POLICY>http://www.server.com/pathto/p3p.xml</POLICY>
  <ACCESS RESOURCE="/profile/interests/*" OPTION="optional">
    <READ/>
    <WRITE/>
    <PURPOSE><p3p:tailoring/></PURPOSE>
  </ACCESS>
  <ACCESS RESOURCE="/profile/demographic/postal/locality"
          GROUP="postal">
    <READ/>
    <PURPOSE><p3p:delivery/></PURPOSE>
  </ACCESS>
  <ACCESS RESOURCE="/profile/demographic/postal/zip"
          GROUP="postal">
    <READ/>
    <PURPOSE><p3p:delivery/></PURPOSE>
  </ACCESS>
  <ACCESS RESOURCE="/profile/payment/creditcard/number"
          ALTERNATIVE="payment">
    <READ OPTION="once-only"><SECURE><READ/>
    <PURPOSE>Bestellung Nr. 12345</PURPOSE>
    <CONDITION>Die Daten werden nach Abschluss der
        Transaktion gelöscht.</CONDITION>
  </ACCESS>
  <ACCESS RESOURCE="/profile/payment/lastschrift"
          ALTERNATIVE="payment">
    <READ OPTION="once-only"><SECURE/><SIGNED/></READ>
    <PURPOSE>Bestellung Nr. 12345</PURPOSE></ACCESS>
</ACCESSREQUEST>
```

ABBILDUNG 4.13: Beispiel Access Request

vorgeschrieben. Daher ist ein allgemeines Schema, wie es z.B. mit XACML verwirklicht ist, nicht ausreichend.

Allerdings ist der Entwurf des AR/AT stark an bestehenden Ansätzen angelehnt und somit keine komplette Neuerfindung. Die gewählte XML-Form ist vergleichbar mit den schon besprochenen XML-Formaten XACML, XML Ticket und den Schemas für Digital Rights Management (DRM), sowohl vom Prinzip als auch einer Abbildung einzelner Elemente, wie z.B. unterschiedlicher Zugriffsmodi, her. Als XML-Dokument ist das AR/AT relativ leicht erweiterbar, um es z.B. für spezielle Anwendungszwecke anpassen zu können. Es erfolgt auch eine Integration von P3P-Terminologie zur Abbildung von Datenschutz-relevanten Sachverhalten. Die Access Tickets ermöglichen damit eine für Benutzer transparente Abbildung der für den Zugriff auf Benutzerprofile wichtigen Aspekte.

4.3 Aushandlung der Zugriffsrechte (Phase I)

Nach der Beschreibung der Access Requests und Tickets, sollen nun in den beiden folgenden Abschnitten die beiden Phasen des Mechanismus erarbeitet werden: Aushandlung von AT's (Kap. 4.3) und Datenzugriff mit den AT's (Kap. 4.4). Zunächst wird die Aushandlung der Rechte beschrieben, also die Ausgestaltung der Phase I.

4.3.1 Herausgabe versus Aushandlung von Access Tickets

Eine Motivation der Aufteilung des Zugriffsschutzmechnismus in zwei Phasen ist es u.a., die Möglichkeiten bzgl. der Phase I (Aushandlung der Zugriffsrechte) möglichst flexibel halten zu können (vgl. 4.1.2.1). Diese Alternativen sollen in diesem Abschnitt diskutiert werden.

Eine erste Möglichkeit wäre, dass Benutzer selbst bei Bedarf Access Tickets herausgeben, unterstützt von einer geeigneten Benutzerschnittstelle. Der Hauptvorteil wäre, dass die Kontrolle des Zugriffs auf jeden Fall beim Benutzer liegt. Außerdem wäre es ein recht einfacher Mechanismus und Benutzer könnten die Ausgabe an den jeweiligen Dienst manuell anpassen. Allerdings hat dies einige schwerwiegende Nachteile:

- Eine manuelle Herausgabe von AT's ist wenig flexibel, da bei jeder Ausgabe genau die Bedingungen spezifiziert werden müssen, die gelten sollen. Es muss dabei vom Benutzer selbst eine Zugriffskontrolle mit expliziter Vergabe der Rechte administriert werden, so dass die in Abschnitt 3.1 dargestellten Nachteile traditioneller Zugriffskontrolle zur Geltung kommen.

- Wie kann der Benutzer bei unbekannten Communities oder anderen Diensten vorgehen? Die Herausgabe von Zugriffsrechte sollte u.a. abhängig vom Zugriffszweck und den Datenschutzpraktiken des Dienstes sein, was der Benutzer bei manueller Rechtevergabe selbst herausfinden und auswerten müsste.

- Benutzer müssen außerdem ein tieferes Wissen und Verständnis des Benutzerprofilverwaltungssystems aufbringen. Eine manuelle Herausgabe von AT's ist wenig geeignet für Benutzer, die sich nicht eingehender mit der Profilverwaltung beschäftigen wollen oder können.

- Eine für den Benutzer relativ aufwändige manuelle Herausgabe von AT's könnte dazu führen, dass Benutzer das System wenig verwenden, so dass ein Problem der Akzeptanz entsteht.

Eine rein manuelle Herausgabe von Rechten oder Administration von Zugriffsrechte ist daher nicht ausreichend. Zumindest sollte der Benutzer bei der Ausgabe der AT's vom System unterstützt werden. Gerade das Automatisieren der Vergabe der Zugriffsrechte ermöglicht es auch Laien, eine sinnvolle und kontrollierte Benutzerprofilverwaltung durchzuführen. Deshalb ist ein Mechanismus, der Rechte anhand von Regeln aushandelt, sinnvoll. Die Regeln könnten dabei von vertrauenswürdigen Organisationen, wie Datenschutzvereinigungen, bereitgestellt werden. Auch unterstützt eine „Aushandlung" zwischen autonomen Komponenten den Ansatz der Trennung von Benutzerprofilverwaltung in Benutzerprofilagenten und – unabhängig davon – die Nutzung der Daten bei den Diensten. Eine völlige Automatisierung der Rechtevergabe ist aber aus folgenden Gründen auch nicht optimal:

- Es könnte möglicherweise schwierig sein, Regelmengen für die Aushandlung zu spezifizieren die die Intentionen des Benutzers exakt abbilden, ohne dass dieser noch in den Aushandlungsprozess eingreifen kann.

- Fortgeschrittene Benutzer möchten mehr Kontrolle auf die Ausgabe von AT's ausüben.

- Durch die Automatisierung könnten eventuell von Benutzer unerwünschte Rechte herausgegeben werden, wenn die Regeln nicht die Intentionen des Benutzers vollständig und korrekt abbilden können.

- Problem der juristischen Relevanz hinsichtlich Datenschutz-Gesetzen, wenn (Benutzer-)Agenten oder andere Programme „selbstständig“ Vereinbarungen aushandeln [CrRe02].

- Benutzer könnten verunsichert werden, wenn sie nicht genau nachvollziehen und ggf. beeinflussen können, wie die Rechte vergeben werden (fehlendes Vertrauen).

Daher erscheint als beste Lösung eine semi-automatische Aushandlung mit Integration von Benutzerinteraktion und zusätzlicher Option für den Benutzer, Rechte selbst festzulegen und AT's ohne „Aushandlung“ herauszugeben. Dies beinhaltet z.B. auch die Möglichkeit für den Benutzer in der Regelmenge festzulegen, dass er bei jeder Zugriffsentscheidung explizit seine Zustimmung geben muss. Des Weiteren kann der Benutzer ausgegebene Rechte jederzeit wieder zurückziehen.

Besonders wichtig ist es im diesem Zusammenhang, auch entsprechende Transparenz-Funktionen für den Benutzer vorzusehen. Entscheidend ist dabei auch die Ausgestaltung des Dialogs des Benutzerprofilagenten mit dem Benutzer.

Im Folgenden wird der semi-automatische Mechanismus zur Aushandlung der Rechte im Detail erarbeitet, wobei insbesondere die dazu notwendigen Zugriffsregeln besprochen werden.

4.3.2 Protokoll

Zunächst wird in diesem Teilabschnitt das zur Aushandlung der Access Tickets nötige Protokoll erarbeitet. Die Aushandlung basiert auf Nachrichten zwischen den Komponenten Benutzerprofilagent (UPA) und Community Agent (CA). Die Interaktion wird dabei im allgemeinen Fall vom CA initiiert, zum Beispiel wegen einer Aktion eines Benutzers, etwa der Nutzung einer Personalisierungsfunktion einer Web-Seite. Das Schema für die Aushandlung sieht den Austausch verschiedener Anforderungen vor. Ein CA hat dabei verschiedene Anfragen (AR's) zur Auswahl, sowie verschiedene, möglicherweise unterschiedlich restriktive P3P-Erklärungen. Die Anfragen werden vom UPA anhand der Zugriffsregeln ausgewertet, wobei auch eine Interaktion mit dem Benutzer erforderlich sein kann. Dies entspricht nicht (uneingeschränkt) dem Paradigma einer völlig autonomen Aushandlung zwischen Agenten, wie es z.B. im Bereich der Künstlichen Intelligenz (KI) untersucht wird, da ein vollständig automatisiertes Aushandlungs-Protokoll wie eben erläutert bei der Verwaltung von persönlichen Daten nicht sinnvoll ist. In dieser Phase I ist auch die manuelle Ausgabe von Access Tickets enthalten.

Folgende Frage wird in dieser Arbeit nicht näher behandelt: Wie findet ein Dienstagent den Benutzerprofilagenten eines Benutzers? Dies wird hauptsächlich auch in den bestehenden Ansätzen zu Identitätsmanagement wie dem Liberty Alliance Project erarbeitet (vgl. Abschnitt 3.3). Bei offenbarter Identität kann man sich einen (föderierten) Verzeichnisdienst (z.B. X.500 oder LDAP) vorstellen, der eine Zuordnung von Benutzerkennungen zu Benutzerprofilagenten gewährleistet. Bei Pseudonymen ist dies in ähnlicher Weise denkbar, nur dass der Identifikator des Benutzers nicht auf seine wahre Identität schließen lässt, sondern eine Kennung eines Anonymisierungdienstes darstellt. Auch bei der Identitätsstufe „anonymous“ ist dies im Prinzip möglich. Die bestehenden Anwendungen zu Identitätsmanagement im Internet betrachten schwerpunktmäßig die Problemstellungen pseudonymer und anonymer Identifikation und Authentifikation, während der Fokus in dieser Arbeit auf Autorisation liegt, also der Abbildung von Zugriffskontrolle und Zugriffsrechten bei Verwaltung von Profilen, unter Berücksichtigung von föderierter Benutzerprofilverwaltung und Aspekte verschiedener Identitäten.

4.3.2.1 Ablauf

Der Ablauf der Aushandlung zwischen UPA und CA ist in Abb. 4.14[2] dargestellt. Gemäß dem Agenten-Paradigma werden die Abläufe als einzelne (asynchrone) Nachrichten dargestellt, eine Zusammenfassung von Nachrichten in synchrone, entfernte Prozeduraufrufe ist natürlich denkbar.

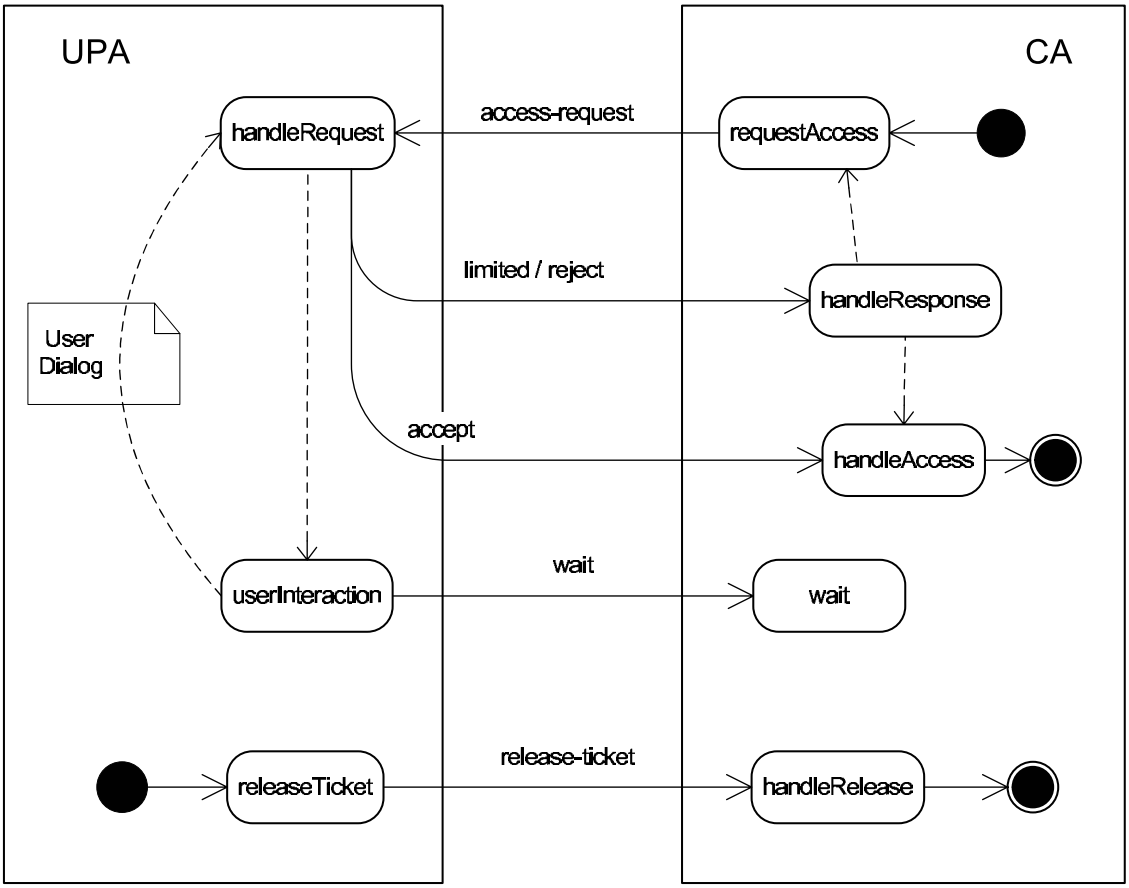

ABBILDUNG 4.14: Ablauf Phase I

4.3.2.2 Nachrichten vom CA zum UPA

```
access-request (ServiceID serviceid, AccessRequest ar)
```

Beschreibung der Nachricht vom CA zum UPA:

- „access-request“: Zugriffsanforderung eines Dienstes mit Parameter[3]:

[2] Die Rechtecke mit abgerundeten Ecken in Abb. 4.14 stellen die Zustände (z.B. “handleRequest“) der beiden Komponenten UPA und CA dar. Ein ausgefüllter Kreis bedeutet, dass der folgende Zustand ein Startzustand des Protokolls ist. Ein aufgefüllter Kreis mit zusätzlichem Kreis stellt einen Endzustand dar. Die durchgezogenen Pfeile sind die Nachrichten des Protokolls zwischen den Komponenten. Ein gestrichelter Pfeil zwischen zwei Zuständen bedeutet einen Übergang von einem Zustand in einen anderen Zustand einer Komponente ohne Nachricht. Dies ist z.B. dann der Fall, wenn beim Auswerten der Zugriffsregeln im UPA eine Benutzerinteraktion erforderlich ist.

[3] Die Parameter sind die nötigen Daten des (entfernten) Funktionsaufrufes. Bei Agentenkommunikation ist dies der Inhalt der Nachricht zwischen den Agenten.

  - serviceid: Identifikator des Dienstes (vgl. Element <SERVICE> im AR), ggf. mit digitalem Zertifikat belegt
  - ar: Der Access Request mit den gewünschten Zugriffsrechten

Die gewünschten Rechte, Ressourcen, Optionen etc. sind im AR enthalten. Der AR enthält darüber hinaus auch einen Verweis auf eine P3P-Datenschutzerkärung des Dienstes, die verpflichtender Bestandteil der Zugriffsanfrage des Dienstes ist.

4.3.2.3 Nachrichten vom UPA zum CA

```
accept (UserID userid, AccessTicket at)
limited (UserID userid, AccessTicket at, String limitedmsg)
reject (UserID userid, String rejectmsg)
wait (UserID userid, AccessRequest ar, String waitmsg)
release-ticket (UserID userid, AccessTicket at)
```

Folgende Nachrichten sind für die Kommunikation vom UPA zum CA möglich, i.d.R. als Reaktion auf einen „access-request“:

- „accept“: Anfrage wird (komplett) erlaubt, ein entsprechendes Access Ticket wird ausgestellt, Parameter:
  - userid: Identifikator des Benutzerprofils (vgl. <USER> im AT)
  - at: Access Ticket mit den genehmigten Zugriffsrechten
- „limited“: Anfrage wird nur zum Teil erlaubt, AT für den genehmigten Teil der Anfrage, sowie der Grund der Ablehnung anderer Anfrageteile, werden mitgeschickt, Parameter:
  - userid
  - at: Access Ticket mit den genehmigten Zugriffsrechten (nur Teil des AR)
  - limitedmsg: Grund der Ablehnung (dies wird nachfolgend noch behandelt)
- „reject“: Anfrage wird nicht erlaubt, CA kann neue Anfrage mit modifiziertem AR und/oder P3P-Erklärung stellen, Parameter:
  - userid
  - rejectmsg: Grund der Ablehnung (dies wird nachfolgend noch behandelt)
- „wait“: Wartezustand wegen notwendiger Benutzerinteraktion, eine weitere Nachricht folgt nach der Entscheidung, Parameter:
  - userid
  - ar: betreffender Access Request
  - (optional:) waitmsg: Begründung des Wartezustandes
- „release-ticket“: Der Benutzer bzw. UPA gibt ohne Anforderung des Dienstes ein AT heraus, Parameter:
  - userid

  - at: (neues) Access Ticket

Der Grund der Ablehnung bei einer „reject“ oder „limited“ Nachricht kann u.a. in folgende Kategorien eingeteilt werden:

- Formale Ursache, z.B. keine P3P-Erklärung vorhanden oder AR nicht korrekt
- Ungenügende P3P-Erklärung (enthält Datenschutzpraktiken, die nicht akzeptabel für den Benutzer sind)
- Zugriff auf bestimmte Attribute kann nicht gestattet werden, der Dienst kann mit einem verändertem AR reagieren (z.B. keine „Weitergabe von Daten“, oder Abwandlung der Optionen eines Zugriffs)

Bei Ablehnung von Zugriffsanfragen kann der CA eine neue Anfrage mit einem modifiziertem AR und/oder P3P-Erklärung stellen. Damit kann eine Annäherung der Anfrage des Dienstes an Benutzer-Präferenzen erreicht werden und es kommt zu einer Aushandlung der Zugriffsrechte zwischen CA und UPA.

Die Nachricht „wait“ dient dazu, dem Dienst mitzuteilen, dass eine Benutzerinteraktion erfolgt und daher eine Zugriffsentscheidung ggf. auf sich warten lässt. Der Dienst kann dann entsprechend reagieren und z.B. die Anfrage aus einer Liste der aktuell zu bearbeiteten Anfragen entfernen. „wait“ ist nicht unbedingt notwendig, es könnte im Rahmen des Protokolls darauf verzichtet werden, da die Nachricht nur zur Information des Dienstes dient.

4.3.2.4 Beschreibung der Zustände

In diesem Teilabschnitt werden die Zustände der Agenten aus Abb. 4.14 erläutert. Mögliche Zustände bei einem Dienstagenten:

- „requestAccess“
  - Startzustand des Protokolls
  - Dienst schickt eine Zugriffsanforderung (AR) an Benutzerprofilagenten
- „handleResponse“
  - CA wertet die Antwort des UPA aus
  - evtl. ergibt sich daraus, die Notwendigkeit, eine weitere, modifizierte Anfrage zu stellen ($\rightarrow$ „requestAccess“)
  - oder für den Dienst sind die gegebenen Rechte ausreichend ($\rightarrow$ „handleAccess“)
- „handleAccess“
  - Anfrage ist OK, Dienst bekommt gewünschte Rechte
  - erhaltendes AT wird abgespeichert, zur späteren Verwendung zum Datenzugriff in Phase II
  - Aushandlungsprotokoll ist beendet
- „wait“

  - Benachrichtigung des UPA, dass Benutzerinteraktion erforderlich ist und UPA daher die Zugriffsanfrage nicht sofort behandeln kann
  - evtl. wird CA die Anfrage aus einer Menge der aktuell offenen Anfragen entfernen
  - UPA wird noch weitere Nachricht mit Zugriffsentscheidung senden (ohne weitere Anfrage des CA)

- „handleRelease“
  - CA erhält ein (manuell ausgegebenes) AT von UPA

Stati beim Benutzerprofilagenten:

- „handleRequest“
  - UPA erhält eine Anforderung eines CA
  - UPA evaluiert diese (anhand von Regeln des betreffenden Benutzers) und schickt eine Antwort
  - evtl. ist Benutzerinteraktion erforderlich → „userInteraction“
  - der Ablauf der Evaluation wird später noch genauer behandelt (siehe Abschnitt 4.5.4)
- „userInteraction“
  - UPA muss auf eine Entscheidung des Benutzers warten
  - nach dem Benutzerdialog geht UPA (erneut) in den Status „handleRequest“, um dem CA anhand des Ergebnisses der Benutzerinteraktion eine weitere Nachricht zu senden
- „releaseTicket“
  - mauelle Herausgabe eines AT an einen Dienst

4.3.3 Zugriffsregeln

Der wichtigste Bestandteil des Zugriffsschutzmechanismus sind geeignete Zugriffsregeln, um den Zugriff auf einzelne Benutzerprofilelemente bei einem „handleRequest“ im UPA entscheiden zu können.

4.3.3.1 Elemente der Entscheidung einer Zugriffsanfrage

Grundsätzlich kann man sich folgende Elemente zur Entscheidung einer Zugriffsanfrage vorstellen:

- Zugriffsregeln: dies wird im Folgenden ausführlich erläutert
- Benutzerinteraktion: Benutzer entscheidet selbst über einen geeigneten Dialog mit dem Benutzerprofilagenten über „Zugriff“ oder „kein Zugriff“
- Konfigurationseinstellungen: gültig für alle Zugriffsanfragen, inklusive allgemeiner Festlegungen (z.B. „ohne <PURPOSE> im AR kein Zugriff“)
- (sinnvolle) Entscheidungen im Algorithmus: z.B. bei „Alternativen“ im AR wird eine der Möglichkeiten ausgewählt oder bei „Gruppierung“ wird bei allen Elementen gleich entschieden, da ein Zugriff auf nur einen Teil der Gruppe nicht sinnvoll ist

- manuelle Einstellungen: dies kann der Benutzer bei einer manuellen bzw. von der Benutzeroberfläche unterstützten Herausgabe von AT's vornehmen

Sicherlich ist es erforderlich, eine Kombination der Konzepte zu verwenden. Ein Vorteil davon ist, dass das eigentliche Regelsystem nicht durch unnötige Parameter überladen wird, wenn einige grundlegende Einstellungen außerhalb des Regelsystems verwaltet werden. Dadurch kann eine größte Flexibilität und auch Transparenz für den Benutzer erreicht werden, da er gewünschte Einstellungen nicht nur über Regeln machen kann, sondern auch z.B. über einfachere Konfigurationseinstellungen.

Im Folgenden werden die Zugriffsregeln im Detail betrachtet. Bei den einzelnen Aspekten wird ggf. jeweils auch auf die weiteren Elemente wie z.B. mögliche Konfigurationeinstellungen eingegangen.

4.3.3.2 Erweiterung des „yes"/„no" Modells der Zugriffskontrolle

In dem betrachteten Szenario macht es Sinn, bei den Zugriffsregeln das einfache Modell von „hat Zugriff" und „hat keinen Zugriff" einer traditionellen Zugriffsentscheidung zu erweitern. Man kann sich vorstellen, weitere *Zugriffsniveaus* einzuführen:

- „no-access": Zugriff auf angefordertes Attribut ist nicht erlaubt
- „if-necessary": Zugriff wird gestattet, „wenn es (unbedingt) erforderlich ist" (zur Erbringung der Personalisierungs- oder sonstigen Funktion eines Dienstes)
- „user-interaction": Zugriffsanforderung kann nicht (automatisch) entschieden werden, der Benutzer trifft die Entscheidung (manuell, über einen geeigneten Dialog mit dem Benutzerprofilagenten)
- „access": Zugriff ist erlaubt (unter Beachtung der angegebenen Bedingungen, z.B. dem Zugriffszweck)

Die Zugriffniveaus entsprechen einer Einstellung in einer Zugriffsregel, dessen Auswirkung gleich im Folgenden erläutert wird. Gründe, warum eine Einführung von Zugriffniveaus hier sinnvoll ist, sind insbesondere:

- Benutzer hat evtl. nicht den Überblick, welche Attribute wirklich erforderlich sind und welche nicht so wichtig wären; ein Benutzer will ggf. Daten nur herausgeben, „wenn dies erforderlich" ist
- Benutzerinteraktion ist hier sehr wichtig, daher soll dies auch explizit in den Regeln ausdrückbar sein
- Untersuchungen bei Groupware haben gezeigt, dass das „yes"/„no" Modell nicht immer ausreichend ist, um bestimmte Szenarien, die dem hier betrachteten vergleichbar sind, benutzergerecht abbilden zu können [StWu98]
- Das einfache „yes"/„no" Modell ist hier auch enthalten, wenn man die zusätzlichen Zugriffniveaus nicht verwenden will; d.h. es ist hier nur eine Erweiterung bestehender Modelle zur Zugriffskontrolle

Darüberhinaus könnte man sich z.B. auch ein „if-possible“ vorstellen. Allerdings gibt es zumindest bei Benutzerprofilverwaltung keine sinnvolle Anwendung dafür, so dass „if-possible“ nicht weiter untersucht wird. Das Modell könnte jedoch prinzipiell noch um weitere Zugriffsniveaus erweitert werden.

Die Zugriffniveaus entsprechen den einstellbaren Präferenzen eines Benutzers in den Zugriffsregeln. Davon zu unterscheiden sind die Optionen bei Zugriffsrechten (vgl. Kap. 4.2.3.2) einer Access Requests: „mandatory“ (Attribut ist verpflichtend, aus Dienst-Sicht), „optional“ (Zugriff ist optional) oder keine Option. Diese Optionen können vom Dienst angegeben werden, um die Wichtigkeit einzelner Benutzerprofilattribute auszudrücken. Die Zugriffsniveaus sind die Möglichkeiten in den Zugriffsregeln, die abhängig von den Zugriffsoptionen sind.

Man braucht nun eine Semantik, um für die Kombinationen der Zugriffsniveaus mit den Anfrage-Optionen eine Entscheidung herbeiführen zu können (siehe Tab. 4.1).

	mandatory	**keine Option**	**optional**
access	Y	Y	Y
user-interaction	UI	UI	N
if-necessary	Y	UI	N
no-access	N	N	N

TABELLE 4.1: Zugriffsniveaus und Optionen

„Y“ in Tabelle 4.1 bedeutet, dass der Zugriff letztlich gestattet wird, „N“ analog dazu, kein Zugriff. Man erkennt, dass bei einer Beschränkung auf die beiden Fälle „access“ und „no-access“ die Tabelle trivial ist. Für die restlichen Fälle muss man eine Semantik festlegen. Wenn dies nicht geschehen kann oder nicht sinnvoll ist, soll der Benutzer explizit nach einer Entscheidung in einem Benutzerdialog gefragt werden („UI“ in der Tabelle). Die konkrete Ausprägung aller Felder in der Tabelle ist dabei für das Modell nicht so entscheidend. Man kann sich z.B. auch vorstellen, dass ausgehend von einer sinnvollen Default-Einstellung (wie in Tab. 4.1 gezeigt), ein Benutzer, der tief in das Zugriffssystem eingreifen will, dies durch eine Konfigurationseinstellung ändern kann.

Es ist wichtig festzuhalten, dass das der „Umweg“ über Zugriffsniveaus bzw. das relativ komplizierte Modell durch ein einfacheres Schema ersetzt werden kann, z.B. von einem Anbieter eines Benutzerprofilagenten(-dienstes), ohne dass das restliche Ansatz modifiziert werden muss.

4.3.3.3 Alternativen für Regelsprache

Man kann sich folgende Alternativen zur Realisierung der Regelsprache zur Auswertung von Access Requests vorstellen:

Neuentwurf einer Regelsprache Eine erste Möglichkeit wäre es, eine Regelsprache, die auf die hier geschilderten Bedürfnisse zugeschnitten ist, neu zu entwerfen. Dies wäre notwendig, wenn die bestehenden Systeme nicht entsprechend verwendet oder erweitert werden können, aber zunächst sollen existierende Ansätze untersucht werden.

Extensible Stylesheet Language Transformation (XSLT) Bei der Zugriffsentscheidung muss im Prinzip ein XML-Dokument (Access Request) in ein anderes XML-Dokument (Access Ticket) transformiert werden, unter Verwendung einer Regelsprache. Allgemein gibt es zur Transformation belie-

biger XML-Dokumente den W3C Standard *Extensible Stylesheet Language Transformation (XSLT)* [XSLT99]. Genauer liest XSLT ein Eingabe-XML-Dokument und ein XSLT-Stylesheet (das selbst wieder ein XML-Dokument ist) und produziert daraus einen Dokumentenbaum, der dann wieder serialisiert (z.B. in eine Datei geschrieben) werden kann. Das Ausgabe-Dokument muss kein XML sein. Das XSLT-Stylesheet enthält dabei eine Menge von Template-Regeln, die aus zwei Teilen bestehen:

- Muster: wird mit den Knoten des Eingabe-Dokuments abgeglichen
- Template: wird ggf. instanziiert, um einen Teil des Zielbaums zu formen

Der XSL-Prozessor vergleicht das Muster mit den Knoten im Eingabe-Dokument und generiert bei einem positiven Vergleich einen Teil des Zielbaums anhand des Templates. Abb. 4.15 zeigt ein Beispiel für eine Template-Regel.

```
<xsl:template match="ACCESS[@RESOURCE='/profile/interests/*']/READ">
   <xsl:copy-of select="."/>
</xsl:template>
```

ABBILDUNG 4.15: Beispiel für eine XSLT Template-Regel

Im Beispiel wird ein Teilbaum der Eingabe (also dem Access Request) bearbeitet, das mit dem angegeben Muster positiv vergleichbar ist (matching). Das ist dann der Fall, wenn es ein <READ>-Element als Kind eines <ACCESS>-Elementes gibt, das ein Attribut „RESOURCE“ mit dem Wert „/profile/interests/*“ hat. Die Schablone kopiert in diesem Fall den aktuellen Teilbaum – also das <ACCESS>-Element – in das Zieldokument, also in unserem Szenario dem Access Ticket mit den genehmigten Rechten. Das bedeutet, die Regel drückt aus: „lesender Zugriff auf die Interessen gestattet“.

Das Anwendungsgebiet von XSLT ist die Transformation von XML-Dokumenten, z.B. von XML nach HTML. Der hier nötige Zweck der Spezifikation von Zugriffsregeln ist von diesem Anwendungsgebiet etwas (zu) weit entfernt. XSLT könnte evtl. als Implementierung von in einer anderen Sprache formulierten Regeln verwendet werden, ist aber nicht zur direkten Spezifikation der Zugriffsregeln geeignet, da es dafür zu unübersichtlich und kompliziert ist. Im Einzelnen:

- Schwierige Formalisierung von Bedingungen: dies muss im „match“-Teil einer Template-Regel als Muster formuliert werden
- Die deklarative Regelsprache ist für Laien nur schwer nachvollziehbar; Regeln wären kaum für Benutzer verständlich, was zu einer schlechten Transparenz führen würde
- Zugriffsniveaus und Benutzerinteraktion sind nicht nahe liegend zu integrieren

Bestehende Systeme zur Zugriffskontrolle Eine weitere Alternative wäre die Verwendung und ggf. Erweiterung einer der bestehenden Formalismen, die in Kap. 3 ausführlich besprochen wurden. Am besten geeignet erscheinen dabei XML-basierte Zugriffskontrolle (z.B. XACML), sowie das formale Modell von Simone Fischer-Hübner. Die generelle Frage dabei ist: wie können ausgehend von den bestehenden Formalismen die Konzepte aus AR/AT integriert werden?

XML-basierte Ansätze zur Zugriffskontrolle wie XACML erscheinen zunächst einmal relativ gut auch für den Zugriff auf Benutzerprofile geeignet. Allerdings enthalten sie keine Privatheits-Aspekte, die integriert werden müssten, z.B. als zusätzliche Bedingungen für einen Zugriff. XACML entspricht

auch mehr einem Access Ticket, weniger den zur Aushandlung notwendigen Regeln. Ein Vorteil von XML-basierten Ansätzen ist es, dass sie leicht erweitert werden können, durch Einführung eines zusätzlichen Namensraumes.

Das Modell von Fischer-Hübner integriert Datenschutz-Aspekte in ein formales Modell zur Zugriffskontrolle. Es gibt Gemeinsamkeiten zu dem hier erarbeiteten Mechanismus der Access Tickets, z.B. ist in beiden Ansätzen eine Zweckbindung bei Zugriffen auf personenbezogene Daten vorgesehen. Allerdings müßte das Modell erweitert und modifiziert werden, um es in dem hier betrachteten Szenario einsetzen zu können. Die formale Vorgehensweise entspricht auch nicht dem hier verfolgten Ansatzes einer Aushandlung von Zugriffsrechten zwischen autonom agierenden Komponenten.

Bei beiden Ansätzen ist außerdem eine Integration von Zugriffniveaus und Benutzerinteraktion schwierig zu realisieren.

APPEL Die viel versprechensde Möglichkeit für die Regelsprache ist die für P3P erwickelte Sprache APPEL (vgl. auch Kap. 3.2.5.3). APPEL erlaubt eine Formalisierung von Datenschutz-Präferenzen zum Abgleich mit P3P-Datenschutzerklärungen. Um die hier nötigen zusätzlichen Aspekte abbilden zu können, stehen zur Erweiterung von P3P und APPEL zwei prinzipielle Alternativen zur Verfügung:

- Verwendung des <EXTENSION>-Mechanismus von P3P
- Erweiterung von APPEL-Regeln mit Hilfe eines weiteren, eigenen XML-Namensraumes

Das <EXTENSION> Tag ist eine im P3P-Standard vorgesehene Option zur Erweiterung des in einer P3P-Erklärung möglichen Vokabulars. Damit können z.B. zusätzliche Bedingungen für einen Datenzugriff formuliert werden. Abb. 4.16 zeigt dies anhand einer Integration eines Zugriffsmodus in einem P3P-Ausdruck, sowie einem entsprechenden Teil einer APPEL-Regel.

In ähnlicher Weise könnten auch die weiteren Komponenten eines AR in die P3P-Erklärung integriert werden. APPEL muss dabei nicht modifiziert werden, um die zusätzlichen Elemente abfragen zu können. Allerdings ist eine Verwendung des Extension-Mechanismus von P3P eigentlich nicht notwendig, da im Access Request als Bestandteil der Zugriffsanfrage die Erweiterungen bereits formalisiert sind.

Daher wird die zweite Alternative, nämlich die Verwendung von AR und unveränderte P3P-Erklärung, zusammen mit einer erweiterten APPEL-Regelmenge betrachtet. Wie die einzelnen Aspekte in APPEL abgebildet werden können, wird in den nächsten Teilabschnitten ausführlich erarbeitet.

Die Vorteile einer Verwendung von APPEL als Regelsprache sind:

- APPEL ist als Sprache für Datenschutz-Präferenzen auf eine Abbildung von Privatheitsaspekten ausgelegt
- Zusätzliche Elemente können über XML-Namensräume einfach integriert werden
- Integration von P3P-Terminologie und -Vokabular
- Konzept für Benutzerinteraktion vorhanden („prompt“ in einer APPEL-Regel)
- Gute Umsetzung der Zugriffniveaus möglich, wie im folgenden Abschnitt gezeigt wird
- Integration der weiteren Aspekte eines AR/AT ist recht einfach möglich

```
<STATEMENT>
  <CONSEQUENCE>
    We use this information when you make a purchase.
  </CONSEQUENCE>
  <PURPOSE><current/></PURPOSE>
  <RECIPIENT><ours/></RECIPIENT>
  <RETENTION><stated-purpose/></RETENTION>
  <DATA-GROUP>
   <DATA ref="#user.name"/>
   <DATA ref="#user.home-info.postal"/>
   <DATA ref="#user.home-info.telecom.telephone"/>
   <DATA ref="#user.home-info.online.email"/>
   <DATA ref="#dynamic.miscdata">
    <CATEGORIES><purchase/></CATEGORIES>
   </DATA>
  </DATA-GROUP>
  <EXTENSION>
   <READ/>
  </EXTENSION>
</STATEMENT>

...

<appel:RULE behavior="request" description="Grant read access">
  <p3p:POLICY>
    <p3p:STATEMENT>
      <p3p:DATA-GROUP>
        <p3p:DATA ref="#user.*"/>
      </p3p:DATA-GROUP>
      ...
      <p3p:EXTENSION>
        <READ>
      </p3p:EXTENSION>
    </p3p:STATEMENT>
  </p3p:POLICY>
</appel:RULE>
```

ABBILDUNG 4.16: Zugriffsmodus in einem P3P-Statement

4.3.4 Spezifikation der Zugriffsregeln in APPEL

Im Folgenden wird zunächst erläutert, wie Access Request und Ticket, Zugriffniveaus, Benutzerinteraktion usw. zusammen mit P3P-Vokabular in APPEL-Regeln abbildbar sind. Dazu müssen die APPEL-Regeln, die (nur) für den Abgleich von P3P-Erklärungen vorgesehen sind, erweitert werden, was durch einen neuen Namensraum erreicht wird.

Wie Benutzer die Regeln in sinnvoller Weise festlegen können, mögliche Benutzerschnittstellen und Aspekte verschiedener Identitätsstufen und Identitäten, wird später betrachtet.

4.3.4.1 Namensräume in einer APPEL-Regelmenge

Eine APPEL-Regelmenge besteht aus einer beliebigen Anzahl von APPEL-Regeln, die von einem <appel:RULESET> Tag umschlossen werden (vgl. auch Abschnitt 3.2.5.3).

APPEL kann als XML-Sprache erweitert werden, indem ein eigener Namensraum hinzugefügt wird, nämlich derjenige, der die Access Requests[4] beschreibt. In Abb. 4.17 wird in der 3.Zeile ein zusätzlicher Namensraums mit dem Präfix „ar“ definiert. Damit kann dann AR-Vokabular verwendet werden, um Bedingungen für APPEL-Regeln auszudrücken.

```
<appel:RULESET xmlns:appel="http://www.w3.org/2002/04/APPELv1"
           xmlns:p3p="http://www.w3.org/2000/12/P3Pv1"
           xmlns:ar="http://www.w3.org/..."
           crtdby="W3C" crtdon="1999-11-03T09:21:32-05:00">

   <appel:RULE behavior="block" description="Service collects
             personal data for 3rd parties" prompt="yes">
       <-- Expressions -->
       <ar:READ/>
   </appel:RULE>
</appel:RULESET
```

ABBILDUNG 4.17: APPEL-Ruleset mit neuem Namensraum

Wie man sieht, wird der APPEL-Namensraum schon um den P3P-Namensraum ergänzt, was auch notwendig ist, um P3P-Vokabular in einer APPEL-Regel verwenden zu können.

4.3.4.2 Aufbau einer APPEL-Regel

Wichtige Elemente einer APPEL-Regel sind (vgl. auch Abb. 4.17):

- Regelkopf <appel:RULE ...>
- Regelrumpf

Regelkopf Wie schon in Abschnitt 3.2.5.3 erläutert, wird im Regelkopf einer APPEL-Regel festgelegt, wie sich ein Benutzeragent verwalten soll, wenn der im Regelrumpf spezifizierte Ausdruck mit positivem Ergebnis ausgewertet werden kann. Es gibt in APPEL 1.0 die folgenden drei „Behaviors“[5], die als Attribut im RULE-Element angegeben werden:

- „request“: Die P3P-Erklärung des Dienst ist OK, auf die betreffende URL soll zugegriffen werden
- „block“: Die Erklärung ist nicht OK, die Ressource soll nicht genutzt werden
- „limited“: Die Erklärung ist teilweise annehmbar, der Benutzeragent könnte beispielsweise mit einer Warnung für den Benutzer reagieren

[4] Bei den Zugriffsregeln werden nur Access Requests betrachtet, keine Access Tickets, da letztere das Ergebnis der Aushandlung darstellen.

[5] Der englische Begriff „Behavior“ wird im folgenden als Bezeichnung der APPEL-„behavior“ beibehalten, da hier kein deutscher Begriff gebräuchlich oder nahe liegend ist.

Ausserdem kann in einem APPEL-Regelkopf noch angegeben werden:

- Mit einem Attribut „prompt“, das den Wert „yes“ oder „no“ annehmen kann[6], ob der Benutzer informiert werden soll; mit Hilfe eines Attributs „promptmsg“ kann der Text spezifiziert werden, der dem Benutzer angezeigt werden soll
- Das Attribut „persona“ kann eine Identität des Benutzers bestimmt werden; auf verschiedene Identitäten und deren Realisierungsmöglichkeiten wird später eingegangen
- Ein Verbindungsoperator: or, and, non-or, non-and, or-exact, and-exact

Die Verbindungsoperatoren können genutzt werden, um bei mehreren Ausdrücken in einem Regelrumpf eine Auswerte-Semantik festlegen zu können. Die Bedeutung der Operatoren ist selbsterklärend, der Defaultoperator ist „and“, d.h. alle Ausdrücke in einem Regelrumpf müssen erfüllt sein, damit die Behavior einer Regel ausgeführt wird.

Regelrumpf Im Rumpf einer Regel kann man Ausdrücke angeben, die erfüllt sein müssen, damit eine Regel ausgeführt wird. APPEL benutzt dabei drei mögliche Arten von Ausdrücken [APPEL02]:

1. „expression“: ein boolscher Ausdruck (der also mit einem Ergebnis „richtig“ oder „falsch“ ausgewertet werden kann), der mit einem XML-Element verglichen wird
2. „attribute expression“: kann mit einem einzelnen XML-Attribut und dessen Wert abgeglichen werden
3. „contained expression“: enthält rekursiv weitere Ausdrücke

```
<POLICY>
    <ENTITY>
      <DATA ref="#business.name">
        W3C
      </DATA>
    </ENTITY>
    <STATEMENT>
      <PURPOSE
        appel:connective="or-exact">
        <current/>
        <delivery/>
      </PURPOSE>
    </STATEMENT>
</POLICY>
```

ABBILDUNG 4.18: Beispiel eines Ausdrucks in einem APPEL-Regelrumpf

Eine „expression“ besteht dabei immer aus:

- einem XML-Element, z.B. <SECURE>

[6] „yes“ oder „no“ bedeutet dabei nicht die mögliche Antwort des Benutzers, sondern ist die Vorgabe für den Benutzeragenten, ob dem Benutzer ein Text angezeigt werden soll.

- keinem oder mehrerer „attribute expressions“, z.B. <READ OPTION=”..”>
- keinem oder mehrerer „contained expressions“
- einem optionalen Verbindungsoperator (entspricht den schon aufgeführten Operatoren in einem Regelkopf)

Abb. 4.18 (nach [APPEL02]) zeigt ein Beispiel eines komplexen Ausdrucks. Als (einziger) „Wild Card“ Operator kann der Stern-Operator „*“ benutzt werden.

Der Inhalt des Ausdrucks bezieht sich dabei im allgemeinen Fall auf die auszuwertende P3P-Datenschutzerklärung. In unserem Szenario wird dies um die Elemente des AR erweitert, so dass z.B. auch der Zugriffsmodus als Bedingung einer Regel verwendet werden kann.

4.3.4.3 Zugriffniveaus in APPEL

Eine Kombination von Behavior und „prompt“ kann nun recht nahe liegend verwendet werden, um die Zugriffssemantik aus Tabelle 4.1 in APPEL zu realisieren (Tab. 4.2).

	mandat.	**k. Opt.**	**option.**	⇔	**APPEL-behav.**	**APPEL-prompt**
access	Y	Y	Y	⇔	request	yes | no
user-interaction	UI	UI	N	⇔	limited	yes
if-necessary	Y	UI	N	⇔	limited	no
no-access	N	N	N	⇔	block	yes | no

TABELLE 4.2: Zugriffsniveaus in APPEL

Eine Angabe „limited“ als APPEL-Behavior mit „prompt=yes“ entspricht also z.B. dem Zugriffsniveau „user-interaction“. Durch eine Spezifikation von „limited“ mit „prompt=no“ in einer Regel wird das Niveau „if-necessary“ umgesetzt. Die „request“ und „block“-Fälle sind eindeutig, in diesen Fällen kann optional eine Nachricht dem Benutzer angezeigt werden. Man sieht beispielsweise, dass mit der Einstellung „if-necessary“ in einer Regel bei einem „verpflichtenden“ Attribut der Zugriff erlaubt wird, während dies bei einem „optionalen“ nicht der Fall ist. Ohne Angabe einer Zugriffsoption wird hier eine Benutzerinteraktion ausgelöst, wobei diese Semantik eventuell in Konfigurationseinstellungen vom Benutzer änderbar sein könnte.

4.3.4.4 Verpflichtende Komponenten

Nach diesen Vorüberlegungen zu Zugriffsregeln in APPEL werden nun die einzelnen Aspekte eines AR erarbeitet. Grundsätzlich werden bei der Auswertung eines AR die <ACCESS> Abschnitte eines AR, also die einzelnen Zugriffsanforderungen, der Reihe nach einzeln ausgewertet.

Das Element <USER> wird später ausführlich im Abschnitt „Identitätsmanagement“ (4.5) behandelt, um die Erarbeitung zunächst etwas zu vereinfachen, indem nur von einer Identität des Benutzers ausgegangen wird.

Zugriffsmodus Zugriffsmodi können in einem APPEL-Regelrumpf als Ausdruck angegeben werden. Mit Hilfe der Verbindungsoperatoren kann z.B. auch ein „Lesen oder Schreiben“ formuliert werden (vgl. Abb. 4.19).

Die Angabe eines Zugriffsmodus stellt eine Erweiterung von APPEL in Richtung einer Sprache für Zugriffskontrolle dar.

```
<appel:RULE behavior="request">
  <ar:ACCESS RESOURCE="/profile/interests/*" appel:connective="or">
    <ar:READ/><ar:WRITE/>
  </ar:ACCESS>
</appel:RULE>
```

ABBILDUNG 4.19: Zugriffsmodi in einer APPEL-Regel

Ressourcen „Ressourcen“, also die Benutzerprofilattribute, auf die zugegriffen werden soll, werden im AR in Form einer XPath Angabe spezifiziert (vgl. Abschnitt 4.2.2.5). In ähnlicher Weise können Teile des Benutzerprofils in P3P bzw. APPEL referenziert werden. Dazu können als 1.Möglichkeit P3P-„Data-Groups“ in folgender Weise in APPEL angegeben werden (Abb. 4.20).

```
<p3p:DATA-GROUP>
   <p3p:DATA ref:"#user.name.*">
</p3p:DATA-GROUP>
```

ABBILDUNG 4.20: Ressource im P3P-Stil

Das bedeutet, dass die APPEL-Regel, die diesen Ausdruck enthält, dann erfüllt wird, wenn in der P3P-Erklärung die „Data-Group“ enthalten ist. Dies entspricht dem Zugriff auf die Ressource „/user/name/*“ in der XPath-Schreibweise.

Als zweite Alternative könnte man direkt den ACCESS-Ausdruck aus einem AR in der Regel angeben (Abb. 4.21).

```
<ar:ACCESS RESOURCE="/user/name/*">
```

ABBILDUNG 4.21: Ressource aus AR

Im Gegensatz zu einem P3P-Datenelement ist bei einem AR ein voller XPath Ausdruck vorgesehen. Beide Varianten sind aber gleichbedeutend, da die Schreibweisen leicht transformiert werden können. In dieser Arbeit wird die zweite Alternative verwendet, da diese etwas näher an der in dieser Arbeit verwendeten Notation liegt.

In einer APPEL-Regel können auch mehrere Ressourcen bzw. <ACCESS> Elemente angegeben werden, die ggf. mit dem Verbindungsoperator „or“ verknüpft werden sollten, damit eine Regel für verschiedene Benutzerprofilteile gilt.

Grundsätzlich bleibt bei der Adressierung von Benutzerprofil-Attributen zu bemerken, dass eine gemeinsame Ontologie zwischen UPA und CA notwendig ist. Wenn es beispielsweise im Profil ein Attribut „Zip-Code“ gibt, der Zugriff aber für „PLZ“ angefordert ist, kann kein Abgleich stattfinden. Dies ist allerdings kein Problem der Privatheit oder der Zugriffskontrolle.

Auswertestrategie für Ressourcen Eine wichtige Frage bei der Auswertung von APPEL-Regeln ist, welche Regel verwendet werden soll, wenn Ressourcen (und sonstige Bedingungen) im Regelrumpf zu mehreren Regeln passen. Dies ist insbesondere dann entscheidend, wenn die Behavior der verschiedenen Regeln unterschiedlich ist (z.B. „request“ und „block“). Für die Auswertung gibt es die folgenden Möglichkeiten:

- Es wird diejenige Regel angewendet, die in der Aufschreibung des Regelsatzes später spezifiziert ist (dies ermöglicht die Formalisierung von zunächst allgemeinen Regeln, dann Ausnahmen davon)
- Es wird diejenige Regel angewendet, die früher im Regelsatz spezifiziert ist („wichtige“ Regeln könnten dann zuerst angegeben werden)
- Es wird diejenige Regel angewendet, dessen Ressourcen „spezieller“ sind (also „/profile/payment/creditcard/number“ hätte Vorrang vor „/profile/payment/*“)
- Es wird diejenige Regel angewendet, die weniger Daten herausgibt (also eine Regel mit Behavior „block“ hat Vorrang vor „request“)

In dieser Arbeit wird die letzte Alternative gewählt, u.a. um die Forderung der „Datensparsamkeit“ von Datenschutz-Gesetzen und -richtlinien zu erfüllen. Außerdem ist es für die Transparenz für den Benutzer nicht gerade förderlich, wenn die Auswertesemantik von der Reihenfolge der Regeln im Regelsatz abhängt.

Wenn gar keine Regel angewendet werden kann, wird grundsätzlich der Zugriff abgelehnt. Bei datenschutzfördernden Systemen ist es wichtig, dass Default-Einstellungen immer „kein Zugriff“ ausdrücken.

Service Eine Bedingung für den zugreifenden Dienst – also <SERVICE> in einem AR – kann auch als Ausdruck im Regelrumpf angegeben werden, analog zum Zugriffsmodus. Der Benutzer kann damit einzelnen vertrauenswürdigen Diensten Rechte einräumen oder auch nicht-vertrauenswürdige Dienste ganz (vom Profilzugriff) ausschließen.

Abb. 4.22 zeigt als Beispiel die Umsetzung der Regel „der Dienst www.badguys.com hat grundsätzlich keinen Zugriff“ als APPEL-Regel. Wenn aufgrund einer anderen Regel eine bestimmte Anforderung gestattet werden würde, wird der Zugriff bei dem angegeben Service trotzdem auf jeden Fall nicht erlaubt.

```
<appel:RULE behavior="block" description="Block untrusted service">
  <ar:SERVICE>http://www.badguys.com</ar:SERVICE>
</appel:RULE>
```

ABBILDUNG 4.22: Beispiel-Regel mit <SERVICE>

Damit ist eine Möglichkeit vorhanden, Regeln auch abhängig vom zugreifenden Dienst zu formulieren, wobei die meisten Regeln wohl eher keine Bedingung <SERVICE> enthalten werden, da die Dienste nicht unbedingt a priori dem Benutzer bekannt sind.

P3P-Datenschutzerklärung Die Beziehung der Zugriffsregeln zur P3P-Datenschutzerklärung spiegelt sich in drei möglichen Bereichen wider:

- Grundsätzliche Vorgabe, dass eine P3P-Erklärung vorhanden sein muss (vgl. 4.2.2.4)
- Abgleich mit allgemeinen Einstellungen des Benutzers (Standard-Anwendung von P3P und APPEL, unabhängig vom Benutzerprofilzugriff)

- Möglichkeit, Bedingungen aus der P3P-Erklärung in den Regeln für Benutzerprofilzugriff zu spezifizieren

Abb. 4.23 zeigt eine Regel für den dritten Fall (nach einem Beispiel aus [APPEL02]). Die Regel drückt folgendes aus: „lesender Zugriff auf demographische Daten ist gestattet, wenn der Dienst ein Gütesiegel einer der beiden angegebenen Organisationen in seiner P3P-Datenschutzerklärung spezifiziert hat“. Außerdem darf der Dienst die Daten nicht an fremde („unrelated“) Dienste weitergeben (beachte Verbindungsoperator „non-and“). Analog könnte z.B. formuliert werden „Zugriff wird nicht gestattet, wenn der Dienst Cookies benutzt“.

```
<appel:RULE behavior="request">
    <p3p:POLICY>
      <p3p:DISPUTES-GROUP appel:connective="or">
        <p3p:DISPUTES resolution-type="independent"
                      service="http://www.privacyprotect.org/*"/>
        <p3p:DISPUTES resolution-type="independent"
                      service="http://www.trustus.org/*"/>
      </p3p:DISPUTES-GROUP>
      <p3p:STATEMENT>
        <p3p:RECIPIENT appel:connective="non-and">
          <p3p:unrelated/>
        </p3p:RECIPIENT>
      </p3p:STATEMENT>
    </p3p:POLICY>
    <ar:ACCESS RESSOURCE="/profil/demograhic/*">
    <ar:READ/></ar:ACCESS>
</appel:RULE>
```

ABBILDUNG 4.23: Beispiel zur Verbindung mit P3P-Erklärung

Dies verdeutlicht das Zusammenspiel von AR (Zugriffsmodus <READ> im Beispiel) und P3P (Bedingungen für Gütesiegel) in APPEL-Regeln.

Zugriffszweck Der Zugriffszweck (<PURPOSE> im AR) kann, analog zum Zugriffsmodus, als Bedingung in einem Regelrumpf aufgenommen werden. Diese Angabe ist optional, obwohl der Zugriffszweck in einem AR verpflichtend ist. Ein <PURPOSE> muss nicht bei jeder Regel angegeben werden, wenn der betreffende Kontext nicht vom Zweck abhängt. Es reicht, dass die obligatorische Angabe des Zugriffszwecks bei der Spezifikation des AR gefordert wird, da damit schon alle gültigen AR's ein <PURPOSE> Element für jeden Zugriffswunsch enthalten müssen. Ansonsten wird der AR schon beim Einlesen mit einer entsprechenden Meldung abgelehnt. Wenn ein Zweck im einem Regelrumpf angegeben ist, wird dies entsprechend den anderen Elementen zur Auswertung einer Regel hinzugezogen.

Gültigkeitkeitszeitraum Eine dedizierte Angabe eines Gültigkeitkeitszeitraums ist nicht in den Regeln sinnvoll. Dies wird vom UPA – ggf. anhand Konfigurationseinstellungen des Benutzers – in das AT eingefügt. Bei einer manuellen Ausgabe von AT könnten Benutzer dies (optional) angeben.

4.3.4.5 Optionale Komponenten des Access Requests in APPEL

Im Folgenden wird dargestellt, wie die weiteren, optionalen Teile eines AR in dem hier zu erarbeiteten Schema ausgedrückt werden können.

Optionen bei Zugriffsanfrage Zunächst einmal gibt es verschiedene Optionen bei einer Zugriffsanfrage, d.h. als Attribut in einem <ACCESS>-Element des AR:

- „optional“/„mandatory“: Dies fließt in die Zugriffniveaus ein und wurde oben schon ausführlich dargestellt
- GROUP: Hier wird eine Entscheidung im Algorithmus getroffen, die Zugriffsentscheidung aller Teile einer Gruppen werden zusammen genommen; d.h. wenn für ein Element der Gruppe der Zugriff verweigert wird, werden alle Gruppenteile im zu erzeugenden Access Ticket weggelassen
- ALTERNATIVE: Es wird derjenige Teil der Alternative in das Access Ticket übernommen, für den das „am wenigsten eingeschränkte“ Zugriffsniveau aus den Regeln abgeleitet werden kann

Die Entscheidung bei „ALTERNATIVE“ ist also abhängig davon, ob z.B. „if-necessary“ oder „access“ in den Regeln ausgedrückt ist. In diesem Fall würde die Alternative mit „access“ gewählt.

Optionen bei Zugriffsrechten Als Optionen bei Zugriffsrechten sind vorgesehen (vgl. 4.2.3.3):

- „once-only“: Ein „einmaliger“ Zugriff kann formalisiert werden, indem in der Regel beim Zugriffsmodus explizit „once-only“ als Attribut spezifiziert wird (Abb. 4.24, erste Regel)
- „distributable“: Genauso kann die Option bezüglich der Weitergabe von Daten im Ausdruck einer Regel angegeben werden; die zweite Regel in Abb. 4.24 formalisiert: „Lesen der Email-Adresse ist nur dann gestattet, wenn sie nicht weitergegeben wird“
- „subscription“: Diese Option ist nicht in den Regeln sinnvoll, sondern in Verbindung mit Benutzerinteraktion

```
<appel:RULE behavior="request">
  ...
  <ar:READ OPTION="once-only"/>
</appel:RULE>
<appel:RULE behavior="request">
  <ar:ACCESS RESOURCE="/profile/email" appel:connetive="non-and">
    <ar:READ OPTION="distributable"/>
  </ar:ACCESS>
</appel:RULE>
```

ABBILDUNG 4.24: Beispiel für Optionen bei Zugriffsrechten

Bei „distributable“ kann optional ein Empfänger angegeben werden. Wenn die Zugriffsentscheidung davon abhängig ist, kann dies in den Regeln spezifiziert werden. Man könnte sich in Verbindung mit Benutzerinteraktion vorstellen, dass der Benutzer eine Weitergabe von Daten an einen bestimmten Empfänger explizit genehmigen könnte.

In einer konkreten Implementierung sollte der Regel-Auswerter bei der Evaluierung erkennen und vermerken, warum eine Regel nicht ausgeführt wird bzw. ein Zugriff nicht gestattet wird, um den Dienst informieren zu können, dass das Problem z.B. die fehlende „once-only“ Option ist. Dies gilt für alle Optionen.

Optionen bzgl. Datenübertragung Abschließend bei den Optionen können in einem AR Optionen bezüglich der Datenübertragung spezifiziert werden:

- <SIGNED>: Signierung von Benutzerprofilinhalten
- <SECURE>: Übertragung über einen sicheren Kanal
- <ENCRYPTED>: verschlüsselte Übertragung
- <ANONYMIZED>: anonymisierte Übertragung

Diese Optionen können wiederum als Bedingungen in einem Regelrumpf angeben werden. Abb. 4.25 zeigt beispielsweise die Formalisierung von: „Der Zugriff auf Kreditkartendaten wird (nur) bei sicherer Übertragung gestattet“.

```
<appel:RULE behavior="request">
  <ar:ACCESS RESOURCE="/profile/payment/creditcard/*">
    <ar:READ><ar:SECURE/></ar:READ>
  </ar:ACCESS>
</appel:RULE>
```

ABBILDUNG 4.25: Beispiel für <SECURE> in APPEL-Regel

Weitere Elemente Als zusätzliche Bedingung kann <CONDITION> in einem AR vorhanden sein. Dies ist nur für eine manuelle Ausgabe von AT’s bzw. in speziellen Anwendungsbereichen vorgesehen. Des Weiteren kann ein <STARTDATE> angeben werden. Dies ist auch für eine manuelle Ausgabe von AT’s gedacht und wird daher nicht in den Aushandlungsregeln abgebildet.

4.4 Datenzugriff (Phase II)

Nachdem jetzt eine Aushandlung von Access Tickets anhand eines Access Requests, einer P3P-Erklärung und Zugriffsregeln erarbeitet wurde (Phase I), soll nun die Zugriffskontrolle mit den AT bei tatsächlichen Datenzugriffen eines Dienstes behandelt werden, also die Verwendung der AT’s. In dieser zweiten Phase werden nur noch Access Tickets betrachtet, keine Access Requests, da sich in dieser Phase Benutzerprofilagent und Dienst schon auf konkrete Zugriffrechte geeinigt haben und es keine Optionen, Wahlmöglichkeiten o.ä. mehr gibt.

Dazu wird zunächst das Protokoll zwischen Benutzerprofil- und Dienstagenten betrachtet.

4.4.1 Protokoll

4.4.1.1 Ablauf

Folgende Protokoll-Elemente zwischen UPA und CA sind notwendig, um die Phase II abwickeln zu können (Abb. 4.26).

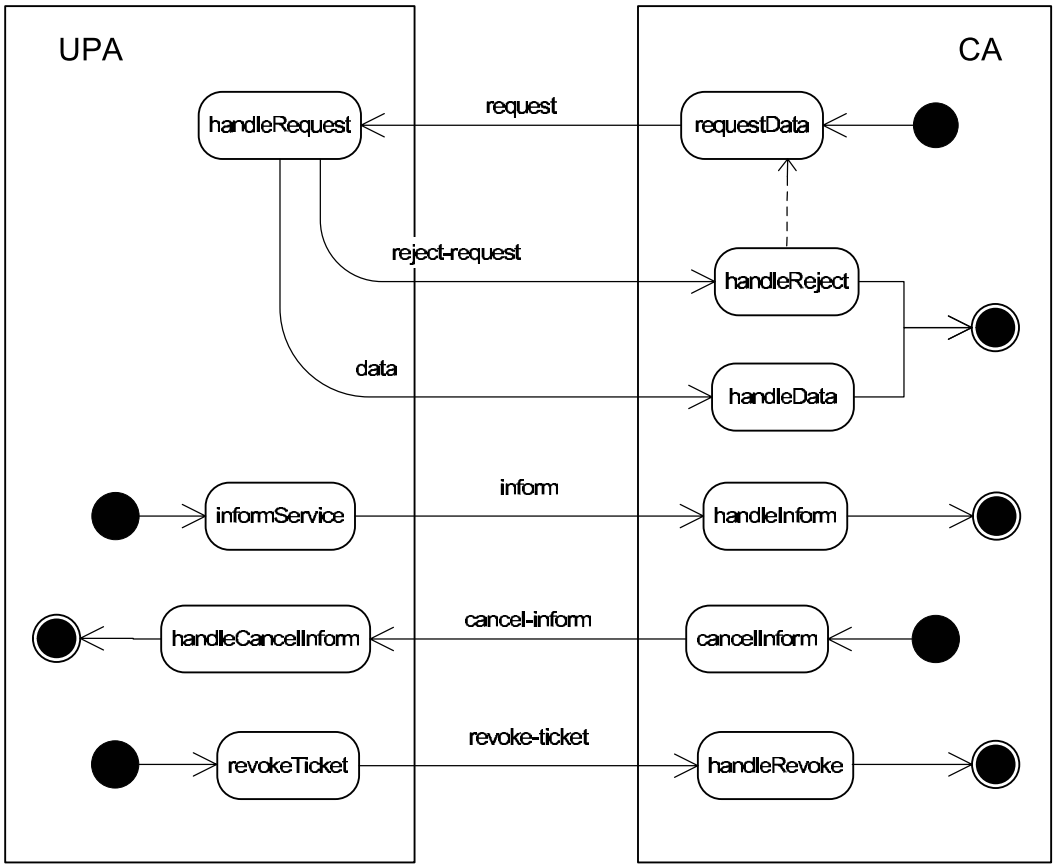

ABBILDUNG 4.26: Ablauf Phase II

4.4.1.2 Nachrichten vom CA zum UPA

```
request (ServiceID serviceid, String resource, AccessMode mode,
          AccessTicket at, String value)
cancel-inform (ServiceID serviceid, String resource, AccessTicket at)
```

Beschreibung der möglichen Nachrichten vom CA zum UPA:

- „request“: Anfrage zum (tatsächlichen) Datenzugriff, Parameter:
  - serviceid: Identifikator des Dienstes, ggf. mit digitalem Zertifikat belegt
  - resource: Pfad zu einer konkreten Ressource, z.B. „/profile/demographic/email“
  - mode: Zugriffsmodus, z.B. „READ“
  - at: Das AT mit den betreffenden Rechten für den angeforderten Zugriff
  - (optional:) value: neuer Wert der Ressource, nur bei Anfragen, die Daten modifizieren
- „cancel-inform“: Dienst will eine Benachrichtigung bei geänderten Daten abbestellen, eine Option „subscription“ muss dazu im AT vorhanden sein, Parameter:
  - serviceid

  - resource: Pfad zu einer Ressource mit „subscription“-Option
  - at

Der Identifikator des Dienstes in der Nachricht „request“ („serviceid“) muss nicht mit dem Identifikator im AT (<SERVICE>) übereinstimmen. Dies ist dann sinnvoll, wenn im AT eine Weitergabe von Daten spezifiziert ist (vgl. dazu die Option „distributable“ bei den Zugriffsrechten, Abschnitt 4.2.3.3). Dann kann das AT weitergegeben werden und auch von einem Dienst verwendet werden, der nicht mit der Angabe bei <SERVICE> identisch ist.

Das Access Ticket muss nicht bei jeder Anfrage (in XML-Form) mitgeschickt werden, es reicht eventuell eine eindeutige ID für jedes AT zu vergeben und damit einen entsprechenden Nachweis der Zugriffsrechte zu machen. Dies ist hier für das Konzept unerheblich, muss aber bei der Implementierung einer verteilten Datenhaltung ggf. beachtet werden.

4.4.1.3 Nachrichten vom UPA zum CA

```
data (UserID userid, String Resource, String data)
inform (UserID userid, String Resource, String data)
reject-request (UserID userid, String rejectmsg)
revoke-ticket (UserID userid, AccessTicket at, String revokemsg)
```

Beschreibung der möglichen Nachrichten vom UPA zum CA:

- „data“: Benutzerprofilagent schickt die angeforderten Daten an den Dienst, Parameter:
  - userid: Identifikator des Benutzerprofils
  - resource: Pfad zur betreffenden Ressource
  - data: Benutzerprofilinformation, z.B. in XML
- „inform“: Benutzerprofilagent schickt geänderte Profildaten einer Ressource mit „subscription“-Option (ohne explizite Anfrage des Dienstagenten), Parameter:
  - userid
  - resource
  - data
- „reject-request“: Zurückweisung der Dienst-Anfrage, z.B. wenn das AT nicht mehr gültig ist oder die angeforderte Ressource im AT nicht abdeckt ist
  - userid
  - (optional:) rejectmsg: Begründung der Ablehnung durch UPA, z.B. „Access Ticket no longer valid“
- „revoke-ticket“: Mitteilung des Benutzerprofilagent an den Dienst, dass das AT zurückgezogen und damit ungültig wird, Parameter:
  - userid
  - at: Das zurückgezogene Access Ticket
  - (optional:) revokemsg: Begründung der Annullierung, z.B. als Klartext-Nachricht des Benutzers

Der Pfad zur Ressource bei der Antwort ist evtl. überflüssig, wenn vom Nachrichtensystem eine Zuordnung von Anfrage zu Antwort gemacht wird. Ansonsten kann ein Dienst auch bei mehreren simultanen Anfragen eine Zuordnung anhand der Antwort ableiten.

Beim Zurückziehen von AT's wird das gesamte Ticket mit den enthaltenen Zugriffsrechten ungültig. Ein Zurückziehen von einzelnen Rechten ist nicht vorgesehen, dazu muss ein neues AT mit modifizierten Rechten vereinbart werden.

Der UPA verwaltet zur Realisierung der Zurückziehbarkeit eine Liste der aktuell gültigen Access Tickets und eine Liste von zurückgezogenen AT's. Letzteres wäre nicht unbedingt nötig, damit der UPA bei einer (fehlerhaften oder irrtümlichen) Anfrage mit einem zurückgezogenen AT dem Dienstagenten eine entsprechende Fehlermeldung schicken kann (z.B. „AT vom Benutzer zurückgezogen", im Unterschied zu „AT unbekannt/ungültig"). Es ist trotz der (notwendigen) Verwaltung einer Liste von aktuell gültigen Access Tickets beim UPA sinnvoll, dass der Dienst bei jedem Zugriff das AT mitschickt, zum Nachweis des einzelnen Zugriffsrechtes. Wie schon erwähnt, muss nicht unbedingt bei jedem Zugriff das komplette AT in XML-Form mitgeschickt werden, es reicht ggf. eine Referenz auf das betreffende AT. Dies ist abhängig von einer konkreten Implementierung.

4.4.1.4 Beschreibung der Zustände

In diesem Teilabschnitt werden die Zustände der Agenten aus Abb. 4.26 erläutert.

Mögliche Stati bei einem Dienstagenten:

- „requestData"
  - CA will auf Daten aus dem Benutzerprofil zugreifen
  - CA sendet dazu konkrete Zugriffsanforderung und entsprechendes AT an UPA
- „handleReject"
  - CA bekommt eine ablehnende Antwort von UPA
  - Möglicherweise kann CA die Anfrage modifizieren und eine erneute Anfrage stellen ($\rightarrow$ „requestData")
- „handleData"
  - CA erhält die anforderten Daten und kann diese weiterverarbeiten
- „handleInform"
  - CA erhält Daten aufgrund einer „subscription" Option
- „cancelInform"
  - Dienst schickt Nachricht an UPA, dass eine Informierung bei geänderten Daten nicht mehr erwünscht und notwendig ist
  - Bestehendes AT bleibt gültig
- „handleRevoke"
  - CA bekommt Nachricht, dass angegebenes AT nicht mehr gültig ist
  - Dienst entfernt AT ggf. aus einer Liste der aktuell gültigen Tickets

  - evtl. ist eine erneute Aushandlung von Rechten erforderlich (siehe Phase I)

Zustände beim Benutzerprofilagenten:

- „handleRequest“
  - UPA erhält eine Zugriffsanforderung eines Dienstes
  - UPA wertet Anforderung aus und sendet die Daten („data“) oder eine Fehlernachricht
  - dies wird später noch detaillierter behandelt, vgl. 4.4.2
- „informService“
  - Ein Teil des Benutzerprofils, für den eine „subscription“-Option besteht, wurde verändert
  - Benachrichtigung von Diensten muss erfolgen
- „handleCancelInform“
  - UPA nimmt Änderung in System auf, z.B. Modifikation einer Liste der zu benachrichtigenden Dienste
- „revokeTicket“
  - Benutzer will ein ausgegebenes AT zurückziehen

4.4.2 Durchführung der Zugriffskontrolle

4.4.2.1 Algorithmus

Der Zugriff in Phase II erfolgt immer nur für eine Ressource für einen bestimmen Zugriffsmodus, vgl. die Methode/Nachricht „request“ in Abschnitt 4.4.1.2. Dazu muss der Dienst ein für den betreffenden Zugriff gültiges AT vorlegen. Dies kann auch ein AT sein, das für einen Dienst ausgestellt wurde, wenn für das betreffende Attribut die Option „distributable“ vorhanden ist und ggf. weitere Bedingungen zutreffen.

Zur Durchführung der Zugriffskontrolle muss der Benutzerprofilagent die folgende Schritte durchführen:

1. Prüfe Gültigkeit des AT’s:
   - Prüfe Authentizität (Signatur) des AT’s: Bei dieser Prüfung muss auch berücksichtigt werden, dass in einem verteilten Benutzerprofilverwaltungssystem das AT evtl. von einem anderen UPA ausgestellt wurde
   - Prüfe, ob AT’s noch nicht zurückgezogen wurde: Ggf. muss der Benutzerprofilagent dazu eine Anfrage bei dem Aussteller des AT machen
2. Prüfe <USER>: Abbruch im Fehlerfall, wenn z.B. die angegebene Identität nicht vorhanden ist
3. Prüfe <SERVICE>: Stimmt das Element mit dem anfragenden Dienst überein?
   - Ja $\rightarrow$ OK
   - Nein $\rightarrow$ Prüfe ob Option „distributable“ vorhanden, Abbruch, falls nicht

4. Prüfe <STARTDATE> und <VALIDITY>: AT noch/schon gültig? (Abbruch im Fehlerfall)
5. Prüfe <POLICY>: Abbruch, falls angegebene P3P-Datei nicht (mehr) vorhanden ist
6. Für jedes <ACCESS>-Element:
   - Prüfe Ressource: Gewünschter Profil-Teil im AT abgedeckt? (Abbruch im Fehlerfall)
   - Prüfe Zugriffsmodus: Gewünschtes Recht abgedeckt? (Abbruch im Fehlerfall)
   - Prüfe Zugriffszweck (<PURPOSE>): Abbruch, falls kein Purpose vorhanden; eine genaue Prüfung des Zugriffszweck erfolgt hier nicht mehr, da dies Bestandteil der Phase I ist
   - Behandlung der Optionen: siehe nachfolgender Abschnitt 4.4.3
7. Zugriff ist OK, Daten können übertragen werden

4.4.2.2 Verteilte Datenhaltung

Eine Trennung von Rechteaushandlung und Datenzugriff, sowie die Formalisierung der Zugriffsrechte in einem digital signierten Access Ticket unterstützt eine verteilte, föderierte Benutzerprofilverwaltung durch eine Vielzahl unabhängiger Benutzerprofilagenten. Zwei Aspekte dabei:

- Verteile Profilhaltung: Gerade durch die Access Tickets wird eine dezentrale Rechteverwaltung ermöglicht, da ein Benutzerprofilagent durch Prüfen der digitalen Signatur eines AT auch die Authentizität eines AT, welches von einem anderen UPA ausgestellt wurde, prüfen kann. Dazu ist eine gemeinsame Public Key Infrastruktur (PKI) bzw. die Integration in ein föderiertes Identitätsmanagementsystem wie Liberty Alliance nötig. Außerdem muss dabei die Möglichkeit der Zurückziehbarkeit von Access Tickets berücksichtigt werden.
- Verteiltes Regelsystem: Ein weitere Frage ist, wie die Zugriffsregeln des Benutzers in einem verteilten Identitätsmanagement verwaltet werden. Hier wird dazu die Annahme gemacht, dass nur ein UPA die Original-Regeln des Benutzers speichert und der Benutzer nur bei diesem Benutzerprofildienst seine Daten pflegt. Andere UPA's müssen bei Bedarf die Benutzerregeln abgleichen. Dazu sind evtl. Methoden aus verteilten Datenbanken anwendbar, um dieses verteilte Regelsystem zu realisieren. Dies ist aber nicht Gegenstand einer genaueren Untersuchung in dieser Arbeit.

4.4.2.3 Caching des Access Tickets

Ein Grund für die Aufteilung in zwei Phasen war es, einen möglichst effizienten Datenzugriff in Phase II zu erlauben. XML erscheint unter diesem Gesichtspunkt als Datenformat für das Access Ticket zunächst nicht so geeignet, da zunächst ein Parsen des AT erfolgen muss, bevor Rechte o.ä. abgefragt werden können. Allerdings muss das Access Ticket nur beim ersten Zugriff eingelesen werden, es kann also ein Caching des eingelesenen AT's im Benutzerprofilagenten erfolgen:

1. Parsen und Auswerten des AT und Aufbau eines AT-Objektes im Speicher
2. Zugriff auf Datenstruktur im Speicher, z.B. zur Prüfung der Rechte für den tatsächlichen Zugriff

Die Machbarkeit einer XML-basierten Zugriffskontrolle wird durch eine prototypische Teil-Implementierung gezeigt (vgl. Abschnitt 5.3.4).

4.4.3 Behandlung der Optionen

Optionen in der Zugriffsanfrage (z.B. „mandatory“) sind beim Datenzugriff nicht mehr vorhanden, da diese nur im AR vorkommen können. Folgende Optionen müssen in Phase II beachtet werden:

- Optionen bei Zugriffsrechten
- Optionen bezüglich Datenübertragung

4.4.3.1 Optionen bei Zugriffsrechten

- Option „distributable“ (Weitergabe von Daten): Dies ist dann der Fall, wenn das AT von einem Dienst weitergegeben wurde. Es muss dann geprüft werden, ob der Zugriffswunsch gestattet werden kann, obwohl der zugreifende Dienst nicht mit dem Dienst, für den das AT ausgestellt wurde, übereinstimmt
- Option „subscribe“: Behandlung wie ein normaler Lese-Zugriff; zusätzlich trägt der Benutzerprofilagent den Wunsch der Benachrichtigung in eine Liste o.ä. ein
- Option „once-only“ (einmaliger Zugriff): Prüfe, ob mit diesem AT schon mal ein Zugriff auf das Element erfolgt ist, wenn ja, wird der Zugriffswunsch abgelehnt. Das AT wird damit nicht (komplett) ungültig, aber das Zugriffsrecht im betreffende <ACCESS> Element kann nicht mehr verwendet werden

Wenn bei verteilter Benutzerprofilverwaltung ein einzelner Benutzerprofilagent nicht prüfen kann, ob mit dem betreffenden Access Tickets bereits ein Zugriff erfolgt ist, ist die beschriebene Behandlung von „once-only“ ist nicht hinreichend. Eine Implementierung kann in diesem Fall nur einen Teil der Konzeptualisierung abdecken, was im Abschnitt „Vertrauensmanagement“ (Kap. 4.5.5), noch genauer behandelt wird.

4.4.3.2 Optionen bezüglich Datenübertragung

Grundsätzlich werden bei den folgenden Optionen geeignete Sicherungs- bzw. Anonymisierungs-Komponenten vom Benutzerprofilagenten für die Datenübertragung zugeschaltet. Im Unterschied zu bestehenden Systemen soll dies abhängig von den im AT spezifizierten Bedingungen für den Datenzugriff automatisch ohne manuelles Eingreifen des Benutzers geschehen.

Dies betrifft die Kommunikation des Benutzerprofilagenten mit den Dienstagenten, nicht die Kommunikation des Benutzer mit seinem Benutzerprofilagenten, da angenommen wird, dass der UPA im „vertrauenswürdigen Bereich“ des Benutzers liegt. Die Kommunikation des Benutzers – bzw. dem Rechner des Benutzers – mit seinem Benutzerprofilagenten z.B. zur Aktualisierung von Profilinformationen könnte über eine gesicherte Verbindung wie SSL erfolgen, um eine Vertraulichkeit gegenüber unbefugten Dritten zu erreichen.

Es gibt folgende Optionen bezüglich der Datenübertragung (vgl. Abschnitt 4.2.3.4): <SIGNED>, <ENCRYPTED>, <SECURE> und <ANONYMIZED>.

Option <SIGNED> Bei dieser Option wird der betreffende Teil des Profils mit einer digitalen Signatur unterschrieben. Der Benutzerprofilagent verwaltet dazu ein Zertifikat des Benutzers. Dies entspricht praktisch der Identitätsstufe „veronymous“, nur das die Option <SIGNED> für einen einzelnen Datenzugriff festgelegt werden kann und die Identitätsstufe auch mehrere Zugriffe umfassen kann.

Eine digitale Signatur von Benutzerprofilteilen kann mit *XML Signature* [XSig02] erstellt werden, weil die Profile in XML-Form dargestellt werden. Dabei wird aus den betreffenden Daten ein Hashwert erstellt, der mit dem privaten Schlüssel des Benutzers verschlüsselt wird. Der Dienst kann durch Anwenden des öffentlichen Schlüssels des Benutzers die Echtheit der Signatur prüfen.

Eine Signatur von Benutzerprofilinhalten ist wichtig für die Integrität und Zurechenbarkeit von Profildaten, also dem Nachweis für den Dienst, dass eine Information wirklich aus dem Profil eines bestimmten Benutzers stammt. Ein Beispiel dafür wäre die Signierung von Zahlungsinformationen wie Bankverbindung zur Abwicklung eines Bankeinzuges. Dies wird im Abschnitt „Vertrauensmanagement“ (4.5.5) nochmals aufgegriffen.

Option <ENCRYPTED> Das Ziel dieser Option ist es, zu verhindern, dass unbefugte Dritte den Inhalt von Nachrichten lesen können (Vertraulichkeit), indem diese z.B. mit symmetrischer Kryptographie verschlüsselt wird. Mit Hilfe einer Kombination von <SIGNED> und <ENCRYPTED> kann der Nachweis der Integrität der Nachricht erreicht werden [Salz03].

Die Durchführung der Verschüsselung kann mit Hilfe von *XML Encryption* [XEnc02] erfolgen. Dazu ist der öffentliche Schlüssel des Dienstes zur Verschlüsselung nötig, damit nur dieser den Inhalt der Nachricht unter Verwendung seine privaten Schlüssels dechiffrieren kann.

Option <SECURE> Diese Option gewährleistet die Vertraulichkeit der Kommunikation zwischen UPA und CA. Es wird dazu ein sicherer Kanal aufgebaut, der die Datenübertragung vor unerlaubtem Abhören schützt. Dies kann z.B. mit dem verbreiteten Secure Socket Layer (SSL) Protokoll realisiert werden.

Der Unterschied von <SECURE> zu <ENCRYPTED> ist, dass bei SSL (oder ähnlichen Verfahren) eine Verschlüsselung zwischen zwei Kommunikationsendpunkten (also auf Kommunikationsebene) erfolgt, während bei <ENCRYPTED> einzelne Teile des Profils verschlüsselt werden, also die Verschlüsselung auf Anwendungsebene geschieht.

Option <ANONYMIZED> Mit Hilfe der Option <ANONYMIZED> kann spezifiziert werden, dass eine Datenübertragung vom Benutzerprofil- zum Dienstagenten in anonymisierter Form geschehen soll. Dies soll gewährleisten, dass Verdeckheit und Unbeobachtbarkeit in offenen Netzen gegenüber unbefugten Dritten sichergestellt werden kann. Im Gegensatz dazu bezieht sich die Identitätsstufe sich auf den (inhaltlichen) Grad an Anonymisierung gegenüber dem Dienst, während die Anonymisierungs-Option den Datenverkehr betrifft.

Eine Realisierung kann durch die Integration eines Anonymisierungsproxies – bzw. dessen Client – in den UPA erfolgen. Ein dafür geeignetes System ist beispielsweise der Java Anon Proxy der TU Dresden (siehe anon.inf.tu-dresden.de/).

4.5 Identitätsmanagement

In diesem Abschnitt werden jetzt die Aspekte verschiedener Identitäten eines Benutzers diskutiert, die in der Aushandlung der Zugriffsrechte noch ausgespart wurden (aus Gründen der Übersichtlichkeit der Darstellung), aber dennoch sehr wichtig für ein Identitätsmanagement im Internet sind. Insbesondere betrifft dies die Berücksichtigung der Identitäten in den Zugriffsregeln, also die Behandlung des Elementes <USER>. Zunächst einige wichtige Vorbemerkungen zu Anonymität und Pseudonymität.

4.5.1 Anonymität und Pseudonymität

Anonymität bei dezentraler Verwaltung von Benutzerprofilen betrifft zwei Aspekte:

- Anonymität gegenüber dem Kommunikationspartner: Ein Dienst soll nicht unbedingt die „wahre Identität“ eines Benutzer erkennen können. Oftmals ist Anonymität oder Pseudonymität völlig ausreichend zur Erbringung auch eines Personalisierungsdienstes. Anonymisierung und Pseudonymverwaltung sind wesentliche Merkmale eines Identitätsmanagement in Hinblick auf Privatheit. In dem hier betrachteten Framework wird dies abgebildet, indem das <USER> Element eines Access Requests oder Tickets nicht unbedingt die offenbarte Identität eines Benutzers enthält, sondern z.B. ein Pseudonym.

- Anonymität gegenüber (ungefugten) Dritten: Dies betrifft die Kommunikation in einem offenen Netz, wie es im Internet der Fall ist. Der Datenverkehr in einem offenen Netz kann möglicherweise abgehört werden, so unbefugte Dritte persönliche Informationen über einen Benutzer gewinnen können. Dies kann durch den Einsatz von Sicherungs- und Anonymisierungskomponenten verhindert werden.

Der zweite Fall, die Anonymität gegenüber unbefugten Dritten bei der Kommunikation zwischen Benutzer- und Dienstkomponenten wird in dieser Arbeit durch die explizite Integration von Sicherungs- und Anonymisierungskomponenten in der Formalisierung der Zugriffskontrolle berücksichtigt (vgl. Kap. 4.2.3.4). Die folgenden Ausführungen zu Identitätsmanagement betreffen daher die Anonymisierung und Pseudonymisierung des Benutzerprofils gegenüber den Dienstagenten.

Man muss sich auch im Klaren darüber sein, dass ein anonymer Benutzerprofilzugriff – d.h. das <USER> Element enthält keine Identität – nicht automatisch bedeutet, dass keine personenbezogenen Daten an einen Dienst übermittelt werden. Benutzerprofilinhalte können auf die Identität eines Benutzers hinweisen, man kann bei Profildaten unterscheiden (nach [Gar02]):

- „Persönliche Daten“: Informationen über eine Person, z.B. dessen Name oder Personalausweisnummer

- „Private Daten“: Persönliche Daten oder Eigenschaften eines Individuums, die nicht allgemein bekannt sind, z.B. Informationen über das Bankkonto einer Person, aber nicht dessen Name im öffentlichen Telefonbuch

- „Personenbezogene Daten“ („personally identifiable information“): Informationen, aus denen man evtl. die Identität der Person ableiten kann, z.B. das Geburtsdatum

- „Anonymisierte Daten“: Informationen, die nicht mehr direkt auf die Person selber schließen lassen (das Gegenteil personenbezogener Daten)

- „Aggregierte Daten“: Statistische Informationen, die aus Benutzerprofil- oder ähnlichen Daten vieler einzelner Personen zusammengefasst wurden

Ein „anonymer“ Zugriff bei den ersten drei Kategorien ist sinnlos, da die Daten Aufschluss über die Identität des Benutzers geben (können). Auch ist die letzte Klasse der aggregierten Daten bei einem Zugriff auf ein einzelnes Benutzerprofil, wie es hier betrachtet wird, nicht relevant. Das folgende bezieht sich daher in erste Linie auf die Kategorie der anonymisierten Daten. Dies können z.B. Interessen, Web-Zugriffslogs oder ähnliches sein. Allerdings lässt oftmals eine geeignete Menge dieser Daten auch wieder Rückschlüsse auf die Person zu, insbesondere bei einer Vermischung von mehreren, getrennt erhobenen, Datenquellen, die für sich alleine betrachtet anonymisierte Daten enthalten.

Mit anderen Worten, aus anonymisierten Daten können leicht personenbezogene Daten werden, die Übergänge dieser Kategorien sind also fließend. Der wichtigste Aspekt von „Anonymität“ ist, dass keine Verbindung zwischen einzelnen Aktionen oder Benutzerprofilattributes des Benutzers hergestellt werden kann.

4.5.2 Identitätsstufen und Identitäten

Eine Identität eines Benutzers ist – etwas vereinfacht – ein bestimmter Teil des Benutzerprofils, z.B. „private“ oder „work“ mit unterschiedlicher Ausprägung der Email-Addresse und anderer Profilattribute. Bezüglich verschiedener Identitäten eines Benutzers ist festzustellen:

- Es gibt unterschiedliche Stufen von Identität oder Anonymität; „pseudonymous“ bedeutet z.B. eine Zuordnung von Transaktionen zu einem Pseudonym, aber keine Aufdeckung der Identität einer Person (vgl. Abschnitt 2.1.1.2)
- Auf den einzelnen Identitätsstufen kann ein Benutzer mehrere Identitäten annehmen, insbesondere natürlich auf die Stufe „pseudonymous“, aber im Prinzip auf allen Stufen

In dieser Arbeit werden folgende Identitätsstufen betrachtet (vgl. Kap. 4.2.3.1), eine weitere Unterscheidung verschiedener Arten von Pseudonymität (vgl. Kap. 2.1.1.2) ist denkbar, aber für das Konzept hier nicht relevant:

- „veronymous“
- „pseudonymous“
- „anonymous“

Verschiedene Identitäten auf der Stufe „veronymous“ können zum Beispiel eine private und berufliche Rolle eines Individuums sein, die ggf. durch verschiedene digitale Zertifikate belegt werden. Bei Anonymisierung wird keine Identifizierung gemacht bzw. es wird automatisch ein neues Pseudonym eines Anonymisierungsdienstes oder ähnliches bei jeder Transaktion des Benutzers angelegt, d.h. auf dieser Stufe ist eine Differenzierung von Identitäten in den Zugriffsregeln nicht sinnvoll oder nötig. Einzelne „pseudonymous“ Identitäten können z.B. Kennung eines Pseudonymisierungsdienstes sein oder den Identitäten von Liberty Alliance entsprechen.

4.5.3 Identitäten und Zugriffsregeln

In folgenden wird erarbeitet, wie Identitätsstufen und Identitäten in die Zugriffsregeln integriert werden können, damit dies bei einer Zugriffsentscheidung berücksichtigt werden kann. Dies war eine wichtige Anforderung in Abschnitt 2.3.

4.5.3.1 Alternativen

Es gibt folgende Alternativen, Identitätsstufen und Identitäten in das erarbeitete Konzept für die Zugriffsregeln zu integrieren:

- Verschiedene Regelmenge für Identitätsstufen
- Verwendung des Persona-Konzeptes von APPEL
- Direkte Abbildung in den Regeln

Verschiedene Regelmengen Ein erster Ansatz ist es, für die unterschiedlichen Identitätsstufen verschiedene, andersartige Regelmengen zu verwalten. Der Vorteil davon ist, dass der Algorithmus zur Auswertung ziemlich einfach wird: Wähle den betreffenden Regelsatz aus und prüfe die Anfrage gegen diese Regelmenge. Die auszuwählende Regelmenge ergibt sich dabei aus der Identitätsstufe der Anfrage. Jede Anfrage bezieht sich nur auf eine Identitätsstufe. Es wäre unsinnig, mehrere Stufen in einer Anfrage vorzusehen, weil dann der Dienst sehr leicht eine Zuordnung verschiedener Identitäten und Identitätsstufen zu einem Benutzer machen könnte. Dies soll gerade durch die Anonymisierung und Pseudonymisierung vermieden werden.

Ein Nachteil dieser Vorgehensweise ist es, dass Redundanz in den Regelmengen vorhanden wäre, wenn Regeln für alle Identitätsstufen gelten sollen und damit in allen Regelmengen enthalten wären. Dies könnte zu einer schlechten Transparenz für den Benutzer führen. Eine geeignete Erweiterung dieses Schemas ist es, die Regeln in einen identitätsunabhängigen Satz und Teilregelmengen für die Regeln, die von der Identitätsstufe abhängen, aufzuteilen. Damit kann unerwünschte Redundanz in den Regelmengen vermieden werden und trotzdem die für die Auswertung der Regeln gut geeignete Systematik beibehalten werden.

Durch die Verwaltung mehrere Regelmengen können also verschiedene Identitätsstufen einfach abgebildet werden. Ein zusätzlicher Mechanismus müsste verwendet werden, wenn Zugriffsentscheidungen von einzelnen Identitäten auf einer Stufe abhängig gemacht werden sollen.

Persona-Konzept in APPEL Wie schon in Abschnitt 4.3.4 erwähnt, gibt es in APPEL das Konzept einer „persona“ zur Kennzeichnung einer bestimmten Identität des Benutzers in einer APPEL-Regel. Abbildung 4.27 zeigt dazu ein Beispiel.

```
<appel:RULE behavior="request" description="Access granted for
            pseudonym nickname123" persona="nickname123">
      <-- Expressions -->
</appel:RULE>
```

ABBILDUNG 4.27: Beispiel für eine Identität in APPEL

Die Ausprägung des Persona-Attributes ist dabei ein eindeutiger Identifikator vom Typ String, der einen Teil des Benutzerprofils kennzeichnet [APPEL02]. Dies betrifft auch mehrere Ausprägungen desselben Profilattributes, z.B. unterschiedliche Werte für die Email-Adresse des Benutzers. Damit können unterschiedliche Zugriffsregeln des Benutzers für einzelne Identitäten spezifiziert werden.

Allerdings ist in APPEL keine Abbildung von Identitätsstufen vorgesehen. Der Persona-Mechanismus müsste dazu erweitert werden, da für „anonymous“ keine explizite Identität angegeben wird.

Direkte Abbildung in einer Regel Eine weitere Alternative zur Abbildung von Identitäten wäre die direkte Abbildung des entsprechenden <USER> Elementes eines AR in einer APPEL-Regel. Dies geschieht analog zu den bereits behandelten Anfragekomponenten als Bedingung einer Regel (vgl. Abb. 4.28).

Man beachte, dass aufgrund der Struktur eines AR die Angabe von <USER> außerhalb des <ACCESS> Elementes erfolgen muss. Daher ist diese Darstellung auch nicht sehr intuitiv, da <USER> für alle Teile einer Anfrage gilt, während die Bedingungen einer APPEL-Regel nur für einen speziellen <ACCESS> Abschnitt des AR spezifiziert werden. Es können auch mehrere <USER> (oder z.B. <ACCESS>) Elemente in einer Regel angegeben werden, ggf. mit Verbindungsoperatoren verknüpft,

```
<appel:RULE behavior="request">
  <ar:USER LEVEL="pseudonymous">nickname123</ar:USER>
  <ar:ACCESS RESOURCE="/profile/interests/private/*">
    <ar:READ/>
  </ar:ACCESS>
</appel:RULE>
```

ABBILDUNG 4.28: Beispiel für Identität als Bedingung einer Regel

um z.B. eine Bedingung „diese Regel soll für die Pseudonyme ”nickname123” und ”mynickname”, sowie für die Identitätsstufe ”anonymous” gelten“ spezifizieren zu können. Grundsätzlich müssen in einer APPEL-Regel alle Elemente des Regelrumpfes erfüllt sein (unter Beachtung der Verbindungsoperatoren), damit die Regel ausgeführt wird.

Ein Vorteil der direkten Abbildung von <USER> wäre, dass über die Verbindungsoperatoren von APPEL eine „oder“-Verbindung von Identitäten erreicht werden könnten. Beispielsweise könnte damit eine Bedingung „Zugriff für Identität ”nickname” oder ”abc123” gestattet“ verwirklicht werden. Im Gegensatz zur Persona-Alternative in APPEL gäbe es hier auch die Möglichkeit einer Abbildung von Identitätsstufen ohne eine konkrete Identität anzugeben. Dazu könnte in Abb. 4.28 der Inhalt des <USER> Elementes leer gelassen werden und nur das Auszeichnungselement <ar:USER LEVEL=”...”/> mit dem Attribut für die Identitätsstufe angegeben werden.

Der Nachteil dieser Vorgehensweise ist es, dass die Regeln unnötig kompliziert werden. Wenn z.B. relativ viele Regeln nur für eine Identitätsstufe gelten sollen, muss dies als Bedingung in allen Regeln angegeben werden. Die Alternative von getrennten Regelmengen für verschiedene Identitätsstufen ist hier nahe liegender und besser.

4.5.3.2 Kombination der Alternativen

Die diskutierten Alternativen können gut kombiniert werden, um eine Abbildung von Identitätsstufen und Identitäten zu erreichen:

- Für die Identitätsstufe wird die Regelmenge eines Benutzers aufgeteilt in:
  - einen allgemeinen Teil, für alle Identitätsstufen
  - zusätzliche Abschnitte, die nur für jeweils eine der Identitätsstufen „veronymous“, „pseudonymous“ und „anonymous“ gelten
- Für verschiedene Identitäten auf einer Stufe wird das Persona-Konzept von APPEL verwendet
- Zusätzlich kann optional die dritte Alternative der direkten Abbildung in der Bedingung einer Regel verwendet werden, um eine logische Verknüpfung von Identitäten als Bedingung einer Regel formalisieren zu können

Damit kann z.B. recht einfach „Zugriff auf Interessen gestattet, wenn dies anonym geschieht“ formalisiert werden, indem eine entsprechende Regeln in die Regelmenge „anonymous“ aufgenommen wird. Weitere Bedingungen, z.B. „Zugriff mit Offenbarung der Identität beim Dienst amazon.de gestattet“, können damit auch sehr einfach spezifiziert werden.

Eine ausdrückliche Default-Identität ist in diesem Ansatz nicht vorgesehen, bei einer Anfrage ohne Angabe einer Identität, wird die Identitätsstufe „anonymous“ verwendet.

4.5.4 Auswerten der Zugriffsregeln

Es soll nun dargestellt werden, wie der Ablauf (im Überblick) im Benutzerprofilagenten bei einer Anfrage eines Dienstagenten erfolgen soll. Die folgenden Schritte sind zur Evaluation eines AR anhand der Zugriffsregeln des Benutzers erforderlich:

1. Prüfe Gültigkeits- und ggf. Startdatum des AR
2. Auswahl der Regelmenge anhand der Identitätsstufe aus dem <USER> Element der Anfrage; die im Folgenden zu benutzende Regelmenge besteht aus einem allgemeinen Teil, plus dem speziellen Teil für die betreffende Identitätsstufe
3. Abruf der P3P-Datenschutzerklärung und Speicherung zum Abgleich in den Regeln, falls nötig (Abbruch, falls keine P3P-Erklärung vorhanden ist)
4. Vorbereitung der Optionen:
   - Attribut „GROUP“: Vermerke, dass diese Elemente gemeinsam geprüft werden müssen
   - Attribut „ALTERNATIVE“: Vermerke, dass diese Elemente Alternativen sind
5. Für jedes <ACCESS> Element im AR (unter Berücksichtigung der Vermerke für die Anfrage-Optionen und der P3P-Erklärung):
   - Für jede APPEL-Regel, prüfe Regelmenge mit APPEL-Validator:
     - Werte Bedingung (Regelrumpf) aus
     - Falls Bedingung erfüllt ist, werte den Regelkopf („behavior“, „prompt“, sowie „persona“) entsprechend der in Abschnitt 4.3.4.3 beschriebenen Semantik aus ($\rightarrow$ „access“, „no-access“ oder „user-interaktion“[7])
   - Falls keine Regel für das <ACCESS> Element erfüllt ist: $\rightarrow$ „no-access“

Bei einer notwendigen Benutzerinteraktion muss zusätzlich die Entscheidung des Benutzers abgewartet und berücksichtigt werden. Für alle <ACCESS> Elemente des AR, für die ein Zugriff erlaubt werden kann, wird ein entsprechender Abschnitt des Access Tickets generiert. Das AT wird schließlich vom Benutzerprofilagenten digital signiert und zum Dienstagenten geschickt.

4.5.5 Vertrauensmanagement

Ein Teil des hier erarbeitete Mechanismus basiert auf Datenschutzerklärungen, Vereinbarungen bezüglich der Verwendung von personenbezogenen Daten etc.. Ein wichtiger Punkt dabei ist, wie die Einhaltung dieser Aspekte überprüft bzw. durch technische oder andere Mittel unterstützt werden kann. Dies soll in diesem Abschnitt unter dem Titel „Vertrauensmanagement“ diskutiert werden.

Der hier erarbeitete Ansatz kann prinzipiell zum Aufbau von Vertrauen[8] zwischen dem Benutzer und Diensten, die personenbezogene Daten verarbeiten beitragen, da es u.a. eine klare Spezifikation von Zugriffsrechten gibt, die z.B. die Verwendung von Daten einschränkt. Dadurch können Missverständnisse vermieden werden. Auch kann der Benutzer durch eine (ausschließlich) manuelle Ausgabe von Access Tickets und die explizite Integration von Benutzerinteraktion sicherstellen, dass die Informationen aus seinem Benutzerprofil – oder auch nur bestimmte, besonders sensible, Daten – nur in seinem Sinne verwendet werden. Dies trägt sicherlich zur Vertrauensbildung bei.

[7] Die Auswertesemantik für den Fall, dass mehrere Regelrümpfe zutreffen, wurde in Abschnitt 4.3.4.4 diskutiert.

[8] Für eine Definition von Vertrauen in diesem Zusammenhang sei auf den Abschnitt 2.2.1 im Grundlagenkapitel verwiesen.

4.5.5.1 Fragestellungen

Die folgenden Fragestellungen sind im betrachteten Umfeld insbesondere interessant:

- Wie kann der Benutzer sicherstellen, dass die Datenschutzerklärungen und sonstigen Bedingungen für einen Benutzerprofilzugriff vom Dienst auch eingehalten werden?
- Wie kann eine (unerlaubte) Weitergabe von persönlichen Daten überprüft werden?
- Wie kann auf der anderen Seite dem Dienst garantiert werden, dass Benutzerprofilinhalte auch wirklich dem angegebenen Benutzer zugeordnet werden können? Der hier vorgestellte Ansatz betrachtet die Privatheit von persönlichen Daten aus Sicht der Benutzer, auf der anderen Seite ist es auch wichtig, Zurechenbarkeit und Integrität in diesem Zusammenhang zu berücksichtigen

Dies ist etwas außerhalb des Fokus dieser Arbeit und wird daher auch nicht erschöpfend behandelt. Allerdings ist dieser Bereich sehr wichtig für die Zukunft, deshalb soll es in dem betrachteten Umfeld angeschnitten werden.

Zunächst sind alle Vereinbarungen, z.B. die P3P-Datenschutzerkärung, „Teil eines Vertrages“ zwischen Dienst und Benutzer. Das bedeutet, ein Nichtbeachten des Dienstes kann zu rechtlichen Konsequenzen führen, z.B. Schadensersatzansprüchen. Ein Benutzer kann (zumindest theoretisch) Rechtsmittel einsetzen, um seine Datenschutzansprüche geltend zu machen. Allerdings gibt es dabei international sehr große Unterschiede, insbesondere hinsichtlich der Durchsetzbarkeit von Datenschutzansprüchen.

Eine Überprüfung der Datenschutzpraktiken eines Dienstes kann unabhängig von gesetzlichen Regelungen durch Gütesiegel unabhängiger Organisationen erfolgen, was im nächsten Teilabschnitt vertieft wird.

4.5.5.2 Gütesiegel

Gütesiegel (privacy seals) sind ein optionaler Teil einer P3P-Datenschutzerklärung und wurden in Abschnitt 3.2.5.4 bei der Besprechung von P3P schon behandelt. Sie ermöglichen eine Überprüfung der Datenschutzpraktiken eines Dienstes durch eine vertrauenswürdige Organisation. Hier soll noch genauer untersucht werden, inwieweit diese Gütesiegel tatsächlich eine Verbesserung der Privatheit erreichen können.

ABBILDUNG 4.29: Gütesiegel

Es werden die 3 wichtigsten Programme betrachtet, nämlich BBBOnLine (www.bbbonline.org), CPA WebTrust (www.webtrust.org) und TRUSTe (www.truste.org). Alle Programme beinhalten die Ausstellung eines Gütesiegels, welches – in unterschiedlicher Weise – die Einhaltung der (auf der Web-Seite) eines Unternehmens veröffentliche Datenschutzerklärung garantieren soll. Die Gütesiegelbetreiber führen dazu ein mehr oder weniger restriktives Auditverfahren durch. Dem Benutzer

wird die Zertifizierung der Web-Site durch ein Siegel (i.d.R. eine Grafik des Ausstellers des Gütesiegels, vgl. Abb. 4.29) auf der Web-Site des Dienstes angezeigt. Die wichtigsten Aspekte werden in Tabelle 4.3 zusammengefasst (nach www.perfectlyprivate.com/newsresources_seals.shtml).

	BBBOnLine	**CPA WebTrust**	**TRUSTe**
Voraussetzungen für die Erlangung des Gütesiegels	Kundenzugriff auf die personenbezogenen Daten möglich, Maßnahmen zur Datensicherheit, Benennung eines verantwortlichen Mitarbeiters, Änderungen der Privacy Policy werden BBBOnLine mitgeteilt, Teilnahme am BBBOnLine Beschwerde-Prozess (s.u.)	Maßnahmen zum Datenschutz, u.a.: Sicherstellung der Korrektheit der Daten, Option für Kunden zum „opt-out“, Mitteilung über die Verwendung von Cookies, Bereitstellung einer Möglichkeit für Kunden, auf ihre Daten zuzugreifen und diese ggf. korrigieren zu können	Wahlmöglichkeit und Zustimmung des Benutzers wie Informationen verwendet werden, Maßnahmen zur Datensicherheit, Sicherstellung der Korrektheit und Qualität und Zugriff zur Korrektur
Prozess der Überprüfung	Dienstbetreiber muss einen 10-seitigen Fragebogen ausfüllen, der von BBBOnLine überprüft wird; laufende, zufällige, Prüfungen	Durchführung eines systematischen Audits mit Bestätigungsvermerk eines CPA Auditor; Rezertifizierung mindestens alle 90 Tage erforderlich	Nicht näher spezifizierte Überprüfung der Datenschutzerklärung (kein systematischer Audit); regelmäßige Reviews
Abwickung von (Kunden-) Beschwerden	BBBOnLine betreibt ein (angesehenes) „privacy dispute resolution center“ zur Aufklärung von Datenschutz-Diskrepanzen und zur Verbesserung der Praktiken eines Dienstes	Keine Abwicklung oder Unterstützung durch CPA WebTrust; der Dienst muss aber Informationen bekannt geben (als Teil der Datenschutzerklärung), wie Beschwerden gelöst werden	Versuch der Auflösung einer Beschwerde mit einem Audit, falls nötig

TABELLE 4.3: Übersicht über Gütesiegel

Ann Cavoukian und Malcolm Crompton kommen bei einer Evaluierung der Programme zu folgendem Ergebnis:

> „At the time of our review, each of the seals had its own strengths. BBBOnLine offered the most customer-friendly dispute resolution system, while WebTrust offered the most rigorous compliance regime. In terms of privacy principles, while TRUSTe scored the highest in our assessment, it is clear that none of the seals required their participants to meet all of the OECD principles. This is a point of concern. Nonetheless, seals are playing a valuable educational role in promoting privacy awareness in the minds of both consumers and businesses alike. This educational role is, in our view, both positive and beneficial.“ (aus: [CaCr00])

Brian Markert fasst unter dem Titel „Privacy seal program caveats“ zusammen:

> „Patrick F. Sullivan, Ph.D provided the following information during the Online Privacy Conference held in Chicago during July 2001 regarding privacy seal programs: ”Seal programs will provide applicants with review criteria based on the general principles and disclosure requirements of the program. These facilitate documenting practices that support disclosures but involve no substantive testing of controls by the seal program, and do not result in an opinion on the compliance of the organization”.
>
> This appears to be true of the privacy seal programs offered by TRUSTe and BBBOnLine. However, qualifying for the WebTrust privacy seal does involve substantive testing and an opinion is provided.“ (aus: [Mar01])

Zusammenfassend lässt sich also feststellen, dass Gütesiegel einen Beitrag zur Überprüfung der Einhaltung von Datenschutzerklärungen leisten können. Der wichtigste Punkt hinsichtlich einer Verbesserung ist wohl eine stärkere Kontrolle der Dienste, die personenbezogene Daten verarbeiten.

4.5.5.3 Technische Aspekte (DRM)

Ein wichtiger Aspekt bei der Durchsetzung und Unterstützung der beschriebenen Lösungsmöglichkeiten ist der Nachweis einer nicht autorisierten Weitergabe von Daten. Eine Möglichkeit dazu ist die Betrachtung von Digital Rights Management (DRM), was in folgenden kurz angesprochen werden soll.

Für textuelle, personenbezogene Informationen wie Geburtsdatum, Adresse o.ä. ist es prinzipiell nicht oder nur sehr eingeschränkt möglich, eine Weitergabe von Daten durch einen gewissen Dienst nachzuweisen. Allerdings ist dies für multimediale Teile eines Benutzerprofils, wie z.B. ein Bild des Benutzers, mit Hilfe von *Digital Watermarking* möglich [PBB02]. Dabei wird einem Dokument ein Wasserzeichen hinzugefügt, so dass eine eindeutige Zuordnung von Dokument zu Dienst gemacht werden kann. Damit kann bei nicht autorisierter Weitergabe der Dienst identifiziert werden, der dafür verantwortlich ist.

In manchen Bereichen ist zur Durchführung einer gewünschten Transaktion unter Umständen kein Zugriff auf persönliche Daten nötig. Zum Beispiel könnte statt dem Zugriff auf die Kreditkarteninformation eines Kunden eine Zahlungsabwicklung über ein anonymes Zahlungssystem erfolgen. Dies könnte man sich auch für andere Teile des Benutzerprofils vorstellen, z.B. das anonymisierte Senden einer Nachricht ohne Offenbarung der Email- oder Postadresse des Benutzers.

4.5.5.4 Zurechenbarkeit und Integrität

Die bisher betrachteten Überlegungen behandeln in erster Linie die Privatheit und Sicherheit aus Sicht des Benutzers. Zusätzliche Anforderungen aus Sicht des Dienstes sind:

- Zurechenbarkeit bzw. Nicht-Abstreitbarkeit (engl. non-repudiation): Nachweis, dass eine Nachricht bzw. Benutzerprofilinhalte wirklich von dem angegebenen Absender stammt, z.B. dass eine Bestellung wirklich von diesem Benutzer abgegeben wurde oder dass Daten zur Bankverbindung eines Kunden für eine Lastschrift wirklich authentisch ist

- Integrität: Nachweis, dass Benutzerprofilinhalte während der Datenübertragung von Benutzerprofilagenten zum Dienst nicht verändert wurden

- Nachweis, dass einzelne Benutzerprofilinhalte, die von dritten Parteien stammen, authentisch sind (dies betrifft z.B. die Authentizität eines Gutscheins als Teil des Benutzerprofils)

Eine Zurechenbarkeit kann in diesem Framework sowohl durch die Identitätsstufe „veronymous“ und die Integration eines digitales Zertifikat des Benutzers zum Nachweis der Identität, als auch durch das Signieren von Benutzerprofildaten (Option <SIGNED>) erfolgen. Integrität kann durch eine Kombination von <SIGNED> und <ENCRYPTED> erreicht werden. Das Benutzerprofil kann außerdem als Inhalt auch Teile enthalten, die von dritten Parteien digital signiert werden. Dadurch kann die Echtheit einer Gutschrift oder ähnliches nachgewiesen werden. Ein Access Ticket kann also drei (verschiedene) digitale Signaturen enthalten: 1.Signatur von Inhalten des Benutzerprofils durch dritte Parteien, 2.Signatur des Benutzers zum Nachweis der Identität, 3.Signatur des (gesamten) AT durch den Benutzerprofilagenten zur Bestätigung der Echtheit des AT.

Die Zurechenbarkeit von persönlichen Daten dient in erster Linie den Schutzzielen der Dienste. Es ist aber auch aus Benutzersicht wichtig, dass entsprechende Mechanismen vorhanden sind, um das Vertrauen in die Personalisierung und das Identitätsmanagement insgesamt zu verbessern und Missbrauch zu verhindern.

4.6 Fazit

Im Folgenden werden wichtige Beiträge dieser Arbeit zusammengefasst, eine ausführliche Evaluierung der Ergebnisse folgt in Kapitel 5.

4.6.1 Wichtige Ergebnisse

In diesem Teilabschnitt sollen einige maßgebliche Beiträge dieser Arbeit zusammengefasst werden, anschließend werden einige Aspekte im Vergleich zu bestehenden Anwendungen aufgezeigt. Ausführliche Vorteile und Alternativen sind bei der Besprechung der jeweiligen Einzelpunkte in diesem Kapitel zu finden. Wichtige Ergebnisse und Neuerungen sind insbesondere:

- Formalisierung von Zugriffsrechten für Benutzerprofile in XML
- Durch eine Kombination von Zugriffskontrolle und Privacy Enhancing Technologies kann eine Verbindung der Eigenschaften und Verknüpfung der Vorteile beider Ansätze erreicht werden
- Einteilung des Benutzerprofilzugriffs in zwei Phasen (vgl. Abschnitt 4.1.2.1), zunächst Aushandlung von Rechten mit Benutzerinteraktion, dann Datenzugriff
- Eine Einführung von Zugriffniveaus ermöglicht u.a. die explizite Berücksichtigung von Benutzerinteraktion in Zugriffsregeln
- Der Benutzer hat bei diesem Ansatz die Kontrolle über sein Profil:
  - Nur mit Erlaubnis in einer Regel oder (ausdrücklicher) Zustimmung per Benutzerinteraktion ist ein Zugriff auf Benutzerprofilinhalte erlaubt
  - Transparenz für Benutzer möglich, wie oben dargestellt
  - Ausgegebene Zugriffsrechte, respektive Access Tickets, sind jederzeit zurückziehbar
- Integrierbarkeit in den wichtigsten aktuellen Ansatz zum Identitätsmanagement (Liberty Alliance) (s.a. Abschnitt 5.2)

4.6.2 Vergleich mit bestehenden Ansätzen

Die bestehenden Anwendungen für Identitätsmanagement wie das Liberty Alliance Projekt haben den Fokus auf Authentifikation, einer Single Sign On Funktion und dem Austausch von Daten zwischen Diensten. Eine Rechtevergabe wie sie in dieser Arbeit entwickelt wurde, ist (noch) nicht enthalten. Diese Autorisation könnte im Prinzip mit XML-basierter Zugriffskontrolle wie XACML gemacht werden, allerdings ist dabei kein Vokabular bezüglich Privatheit und Datenschutz vorhanden, während dies bei P3P und APPEL als Basis des vorgestellten Ansatzes bereits enthalten ist.

Es erfolgt weiterhin eine Erweiterung bestehender Modelle zur Zugriffskontrolle, u.a. durch neue Zugriffsrechte bzw. Optionen, die bei der Verwaltung von Benutzerprofilen sinnvoll und notwendig sind, sowie eine Modellierung des Zugriffszwecks. Eine Einteilung von Diensten in einzelne Gruppen und Zuordnung von Rechten zu Rollen o.ä. wie bei Role-Based Access Control (RBAC) ist in dem hier betrachteten Szenario nicht sinnvoll, da nur in Ausnahmefällen davon ausgegangen werden kann, dass Datenschutzpraktiken und Anfragekontext zweier Dienste identisch sind und somit einer Gruppe zugeordnet werden könnten. Auch ist es bei traditioneller Zugriffskontrolle schwieriger, ausgegebene Rechte wieder zurückzunehmen.

Bezüglich der in Abschnitt 3.1.3 diskutierten Modelle benutzerbestimmte (DAC) und systembestimmte (MAC) Zugriffskontrolle läßt sich feststellen, dass es sich hier eher um ein benutzerbestimmtes Modell handelt, da kein Administrator o.ä. Rechte festlegt, sondern die Rechtevergabe aufgrund von „Besitzer“ der Daten ausgewählten Regeln geschieht. Dies ist ein Vorteil im Vergleich zu systembestimmten Modellen.

Ein weiterer Punkt ist die Möglichkeit, Anonymisierungs- und Sicherheitskomponenten explizit zu fordern und dies als Bedingung für den Datenzugriff zu formalisieren. Dies ist z.B. auch nicht in P3P oder APPEL enthalten[9], es gibt in APPEL keine Möglichkeit „Datenzugriff ist nur bei sicherem Kanal möglich“ zu formalisieren. Bisher ist dazu immer eine manuelle Entscheidung und Zuschaltung entsprechender Komponenten des Benutzers nötig.

4.6.3 Abgleich mit den Anforderungen

Abschließend soll der erarbeitete Ansatz mit den Anforderungen aus Abschnitt 2.3 abgeglichen werden, indem die dabei wichtigsten Punkte aufgeführt werden.

Autorisierungsziele

- Flexibilität bei der Zugriffskontrolle, u.a. durch die Aufteilung in zwei Phasen und die Aushandlung
- Möglichkeit, Optionen wie „Zugriff nur bei gesicherter Übertragung“ zu realisieren
- Möglichkeiten zur zeitlichen Begrenzung und Zurückziehbarkeit von Rechten
- Möglichkeit, Zugriffsregeln sowohl für bestimmte Dienste oder auch unabhängig von einem konkreten Dienst zu formulieren

Identitätsziele

- Mechanismen für Identitätsmanagement, z.B. der Möglichkeit, Zugriffsregeln in Abhängigkeit einer Identität zu formulieren

[9] Dies liegt an einem generellen Verzicht, Angaben zu Datentransfer in das P3P-Vokubular mit aufzunehmen.

- Möglichkeit für Benutzer, anonym oder unter Verwendung eines Pseudonyms zu interagieren (Integration von Identitätsstufen)

Vertraulichkeitsziele

- Möglichkeit zur Festlegung von Zugriffsrechten für Benutzerprofil-Attribute (Zugriffskontrollsystem)
- Integration gesicherter und anonymisierter Kommunikationsbeziehungen

Absicherungsziele

- Speicherung des Profils unter der Kontrolle des Benutzers
- Zweckbindung von Profilzugriffen
- Kontrolle der „Weitergabe von Daten“
- Möglichkeiten der Zurechenbarkeit und Unabstreitbarkeit
- Möglichkeit der Signierung von Zugriffsrechten (durch den Benutzerprofilagenten)

Transparenzziele

- Möglichkeit für Benutzer, Zugriffe überwachen zu können, Protokollierung aller Zugriffe
- Möglichkeit, vergebene Zugriffsrechte jederzeit überprüfen und ggf. zurückziehen zu können
- Integration von Gütesiegeln vertrauenswürdiger Institutionen

Kapitel 5

Evaluierung und Systementwurf

„The psychological effects of life on the screen can be complicated: a safe place is not all that is needed for personal change. [...] When people adopt an online persona they cross a boundary into highly charged territory. Some feel an uncomfortable sense of fragmentation, some a sense of relief. Some sense the possibilities for self-discovery, even self-transformation.“
Aus: [Tur96]

Nachdem ein Mechanismus erarbeitet wurde, der die Anforderungen hinsichtlich Privatheit bei einer dezentralen Verwaltung von Benutzerprofilen erfüllt, soll dieser nun evaluiert werden. Dazu wird zunächst die Umsetzbarkeit des Ansatzes in Benutzerschnittstellen untersucht, wobei insbesondere eine Betrachtung der Transparenz für den Benutzer wichtig ist. Dann wird eine mögliche Integration in einen aktuellen und relevanten Standard (Liberty Alliance) gezeigt, sowie ein Systementwurf und eine (Teil-)Implementierung in dem konkreten Projektumfeld dargestellt. Abschließend wird eine mögliche Übertragbarkeit des Ansatzes auf andere Anwendungsbereiche betrachtet.

5.1 Umsetzbarkeit in Benutzerschnittstellen

Ein wichtiger Aspekt eines Identitätsmanagementsystems ist die Möglichkeit für den Benutzer, sein Profil zu pflegen, Zugriffsregeln festzulegen und andere Konfigurationseinstellungen zu machen. Dazu sind geeignete Benutzerschnittstellen erforderlich. In diesem Abschnitt werden einige Grundsätze dazu erarbeitet, die die Grundlage für eine Implementierung und empirische Untersuchung bilden könnten.

5.1.1 Regelerstellung und -pflege

Zunächst stellt sich die Frage, wie Benutzer ihre Regeln erstellen und pflegen können. Eine geeignete Grundmenge an Regeln könnte dazu von vertrauenswürdigen Institutionen und Organisationen bereitgestellt werden, z.B. von den Organisationen, die auch die Gütesiegel herausgegeben. Zusätzlich muss der Benutzer die Regeln an seine persönlichen Präferenzen anpassen können. Dazu ist es erforderlich, eine suggestive Benutzerschnittstelle anzubieten, wozu in diesem Teilabschnitt einige Grundsätze erläutert werden sollen.

Die Benutzerschnittstelle kann dem Benutzer ausgehend von einer geeigneten Regelmenge zu jedem Attribut eine sinnvolle Menge von Regeln für dieses Attribut vorschlagen. Dies sollte abhängig

vom betreffenden Datum sein, Informationen zur Sozialversicherung erfordern sicherlich einschränkendere Regeln als Daten, die öffentlich zugänglich sind. In Abb. 5.1 ist beispielhaft eine mögliche Benutzerschnittstelle angedeutet[1].

ABBILDUNG 5.1: Festlegung von Zugriffsregeln

Es könnte zu jedem Attribut eine sinnvolle Menge von Regeln für den Benutzer angeboten werden, z.B. mit Hilfe einer Auswahlbox wie in Abb. 5.1 gezeigt. Jedem Attribut wird dabei also eine Regel zugeordnet. Einige weitere Aspekte dabei:

- Geeignete Default-Einstellungen sind wichtig, z.B. eine Voreinstellung von „gesicherter Verbindung“ für Kreditkarten- oder anderen Zahlungsdaten
- Elemente der Benutzeroberfläche sollten vom Kontext abhängig sein, z.B. beim Zugriff auf den Namen des Benutzers sollte keine Regel „anonymisierter Zugriff“ angeboten werden
- Verschiedene Modi sind sinnvoll: z.B. „beginner“ - „normal“ (etwa wie in Abb. 5.1) - „expert“ (genauere Einstellungen möglich) - „manual rule creation“
- Keine Überladung der Oberfläche, manche Einstellungen könnten mit Hilfe eines weiteren Dialogfensters (Schaltfläche „Erweitert ...“) angeboten werden
- Ein angemessenes Abstraktionsniveau ist wichtig [PSWW00]: technische Details sollten vorborgen werden können, zumindest für den nicht fortgeschritten Benutzer, z.B. muss das Konzept der Zugriffniveaus nicht explizit in der Benutzerschnittstelle abgebildet werden, sondern dies kann implizit in den Regelvorschlägen enthalten sein

Der Regelsatz sollte dynamisch erweitert werden können, z.B. wenn der Benutzer bei einer Benutzerinteraktion explizit Erlaubnis für einen bestimmten Zugriff gibt und über eine zusätzliche Bestätigung zum Ausdruck bringt, dass dies auch für zukünftige Zugriffe gelten soll. Dann kann die Regelmenge um eine entsprechende Regel erweitert werden. Wichtig dabei ist es, dass getätigte Einstellungen leicht wieder rückgängig gemacht werden können, was bei vielen bestehenden Systemen nicht der Fall ist.

Eine genauere Analyse einer konkreten graphischen Benutzeroberfläche ist nicht im Fokus dieser Arbeit, für einige allgemeine Gesichtspunkte sei auf den Ausblick (vgl. Abschnitt 6.2) verwiesen.

[1] Es wurde in dem Beispiel kein Wert auf eine sinnvolle Anordnung und Ausgestaltung der Texte und graphischen Elemente gelegt.

5.1.2 Transparenz für Benutzer

Neben einer Benutzerschnittstelle zur Festlegung von Regeln ist es wichtig, Funktionen zur Unterstützung der Transparenz des Benutzers vorzusehen. Dabei sollte u.a. die Möglichkeit bestehen, vergebene Zugriffsrechte jederzeit überprüfen und (tatsächliche) Zugriffe überwachen zu können (vgl. Anforderungen, Abschnitt 2.3). Hierfür ist eine Aufzeichnung (Log) der folgenden Aktionen nötig:

- Historie der ausgegebenen Access Tickets (insbesondere natürlich die Menge der derzeit gültigen AT's)
- Aufzeichnung der Aushandlung der Zugriffsrechte (zur Nachvollziehung für den Benutzer)
- Protokoll von (erlaubten und abgelehnten) Datenzugriffen

Dazu gehört eine Integration von (konfigurierbaren) Benachrichtigungsfunktionen für den Benutzer, um die *Awareness* zu unterstützen. So könnte ein Benutzer zum Beispiel einstellen, bei jeder (automatisch ausgehandelten) Ausgabe eines Access Tickets eine Email mit den vergebenen Zugriffsrechten geschickt zu bekommen.

Wichtig ist es auch hier, eine geeignete Benutzeroberfläche vorzusehen. Zum Beispiel, dass es die Benutzerschnittstelle ermöglicht, aus der Zugriffshistorie eine Funktion aufzurufen, um eine zu einem Log-Eintrag gehörende Regel oder AT zu ändern. Man könnte sich auch eine Simulation der Wirkung von Regeln in Hinblick auf konkrete Anfragen vorstellen. So könnte ein Benutzer überprüfen, wie sich z.B. unterschiedliche Regelmengen oder abgeänderte Regeln auf (typische) Anfragen von Diensten auswirken.

5.1.3 Funktionen Personal Identity Assistant

Die dargestellten Funktionen müssen in ein Personal Identity Assistant Werkzeug integriert werden (vgl. dazu die Grafik in der Einleitung zu Kapitel 4, Abschnitt 4.1.2.1). Zu den Aufgaben des PIA gehören im Detail:

- Schnittstelle für den Benutzer zur
  - Pflege seines Benutzerprofils
  - Auswahl und Festlegung von Zugriffsregeln
  - Abfrage und Auswertung von Log-Einträgen (Transparenz)
- Abwicklung des Dialogs zur Benutzerinteraktion in Phase I
- Identitätsmanager für den Benutzer:
  - Auswahl aus verschiedenen Pseudonymen etc.
  - Verwaltung von digitalen Zertifikaten des Benutzers
- (Automatische) Integration von Anonymisierungs- und Sicherheitskomponenten, z.B.
  - Client eines Anonymisierung-Proxies (z.B. JAP der TU Dresden)
  - Secure Socket Layer (SSL) für gesicherte Verbindung auf Netzwerkebene
- Generierung von (Dienst-übergreifenden) Log-Dateien

- Möglichkeit zur Integration von individualisierten, Client-seitigen Diensten, wie personalisierte Suchmasken für Internet Recherchen usw.

Auch zu beachten ist dabei die Authentifikation des Benutzers am Identitätsmanager bzw. Benutzerprofilagenten. Dies wird in dieser Arbeit nicht näher behandelt, aber könnte z.B. mit Hilfe einer SmartCard, die ein digitales Zertifikat des Benutzers enthält, biometrischen Verfahren und/oder Passwörtern erfolgen.

Bisher wird die aufgeführte Funktionalität eines Personal Identity Assistants durch verschiedene, unabhängige Programme bereitgestellt. Der Benutzer muss z.B. den JAP-Client installieren und manuell eine Anonymisierung vornehmen. Sinnvoll wäre eine Integration von Identitätsmanagementfunktionen in Web-Browser wie z.B. dem (Open Source) Mozilla. Bisher enthalten Browser in Bezug auf Sicherheit und Privatheit nur einen Teil der folgenden Funktionen: Verwaltung von Benutzernamen und Passwörter, Cookie-Management, Integration von P3P, eine Konfiguration allgemeiner Sicherheitsniveaus (beim Microsoft Internet Explorer), sowie die Möglichkeit einzelne Sicherheitsrelevante Einstellungen wie (De-)Aktivierung von Java, JavaScript, Active-X etc. vorzunehmen. Alternativ könnten die in Abschnitt 3.3 diskutierten Anwendungen für Identitätsmanagement im Internet für diese Aufgabe erweitert werden.

Sehr wichtig ist in diesem Zusammenhang auch die konkrete Ausgestaltung der (i.d.R. graphischen) Benutzeroberfläche für die skizzierten Benutzerschnittstellen. Dies gilt allgemein für Privacy Enhancing Technologies und Sicherheits-Werkzeuge, wozu eingehende Untersuchungen und Feldversuche erforderlich sind. Eine genauere Betrachtung liegt nicht im Fokus dieser Arbeit, im Ausblick wird auf einige Problematiken und Grundsätze hingewiesen (vgl. Kap. 6.2).

Als Fazit lässt sich festhalten, dass mit den beschriebenen Konzepten eine Grundlage für eine größtmögliche Transparenz für den Benutzer geschaffen wurde. Im Vergleich zu anderen Ansätzen sind z.B. die tatsächlich vorhandenen Rechte von Diensten durch eine Abfrage der gültigen Access Tickets jederzeit einfach nachvollziehbar.

5.2 Integration in Liberty Alliance

Neben der Umsetzbarkeit des Mechanismus dieser Arbeit in konkrete Benutzerschnittstellen ist auch die Integration und Kompabilität in bestehende Standards zum Identitätsmanagement ein wichtiger Gesichtpunkt der Evaluierung. Der wichtigste Aspekt dabei ist ein Abgleich mit der Liberty Alliance Spezifikation.

Wie in Abschnitt 3.3 dargestellt, konzentriert sich das Liberty Alliance Projekt bisher auf die Spezifikation eines SSO-Dienstes mit dem Schwerpunkt auf eine kontrollierte Verbindung von verschiedenen Identitäten. Dies betrifft also die Authentifikation von Benutzern, während in dieser Arbeit der Fokus auf der Autorisation liegt. Eine Unterstützung von Autorisation ist für zukünftige Versionen von Liberty Alliance in folgender Art und Weise vorgesehen:

> „The next Liberty Alliance specification will include the notion of a ”container” for expressing privacy rights. P3P and other languages might be plugged into this containers. The spec will allow service providers to make requests for specific data elements that will be used in specific ways. A simple negotiation can then take place.“ (aus dem Bericht zum Workshop on the Future of P3P, Washington DC, Nov. 2002, verfügbar unter www.w3.org/2002/12/18-p3p-workshop-report.html)

Es werden also verschiedene Rechtesprachen in Liberty Alliance zum Einsatz kommen können. Dabei bietet sich die Verwendung der in dieser Arbeit erarbeiteten Access Tickets an, weil diese

zum dezentralen Zugriff auf Benutzerprofile entwickelt wurden, wie es auch für die Verwaltung von Identitäten bei Liberty Alliance benötigt wird. Eine Verwendung von Access Tickets ist dabei noch zweckmäßiger als P3P-Datenschutzerklärungen, da erstere um wichtige zusätzliche Aspekte erweitert wurden, wie ausführlich dargestellt wurde. Das beschriebene Protokoll zur Aushandlung dieser Rechte kann wohl auch bei einer Integration in Liberty Alliance in ähnlicher Weise verwendet werden, da die Möglichkeit einer „einfachen Aushandlung“ vorgesehen ist.

Das Liberty Alliance System von virtuellen Identitäten könnte außerdem für eine pseudonyme Authentifikation auch in dieser Arbeit verwendet werden. Der hier beschriebene Ansatz macht keine Einschränkung wie eine Realisierung von Pseudonymen auszusehen hat.

Damit steht der erarbeitete Ansatz nicht in Konkurrenz zu Liberty Alliance, sondern ist im Gegenteil eine sinnvolle Ergänzung dazu. Eine konkrete Integration des eigenen Mechanismus in das Liberty Alliance Framework ist als Erweiterung und Fortführung der hier präsentierten Arbeit für die Zukunft geplant.

5.3 Systementwurf und Implementierung

5.3.1 Projektumfeld

Im Folgenden wird die Implementierbarkeit des Ansatzes dieser Arbeit im Projekt Cobricks gezeigt. Dazu wird zunächst das Projektumfeld etwas näher vorgestellt, dann ein Systementwurf in diesem Umfeld erarbeitet und schließlich kurz auf eine prototypische Teil-Implementierung hingewiesen.

Das Ziel des Projektes Cobricks (siehe u.a. [Koch02, KoWö01]) ist es, generische Komponenten zur modularen Entwicklung von Community-Unterstützungssystemen zu identifizieren und zu realisieren. Zu diesen Komponenten gehören z.B. ein „Item Manager“, der für die Verwaltung von Community-Informationen – wie etwa von den Community-Mitgliedern gemeinsam genutzte Literaturreferenzen oder Hyperlinks – zuständig ist, sowie eine Komponente „User Profil Manager“ als Schnittstelle zum Benutzerprofilzugriff in einem Community Agenten.

Darüber hinaus gibt es Komponenten für einen Benutzerprofilagenten. Dabei werden Schnittstellen zum Profilzugriff bereitgestellt, die u.a. über Agentenkommunikation (FIPA) oder Web Services (SOAP) verfügbar sind. Ein UPA speichert Benutzerprofile in XML-Form, wie im Grundlagen-Kapitel (Abschnitt 2.1) gezeigt, dessen Struktur mit einer Ontologie für die Inhalte der Community verbunden ist. Außerdem werden Mechanismen zur Authentifikation angeboten.

Zum Profilzugriff für CA's werden von einem UPA u.a. die folgenden Methoden bereitgestellt:

```
String getDataGroup (String identityID, String datagrouppath)
void putDataGroup (String identityID, String datagrouppath, String xml)
```

Diese Methoden dienen der Abfrage (Zugriffsmodus <READ>) und der Modifikation (<WRITE>, <APPEND>, <CREATE>) von Profilen. Dazu wird ein Identifikator (dies entspricht der Angabe in einem <USER> Element eines Access Tickets) sowie der Pfad zum gewünschten Profilteil (z.B. „/profile/demographic/email“) übergeben. Bei putDataGroup wird zusätzlich der neue Benutzerprofilinhalt in XML-Form mitgeschickt. Das Ergebnis beim „get“ wird in XML als String zurückgeliefert.

Um jetzt in diesem Framework eine Zugriffskontrolle zu integrieren, wird eine Komponente „Privacy Engine“ als Bestandteil des Benutzerprofilagenten entworfen, die die Mechanismen in dieser Arbeit in ein Softwaremodul umsetzt.

5.3.2 Komponenten der Zugriffskontrolle

Neben den Komponenten ist auch eine Betrachtung der benötigten Datenstrukturen wichtig.

5.3.2.1 Komponenten

Die Privacy Engine (PE) hat folgende Teil-Komponenten (vgl. Abb 5.2):

- Rule Engine (RE): Realisiert die Aushandlung der AT's, also die Phase I
- Ticket Decision Point (TDP): Prüft Access Ticket einer Anfrage (Phase II) und leitet erlaubte Zugriffe an den TEP weiter
- Ticket Enforcement Point (TEP): Macht den Benutzerprofilzugriff, realisiert dabei u.a. die Wahl des Kommunikationskanals (in Abb. 5.2 wird dies durch unterschiedliche Pfeile veranschaulicht)
- Logger (LOG): Wird von den anderen Komponenten benutzt, um eine Historie der Aktionen zur Aushandlung und zum Datenzugriff aufzuzeichnen (vgl. die Bemerkungen zur Transparenz für den Benutzer, Abschnitt 5.1.2)

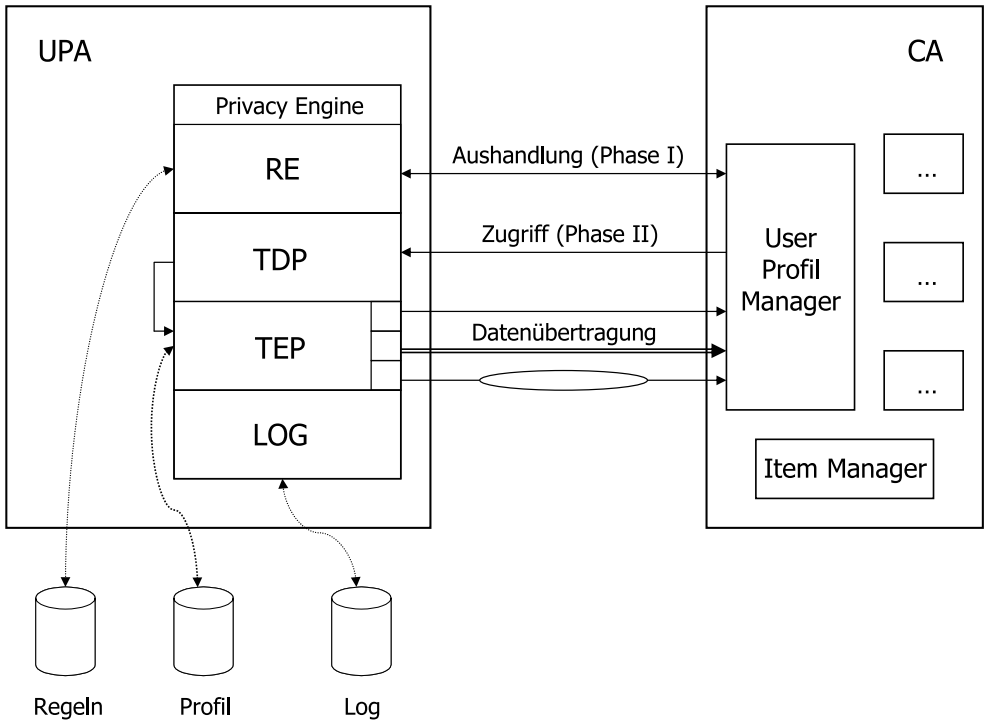

ABBILDUNG 5.2: Komponenten

Der grundsätzliche Ablauf bei der Phase I (RE) bzw. II (TDP) wurde schon im Abschnitt 4.5.4 (Phase I) bzw. 4.4.2 (Phase II) dargestellt.

5.3.2.2 Datenhaltung

Folgende Daten müssen persistent abgelegt werden (vgl. Abb 5.2):

- Benutzerprofile (Zugriff vom TEP):

  - Speicherung der Profile in XML-Form
  - Liste der Benutzerprofilattribute mit „subscription„ Option eines Dienstes[2]

- Regeln: Zugriffsregeln der Benutzer, für die Aushandlung (RE)
- Log: Transaktions-Historie, dies beeinhaltet:
  - Access Tickets: dies beinhaltet sowohl die aktuell gültigen, als auch abgelaufene und zurückgezogene[3]
  - Aushandlung
  - Tatsächliche (und abgelehnte) Zugriffe

5.3.3 Entwurf von AccessRequest und AccessTicket Klassen

Alle Komponenten müssen Access Requests und/oder Tickets verarbeiten, daher wird in diesem Teilabschnitt ein Entwurf entsprechender Klassen als wichtigster Bestandteil einer Implementierung angegeben.

In Abb. 5.3 ist das entsprechende Klassendiagramm gezeigt[4]. Nachdem bei Access Request und Ticket die meisten Bestandteile identisch sind, wurden sie als eine Generalisierung einer gemeinsamen, abstrakten Oberklasse `AccessToken` entworfen. Die Komponenten von AR und AT sind als Attribute der Klasse modelliert. <USER> und <SERVICE> wurden ebenfalls als Ableitung von einer abstrakten Oberklasse definiert.

Der Übersicht halber sind von den drei Arten von Optionen nur die Optionen bezüglich der Datenübertragung genauer als (Hilfs-)Klasse angegeben. Es ist sinnvoll, auch die anderen Optionen als eigene Klassen zu modellieren, damit z.B. eine Erweiterung leicht realisiert werden kann. In diesen Klassen für die Optionen können auch u.a. sinnlose Kombinationen von Optionen (vgl. die entsprechende Abschnitte in Kap. 4.2) verhindert werden.

5.3.4 Implementierung

In diesem Teilabschnitt wird eine prototypische Teil-Implementierung einer AT Klasse auf Basis des erläuterten Systementwurfs dargestellt. Dies soll die Implementierbarkeit in einem größeren Projektumfeld zeigen und damit zur Evaluierung der Lösung beitragen. Die Implementierung wurde z.T. im Rahmen eines „Systementwicklungsprojektes“ eines Studierenden abgewickelt.

Es wurde ein Teil von Phase II im TDP realisiert. Dazu wurden die schon vorgestellten Methoden zum Profilzugriff um einen Parameter accessticket erweitert:

```
String getDataGroup (String identityID, String datagrouppath,
                     String accessticket)
void putDataGroup (String identityID, String datagrouppath, String xml,
                     String accessticket)
```

[2] Damit bei einer Änderung von Attributen die betreffenden Dienste benachrichtigt werden können.

[3] Eine Liste der zurückgezogenen Access Tickets ist notwendig, damit der UPA eine entsprechende Prüfung durchführen kann, vgl. Abschnitt 4.4.1.2.

[4] Auf eine vollständig Darstellung von Methoden, Typ-Angaben von Attributen usw. wurde zugunsten der Anschaulichkeit des Diagramms verzichtet.

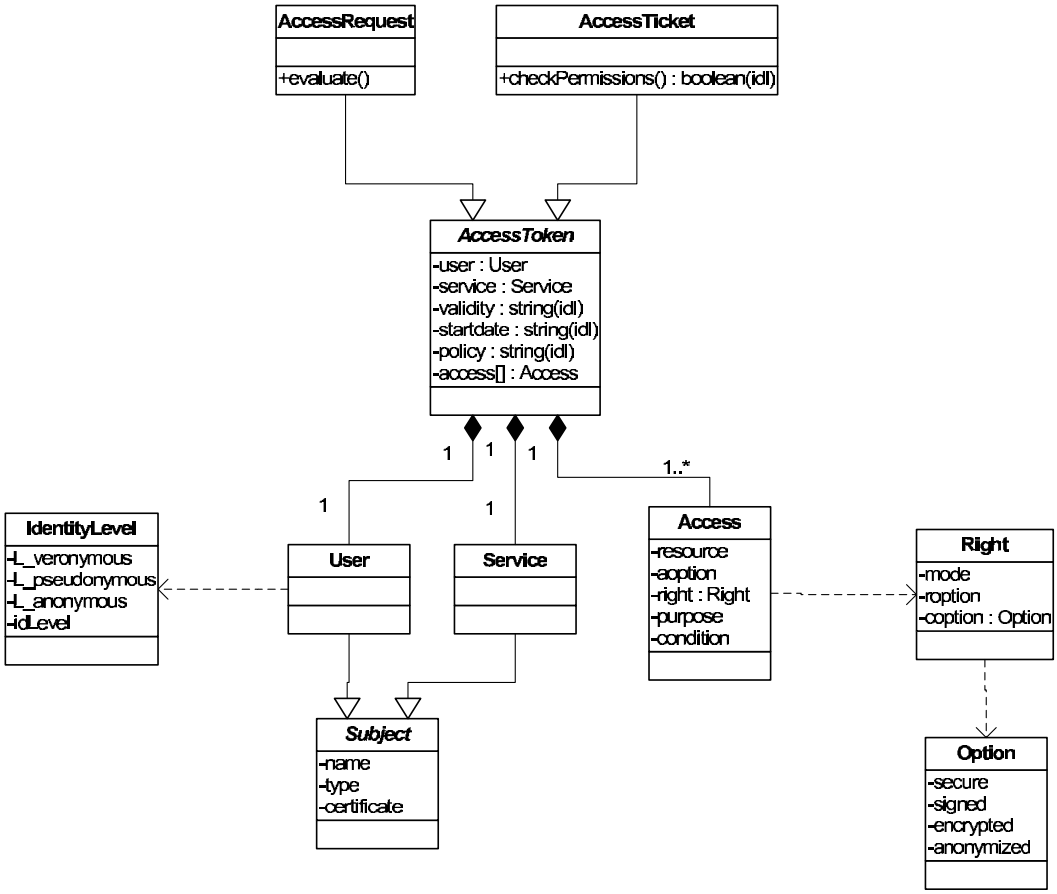

ABBILDUNG 5.3: Klassendiagramm Access Request und Access Ticket

Diese Methoden müssen vor dem Profilzugriff die Gültigkeit des übergebenen Access Tickets kontrollieren. Dazu wurde u.a. eine Methode checkPermissions() in der Klasse `AccessTicket` implementiert, die prüft, ob ein konkreter Zugriffswunsch auf einen bestimmten Benutzerprofilabschnitt durch die Zugriffsrechte im AT abgedeckt ist.

Das AT wird dabei eingelesen und vom XML- bzw. String-Format in ein `AccessTicket` Objekt umgewandelt. Um bei Folgezugriffen nicht erneut dasselbe AT parsen zu müssen, werden die AT-Objekte im Speicher verwaltet. Im Prototypen wurde dazu eine Hashtable verwendet, die die letzten n (z.B. 25) AT's speichert. Erste Tests lassen erkennen, dass ein Zugriff mit der Auswertung des AT's nur unwesentlich langsamer ist, als ohne diese Zugriffskontrolle.

Diese prototypische Implementierung kann damit einen Beitrag leisten, um die die Machbarkeit einer XML-basierten Zugriffskontrolle in einem größeren Framework zur dezentralen Benutzerprofilverwaltung zu zeigen. Derzeit wird ein Client-seitiger Identitätsmanager entwickelt (also eine Implementierung des Personal Identity Assistants), der mit Hilfe der Web Services Schnittstelle des Benutzerprofilagenten eine konkrete Benutzeroberfläche unter Berücksichtigung der in Abschnitt 5.1 erarbeiten Aspekte für den Benutzer zur Verfügung stellen soll.

5.4 Übertragbarkeit auf andere Anwendungsbereiche

Abschließend zur Evaluierung soll die Übertragbarkeit der Lösung auf andere Anwendungsbereiche untersucht werden. Im Prinzip kommt ein vergleichbarer Ansatz dann in Frage, wenn sensible Informationen – insbesondere personenbezogene Daten – verwaltet werden und eine Rollen-basierte Zugriffskontrolle nicht sinnvoll ist, da eine Zuordnung von Rollen oder Gruppen schwierig ist. Dies soll anhand zweier Projekte verdeutlicht werden, *Cosmos* und *GeoPortal*, welche am Lehrstuhl für Angewandte Informatik / Kooperative Systeme, Prof. Dr. J. Schlichter (siehe www11.in.tum.de), durchgeführt werden.

Das Ziel der Aktivitäten im Projekt Cosmos („Community Online Services and MObile Solutions“) (www.cosmos-community.org) ist die „Entwicklung generischer Dienstleistungskonzepte und Technologien für den Betrieb von mobilen Communities“. Dies baut auf der Nutzung von dezentralen Benutzerprofilen auf und hat das Projekt Cobricks als technische Basis. Ein Anwendungsszenario ist z.B. die Kontaktfindung in einer Community von Studenten einer Stadt. Ein Student soll sich dabei beispielsweise eine Liste seiner Freunde, die sich im Moment in seiner Nähe befinden, auf seinem mobilen Endgerät anzeigen lassen können, um mit diesen spontan Kontakt aufnehmen zu können. Dazu ist eine Zugriffskontrolle z.B. für die Abfrage des Aufenthaltsorts einer Person notwendig, weil ein Benutzer sicherlich nicht mit einer sonst möglichen Überwachung durch andere Community-Mitglieder einverstanden wäre.

Für diesen Zweck könnte das Modell einer Aushandlung von Zugriffsrechten dieser Arbeit verwendet und im Vokabular angepasst und erweitert werden. Eine sinnvolle Zugriffregel wäre z.B.: „Zugriff auf meinen aktuellen Aufenthaltsort ist nur gestattet, wenn der Zugreifer in meiner Liste von Freunden enthalten ist und sich gerade in meiner Nähe aufhält“. Wichtig dabei wäre u.a. eine Integration von Ortsabhängigkeit in die Zugriffskontrolle, eine stärkere Betrachtung von Benutzer-zu-Benutzer Beziehungen (in dieser Arbeit wird eine Benutzer-zu-Dienst Sichtweise verfolgt) und eine Berücksichtigung der Beschränkungen von Displays mobiler Endgeräte für die Benutzerschnittstelle.

Im Projekt GeoPortal (www.gis1.bv.tum.de/Forschung/Projekte/GeoPortal/GeoPortal.htm) wird untersucht, wie mit Hilfe von Web-Portalen eine Vermittlung von Anbietern (z.B. die Vermessungsverwaltung eines Bundeslandes) und Nutzern (z.B. ein Ingenieur-Büro) geographischer Daten realisiert werden kann. Die Grundidee des GeoPortals ist die verteilte Datenhaltung: Die vermittelten Daten verbleiben bei den Datenanbietern, die sie erfassen und pflegen. Autorisierungs-Probleme in diesem Zusammenhang sind z.B., dass ein Mitglied einer Gemeindeverwaltung nur Zugriff auf Daten seines Verwaltungsbereiches haben darf oder der Eigentümer eines Grundstücks nur Informationen zu den Nachbargrundstücken abfragen darf.

Eine Zugriffsentscheidung kann dabei von vielen Faktoren abhängig sein, die evtl. auch eine Prüfung durch eine staatliche Stelle umfassen könnte. Es wäre daher wohl auch hier eine Trennung von Rechteaushandlung und Datenzugriff sinnvoll. Dabei könnte ein möglicher Zugreifer eine Art Access Ticket von einem Autorisierungs-Server erhalten, mit dem dann der Zugriff auf die Daten in der eigentlichen GeoPortal-Anwendung unter einem angegebenen Kontext möglich wäre. Die Formalisierung der Bedingungen für den Datenzugriff (z.B. „Eigentümer des Nachbargrundstücks“) könnte in Zugriffsregeln in ähnlicher Art und Weise erfolgen, wie in dieser Arbeit Bedingungen für den Benutzerprofil (z.B. der Zweck eines Benutzerprofilzugriffs) modelliert wurden.

Wichtige Aspekte, die man sowohl in dem in dieser Arbeit betrachteten Szenario als auch den kurz vorgestellten Projekten abstrahieren kann, sind insbesondere eine Dynamik, Kontextabhängigkeit und Verteiltheit der Zugriffskontrolle.

Die Dynamik kommt dadurch zustande, dass eine Zugriffsentscheidung nicht immer durch die (statische) Angabe einer Subjekt-Modus-Objekt Beziehung wie in den meisten traditionellen Ansät-

zen zur Zugriffskontrolle gemacht werden kann. Eine Trennung der Zugriffskontrolle in eine Rechteaushandlung und dem eigentlichen Zugriff (zwei Phasen), sowie eine Spezifikation von Zugriffsregeln, ermöglicht eine Zugriffsentscheidung unter Berücksichtigung komplexer Sachverhalte oder einer Benutzerinteraktion, und trotzdem einen effizienten Datenzugriff mit den ausgehandelten Rechten.

Eine Kontextabhängigkeit wird sichtbar, wenn eine Zugriffsentscheidung von gewissen Bedingungen oder Einschränkungen in der Anwendungsdomäne abhängt. Dies ist z.B. die „Übertragung über einen gesicherten Kanal“. Durch eine Modellierung der Anwendungsdomäne können Voraussetzungen eines Zugriffs formalisiert werden, wie dies in dieser Arbeit für die Domäne (dezentrale) Benutzerprofilverwaltung durchgeführt wurde. Das Prinzip einer Abhängigkeit eines Datenzugriffs vom Kontext kann auch auf andere Domänen übertragen werden.

Eine Verteiltheit der Zugriffskontrolle kann sich ergeben, wenn eine Zugriffsentscheidung nicht lokal getroffen werden kann, sondern in einem verteilten Umfeld nötig ist. Dabei muss die Entität, die die Zugriffsentscheidung trifft, nicht mit der Komponente, die einen Datenzugriff durchführt, identisch sein. Die Grundidee dazu ist die Formalisierung von Zugriffsrechten in einem XML-Dokument, welches vom Besitzer der Daten oder einer vertrauenswürdigen dritten Partei digital signiert wird. Dies ist sicherlich auch auf andere Anwendungsbereiche übertragbar.

Obwohl die Übertragbarkeit hier nicht annähernd gründlich behandelt werden konnte, zeigt sich doch, dass eine Anwendung des Ansatzes dieser Arbeit auf mehr oder weniger ähnliche Anwendungsbereiche möglich sein könnte. Eine direkte Anwendung der Access Tickets, also des speziellen Datenformats ist dabei wohl nicht sinnvoll, aber ein Transfer der dargestellten Grundideen. Im Ausblick wird noch etwas näher auf die Untersuchung von Privatheit bei mobilen und ubiquitären Systemen hingewiesen, da dies eine logische Fortführung dieser Arbeit ist (vgl. Abschnitt 6.3.4). Eine Untersuchung von Gesichtspunkten der Privatheit ist bei mobilen Systemen sehr wichtig, nicht nur hinsichtlich Zugriffskontrolle auf Benutzerprofile, sondern insbesondere auch in Bezug auf Fragen der Anonymität.

Kapitel 6

Zusammenfassung und Ausblick

„The loss of personal privacy is the Number One concern of Americans as the 21st century approaches.“
Wall Street Journal/NBC News Poll, September 1999

In diesem abschließenden Kapitel werden die Ergebnisse dieser Arbeit kurz zusammengefasst. Außerdem wird ein Ausblick auf mögliche weiterführende Arbeiten gegeben. Dabei ist insbesondere eine Untersuchung von Benutzerschnittstellen für Privacy Enhancing Technologies wichtig, daher wird dies etwas ausführlicher dargestellt.

6.1 Zusammenfassung

Der Ausgangspunkt dieser Arbeit waren fehlende Mechanismen, um die Privatheit bei dezentraler Verwaltung von Benutzerprofilen sicherzustellen, insbesondere hinsichtlich Autorisation von Profilzugriffen. Bestehende Modelle und Systeme sind nicht ausreichend, um die Aufgabenstellung zu erfüllen, wie in Kapitel 3 gezeigt wurde.

Darauf aufbauend wurde daher in Kapitel 4 ein Zugriffsschutzmechanismus entwickelt, der eine Formalisierung von Zugriffsrechten und -regeln in dem betrachteten Szenario ermöglicht. Dabei wurden Aspekte von Privacy Enhancing Technologies wie Anonymisierung, eine Verwaltung verschiedener Identitäten eines Benutzers oder die maschinenlesbaren Datenschutzerklärungen des P3P-Standards in eine Zugriffskontrolle integriert.

Der Kern der Arbeit sind so genannte Access Tickets, die eine XML-basierte Formalisierung von Zugriffsrechten für Benutzerprofile konstituieren. Diese Access Tickets werden in einer ersten Phase zwischen Benutzerprofilagent und Dienstagent ausgehandelt, wobei u.a. auch eine Benutzerinteraktion möglich ist. Die Zugriffsregeln sind dabei APPEL-Regeln zur Spezifikation von Datenschutzpräferenzen, die um weitere notwendige Aspekte erweitert wurden.

Der Zugriff in Phase II ist dann vergleichbar mit einer XML-basierten Zugriffskontrolle unter Berücksichtigung zusätzlicher Gesichtspunkte, die bei einer dezentralen Verwaltung eine Rolle spielen. Dies sind u.a. Optionen bei Zugriffsrechten zur Datenweitergabe und dem verwendeten Kommunikationskanal, eine zeitliche Begrenzbarkeit und Zurückziehbarkeit von Rechten und eine Berücksichtigung verschiedener Identitätsstufen.

Die gefundene Lösung wurde evaluiert, indem eine Umsetzbarkeit in Benutzerschnittstellen, die Vereinbarkeit mit dem Liberty Alliance Standard für föderiertes Identitätsmanagement und eine konkrete Integration im Projektumfeld gezeigt wurde. Zu letzterem wurde ein Komponenten- und Syste-

mentwurf, sowie eine prototypische Teil-Implementierung gemacht, um die Implementierbarkeit der erarbeiteten Konzepte zu verifizieren.

Ein Beitrag der Lösung besteht auch darin, dass einige auf andere Anwendungsbereiche übertragbare Grundideen in der dezentralen Autorisation eines Zugriffs auf sensible Daten identifiziert werden konnten. Dazu gehören eine stärkere Berücksichtigung von Dynamik (u.a. durch Aufteilung der Zugriffskontrolle in zwei Phasen), Kontextabhängigkeit (dies betrifft die Modellierung der Anwendungsdomäne, hier z.B. eine Abhängigkeit der Zugriffsgewährung von der Verwendung von personenbezogenen Daten) und Verteilung (durch digital signierte Zugriffsrechte in XML-Form) in der Zugriffskontrolle.

Außerdem konnten einige bisher nicht oder wenig betrachtete Neuerungen in der Zugriffskontrolle erarbeitet werden, nämlich eine Einführung von Zugriffniveaus als Erweiterung des traditionellen „yes“/„no“ Modells der Zugriffskontrolle oder detaillierter Optionen der Anwendungsdomäne zu Zugriffsrechten, in dem betrachteten dieser Arbeit Umfeld Aspekte der Privatheit beim Benutzerprofilzugriff.

6.2 Benutzerschnittstellen für Privacy Enhancing Technologies

Wie in Abschnitt 5.1 angeschnitten wurde, sind Benutzerschnittstellen und -oberflächen für Benutzerprofilverwaltung und Identitätsmanagement sehr wichtig und erfordern eingehender Untersuchungen, insbesondere eine umfassende Evaluation aus Nutzer-Sicht und Feldversuche. Da dies als kleiner Teil dieser Arbeit nicht erschöpfend behandelt werden kann, soll hier kein eigener Vorschlag einer Benutzeroberfläche o.ä. entwickelt werden, sondern in diesem Ausblick nur auf einige Problematiken und Grundsätze allgemein für Privacy Enhancing Technologies hingewiesen werden.

6.2.1 Problematik

Das Thema Benutzerschnittstellen in dem betrachteten Kontext ist auf der einen Seite von hohem Stellenwert, auf der anderen Seite aber auch sehr schwierig:

- Oftmals sind Client-seitige Einstellungen erforderlich, d.h. die Konfiguration muss jeder Benutzer auf seinem eigenen Rechner machen.

- Aufgrund fehlerhafter oder unbeabsichtiger Konfiguration veröffentlichte Daten können nicht einfach wieder zurückgenommen werden, da z.B. eine Email-Adresse eventuell schon in Datenbanken von Werbefirmen aufgenommen wurde. Auch ist es i.d.R. schwierig, Verletzungen von Privatheit und Datenschutz nachzuweisen.

- Sicherheit und Privatheit sind keine Primärziele, d.h. Benutzer arbeiten nicht mit Sicherheitswerkzeugen, um produktiv zu sein. Daher ist davon auszugehen, dass Benutzer generell wenig Motivation haben, sich genauer mit dem Umgang mit Sicherheitsanwendungen zu befassen [MJM01].

- Rechtliche Aspekte: Wie muss dem Benutzer z.B. eine Datenschutzerklärung präsentiert werden, damit man von Zustimmung im datenschutzrechtlichen Sinne ausgehen kann? Auch stellt sich die Frage, wie die Relevanz von ausgehandelten Vereinbarungen von Benutzeragenten juristisch zu bewerten ist [CrRe02].

- Die Default-Einstellungen von verbreiteten Anwendungen, z.B. Microsoft Internet Explorer, sind entscheidend. Untersuchungen zeigen, dass Benutzer nur relativ selten die vorkonfigurierten Einstellungen ändern (vgl. z.B. [BJL01]).

- Die Sicherheit eines Gesamtsystems ist (höchstens) so groß wie die Sicherheit der schwächsten Komponente.

6.2.2 Fallbeispiel

Anhand eines Fallbeispiels der Oberfläche eines PET-Werkzeuges soll im Folgenden gezeigt werden, welche Aspekte eine Rolle spielen können. Es wurde der Java Anonymisierungsproxy (JAP) der TU Dresden (vgl. anon.inf.tu-dresden.de) gewählt, weil dies ein relativ typisches Beispiel für ein PET-Werkzeug ist, auch in den eigenen Ansatz zur Anonymisierung integriert werden könnte und ausserdem eine recht durchdachte Benutzeroberfläche aufweist.

Der JAP ist ein Werkzeug, der auf einem (Client-)Rechner installiert wird, um als Proxy für die Kommunikation zwischen einem lokal ausgeführten Programm, das einen Internet-Zugriff voraussetzt (beispielsweise einem Web-Browser) und einem Anonymisierungs-Server zu dienen [Rol02]. JAP arbeitet auf Basis des in Abschnitt 3.2.4.3 erläuterten Mix-Konzeptes. Die JAP-Oberfläche gliedert sich im wesentlichen in ein Hauptfenster, siehe Abb. 6.1 (von anon.inf.tu-dresden.de/screenshot.html), das ständig über den Status der Verbindung zum Mix-Server informiert und einem Konfigurations-Menü für detailliertere Einstellungen.

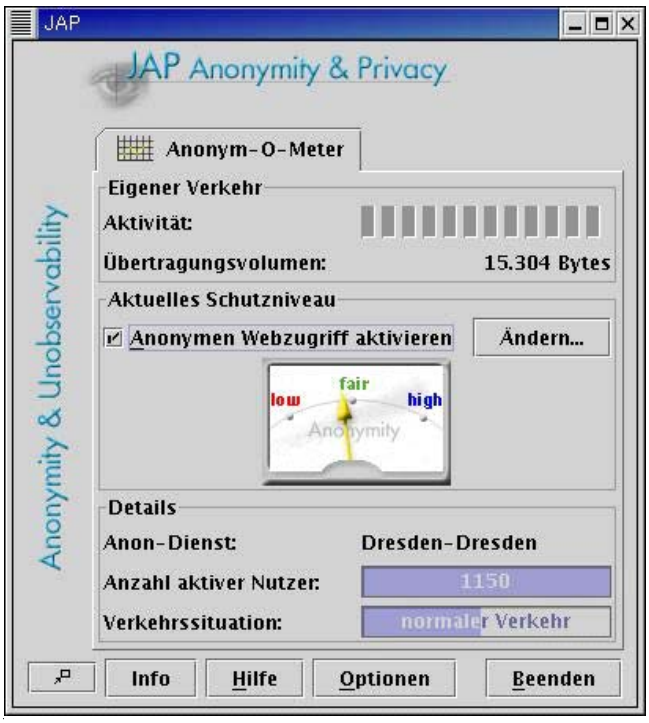

ABBILDUNG 6.1: Java Anon Proxy

Als positiv kann man beim JAP-Hauptfenster festhalten [Rol02]:

- Alle relevanten Informationen sind zentral im Hauptfenster zusammengefasst.
- Es wird eine anschauliche Abstraktion der relativ komplizierten Technik gemacht: Im so genannten „Anonym-O-Meter“ wird der momentane Grad an Anonymität angezeigt, ohne auf technische Details einzugehen.
- Die für die Oberflächenelemente verwendeten Begriffe sind weitgehend selbsterklärend.
- JAP hat eine sinnvolle Default-Konfiguration. Der Benutzer kann sofort das Werkzeug nutzen oder testen, ohne zunächst komplizierte Voreinstellungen machen zu müssen.

Negativ fällt bei der Evaluierung von JAP auf:

- Es erfolgt keine Integration in das Anwendungsprogramm, der Benutzer muss zwischen dem Web-Browser oder einer anderen Anwendung und dem JAP-Werkzeug wechseln, wenn z.B. die Anonymisierung aktiviert werden soll.
- Es gibt keine kontextsensitive Hilfe.
- Die Ablauf der Konfiguration und die Zuordnung von Oberflächenelementen ist nicht immer ganz logisch. Zum Beispiel führen die (verschiendene) Schaltflächen „Ändern...“ und „Optionen“ (vgl. Abb. 6.1) zu dem genau gleichen Konfigurationsmenü.

6.2.3 Einige Grundsätze und Kriterien

In der Literatur finden sich einige Grundsätze, die für Benutzerschnittstellen von Privacy Enhancing Technologies relevant sind. Lau et.al. wenden Kriterien von Victoria Bellotti für „Privacy Interfaces for Information Management“ an [LEW99, Bel97], unter anderem:

- Fehlerfreiheit und Vertrauenswürdigkeit des Systems („fail safety“ und „trustworthiness“)
- adäquate Wahl des Zeitpunkts von Benutzerdialog („appropriate timing“)
- Darstellung aussagekräftiger Informationen („meaningfullness“)
- kein aufdringlicher Charakter der Benutzerschnittstelle („unobstrusiveness“)
- Flexibilität und Anpassbarkeit der Oberfläche („flexibility“)
- geringer Aufwand für den Benutzer („low effort“)
- einfache Erlernbarkeit der Benutzerschnittstelle („learnability“)

Folgenden Punkte können außerdem beim Entwurf von Benutzerschnittstellen in Hinblick auf Privatheit wichtig sein [Rol02]:

- Übersichtlichkeit: Komplexitätsreduzierung erforderlich [MJM01], Verringerung der Funktionalität zugunsten der Überschaubarkeit nötig
- Benutzerschnittstelle soll Datensparsamkeit unterstützen
- Unterscheidung zwischen ungeübten und erfahrenen Benutzer ist nötig

- Selbstbeschreibungsfähigkeit der Oberfläche ist wichtig (z.B. Text anstelle nicht aussagekräftiger Symbole)
- Ausgleich zwischen manueller Konfiguration und automatischen Abläufen

Eine mögliche Beeinflussung des Anwenders spielt eine signifikante Rolle:

- Opt-in versus Opt-out ist ein zentraler Punkt im Datenschutz. Eine Opt-in Wahl, d.h. bei der Verarbeitung von personenbezogenen Daten ist eine explizite Aktion des Benutzers zur Zustimmung notwendig, ist dabei immer vorzuziehen
- Default-Einstellungen sind sehr wichtig, diese sind unter Privatheits-Aspekten zu wählen
- Die Art der Fragestellung kann entscheidend sein. Beispielsweise zeigen die Untersuchungen von Bellman et.al. unter anderem folgendes [BJL01]: Bei einer Auswahl „Do NOT notify me about more health surveys“ antworteten 70,8% der Befragten mit „Ja“ (ohne Voreinstellung), bei der Option „Notify me about more health surveys“ wählten 88,5% ebenfalls „Ja“, obwohl beide Alternativen offensichtlich gegensätzlich sind

Wie erwähnt sind weitere Untersuchungen nötig, um diese und andere Grundsätze in konkrete Benutzeroberflächen umsetzen und evaluieren zu können. Auch ist eine stärkere Integration von einzelnen PET-Werkzeugen erforderlich, um ein benutzerzentiertes und Datenschutz-orientiertes Identitätsmanagement zu ermöglichen.

6.3 Ausblick

Abschließend werden einige ausblickende Bemerkungen gemacht und weitere Bereiche für zukünftige Forschungen im Umfeld dieser Arbeit vorgestellt.

6.3.1 Anonymität versus Zurechenbarkeit

Die vorliegende Arbeit versucht die Privatheit im Internet zu verbessern, dabei stellt sich eine wichtige Frage: Ist eine größtmögliche Privatheit im Internet immer und uneingeschränkt gut?

Wie dargestellt gibt es Zielkonflikte zwischen Benutzern und Diensten bezüglich der Privatheit von Benutzerprofilen. Dies ist auch in einem größeren Zusammenhang in Bezug auf Anonymität und Zurechenbarkeit der Fall. David Davenport argumentiert bezüglich der Annahme, dass Anonymität ein „strong human and constitutional right“ im Internet sei:

> „This view is fundamentally mistaken; by allowing anonymous communication we actually risk an incremental breakdown of the fabric of our society. The price of our freedoms is not, I believe, anonymity, but accountability. Unless individuals and, more importantly, governments can be held accountable, we lose all recourse to the law and hence risk our very freedom.“ (aus: [Dav02])

Davenport führt weiter aus, dass Anonymität nur die Tür für Kriminalität im Internet wie Viren-Verbreitung, Denial of Service Angriffen, Kreditkartenmissbrauch oder Diebstahl der Identität öffne und dass ein Grundrecht auf Rede- und Pressefreiheit nicht Anonymität impliziere [Dav02]. Diese Diskussion ist insbesondere unter den Nachwirkungen der Terroranschläge vom 11.September 2001 zu sehen. In letzter Zeit sind in Folge der Angst vor Terrorismus Programme angelaufen, die Überwachungssysteme im Internet realisieren sollen. Dazu gehört das „Total Information Awareness“ (TIA)

System (siehe z.B. www.darpa.mil/iao/TIASystems.htm), das versuchen soll, mit Hilfe von Mustererkennungsverfahren und Sprachübersetzungssystemen terroristische Aktivitäten im Internet zu erkennen.

Die Abwägung zwischen Anonymität, Zurechenbarkeit und Überwachung im Internet ist eine wichtige Fragestellung für die nahe Zukunft, sowohl in der gesellschaftlichen Diskussion, der gesetzlichen und rechtlichen Entwicklung, als auch des technologischen Fortschritts von Überwachungssystemen auf der einen und Anwendungen zur Anonymisierung und Identitätsmanagement im Internet auf der anderen Seite. Bezüglich letzteren muss sich auch zeigen, inwieweit sich Ansätze wie das Liberty Alliance Projekt in der Praxis bewähren und durchsetzen können.

Ein kontrolliertes Identitätsmanagement, wie zum Teil in dieser Arbeit diskutiert, ist dabei besser als rein anonymisiertes Internet und könnte einen Ausgleich zwischen einem Recht auf Privatheit und Forderungen nach Zurechenbarkeit im Internet schaffen.

6.3.2 Empirische Untersuchung

In Abschnitt 5.1 wurde als Teil einer Evaluierung die Umsetzbarkeit des Ansatzes in dieser Arbeit in Benutzerschnittstellen gezeigt und in 6.2 einige Grundsätze dazu vorgestellt. Wichtig dabei ist es, entsprechende empirische Studien als Nachweis zu machen. Das Kapitel 5 kann als Grundlage einer Implementierung dienen, die im Rahmen von weitergehenden Untersuchungen und Feldstudien – insbesondere hinsichtlich Benutzerschnittstellen – getestet werden könnte. Wie in 6.2 erläutert, ist dies allgemein für Privacy Enhancing Technologies notwendig, wo empirische Untersuchungen noch weitgehend fehlen.

6.3.3 Zukunft von P3P und APPEL

Ein Problem bei P3P ist, dass zwar einige große Web-Sites eine Datenschutzerklärung auch in P3P-Form veröffentlich haben, aber insgesamt die Verbreitung von P3P noch nicht befriedigend ist. Studien zeigen jedoch, dass die Akzeptanz von P3P stetig steigt[1]. Nachdem die Hersteller der verbreiteten Web-Browser, Microsoft und Mozilla/Netscape, eine Integration von P3P durchgeführt oder geplant haben und sich auch an der Weiterentwicklung von P3P beteiligen, ist wohl für die Zukunft eine größere Bedeutung von P3P zu erwarten.

Eine mögliche Erweiterung von P3P ist, das Vokabular mit Hilfe von Semantic Web[2] Technologien zu definieren, insbesondere der Definition einer Ontologie für Datenschutz („Privacy Ontology“). Das Datenschutz-Vokubular ist grundsätzlich schon lange etabliert und in Datenschutzgesetzen verankert. Ein Vorteil davon wäre, dass eine bessere Formalisierung von z.B. dem Zweck einer Verwendung von personenbezogenen Daten möglich wäre. Die bisherige Auswahl aus einer mehr oder weniger willkürlichen Liste vorgegebener Gründe der Datenverwendung in P3P ist ein erster Schritt, aber verbesserungsfähig. Dabei könnten auch Ideen der hier vorgestellten Arbeit einfließen, da z.B. Identitätsstufen oder Optionen bezüglich des Datentransfers noch nicht in P3P oder APPEL vorhanden sind, aber sicherlich in einer Ontologie für Privatheit sinnvoll wären.

Es könnte dann auch APPEL durch eine bessere Regelsprache für Datenschutzpräferenzen ersetzt werden. APPEL wird bisher in der Praxis wenig eingesetzt, wohl auch weil es ein relativ technisches Prinzip des Matchings von Bedingungen für Regeln verfolgt. Eine eventuell neue und verbesserte Re-

[1] Die Informationen in diesem Teilabschnitt stammen vom Workshop on the Future of P3P, Washington DC, Nov. 2002.

[2] „The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.“ [BHL01]

gelsprache könnte über einen Austausch der Zugriffsregeln gut in den Ansatz dieser Arbeit integriert werden.

6.3.4 Privatheit im mobilen und ubiquitären Umgebungen

Wie bereits bei der Diskussion der Übertragbarkeit der Lösungsideen dieser Arbeit auf andere Anwendungsbereiche (Abschnitt 5.4) erwähnt, wäre eine mögliche Weiterentwicklung eine Anwendung für mobile Systeme zur Personalisierung. Eine dezentrale Speicherung von Benutzerdaten in vertrauenswürdigen Benutzerprofilagenten ist auch in Szenarien mit mobilen Endgeräten sinnvoll, da mobile Endgeräte i.d.R. eine beschränkte Technik aufweisen und nicht zur Verwaltung von (kompletten) Benutzerprofilen verwendet werden können.

Mit einer zunehmenden Miniaturisierung der Computertechnologie wird außerdem in Zukunft eine Integration von Prozessoren und kleinsten Sensoren in Alltagsgegenständen möglich werden [MaLa01]. Dieser so genannte „Ubiquitous Computing“ Bereich stellt ganz neue Herausforderungen hinsichtlich der Privatheit bis hin zu einem Potential für einen „Orwell’schen Überwachungsstaat“. Wichtige Fragestellungen in diesem Zusammenhang sind u.a.:

- Wie kann eine mobile Vertraulichkeit erreicht werden? Mechanismen zur Verschlüsselung und Anonymisierung wie im World Wide Web stehen nicht in vergleichbarer Weise in mobiler Kommunikation zur Verfügung.
- Wie kann mit der riesigen Menge an (personenbezogenen) Daten umgegangen werden, die durch eine „Computerisierung“ von Alltagsgegenständen anfallen?
- Wie kann eine notwendige Berücksichtigung von Konzepten wie „örtlicher Lage“ und „(räumlicher) Nähe“ erfolgen?

Eine mögliche Weiterführung dieser Arbeit wäre es, eine Anwendbarkeit und Erweiterung hinsichtlich der Verbesserung der Privatheit bei mobilen und ubiquitären Systemen zu untersuchen. Ein zu den Ideen in dieser Arbeit grob vergleichbarer Ansatz, da auch P3P und Ideen des Identitätsmanagement verwendet werden, ist das *privacy awareness system (pawS)* von Marc Langheinrich [Lan02]. Einen weiteren, ähnlichen Ansatz zum Identitätsmanagement in mobilen Umgebungen beschreiben Jendricke et.al. in [JKZ02].

Der Ausgangspunkt bei pawS ist, dass eine vollständige Anonymität in einer Welt vernetzter Alltagsgegenstände nicht möglich oder durchsetzbar ist. Stattdessen wird ein „notice & choice“ Prinzip verfolgt, vergleichbar zu und unter Verwendung von P3P. Darüber hinaus besteht das System aus:

- „Policy Announcement Mechanism“: ersetzt den HTTP-basierten Mechanismus in P3P zur Übertragung von Datenschutzerklärungen
- „Privacy Proxies“: sind durchgehend laufenden Dienste, die den Transfer von Daten und P3P-Erklärungen erledigen und Zugriffsfunktionen für den Benutzer bereitstellen
- „Privacy-Aware Database“: zeichnet alle Privatheits-relevanten Transaktionen und soll damit die Transparenz für den Benutzer verbessern

Ein Privacy Proxy entspricht dem Benutzerprofilagenten (zusammen mit einer Benutzerschnittstelle) in unserem Szenario. Dabei sollen u.a. auch APPEL-Regeln verwaltet werden, die ausgehend von einer generellen Grundmenge vom Benutzer schrittweise erweitert werden können, und eine Abfrage der Transaktionshistorie ermöglicht werden.

Was bei pawS fehlt, ist eine Integration von Aspekten der Zugriffskontrolle für den Zugriff auf personenbezogene Daten, wie es in der vorliegenden Arbeit für ein nicht-mobiles Szenario realisiert wird. Ausgehend von dem Ansatz in dieser Arbeit die Unterstützung von Privatheit in mobilen und ubiquitären Anwendungen zu untersuchen, erscheint daher zusammenfassend als interessanter und relevanter Bereich für zukünftige Forschungsarbeiten.

6.3.5 Vertrauensmanagement

Die in dieser Arbeit erläuterten Funktionen zur Sicherstellung von Privatheit in dem betrachteten Szenario sollen dafür sorgen, dass Benutzerdaten nur so von Diensten verwendet werden, wie es vom Inhaber der betreffenden Identität gewünscht ist. Dies kann als Teil von „Vertrauen“ aufgefasst werden, denn Vertrauen zwischen Dienstanbieter und Individuen ist eigentlich das angestrebte Ziel. Vertrauen wird dabei nur langsam aufgebaut, kann aber schnell wieder verloren werden.

Verschiedene Ansätze versuchen den Aufbau von Vertrauen bei elektronischer Interaktion zu unterstützen. Dies ist auch notwendig, weil das Internet nicht nur Menschen verbindet, sondern sie auch trennt [Gri01], da z.B. eine persönliche Beratung in einem Fachgeschäft bei E-Commerce nicht möglich ist. Der Ansatz von Winslett et.al. [WYS+02] soll z.B. den Aufbau von Vertrauen durch einen Verhandlungsprozess ermöglichen, bei dem schrittweise Identitätsnachweise und (Zugriffsrechte für) sensitive Daten ausgetauscht werden.

Oftmals findet man im Internet auch ein Bewertungsschema oder ähnliches, die einem Kunden bei einer Entscheidung, welchem Verkäufer er vertrauen soll, unterstützen sollen. Dieser Vertrauensaufbau ist aber immer mit einem Verlust an Privatheit verbunden. Das Zusammenspiel von Vertrauen mit Privatheit wird noch kaum in bestehenden Anwendungen oder Ansätzen in der Forschung betrachtet. Es ist eine stärkere Integration von Aspekten der Privatheit in Systeme, die Vertrauen aufbauen sollen, nötig:

> „Privacy guarantees for parties negotiating trust are an interesting area for future research.“ (aus: [WYS+02])

Anhang A

Syntax Access Request

Hier ist die Syntax eines Access Requests in BNF-ähnlicher Notation, wie sie auch für die Standards des World Wide Web Consoriums (W3C) verwendet wird, abgedruckt. Das Access Ticket wird analog dazu definiert.

```
accessrequest = ‘<ACCESSREQUEST>‘
                  user
                  service
                  validity
                  [startdate]
                  policy
                  1*access
                ‘</ACCESSREQUEST>‘

user = ‘<USER‘ [TYPE="X.500" | TYPE="Cert"] [level]‘>‘ username ‘</USER>‘
username = string
level = ‘LEVEL="‘(‘veronymous‘ | ‘pseudonymous‘ | ‘anonymous‘)‘"‘

service = ‘<SERVICE‘ [TYPE="X.500" | TYPE="Cert" | TYPE="P3P"] ‘>‘ sname
          ‘</SERVICE>‘
sname = string

validity = ‘<VALIDITY>‘ (datestring | ‘infinite‘) ‘</VALIDITY>‘
datestring = date [time]
date = string   ; e.g. 20-03-2002
time = string   ; e.g. 12:00

startdate = ‘<STARTDATE>‘ datestring ‘</STARTDATE>‘

policy = ‘<POLICY>‘ uri ‘</POLICY>‘
uri = string   ; e.g. http://www.server.com/pathto/p3p.xml

access = ‘<ACCESS‘ resource option ‘>‘
           1*right
           purpose
           [condition]
         ‘</ACCESS>‘
resource = ‘RESOURCE="‘ res ‘"‘
res = string   ; XPath,  e.g. /profile/interests/*
```

```
option = [‘OPTION="optional"‘ | ‘OPTION="mandatory"‘]
         [‘GROUP="‘groupname‘"‘]
         [‘ALTERNATIVE="‘altname ‘"‘]
groupname = string
altname = string
right = (read | otherright)
read = (‘<READ‘ [roption] ‘/>‘ |
        ‘<READ‘ [roption] ‘>‘ commopt ‘</READ>‘)
roption = ‘OPTION="‘(‘once-only‘ | ‘distributable‘ | ‘subscription‘)‘"‘ |
            [RECIPIENT = recipient]
recipient = string   ; see [P3P02], ch. 3.3.5
commopt = (‘<SIGNED>‘ | ‘<SECURE‘ [‘TYPE="SSL"‘] ‘>‘|
           ‘<ENCRYPTED>‘ | ‘<ANONYMIZED>‘)
otherright = (‘<‘oright‘/>‘ | ‘<‘oright ‘>‘ [commopt] ‘<‘oright‘/>‘)
oright = (‘WRITE‘ | ‘APPEND‘ | ‘DELETE‘ | ‘CREATE‘)
purpose = string
condition = string
```

Beschreibung des verwendeten Formats:

- `‘text‘`: „Schlüsselwort“
- `elem1 elem2`: Beide Elemente in Sequenz
- `[elem]`: Optionales Element
- `*elem`: Element kann beliebig oft vorkommen
- `1*elem`: Element kann beliebig oft vorkommen, aber mindestens einmal
- `(elem1 | elem2)`: Entweder elem1 oder elem2 muss vorkommen
- `string`: Sequenz beliebiger Zeichen bzw. PCDATA in XML

Anhang B

Syntax Access Ticket

```
accessticket = ‘<ACCESSTICKET>‘
                 user
                 service
                 validity
                 [startdate]
                 policy
                 1*access
               ‘</ACCESSTICKET>‘

user = ‘<USER‘ [TYPE="X.500" | TYPE="Cert"] [level]‘>‘ username ‘</USER>‘
username = string
level = ‘LEVEL="‘(‘veronymous‘ | ‘pseudonymous‘ | ‘anonymous‘)‘"‘

service = ‘<SERVICE‘ [TYPE="X.500" | TYPE="Cert" | TYPE="P3P"] ‘>‘ sname
          ‘</SERVICE>‘
sname = string

validity = ‘<VALIDITY>‘ (datestring | ‘infinite‘) ‘</VALIDITY>‘
datestring = date [time]
date = string   ; e.g. 20-03-2002
time = string   ; e.g. 12:00

startdate = ‘<STARTDATE>‘ datestring ‘</STARTDATE>‘

policy = ‘<POLICY>‘ uri ‘</POLICY>‘
uri = string   ; e.g. http://www.server.com/pathto/p3p.xml

access = ‘<ACCESS‘ resource ‘>‘
           1*right
           purpose
           [condition]
         ‘</ACCESS>‘
resource = ‘RESOURCE="‘ res ‘"‘
res = string   ; XPath,  e.g. /profile/interests/*
right = (read | otherright)
read = (‘<READ‘ [roption] ‘/>‘ |
        ‘<READ‘ [roption] ‘>‘ commopt ‘</READ>‘)
roption = ‘OPTION="‘(‘once-only‘ | ‘distributable‘ | ‘subscription‘)‘"‘ |
```

```
            [RECIPIENT = recipient]
recipient = string   ; see [P3P02], ch. 3.3.5
commopt = (‘<SIGNED>‘ | ‘<SECURE‘ [‘TYPE="SSL"‘] ‘>‘|
          ‘<ENCRYPTED>‘ | ‘<ANONYMIZED>‘)
otherright = (‘<‘oright‘/>‘ | ‘<‘oright ‘>‘ [commopt] ‘<‘oright‘/>‘)
oright = (‘WRITE‘ | ‘APPEND‘ | ‘DELETE‘ | ‘CREATE‘)
purpose = string
condition = string
```

Literaturverzeichnis

[APPEL02] A P3P Preference Exchange Language 1.0 (APPEL 1.0). W3C Working Draft, Apr. 2002, http://www.w3.org/TR/P3P-preferences/[1]

[Abe02] Federated Identity Systems. AberdeenGroup White Paper, Boston MA, May 2002, http://wwws.sun.com/software/sunone/wp-federatedid.pdf

[ACR99] Ackerman, M.S.; Cranor, L.F.; Reagle, J.: Privacy in E-Commerce: Examining User Scenarios and Privacy Preferences. In: Proc. ACM Conference on Electronic Commerce, Nov. 1999

[AKSX03] Agrawal, A., Kiernan, J.; Srikant, R.; Xu, Y.: Implementing P3P using database technology. In: 19th Int'l Conference on Data Engineering, Bangalore, India, Mar. 2003

[Bäu00] Bäumler, H. (Hrsg.): E-Privacy. Vieweg, 2000

[BBB+97] Bayardo, R. J.; Bohrer, W.; Brice, R.; Cichocki, A.; Fowler, J.; Helal, A.; Kashyap, V.; Ksiezyk, T.; Martin, G.; Nodine, M.; Rashid, M.; Rusinkiewicz, M.; Shea, R.; Unnikrishnan, C.; Unruh, A.; Woelk, D.: InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments. In: Proc. ACM SIGMOD International Conference on Management of Data, Tucson AZ, 1997, S. 195-206

[BeLa73] Bell, D.E.; LaPadula, L.: Secure Computer Systems: A Mathematical Model. Mitre Corp., Bedford MA, 1973

[Bel97] Bellotti, V.: Design for Privacy in Multimedia Computing and Communications Environments. In: Agre, P.; Rotenberg, M. (Hrgs.): Technology and Privacy: The New Landscape, MIT Press, Cambridge MA, 1997

[BJL01] Bellman, S.; Johnson, E.; Lohse, G.: To opt-in or opt-out? It depends on the question. In: Communications of the ACM, Vol. 42, Nr. 2, Feb. 2001, S. 25-27

[BHL01] Berners-Lee, T; Hendler, J.; Lassila, O.: The Semantic Web. In: Scientific American, May 2001

[BFK00] Berthold, O.; Federrath, H; Köhntopp, M.: Project „Anonymity and Unobservability in the Internet". In: Proc. of the Tenth Conference on Computers, Freedom & Privacy (CFP 2000), 2000, S. 57-65

[BeKö00] Berthold, O.; Köhntopp, M.: Identity Management Based On P3P. In: Workshop on Design Issues in Anonymity and Unobservability, Berkeley CA, Jul. 2000

[1] *Alle URLs wurden im April 2003 geprüft.*

[BCF01] Bertino, E.; Castano, S.; Ferrari, E.: Securing XML Documents with Author-X. In: IEEE Internet Computing, Vol. 5, No. 3, May 2001, S. 21-31

[BKL+01] Borghoff, U.M.; Koch, M; Lacher, M.S.; Schlichter, J.H.; Weißer, K.: Informationsmanagement und Communities – Überblick und Darstellung zweier Projekte der IMC-Gruppe München. In: Informatik Forschung und Entwicklung, Springer, Jul. 2001, S.103-109

[BSG00] Boucher, P.; Shostack, A.; Goldberg, I.: Freedom System 2.0 Architecture. White paper, Zero-Knowledge Systems Inc., Dec. 2000

[CaCr00] Cavoukian, A.; Crompton, M.: Web Seals: A Review of Online Privacy Programs. In: 22nd International Conference on Privacy and Personal Data Protection, Venice, Sep. 2000, http://www.privacy.gov.au/publications/seals.html

[Cha81] Chaum, D.: Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms. In: Communications of the ACM, Vol. 24, No. 2, Feb. 1981

[Cla99] Clarke, R.: Internet Privacy Concern Confirm the Case for Intervention. In: Communications of the ACM, Vol. 42, No. 2, Feb. 1999, S. 60-67

[CC98] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements. Version 15408-2 FDIS, ISO/IEC SC27 N2162, 1998

[Cra99] Cranor, L.F.: Agents of Choice: Tools that Facilitate Notice and Choice about Web Site Data Practices. In: Proc. 21st Intl. Conf. on Privacy and Personal Data Protection, Hong Kong, China, Sep. 1999

[Cra00a] Cranor, L.F.: Internet Privacy and WWW. Tutorial, WWW9 Conference, May 2000, http://www.research.att.com/projects/p3p/p3p-www9.ppt

[Cra00b] Cranor, L.F.: Gateway: Platform for Privacy Preferences – P3P . In: Datenschutz und Datensicherheit, Vieweg, 2000, http://www.datenschutz-und-datensicherheit.de/jhrg24/p3p.htm

[CrRe02] Cranor, L.; Reidenberg, J.: Can user agents accurately represent privacy notices? In: Proc. 30th Research Conference on Communication, Information and Internet Policy, Alexandria VA, 2002

[Cur02] Curtin, M.: Developing Trust: Online Privacy and Security. Apress, Berkeley CA, 2002

[DVPS00] Damiani, E.; Vimercati, S.D.C.; Paraboschi, S.; Samarati, P.: Securing XML Documents. In: 7th International Conference on Extending Database Technology, Lecture Notes on Computer Science, Vol. 1777, Springer, Mar. 2000

[Dav02] Davenport, D.: Anonymity on the Internet: Why the Price May Be Too High. In: Communications of the ACM, Vol. 45, No. 4, Apr. 2002, S. 33-35

[DGLP97] Dunn, M.; Gwertzmann, J.; Layman, A.; Partovi, H.: Privacy and Profiling on the Web. W3C Note, Jun. 1997, http://www.w3.org/TR/NOTE-Web-privacy.html

[Dys02a] Dyson, E.: Digital Identity Management. Release 1.0, Vol. 20, No. 6, Jun. 2002

[Dys02b] Dyson, E.: Personal Identity Management: The Applications. Release 1.0, Vol. 20, No. 7, Jul. 2002

[Eck01] Eckert, C.: IT-Sicherheit. Oldenbourg, 2001

[Epic01] Electronic Privacy Information Center: Pretty Poor Privacy: An Assessment of P3P and Internet Privacy. June 2000, http://www.epic.org/reports/prettypoorprivacy .html

[FeBe00] Federrath, H; Bertold, O: Identitätsmanagement. In: [Bäu00], S. 189-204

[FeMa98] Federrath, H.; Martuis, K.: Anonymität und Authentizität im World Wide Web. In: ITG-Fachbericht 153, Vorträge der ITG-Fachtagung „Internet – frischer Wind in der Telekommunikation“, VDE-Verlag, 1998, S. 91-101

[FFSS01] Feigenbaum, J.; Freedman, M.; Sander, T.; Shostack A.: Privacy Engineering for Digital Rights Management. In. Proc. Security and Privacy in Digital Rights Management, Philadelphia PA, Nov. 2001

[FeKu92] Ferraiolo, D; Kuhn, R.: Role-Based Access Controls. In: Proc. 15th National Computer Security Conference, Baltimore MD, Oct. 1992

[FLM97] Finin, T.; Labrou, Y.; Mayfield, J.: KQML as an Agent Communication Language. In: Bradshaw, J.: Software Agents, MIT Press, S. 291-316

[FIPA99] FIPA 99 Specification. Technical report, FIPA, 1999

[FiHü01] Fischer-Hübner, S.: IT-Security and Privacy. Lecture Notes in Computer Science, Vol. 1958, Springer, 2001

[FHO98] Fischer-Hübner, S.; Ott, A.: From a Formal Privacy Model to its Implementation. In. Proc. National Information Systems Security Conference (NISSC 98), 1998

[For01] Forrester Research: Surviving the Privacy Revolution. Report, Feb. 2001

[FKK96] Freier, A.O; Karlton, P.; Kocher, P.C.: The SSL Version 3.0. Internet-Draft, Netscape Corp., 1996, http://home.netscape.com/eng/ssl3/draft302.txt

[FNS99] Fujimura, K.; Nakajima, Y.; Sekine, J.: XML Ticket: Generalized Digital Ticket Definition Language. W3C Signed XML Workshop, Apr. 1999, http://www.w3.org/DSig/signed-XML99/pp/NTT_xml_ticket.html

[FuNa98] Fujimura, K; Nakajima, Y: General-purpose Digital Ticket Framework. In: 3rd USENIX Workshop on Electronic Commerce, Aug. 1998, S. 177-186, http://www.usenix.org/publications/library/proceedings/ec98/fujimura.html

[GGK+99] Gabber, E.; Gibbons, P.B.; Kristol, D.M.; Matias, Y.; Mayer, A.: Consistent, yet Anonymous, Web Access with LPWA. In: Communications of the ACM, Vol. 42, No. 2, Feb. 1999, S. 42-47

[Gar94] Garfinkel, S.L.: PGP – Pretty Good Privacy. O’Reilly, 1994

[Gar02] Garfinkel, S.L.: Web Security, Privacy and Commerce. O’Reilly, 2002

[GGPS97] Gattung, G.; Grimm, R.; Pordesch, U.; Schneider, M. J.: Persönliche Sicherheitsmanager in der virtuellen Welt. In: Müller, G.; Pfitzmann, A. (Hrsg.): Mehrseitige Sicherheit in der Kommunikationstechnik, Addison Wesley, Bonn, Reading, 1997

[GRS99] Goldschlag, D.; Reed, M.; Syverson, P.: Onion Routing for Anonymous and Private Internet Connections. In: Communications of the ACM, Vol. 42, No. 2, 1999, S. 39-41

[Gri01] Grimm, R.: Vertrauen in E-Commerce: Wie sicher soll E-Commerce sein? In: [MüRe01]

[GrRo00] Grimm, R; Rossnagel, A.: P3P and the Privacy Legislation in Germany: Can P3P Help to Protect Privacy Worldwide? In Proc. ACM Multimedia, Nov. 2000, http://sit.gmd.de/ grimm/texte/P3P-Germany-e.pdf

[GMM99] Guttman, R.; Moukas, A.; Maes, P.: Agent-mediated Electronic Commerce: A Survey. In: Knowledge Engineering Review, Jan. 1998

[HaSi99] Hagel, J.; Singer, M.: Net Worth: Shaping Markets When Customers Make the Rules. Harvard Business School Press, 1999

[HSD98] Howes, T.; Smith, M.; Dawson, F.: MIME Content-Type for Directory Information (vCARD Specification). RFC 2425, Sep. 1998

[Ian01] Iannella, R: Representing vCard Objects in RDF/XML. W3C Note, Feb. 2001, http://www.w3.org/TR/vcard-rdf

[JKZ02] Jendricke, U.; Kreutzer, M.; Zugenmaier, A.: Mobile Identity Management. In: Proceedings of Workshop on Security in Ubiquitous Computing (UBICOMP), Göteborg, Schweden, 2002

[Kes00] Kesdogan, D.: Privacy im Internet – Vertrauenswürdige Kommunikation in offenen Umgebungen. DuD-Fachbeiträge, Vieweg, 2000

[Köh99a] Köhntopp, M: Pseudonymität – Technik und Recht. Folien für einen Kurzvortrag auf dem DASIT-Treffen in Frankfurt am 13. Dezember 1999

[Köh99b] Köhntopp, M.: Technische Randbedingungen für einen datenschutzgerechten Einsatz biometrischer Verfahren. In: Proc. zur Arbeitskonf. Sicherheitsinfrastrukturen, Vieweg, Wiesbaden, Mar. 1999

[Köh00] Köhntopp, M.: Identitätsmanagement. In: Bäumler, H.; Breinlinger A.; Schrader, H.-J. (Hrsg.): Datenschutz von A-Z, Luchterhand, Neuwied, 2000

[Koch01a] Koch, M.: Kollaboratives Filtern. In: Schwabe, G.; Streitz, N.; Unland, R (Hrsg.): CSCW-Kompendium, Springer Verlag, Berlin, 2001, S. 351-357

[Koch01b] Koch, M.: Community-Support-Systeme. In: Schwabe, G.; Streitz, N.; Unland, R (Hrsg.): CSCW-Kompendium, Springer Verlag, Berlin, 2001, S. 296-296

[Koch02] Koch, M.: An Architecture for Community Support Platforms – Modularization and Integration. In: Luczak, H.; Cakir, A.E.; Cakir, G. (Hrgs.): Proc. 6th Intl. Conf. on Work With Display Units – World Wide Work (WWDU2002), Berchtesgaden, May 2002, S. 533-535

[KLW01a] Koch, M.; Lacher, M; Wörndl, W.: Das CommunityItemsTool – Interoperable Unterstützung von Interessens-Communities in der Praxis. In: Britzelmaier, B.; Geberl, S.; Weinmann, S. (Hrsg.): Proc. 3. Liechtensteinisches Wirtschaftsinformatik-Symposium, Teubner, Stuttgart, 2001, S. 147-157

[KLW01b] Koch, M.; Lacher, M.; Wörndl, W: The CommunityItemsTool – Interoperable Community Support in Practice. Proc. WETICE-2001 (Workshop on Web-based Infrastructures and Coordination Architectures for Collaborative Entreprises), Cambridge, MA, Jun. 2001

[KoWö01] Koch, M.; Wörndl, W.: Community Support and Identity Management. In: Proc. Europ. Conference on Computer-Supported Cooperative Work (ECSCW2001), Bonn, Germany, Sep. 2001

[KoRu00] Kormann, D. P.; Rubin, A. D.: Risks of the Passport Single Signon Protocol. IEEE Computer Networks, Vol. 33, 2000, S. 51-58, http://avirubin.com/passport.html

[KuHa00a] Kudo, M.; Hada, S.: XML Access Control. Proposal, Oct. 2000, http://www.trl.ibm.com/projects/xml/xacl/xmlac-proposal.html

[KuHa00b] Kudo, M.; Hada, S.: XML Document Security Based on Provisional Authorization. In: Proc. 7th ACM Conference on Computer and Communications Security, Athens, Greece, Nov. 2000

[Kuh01] Kuhlen, R.: Privacy Sicherung in der Wirtschaft. Juni 2001, http://www.ib.huberlin.de/ kuhlen/VERT01/trust-v6-1-v-privacy-sicherung-wirtschaft060601.pdf

[Kyus98] Kyas, O.: Sicherheit im Internet. Internat. Thomson Publ., Bonn, 1998

[Lan00] Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein: Neuer Standard für Online-Privacy in Deutschland vorgestellt. Pressemeldung, Aug. 2000, http://www.datenschutzzentrum.de/somak/somak00/p3p_pm.htm

[LKW01] Lacher, M.; Koch, M; Woerndl, W.: A Framework for Personalizable Community Web Portals. In: Proc. HCI International, New Orleans LA, Aug. 2001

[LWKB00] Lacher, M.; Wörndl, W.; Koch, M.; Brede, H.: Ontology Mapping in Community Support Systems. Technical Report 2-2000, Dept. of Computer Science, TU Muenchen, May 2000

[Lan02] Langheinrich, M.: A Privacy Awareness System for Ubiquitous Computing Environments. In: Borriello, G; Holmquist, L.E. (Hrgs.): 4th International Conference on Ubiquitous Computing (UbiComp2002), Springer-Verlag, LNCS 2498, Sep. 2002, S. 237-245

[LEW99] Lau, T.; Etzioni, O.; and Weld, D.: Privacy Interfaces for Information Management. In: Communications of the ACM, Vol. 42, No. 10, Oct. 1999, S. 89-94

[Lau00] Laukka, M.: Criteria for Privacy Supporting System. In: Proceedings of NordSec2000, Reykjavik, Iceland, 2000

[Les01] Lester, T.: The Reinvention of Privacy. The Atlantic Monthly, Mar. 2001, http://www.theatlantic.com/issues/2001/03/lester-p1.htm

[Lib03] Liberty Alliance Specifications, Version 1.1, Jan. 2003, http://www.projectliberty.org/specs/index.html

[LiLo98] Lin, D.; Loui, M.C.: Taking the Byte Out of Cookies: Privacy, Consent, and the Web. In: Proc. ACM Policy, Washington, May 1998

[Lor87] Lorenzen, P.: Lehrbuch der konstruktiven Wissenschaftstheorie. BI Wissenschaftsverlag, Mannheim, 1987

[LuKö94] A.L. Luft, R. Kötter: Informatik Ű Eine moderne Wissens-technik. BI Wissenschaftsverlag, Mannheim, 1994

[MaAd02] Madsen, P.; Adams, C.: Privacy and XML, Part I. XML.com, Apr. 2002, http://www.xml.com/pub/a/2002/04/17/privacy.html

[MGM99] Maes, P.; Guttman, R. H.; Moukas, A. G.: Agents that Buy and Sell. In: Communications of the ACM, Vol. 42, No. 3, Mar. 1999

[Mar01] Markert, B.: Comparison of Three Online Privacy Seal Programs. SANS Institute, Aug. 2001, http://www.sans.org/rr/privacy/seal.php

[MJM01] tom Markotten, D.G.; Jendricke, U.; Müller, G.: Benutzbare Sicherheit – Der Identitätsmanager als universelles Sicherheitswerkzeug. In: [MüRe01]

[MaLa01] Mattern, F.; Langheinrich, M.: Allgegenwärtigkeit des Computers – Datenschutz in einer Welt intelligenter Alltagsdinge. In: [MüRe01]

[Mic01] Building User-Centric Experiences – An Introduction to Microsoft HailStorm. Microsoft white paper, Mar. 2001

[MBG+02] Mont M.; Bramhall P.; Gittler M.; Pato, J.; Rees, O.: Identity Management: a Key e-Business Enabler. Tech. Rep. HBL-2002-164, HP Laboratories Bristol, Jun. 2002

[MAIO97] Mynatt, E.D.; Adler, A.; Ito, M.; Oday, V.L.: Design for Network Communities. In: Proc. ACM SIGCHI Conf. On Human Factors in Computer Systems, 1997

[MüRa99] Müller, G.; Rannenberg, K. (Hrsg.): Multilateral Security in Communications. Addison-Wesley, 1999

[MüRe01] Müller, G.; Reichenbach, M. (Hrsg.): Sicherheitskonzepte für das Internet. Springer, 2001

[MuSc00] Mulligan, D.; Schwartz, A.: Your place or mine? Privacy Concerns and Solutions for Server and Client-side Storage of Personal Information. In: Proc. Computers, Freedom and Privacy, Toronto ON, Canada, Apr. 2000

[Opp97] Oppliger, R.: IT-Sicherheit. DuD-Fachbeiträge, Vieweg, 1997

[OECD80] Organization for Economic Cooperation and Development (OECD): Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. Paris, 1980

[P3P02] P3P: The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, Apr. 2002, http://www.w3.org/TR/P3P/

[Pfi90] Pfitzmann, A.: Diensteintegrierende Kommunikationsnetze mit teilnehmerüberprüfbarem Datenschutz. IFB 234, Springer, 1990

[PfKö01] Pfitzmann, A.; Köhntopp, M.: Anonymity, Unobservability, and Pseudonymity – A Proposal for Terminology. In: Federrath, H. (Hrsg.): Designing Privacy Enhancing Technologies, Proc. Workshop on Design Issues in Anonymity and Unobservability; Lecture Notes in Computer Science, Vol. 2009, Springer, 2001

[PPSW99] Pfitzmann, A.; Pfitzmann, B.; Schunter, M.; Waidner, M.: Trustworthy User Devices. In: [MüRa99], S. 137-156

[PPW88] Pfitzmann, A.; Pfitzmann, B.; Waidner, M.: Datenschutz garantierende offene Kommunikationsnetze. In: Informatik-Spektrum 11/3, 1988, S. 118-142

[PSWW00] Pfitzmann, A.; Schill, A.; Westfeld, A.; Wolf, G.: Mehrseitige Sicherheit in offenen Netzen. DuD-Fachbeiträge, Vieweg, 2000

[PBB02] Piva, A; Bartolini, F; Barni, M.: Managing Copyright in Open Networks. IEEE Internet Computing, Vol. 6, Issue 3, May 2002

[Reg02] Regan, K.: Microsoft To Phase Out Passport Payment Service. E-Commerce Times, Sep. 2002, http://www.ecommercetimes.com/perl/story/19268.html

[ReRu97] Reiter, M.K.; Rubin, A.D.: Crowds: Anonymity for Web Transactions. Technical Report 97-15, DIMACS, Aug. 1997

[Rös97] Röscheisen, M.: A Network-Centric Design for Relationship-based Right Management. Ph.D. Dissertation, Computer Science Department, Stanford University, 1997

[RW97a] Röscheisen, M.; Winograd, T.: A Network-Centric Design for Relationship-based Security and Access Control. In: Journal of Computer Security, Special Issue on Security in the World Wide Web, 1997

[RW97b] Röscheisen, M.; Winograd, T.: The Stanford FIRM Framework for Interoperable Rights Management. Forum on Technology-based Intellectual Property Management, Washington DC, 1997

[Rol02] Rollar, T.: Benutzerschnittstellen für Privacy Enhancing Technologies. Diplomarbeit, TU München, 2002

[Salz03] Salz, R.: Securing Web Services. XML.com, Jan. 2003, http://www.xml.com/pub/a/2003/01/15/ends.html

[SaHa00] Samuels, R.; Hawco, E.: Untracable Nym Creation on the Freedom 2.0 Network. White paper, Zero-Knowledge Systems Inc., Nov. 2000

[SaSa94] Sandhu, R.; Samarati, P.: Access Control: Principles and Practice. IEEE Communications Magazine, Vol. 32, No. 9, Sep. 1994, S. 40-48

[ScEn00] Schulze, G.; Enzmann, M.: Datenschutz im Internet. In: Der GMD-Spiegel, Jan. 2000, S.42-44

[Sie01] Beschreibung Platform for Privacy Preferences Project. Mai 2001, http://www.infoserversecurity.org/p3p.php

[SoCr98] Soltysiak, S. J.; Crabtree I. B.: Knowing Me, Knowing You. Practical Issues in the Personalisation of Agent Technology. In: Proc. Third Intl. Conf. in the Practical Applications of Agents and Multi-Agent Technology (PAAM-98), Mar. 1998

[StWu98] Stiemerling, O.; Wulf, V.: Beyond ”Yes or No” – Extending Access Control in Groupware with Negotiation and Awareness. In: Proceedings of COOP ’98, Cannes, France, 1998

[SNS98] Steiner, J.G.; Neuman, C.; Schiller, J.I.: Kerberos: An Authentication Service for Open Network Systems. In: USENIX Conference Proceedings, Winter 1988.

[Sur02] Sury, U.: Biometrische Verfahren und Recht. Informatik Spektrum, Band 25, Heft 4, Aug. 2002, S. 322-324

[Tät00] 22. Tätigkeitsbericht des Unabhängigen Landeszentrums für Datenschutz Schleswig-Holstein. Landtagsdrucksache 15/10, Apr. 2000

[ThSt00] Thiel, U.; Stein, A.: Intelligent E-Commerce with Guiding Agents based on Personalized Interaction Tools. In: Stanford-Smith, B.; Kidd, P.T. (Hrsg.): E-business: Key Issues, Applications and Technologies, IOS Press, Amsterdam, NL, S. 760-766

[Tur96] Turkle, S.: Who Am We? Wired Magezine, Issue 4.01, Jan. 1996

[Wag01] Wagner, M.: Shared Directories Need Shared Control. Information Week, Jul. 2001, http://www.internetweek.com/newslead01/lead072601.htm

[WaBr1890] Warren, S.D.; Brandeis, L.D.: The Right to Privacy. Harvard Law Review 193, 196, 1890

[Wes67] Westin, A.: Privacy and Freedom. New York, 1967

[WYS+02] Winslett, M.; Yu, T.; Seamons, K.; Hess, A.; Jacobsen, J., Jarvis, R.; Smith, B.; Yu, L.: Negotiating Trust in the Web. IEEE Internet Computing, Vol. 6, Issue 6, Nov. 2002

[Wir99] Wirtz, B.: Biometrische Verfahren – Überblick, Evaluierung und aktuelle Themen. In: DuD, Vieweg, 3/99, S. 129-134

[Wör01] Wörndl, W.: Privatheit und Zugriffskontrolle bei Agenten-basierter Verwaltung von Benutzerprofilen. Tech. Rep. TUM-I0106, Institut für Informatik, TU München, Nov. 2001

[Woe02] Woerndl, W.: Using P3P to Negotiate Access Rights To User Profiles. Workshop on the Future of P3P, Washington DC, Nov. 2002

[WöKo02] Wörndl, W.; Koch, M.: Privatheit bei Verwaltung von Benutzerprofilen. Proc. Mensch und Computer 2002, Teubner, Hamburg, Sep. 2002

[WoPf00] Wolf, G.; Pfitzmann, A.: Charakteristika von Schutzzielen und Konsequenzen für Benutzungsschnittstellen. In: Informatik Spektrum, Vol. 23, No. 3, Jun. 2000, S. 173-191

[XEnc02] XML Encryption Syntax and Processing. W3C Recommendation, Dec. 2002, http://www.w3.org/TR/xmlenc-core/

[XML00] Extensible Markup Language (XML) 1.0. W3C Recommendation (2nd Edition), Oct. 2000, http://www.w3.org/TR/REC-xml

[XPa99] XML Path Language (XPath) 1.0. W3C Recommendation, Nov. 1999, http://www.w3.org/TR/xpath

[XSig02] XML Signature Syntax and Processing. W3C Recommendation, Feb. 2002, http://www.w3.org/TR/xmldsig-core/

[Xrml00] XrML Specification Version 1.03., 2000, http://www.xrml.org/

[XSLT99] XSL Transformations (XSLT). W3C Recommendation, Nov. 1999, http://www.w3.org/TR/xslt

[Zeh02] Zehentner, J.: Privatheit bei Anwendungen für Identitätsmanagement im Internet. Diplomarbeit, TU München, 2002