Eine policybasierte Zugriffskontrollarchitektur für das Multi Service Internet

vom Fachbereich 18
der Technischen Universität Darmstadt
genehmigte Dissertation
zur Erlangung des Grades eines
Doktor-Ingenieur (Dr.-Ing.)

von

Diplom Wirtschaftsinformatiker Christoph Rensing
geboren am 4.7.1967 in Düsseldorf

Darmstadt 2003
Hochschulkennziffer D17

Vorsitzender:	Prof. Dr. Oskar von Stryk
Referent:	Prof. Dr. Ralf Steinmetz
Korreferent:	Prof. Dr. Burkhard Stiller
Tag der Einreichung:	06. Juni 2003
Tag der Disputation:	15. Juli 2003

Eidesstattliche Erklärung

Ich versichere hiermit an Eides statt, dass ich die vorliegende Arbeit allein und nur unter Verwendung der angegebenen Literatur verfasst habe.

Darmstadt, den 06. Juni 2003

Diplom Wirtschaftsinformatiker Christoph Rensing

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1: Einleitung

Das “Internet” war ursprünglich ein rein von öffentlichen Institutionen betrieben, genutzt und finanziertes Kommunikationsnetz. Es hat sich zu einem Netz entwickelt, in welchem privatwirtschaftliche Institutionen und Privatpersonen als Anbieter und Nutzer von Diensten auftreten. Das Internet kann heute als ein Markt mit verschiedensten Diensten angesehen werden, der auf offenen Protokollen und Technologien sowie vielfältigen Infrastrukturkomponenten basiert. Es ist das Multi Service Internet. Der Kunde hat die Möglichkeit, aus einem globalen Angebot verschiedener Dienstanbieter einzelne Dienste auszuwählen und diese nach seinen Bedürfnissen zu kombinieren. Er kann dabei schnell und problemlos zwischen Anbietern vergleichbarer Dienste wechseln.

Privatwirtschaftliche Institutionen, die zwischenzeitlich sowohl weite Teile der Internet-Infrastruktur betreiben als auch elektronische Inhalte und vielfältige Anwendungen über das Internet anbieten, wollen und müssen Einnahmen erzielen. Nur so können sie auf Basis ihrer Investitionen in die technologische Infrastruktur und unter Berücksichtigung der Kosten für die Bereitstellung der Dienste Gewinne erwirtschaften. Geschäftsmodelle der Dienstanbieter beschreiben u.a., welche Dienste angeboten und wie die Einnahmen erzielt werden sollen. Neben Ertragsmodellen, die Einnahmen auf Basis von Werbung vorsehen, gewinnen solche Modelle zunehmend an Bedeutung, die Einnahmen auf Basis von Transaktionen oder Abonnements realisieren [Bir02]. Die Nutzung der Dienste wird kostenpflichtig. Damit wird es notwendig, den Zugriff auf die Dienste zu kontrollieren und sie abzurechnen. Solange sowohl die Nutzung des Internets als reines Kommunikationsnetz als auch die Nutzung der im Internet angebotenen Anwendungen und Inhalte kostenfrei war, war eine Kontrolle des Zugriffs in der Regel nur dann notwendig, wenn über das Internet ein Zugriff auf vertrauliche Informationen oder andere private Ressourcen möglich war oder wenn die Dienste gegen verschiedenen Formen von böswilligen Angriffen zu sichern waren.

1.1 Motivation

Der Markt der Internet-Dienste zeichnet sich durch eine hohe Wettbewerbsintensität und eine hohe Angebotstransparenz aus. Die Dienste konkurrierender Anbieter sind häufig, u.a. beim Angebot von Zugängen zum Internet, vergleichbar oder nahezu identisch. Eine Differenzierung der Anbieter erfolgt in diesem Fall primär über den Preis. Anbieter von Diensten im Internet sind daher gezwungen, wollen sie auf diesem Markt wirtschaftlich überleben, schnell neue Dienste unter Nutzung neuer Technologien zu entwickeln oder bestehende Dienste zu neuen Mehrwertdiensten zu kombinieren und ihre Ertrags- und Geschäftsmodelle flexibel den Marktsituationen anzupassen [MS02].

Parallel zur Kommerzialisierung des Internets vollzieht sich eine weitere Entwicklung: Der Dienstnutzer wird zunehmend mobil und möchte an jedem Ort und zu jeder Zeit auf die Internet-Infrastruktur und die angebotenen Dienste möglichst in der gewohnten Qualität zugreifen [VVK00]. Neue Technologien im Bereich des Zugangs zur Internet-Infrastruktur wie Funktelefonnetze, Hot Spots oder Adhoc-Netze machen dies möglich.

Unter diesen zwei Aspekten, der Kommerzialisierung und der Mobilität, gewinnen die Zugriffskontrolle und die Abrechnung immer mehr an Bedeutung. Aufgrund der hohen Dynamik der technologischen und ökonomischen Entwicklungen besteht die Notwendigkeit, ein flexibel konfigurierbares und für verschiedene Dienste und Nutzungsszenarien einsetzbares Zugriffskontroll- und Abrechnungssystem bereitzustellen. Die im Rahmen der Arbeit entwickelte Zugriffskontrollarchitektur legt die konzeptionelle Basis für ein solches System.

Die heute existierenden Systeme entstanden mit der einsetzenden Kommerzialisierung des Internets primär in zwei Bereichen: Sie dienen zur Kontrolle und zur Abrechnung des Zugriffs auf die von privaten Anbietern bereitgestellten Infrastruktur-Komponenten des Internets und auf kostenpflichtige Inhalte und Dienste. Eine Kontrolle des Zugriff auf die Infrastruktur-Komponenten muss dort erfolgen, wo diese öffentlich zugänglich sind. Dabei handelte es sich zunächst vorzugsweise um sogenannte Einwahlknoten, die über eine Telefonverbindung angewählt werden können. Die für dieses Szenario entwickelte Form der Zugriffskontrolle wurde mit der Entwicklung neuer Technologien wie Funktelefonnetzen, öffentlichen lokalen Funknetzen und Mobile IP, bis heute angepasst und erweitert[CLG+02]. Eine Kontrolle des Zugriffs auf kommerziell angebotene Inhalte und Anwendungen und deren Abrechnung erfolgen zumeist durch die Serveranwendung, welche die Dienste bereitstellt. Die Zugriffskontrolle dient dann oftmals primär dazu den vorab registrierten Dienstnutzer zu identifizieren, um ihm die Dienstnutzung zurechnen und nachfolgend in Rechnung stellen zu können. Alternativ kann eine Prüfung der Kreditwürdigkeit bzw. Zahlungsfähigkeit des Dienstnutzers erfolgen. Dazu legt der Dienstnutzer z.B. die Daten seiner Kreditkarte vor oder verwendet elektronische Bezahlverfahren [Mül02]. Ein Anbieter unterschiedlicher Dienste muss, um den Zugriff auf diese kontrollieren und abrechnen zu können, verschiedene Systeme und Verfahren einsetzen. Zusätzlich müssen bei der Entwicklung neuer Dienste jeweils die Verfahren zu Zugriffskontrolle und Abrechnung berücksichtigt und als Teilfunktion der Anwendung realisiert werden.

Bestehende Systeme zur Zugriffskontrolle und Abrechnung unterstützen also weder die für Anbieter von Internet-Diensten erforderliche Flexibilität zur Anpassung an neue Dienste und zur Gestaltung von unterschiedlichen Geschäftsmodellen, noch die Kontrolle des Zugriffs auf die Internet-Infrastruktur unabhängig von der genutzten Technologie.

1.2 Vision und Ziele

Diese Unzulänglichkeiten bestehender Zugriffskontrollsysteme bildet den Ausgangspunkt der Arbeit. Die Vision besteht darin, dass ein Internet-Dienstanbieter, basierend auf den technischen Potentialen, die das Internet bietet, allein unter Berücksichtigung ökonomischer Kriterien und unabhängig von technischen Rahmenbedingungen, die sich durch die Notwendigkeit einer

Zugriffskontrolle und Abrechnung ergeben, neue Dienste entwickeln, bestehende Dienste zu neuen Diensten kombinieren und diese den Dienstnutzern anbieten kann. Zugleich kann er auf geänderte Marktbedingungen schnell und flexibel durch eine Änderung seiner Geschäftsmodelle reagieren. Dazu steht ihm ein flexibel konfigurierbares System zur Verfügung, welches die Zugriffskontrolle und Abrechnung nach den Vorgaben des Geschäftsmodells ausführt und die Dienstnutzung berechnet. Die Zugriffskontrolle erfolgt unabhängig vom Aufenthaltsort des Dienstnutzers und der von ihm verwendeten Zugangstechnologie.

Zielsetzung der Arbeit ist es, eine Architektur als konzeptionelle Basis eines neuartigen generischen Zugriffskontroll- und Abrechnungssystems zu entwickeln, welches dem Internet-Dienstanbieter die geforderte Flexibilität bietet, so dass er nur noch ein Zugriffskontroll- und Abrechnungssystem betreiben muss. Dabei werden in der Arbeit vier Teilziele verfolgt: Die Architektur soll:

- generisch sein hinsichtlich der zu kontrollierenden Dienste,
- generisch sein hinsichtlich der abbildbaren Geschäftsmodelle und
- generisch sein hinsichtlich der eingesetzten kryptographischen Verfahren sowie
- unabhängig vom Aufenthaltsort des Dienstnutzers einsetzbar sein.

Mittels eines einheitlichen Zugriffskontroll und Abrechnungssystems soll der Zugriff des Dienstnutzers

- auf die Internet-Infrastruktur und
- auf im Internet angebotene Anwendungen und Inhalte

kontrolliert und abgerechnet werden können. Das System muss flexibel konfigurierbar sein, so dass ein Internet-Dienstanbieter zur Realisierung verschiedener Geschäftsmodelle jeweils bestimmen kann, welche Kriterien für die Zugriffskontrollentscheidung berücksichtigt werden und wie die Abrechnung erfolgt. Die Zugriffskontrollarchitektur soll zudem einen Austausch von Komponenten zur Realisierung der Zugriffskontrolle ermöglichen, sofern sie auf kryptographischen Verfahren basieren. Dies ist vorteilhaft, da für sicher gehaltene kryptographische Verfahren von einem Tag auf den anderen als weniger sicher oder unsicher eingestuft werden können und somit durch neue Verfahren zu ersetzen sind.

1.3 Abgrenzung

Die Architektur beschränkt sich auf eine Kontrolle des Zugriffs und auf die Abrechnung von solchen Diensten, die für den Dienstnutzer einen ökonomischen Wert darstellen und denen ein Preis zugeordnet wird, wie z.B. einer Videokonferenz oder einem elektronischen Zeitungsartikel. Die Kontrolle des Zugriffs auf Dienste, die für den Dienstnutzer keinen eigenen bezifferbaren Wert darstellen, die zur Bereitstellung der oben genannten Dienste aber genutzt werden, wird nicht betrachtet. Zu solchen Diensten gehört z.B. der Transport einzelner IP-Pakete. Diese Dienste werden in der Regel nicht einzeln abgerechnet. Die Kontrolle des Zugriffs auf solche Dienste muss sehr schnell durchführbar und mit geringen Transaktionskosten verbunden sein.

Die Architektur soll außerdem nur den Zugriff auf Dienste kontrollieren, soweit der Zugriff in einer autorisierten Form erfolgt. Eine autorisierte Form liegt dann vor, wenn der Dienstnutzer eine

Anfrage an den Dienstanbieter mittels eines für diesen Dienst spezifizierten Protokolls stellt. Der Schutz der zur Diensterbringung genutzten Komponenten (Systemsicherheit) und der genutzten Kommunikationsverbindungen (Kommunikationssicherheit) sind nicht Gegenstand der Arbeit.

1.4 Eigener Ansatz und Vorgehen

Die möglichen Formen der Zugriffskontrolle werden innerhalb der Arbeit bestimmt und klassifiziert. Sie müssen alle von einem generischen Zugriffskontroll- und Abrechnungssystem, welches die Forderung nach einer Unterstützung möglichst vieler Geschäftmodelle der Dienstanbieter erfüllt, realisiert werden. Um eine integrierte Sichtweise auf die angebotenen Dienste und die Funktionen der Zugriffskontrolle und Abrechnung zu ermöglichen, wird ein Policy-Modell gebildet. Dieses Modell fasst zum einen die Komponenten von Geschäftsmodellen, zum anderen die Teilfunktionen der Zugriffskontrolle und der Abrechnung in einem Policy-Modell zusammen. Die einzelnen Elemente des Modells werden formal als Policies beschrieben.

Die Arbeit betrachtet Zugriffskontrolle und Abrechnung als eigenen Dienst und nicht als Teilfunktion der Realisierung der vom Dienstnutzer angefragten Dienste. Der Anbieter von Internet-Diensten fordert in einer Client-Rolle die Zugriffskontroll- und Abrechnungsdienste von einem eigenen Server-System an. In der Architektur erfolgt somit eine Separierung der Zugriffskontrolle und Abrechnung als Dienst von den eigentlichen Internet-Diensten.

Als weitere Methoden werden die Modularisierung von Funktionen und ein policybasiertes Management des Systems angewandt: Alle in einem generischen System notwendigen Formen von Zugriffskontroll- und Abrechnungsdiensten werden in ihre Teilfunktionen zerlegt, die von jeweils einem eigenen Modul realisiert werden. Dies erlaubt es dem Dienstanbieter, in Abhängigkeit von seinem Geschäftsmodell zu bestimmen, welche Teilfunktionen zur Kontrolle des angefragten Dienstes ausgeführt werden sollen und wie der Dienst abzurechnen ist. Der Dienstanbieter spezifiziert dies mit Hilfe von Policies, die beim Zugriffskontroll- und Abrechnungs-Server hinterlegt sind. Die Policies werden jeweils zum Zeitpunkt einer Dienstanfrage dynamisch ausgewertet und bestimmen die vom Server auszuführenden Funktionen. Das gesamte Zugriffskontroll- und Abrechnungssystem wird so mit Hilfe von Policies flexibel konfiguriert.

1.5 Aufbau der Arbeit

Ein realistisches Anwendungsszenario, welches verschiedene Internet-Dienste umfasst, veranschaulicht in Kapitel 2 das Problemfeld der Arbeit. Weiterhin werden in diesem Kapitel die grundlegenden, für das Verständnis der Arbeit notwendigen Begriffe anhand eines eigenen Internet-Dienstmodells definiert.

Die verschiedenen Teilfunktionen der Zugriffskontrolle und unterschiedliche Verfahren zu ihrer Realisierung werden in Kapitel 3 vorgestellt. Existierende Zugriffskontrollsysteme für verschiedene Klassen von Diensten werden anhand einzelner Dienste des Anwendungsszenarios erörtert und klassifiziert.

Kapitel 4 untersucht die Abhängigkeiten zwischen möglichen Ausprägungen eine Geschäftsmodells und den Formen der Zugriffskontrolle und Abrechnungsfunktionen. Dazu werden ein Policy-Modell und XML-basierte Sprachen zur Spezifikation der verschiedenen Policies definiert.

In Kapitel 5 wird die neue Architektur, A^X-Architektur genannt, vorgestellt. Zunächst wird begründet, welche Konzepte beim Design der Architektur verwendet wurden. Aus diesen Konzepten definiert sich ein abstraktes Bild der Architektur und deren Funktionsweise. Dieses abstrakte Bild wird in der eigentlichen Beschreibung der generischen policybasierten Architektur konkretisiert. Dazu werden die Komponenten der Architektur und ihre Funktionen genau analysiert, die Systemschnittstellen betrachtet und Datenobjekte und Protokollnachrichten definiert. Die Funktionsweise des Zugriffskontroll- und Abrechnungssystems wird zum Abschluß detailliert erklärt.

In Kapitel 6 wird die Architektur bewertet. Dazu wird die Nutzung der Architektur in repräsentativen Anwendungsfällen erläutert und damit die Erfüllung der funktionalen Anforderungen in Vergleich zu existierenden Systemen überprüft. Außerdem werden die allgemeinen Anforderungen an Softwaresysteme validiert, wobei ein Vergleich der Leistungsfähigkeit der Architektur mit existierenden Systemen im Mittelpunkt steht.

Kapitel 7 fasst die wichtigsten Ergebnisse der Arbeit zusammen und beschreibt mögliche weiterführende Arbeiten.

Kapitel 2: Internet-Dienste und Aufgaben eines Anbieters von Internet-Diensten

Die Aufgabe eines generischen Zugriffskontroll- und Abrechnungssystems für Dienstanbieter im Multi Service Internet besteht in der Kontrolle des Zugriffs auf verschiedene, im Internet angebotene Dienste sowie deren Abrechnung. Bevor die in der Arbeit für ein solches System entwickelte Architektur vorgestellt wird, ist zunächst zu erläutern, welches Verständnis von Dienst der Arbeit zugrunde liegt, welche Internet-Dienste betrachtet werden und welche Funktionen ein Dienstanbieter innerhalb der Zugriffskontrolle und Abrechnung durchführen muss. Diese Erläuterungen erfolgen anhand eines vorab definierten Internet-Dienstmodells. Zum Abschluss des Kapitels wird das Paradigma des policybasierten Netzwerkmanagements als weitere Grundlage zum Verständnis der Arbeit erläutert. Es stellt eine wichtige konzeptionelle Basis der generischen Zugriffskontrollarchitektur dar. Das Kapitel beginnt mit der Beschreibung eines Anwendungsszenarios und mehrerer Anwendungsfälle, die das Problemfeld der Arbeit illustrieren und im Weiteren mehrfach zur Veranschaulichung genutzt werden.

2.1 Anwendungsszenario und Anwendungsfälle

Das Anwendungsszenario beschreibt die an den verschiedenen Anwendungsfällen beteiligten Organisationen und die zur Kommunikation innerhalb und zwischen den Organisationen genutzte Netzinfrastruktur. Es ist in Abbildung 1 dargestellt.

Die Wirtschaftsprüfungskanzlei Huber & Partner hat ihren Hauptsitz in Köln und Außenstellen in Dortmund und Münster. Innerhalb der Standorte existiert ein lokales Intranet auf Basis eines geswitchten Netzes sowie eines lokalen Funk-Netzes. Die Anbindung der lokalen Netze an das Internet erfolgt über eine Standleitung zu einem regionalen Anbieter NRW-On. Weiterhin besitzen die Mitarbeiter der Kanzlei ein Laptop und ein GSM-Mobiltelefon sowie einen entsprechenden Anbietervertrag mit T-Mobile. Der Web-Auftritt der Kanzlei ist bei der WWW-OK gehostet, die auch einen Mail-Server für die Kanzlei betreibt. Die Firma Recht&Urteil GmbH bietet über einen Web-Server aktuelle Gesetzestexte und Urteile in Steuerfragen an.

Die einzelnen Anwendungsfälle beschreiben die Nutzung der von den im Anwendungsszenario beschriebenen Organisationen angebotenen Dienste.

1. Anwendungsfall. *Herr Grimm, Mitarbeiter der Kanzlei Huber & Partner, sitzt in seinem Büro im Hauptsitz in Köln an seinem PC. Für eine Bearbeitung der Mandantendaten mit dem auf dem Intranet-Server eingerichteten Programm Rechnungswesen-Deluxe, muss sich seine Smartcard*

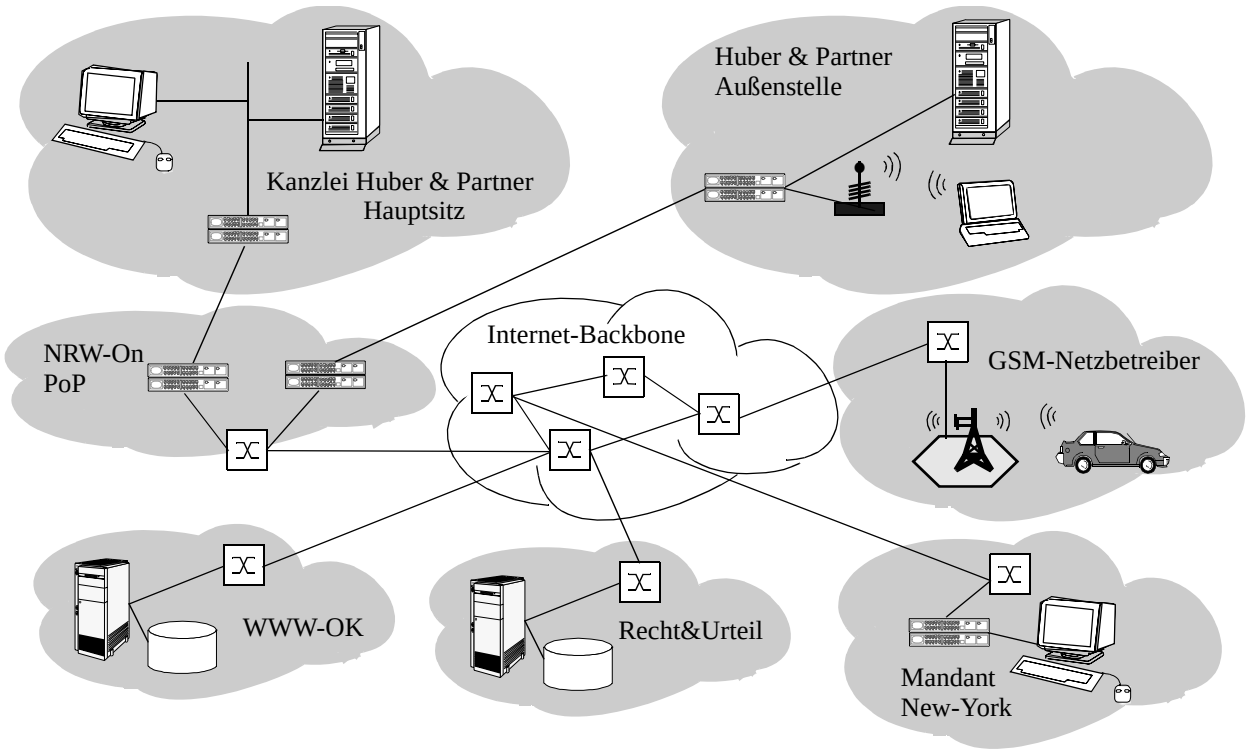

Abbildung 1: Anwendungsszenario

im Smartcard-Leser seines PCs befinden und über die Eingabe einer persönlichen Nummer (PIN) freigegeben sein.

2. Anwendungsfall. *Herr Grimm arbeitet in der Außenstelle in Dortmund an seinem Laptop, das mit einer Funknetzkarte ausgestattet ist. Um eine Verbindung zum Netz zu erhalten, kann er seine, auch im Hauptsitz gültige Benutzerkennung und sein Passwort nutzen.*

3. Anwendungsfall. *Herr Grimm ist zu einem Mandanten in Belgien unterwegs. Um noch schnell seine E-Mails zu lesen, nutzt er sein GSM-Handy über das Netz des belgischen Anbieters Belgacom. Belgacom besitzt ein Roaming-Abkommen mit T-Mobile und bietet seinen GSM-Kunden auch einen Call-by-Call Internet-Einwahlknoten an. Um die E-Mails abzurufen, muss er in der Web-Mail Anwendung von WWW-OK seine Benutzerkennung und sein Passwort eingeben.*

4. Anwendungsfall. *Herr Grimm möchte den privaten Bereich des Web-Auftritts der Kanzlei um ein Fotoalbum vom letzten Betriebsausflug erweitern. Die WWW-OK bietet dazu eine webbasierte Anwendung an. Herr Grimm muss sich nur mit einer Administratoren-Benutzerkennung und Passwort bei WWW-OK anmelden, die Anwendung auswählen und die digitalen Bilder auf den Rechner von WWW-OK laden. Die Bilder werden dann durch einen eigenen Webservice komprimiert und zusätzlich wird eine HTML-Übersichtsseite mit Thumbnails aller Bilder erstellt.*

5. Anwendungsfall. *Im Steuerwesen ändert sich regelmäßig etwas. Die aktuellsten Gesetzestexte und Urteile stellt die Recht&Urteil GmbH im Internet zur Verfügung. Herr Grimm recherchiert regelmäßig in der kostenfrei zugänglichen Datenbank. Um auf die gesamten Urteilstexte zugreifen zu können, muss er wiederum eine Benutzerkennung und ein Passwort oder seine Kreditkar-*

teninformationen eingeben. Die jeweiligen Gebühren für die Texte werden im ersten Fall von einem Guthabenkonto der Kanzlei abgebucht.

6. Anwendungsfall. *Ein Mandant der Kanzlei Huber&Partner, mit einer Niederlassung in den USA, hat große Schwierigkeiten mit den Finanzbehörden. Eine Lösung für die Probleme soll im Rahmen einer Videokonferenz zwischen dem Büro in Köln, dem Hauptsitz des Mandanten in Dortmund und der Niederlassung in New-York diskutiert werden. NRW-On bietet dafür eine IP-basierte Videokonferenz an und reserviert für die Verbindung über den Atlantik eine ausreichende Übertragungskapazität bei einem Backbone-Anbieter.*

Die Anwendungsfälle beschreiben verschiedene Dienste, in denen mannigfaltige Mechanismen zur Kontrolle des Zugriffs sowie unterschiedliche Gebührenmodelle und Verfahren zur Bezahlung der teilweise kostenpflichtigen Dienste genutzt werden. Die Realisierung der Zugriffskontrolle in den einzelnen Anwendungsfällen wird bei der Untersuchung bestehender Architekturen und Verfahren der Zugriffskontrolle im Folgenden genauer betrachtet. Zusätzlich werden die identischen Anwendungsfälle zur Beschreibung der Funktionsweise der generischen Zugriffskontrollarchitektur und zum Vergleich mit den existierenden Lösungen genutzt.

2.2 Internet-Dienste im Internet-Dienstmodell

Als Grundlage zum Verständnis der weiteren Arbeit ist die Definition grundlegender Begriffe notwendig. Der zentrale Begriff der Arbeit ist der des *Dienstes*. Da das Verständnis von Dienst innerhalb der Arbeit von dem in Modellen für Kommunikationsnetze üblichen abweicht, werden Dienst und weitere Begriffe nachfolgend definiert und umfassen anhand eines Internet-Dienstmodells und mehreren Beispielen erläutert.

Der Begriff *Dienst* wird in verschiedensten Bereichen verwendet. Eine allgemeine Definition von *Dienst* ist in Meyers Lexikon [Gri92] wie folgt gegeben:

> ***Dienst**, allg. die Erfüllung von Pflichten; im religiösen Bereich der Gottes-D.; im karitativen Bereich der D. am Nächsten; im berufl. Bereich die Verrichtung der zu erbringenden Leistung.*

Im Zusammenhang der Arbeit soll im weiteren die folgenden Definitionen, angelehnt an [Ste00], genutzt werden.

> *Ein **Dienst** definiert eine Menge von in sich zusammenhängenden Funktionen, die einem Nutzer von einem Anbieter angeboten und erbracht werden.*
>
> *Ein **Internet-Dienst** ist ein Dienst, der über das Internet unter Nutzung von Internet-Protokollen angefragt und erbracht wird.*
>
> *Ein **Dienstnutzer** ist derjenige, der einen Dienst in Anspruch nimmt.*
>
> *Ein **Internet-Dienstanbieter** stellt den Dienstnutzern Internet-Dienste zur Verfügung.*
>
> *Mit **Dienstinfrastruktur** werden die technischen, personellen und organisatorischen Ressourcen bezeichnet, die ein Dienstanbieter betreibt um mit deren Hilfe einen Dienst zu erbringen.*

*Das **Multi Service Internet** bezeichnet die Menge der Internet-Dienste, die von Internet-Dienstanbietern angeboten werden, sowie die dafür notwendige Dienstinfrastruktur der Internet-Dienstanbieter.*

Da es sich bei Internet-Diensten um elektronische Dienste handelt, muss ein Dienstnutzer über ein rechnerbasiertes Gerät verfügen, welches ihm die Nutzung der elektronischen Dienste ermöglicht. Dieses elektronische Gerät und die Dienstinfrastruktur sind technische Systeme. Dienstanbieter und Dienstnutzer sind in Abgrenzung dazu Rollen.

Die verschiedenen Definitionen lassen sich in einem einfachen Modell zusammenfassen, wie in Abbildung 2 dargestellt.

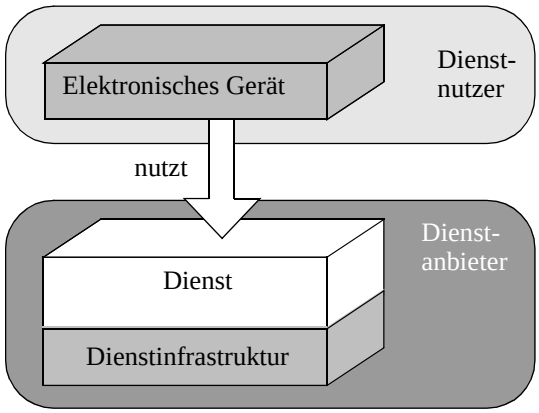

Abbildung 2: Einfaches Dienstmodell

Zur Erbringung der Dienste kann ein Dienstanbieter, bzw. genauer dessen Dienstinfrastruktur, wiederum die Dienste anderer Dienstanbieter nutzen. In Abbildung 3 nutzt A die Dienste von B, die dessen Dienstinfrastruktur unter Nutzung der Dienste von C erbringt. B nimmt in diesem Fall die Rolle des Dienstnutzers und Dienstanbieters an.

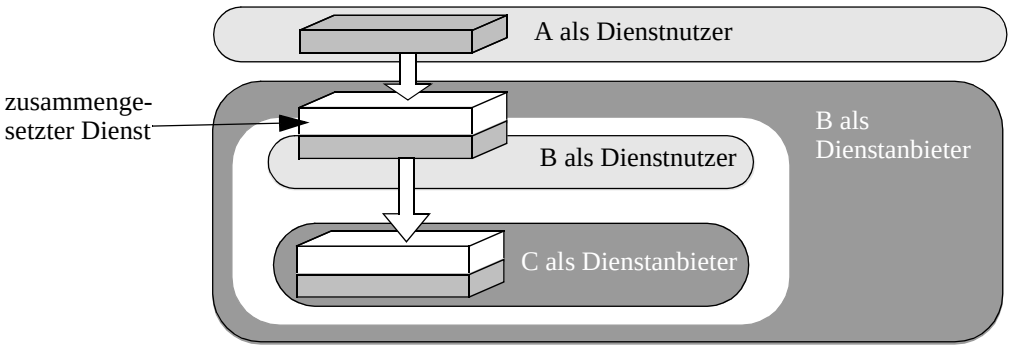

Abbildung 3: Erweiteres Dienstmodell

Werden Dienste unter Nutzung der Dienste eines weiteren Dienstanbieters erbracht, so kann das für den übergeordneten Dienstnutzer transparent geschehen, d.h. dieser erkennt nicht, dass zur Diensterbringung ein Dienst eines anderen Anbieters genutzt wird.

*Ein **zusammengesetzter Dienst**, ist ein Dienst, der sich aus einzelnen Diensten mehrerer Dienstanbieter transparent zusammensetzt.*

In Abbildung 3 erbringt B einen zusammengesetzten Dienst für A. Dienstnutzer, die keine Dienste anbieten, lassen sich dann wie folgt von allgemeinen Dienstnutzern abgrenzen.

> *Ein **Endnutzer** ist ein Dienstnutzer, der keine eigenen Dienste anbietet und nie die Rolle eines Dienstanbieters einnimmt.*

Das Dienstmodell erlaubt es, dass Dienste eines Dienstanbieters, sowohl als Bestandteil eines zusammengesetzten Dienstes eines anderen Dienstanbieters, als auch als eigener Dienst angeboten werden. Dies ist in Abbildung 4 für den Dienst von C dargestellt, der von B zur Erbringung eines zusammengesetzten Dienstes und von D unmittelbar genutzt wird.

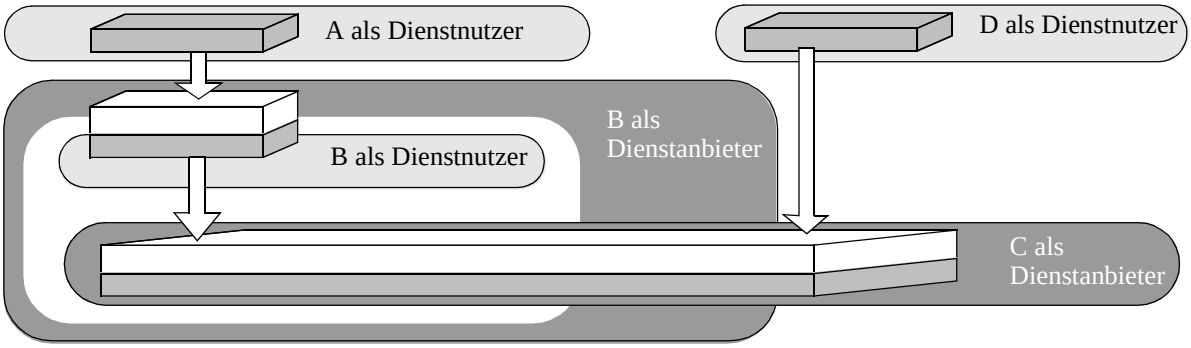

Abbildung 4: Nutzung eines Dienstes als zusammengesetzter und einzelner Dienst

Innerhalb des Internets finden sich diese Strukturen und die Nutzung zusammengesetzter Dienste sehr häufig. Das gilt auch für die beschriebenen Anwendungsfälle. Das Dienstmodell des Anwendungsfalls 3, in welchem Herr Grimm mittels einer Verbindung über sein GSM-Mobiltelefon und einen Call-by-Call Internetzugang die E-Mails über einen Web-Mail Dienst von WWW-OK bearbeitet, zeigt Abbildung 5.

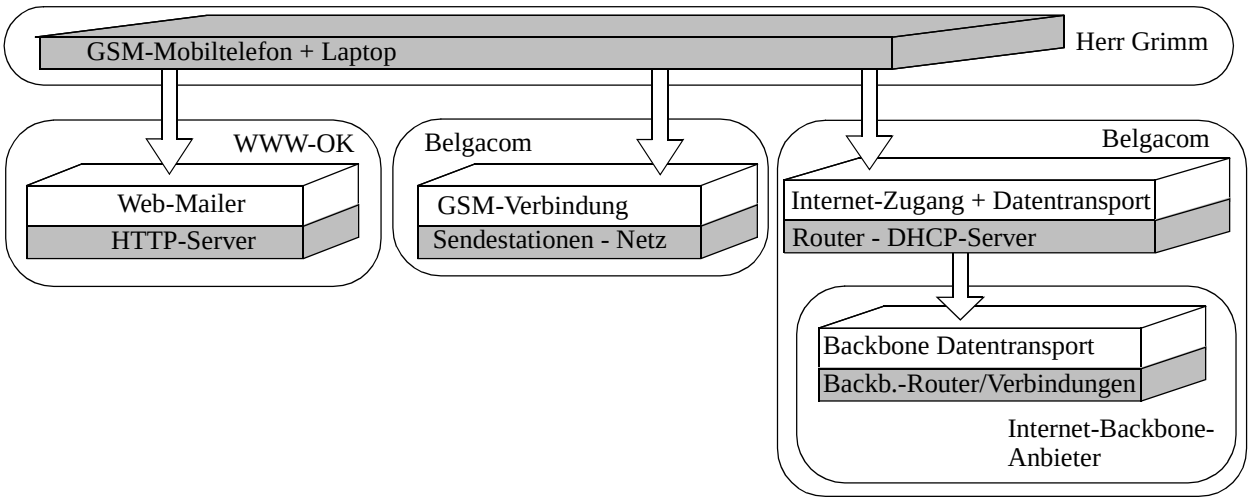

Abbildung 5: Internet-Dienstmodell im Anwendungsfall 3

Herr Grimm verwendet dabei einen zusammengesetzten Dienst, bestehend aus einem Zugang zum Internet und den Datentransport im Internet. Dabei handelt es sich um eine Vereinfachung, denn im Internet-Backbone können durchaus mehrere Dienstanbieter an der Erbringung des Datentransports beteiligt sein.

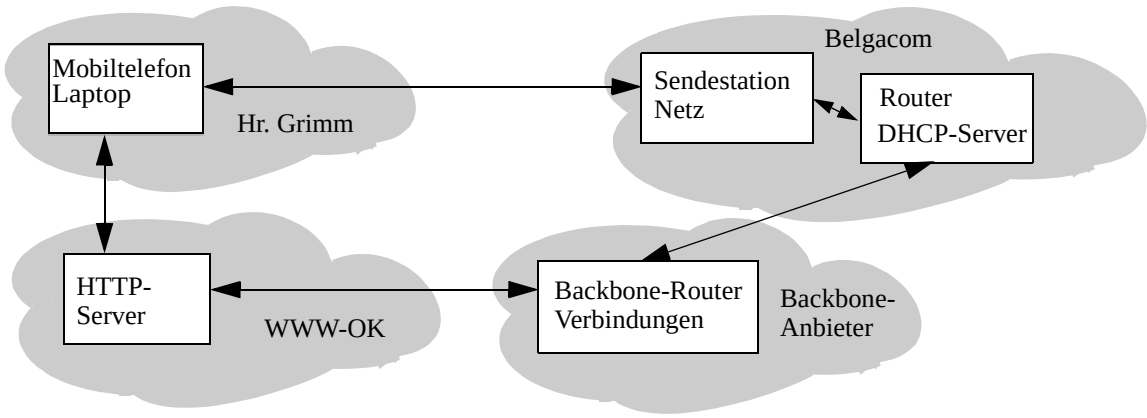

Abbildung 6: Organsisationsmodell im Anwendungsfall 3

Das Internet-Dienstmodell wird überlagert von einem Organisationsmodell, in welchem eine Zuordnung der Dienstanbieter, Dienste, Dienstnutzer und der notwendigen Dienstinfrastruktur zu Organisationen erfolgt. Dieses ist wiederum für den Anwendungsfall 3 in Abbildung 6 illustriert.

Weiterhin wir deutlich, dass es im Internet Dienste gibt, die von einem Endnutzer nie direkt genutzt werden, wie der des Backbone-Datentransports.

Das Dienstmodell beschreibt die im Internet auftretenden Dienstbeziehungen. Es wird daher nachfolgend Internet-Dienstmodell genannt. Das Internet-Dienstmodell erlaubt eine ökonomische Sicht auf das Internet, indem es die Wertschöpfung durch einzelne Dienstanbieter beschreibt. Alle Dienste können unabhängig von einander angeboten, realisiert und berechnet werden. Es unterscheidet sich damit von geschichteten Kommunikationsmodellen, wie dem ISO Open Systems Interconnection (OSI)-Referenzmodell [ISO94] oder dem Internet-Modell [Tan02], welchen ein technologischer Blickwinkel zugrunde liegt. In diesen Modellen nutzt der Endnutzer immer Dienste der obersten Ebene, der Anwendungsschicht.

Mit der zuvor vorgenommenen Definition von Internet-Diensten erfolgt eine Abgrenzung zu den weithin verwendeten Begriffen e-Commerce und Internet-Ökonomie, die hier nochmals explizit herausgestellt werden soll. Zum e-Commerce zählen auch solche Dienste, die das Internet nur für einzelne betriebswirtschaftliche Funktionen, z.B. als Marketingkanal oder Bestellweg nutzen, aber physikalische Güter als primären Wert für den Kunden anbieten [CSW97]. Die Internet-Ökonomie umfasst zudem noch Anbieter von Technologien für das Internet, wie Hersteller von Internet-Software und Netzwerkkomponenten sowie von Dienstleistungen, wie Web-Design oder Programmierung [CW00].

2.3 Klassifikation von Internet-Diensten

Um in den folgenden Kapiteln die Anforderungen an eine generische Zugriffskontrollarchitektur unter technischen Aspekten systematisch bestimmen zu können, sollen die Internet-Dienste klassifiziert werden. Anhand dieser Einteilung werden zudem bestehende Lösungen zur Realisierung einer Zugriffskontrolle für Internet-Dienste eingeordnet.

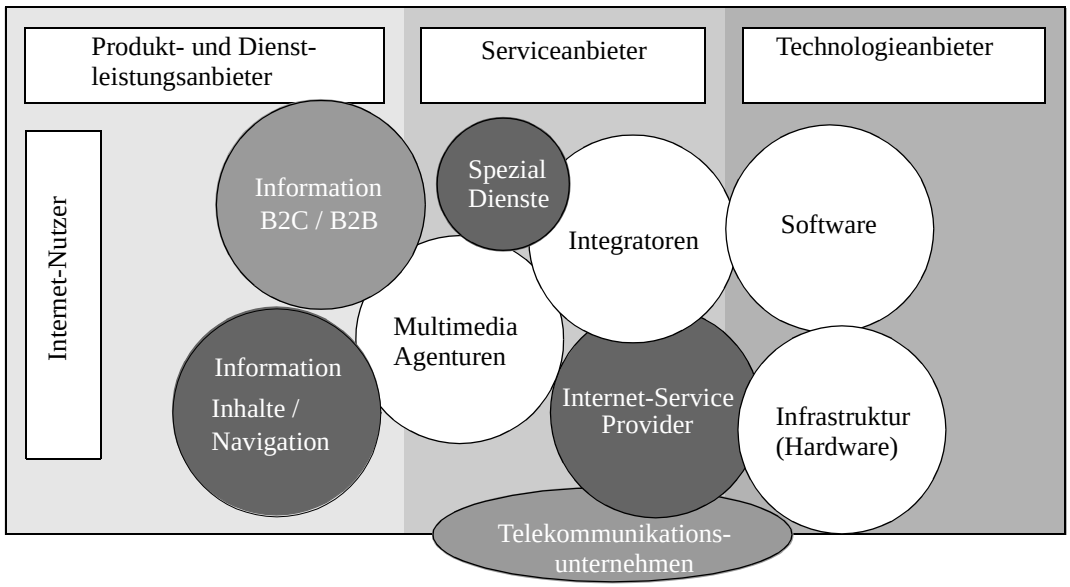

Abbildung 7: Segmentierung der Internet-Ökonomie in Anlehnung an [Kra00]

In der Literatur finden sich verschiedene Einteilungen für die umfassenderen Bereiche Internet-Ökonomie und e-Commerce. Zumeist erfolgt sie anhand der Geschäftsschwerpunkte von Unternehmen. In [Kra00] werden, wie auch in Abbildung 7 gezeigt, die Hauptgruppen der *Technologieanbieter*, *Serviceanbieter* sowie *Produkt- und Dienstleistungsanbieter* unterschieden.

Eine andere Segmentierung der Internet-Ökonomie in Form eines Schichtenmodells wird in [BWY00] und [BW01] vorgenommen. Auf der untersten Ebene *Internet-Infrastruktur* finden sich die Anbieter von Hardware, insbesondere Netzwerkkomponenten, Telekommunikationsanbieter, Backbone-Anbieter und Anbieter von Zugängen zum Internet. Die zweite Ebene *Anwendungen* umfasst Anbieter von Softwareprodukten für das Internet und zur Erstellung von multimedialen Inhalten. Zur dritten Ebene zählen beispielsweise Portale oder verschiedene Verzeichnisdiente. Auf der obersten Ebene sind Online Händler, Anbieter von digitalen Gütern aber auch Anbieter von nicht digitalen Gütern zu finden.

[Met00] unterscheidet drei Gruppen: die *Inhaltsanbieter*, die digitale Güter anbieten, die *Umgebungsanbieter*, welche die Arbeitsumgebung des Internet-Nutzers z.B. in Form eines Browsers gestalten und die *Infrastrukturbetreiber*. Zu den Infrastrukturbetreibern zählt er, neben den Internet Service Providern und den Business Service Providern, auch die Telekommunikationsunternehmen.

Die verschiedenen existierenden Einteilungen sind für die weitere Arbeit ungeeignet, da sie Dienste einbeziehen, die aus der verwendeten Definition von Internet-Diensten fallen. In Abbildung 7 sind die Internet-Dienste im Sinne der Arbeit dunkel dargestellt. Telekommunikationsunternehmen und Informationsanbieter für Business-to-Business (B2B) oder Business-to-Customer (B2C) Dienste zählen nur teilweise dazu. Ein weiterer Nachteil der vorgestellten Einteilungen liegt in der starken Orientierung an ökonomischen Aspekten, die für die Untersuchung der technischen Anforderungen an die Zugriffskontrolle ungeeignet ist.

Daher wird für die weitere Arbeit eine eigene Unterteilung der Dienste in fünf verschiedene Dienstklassen definiert. Einer Dienstklasse werden solche Dienste zugeordnet, die sich dadurch auszeichnen, dass zu Ihrer Realisierung und Bereitstellung vergleichbare Technologien verwendet werden und dass sie zugleich ähnliche technische Anforderungen an eine Zugriffskontrollarchitektur stellen.

> ***Verbindungsdienste*** *stellen eine physikalische und logische Verbindung zwischen dem Gerät eines Dienstnutzers und dem eines Dienstanbieters oder zwischen den Geräten zweier Dienstnutzer zur Verfügung.*

Bei den Verbindungsdiensten handelt es sich in der Regel nicht um Internet-Dienste im definierten Sinne, da sie nicht über ein Internet-Protokoll vom Dienstnutzer angefragt werden, auch wenn zu ihrer internen Realisierung Internet-Protokolle genutzt werden. In zukünftigen Mobiltelefonnetzen der dritten Generation muss dieses nicht mehr gelten, da hier zunehmend Internet-Protokolle Verwendung finden [GPO03]. Beispiele für Verbindungsdienste sind Telefonverbindungen im Fest- und Mobilnetz zwischen zwei Endgeräten von Endnutzern oder zwischen dem Endgerät eines Endnutzer und dem eines Anbieters von Internet-Diensten oder die Errichtung von Standleitungen. Der Nutzer eines Verbindungsdienstes muss nicht notwendigerweise Endnutzer, sondern kann auch zugleich Anbieter von Internet-Diensten sein. Ein Verbindungsdienst ist somit auch eine Satellitenverbindung oder ein Transatlantikkabel zwischen zwei Routern von Internet-Dienstanbietern.

> ***Internet-Zugangs- und Basistransportdienste*** *stellen einen Router im Internet sowie einen Best-Effort Transport von Daten zwischen Rechnern innerhalb des Internets zur Verfügung.*

Internet-Zugangs- und Basistranportdienste[1] verwenden zu Ihrer Realisierung immer Verbindungsdienste. Außerdem muss ein Dienstnutzer zur Nutzung eines Internet-Zugangs- und Basistransportdienst auch zeitgleich einen Verbindungsdienst nutzen. Diesen kann er von einem Verbindungsdienstanbieter einer anderen Organisation oder vom Internet-Zugangsanbieter beziehen. Die Definition umfasst auch den Transport von Daten im Internet-Backbone. Der Zugang zu einem Router im Internet und der Datentransport wir dann nicht von einem Endnutzer sondern von einem anderen Anbieter genutzt.

> ***QoS-Transportdienste*** *realisieren einen Transport von Daten zwischen zwei oder mehreren Rechnern innerhalb des Internets, der bestimmte festgelegte Qualitätsparameter einhält.*
>
> ***Inhaltsdienste*** *bieten jedwede Art digitaler Informationen an, die bereits vor der Diensterbringung beim Anbieter gespeichert sind.*
>
> ***Anwendungsdienste*** *sind solche Internet-Dienste, die keiner der andern Gruppen zuzuordnen sind. Es handelt sich um Kommunikations- oder Verzeichnisdienste, transaktions- oder datenstromorientierte Dienste, Rechen- oder Speicherdienste und interaktive Dienste.*

1. Im folgenden werden Internet-Zugangs- und Basistransportdienste zumeist nur als Internet-Zugangsdienste bezeichnet. Sie umfassen aber auch dann einen Best-Effort Datentransport.

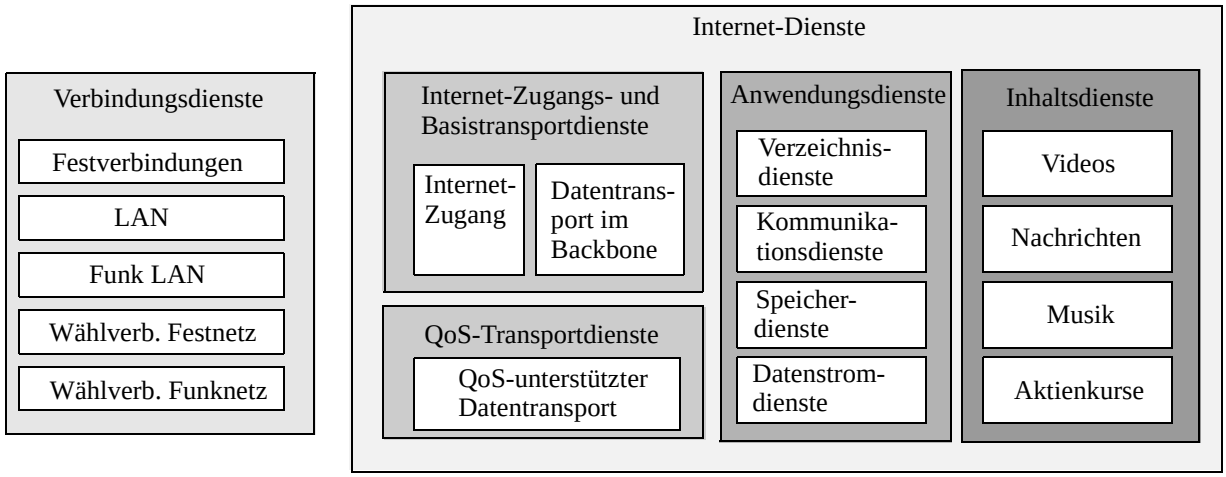

Abbildung 8: Beispiele für Endnutzerdienste

Für die Dienste dieser fünf Dienstklassen gilt übereinstimmend, dass sie einem Endnutzer einen direkten Mehrwert im Sinne einer Leistungserbringung bieten und er daher gegebenenfalls dazu bereit ist, für einen solchen Dienst Gebühren zu bezahlen.

> *Ein **Endnutzerdienst** ist ein Verbindungdienst, ein Internet-Zugangs- und Basistransportdienst, ein QoS-Transportdienst, ein Anwendungsdienst oder ein Inhaltsdienst. Ein Endnutzerdienst wird mit Hilfe der **Endnutzerdienstinfrastruktur** erbracht.*

Verschieden Beispiele für Dienste der einzelnen Dienstklassen sind in Abbildung 8 dargestellt.

Äquivalent zu den Dienstklassen können auch Dienstanbieter in fünf verschiedenen Gruppen, nämlich Verbindungsdienstanbieter, Internet-Zugangsanbieter, QoS-Transportdienstanbieter, Anwendungsdienstanbieter und Inhaltsdienstanbieter eingeteilt werden. Dabei ist zu beachten, dass Internet-Dienstanbieter oftmals nicht nur Dienste einer Dienstklasse, sondern mehrerer Dienstklassen oder auch zusammengesetzte Dienste, die Dienste mehrerer Dienstklassen umfassen, anbieten. Sie können dann nicht eindeutig einer Gruppe zugeordnet werden. Ein Anbieter von QoS-Transportdiensten wird in aller Regel auch einen Best-Effort Datentransport und somit Dienste der Klasse Internet-Zugang und Basistransport bereitstellen. Im Anwendungsszenario sind *T-Mobile* und *Belgacom* als Betreiber eines GSM-Netzes sowohl Anbieter von Verbindungsdiensten, als auch eines Internet-Zugangs- und Basistransportdienstes. *Recht&Urteil* bietet mit dem Such-Interface nach Gesetzestecten und Urteilen einen Anwendungsdienst und mit der Möglichkeit Urteilstexte herunterzuladen einen Inhaltsdienst an.

Anbieter klassischer Telekommunikationstechnologien ausserhalb des Internets erbringen in aller Regel nur zusammengesetzte Dienste, von der Anwendung oder dem Inhalt bis zur Verbindung. Telefongesellschaften beispielsweise bieten einen Anwendungsdienst, nämlich die Telefonie, an. Sie fassen diesen vertikal mit Verbindungs- und Transportdiensten, genannt Trägerdiensten, und historisch sogar mit der Bereitstellung von Endgeräten zusammen. Radio- und Fernsehgesellschaften bieten einen Inhaltsdienst, ihr Programm, an und verknüpften diesen Dienst ebenfalls mit untergeordneten Diensten zum Transport der Programme, abgesehen von den

privatisierten Kabelnetzen. Der reine Inhaltsdienst der Fernsehgesellschaft stellt für den Endnutzer, den Zuschauer, keinen Wert dar, so lange das Programm nicht übertragen wird.

Erst die offenen Protokolle und Technologien des Internets ermöglichen Anbietern, nur die Dienste einzelner Dienstklassen anzubieten, da diese trotzdem einen Mehrwert für den Endnutzer darstellen, wie im Internet-Dienstmodell beschrieben. Dies führte in der vergangenen Dekade dazu, dass eine Vielzahl von kommerziellen und nicht kommerziellen Anbietern heute in Konkurrenz zueinander ihre Dienste offerieren. Gleichzeitig können sich Anbieter von Verbindungsdiensten und von Internet-Zugangs- und Basistransportdiensten nur noch über den Preis von ihren Mitbewerbern abgrenzen, da der angebotene Dienst vergleichbar und nahezu identisch ist. Aus diesem Grund besteht für diese Anbieter wiederum die Notwendigkeit, verschiedene Dienste insbesondere auch mit Anwendungs- und Inhaltsdiensten zu kombinieren, um sich so von Konkurrenten zu differenzieren.

2.4 Zugriffskontrolle auf Internet-Dienste

Die in Kapitel 2.1 beschriebenen Anwendungsfälle veranschaulichen bereits die Vielfalt einer Zugriffskontrolle auf Internet-Dienste. So wird der Zugriff auf die Dienste mittels der Verwendung von Benutzerkennung und Passwort oder einer Smartcard kontrolliert. Beim kostenpflichtigen Download von Texten prüft Recht&Urteil vor der Diensterbringung, ob Herr Grimm bereits vorab bezahlt hat oder ob er kreditwürdig ist.

Allgemein wird eine Zugriffskontrolle in verschiedensten Umfeldern eingesetzt. Es gibt beispielsweise eine physikalische Zugriffskontrolle beim Einlaß von Personen in geschützte Gebäude oder beim Zugriff auf Daten und Rechner. Für Kommunikationsnetze definiert die ITU [ITU91] Zugriffskontrolle wie folgt:

> ***Zugriffskontrolle****: Die Vermeidung einer nicht autorisierten Nutzung einer Ressource und die Vermeidung der Nutzung eine Ressource in einer nicht autorisierten Art und Weise.*

Diese Definition wird für die vorliegende Arbeit auf Internet-Dienste als Ressource eingeschränkt. Auch der Schutz vor einer Nutzung der Ressourec in nicht autorisierter Art und Weise liegt nicht im Fokus der Arbeit:

> ***Zugriffskontrolle auf Internet-Dienste:*** *Die von einem Dienstanbieter durchgeführte Vermeidung der nicht autorisierten Nutzung von Internet-Diensten.*

Die Zugriffskontrolle muss durchgeführt werden, bevor ein Internet-Dienst erbracht wird. Der Nutzung eines Internet-Dienstes geht in vielen Fällen eine explizite Dienstanfrage durch den Dienstnutzer voraus. Die Dienstanfrage erfolgt oftmals, insbesondere bei Diensten mit einer zeitlichen Dauer, mittels eines sogenannten Signalisierungsprotokolls, wohingegen für die eigentliche Diensterbringung ein anderes Protokoll verwendet wird. Gegebenenfalls werden mehrere Nachrichten zur Aushandlung der Dienste oder zur Übermittlung weiterer Informationen ausgetauscht, bevor der Dienst erbracht wird. Die Zugriffskontrolle wird dann durch ein Signalisierungsnachricht des Dienstnutzers ausgelöst und mittels der Dienstinfrastruktur des Dienstanbieters realisiert, wie in Abbildung 9 dargestellt. Das Ende der Diensterbringung oder

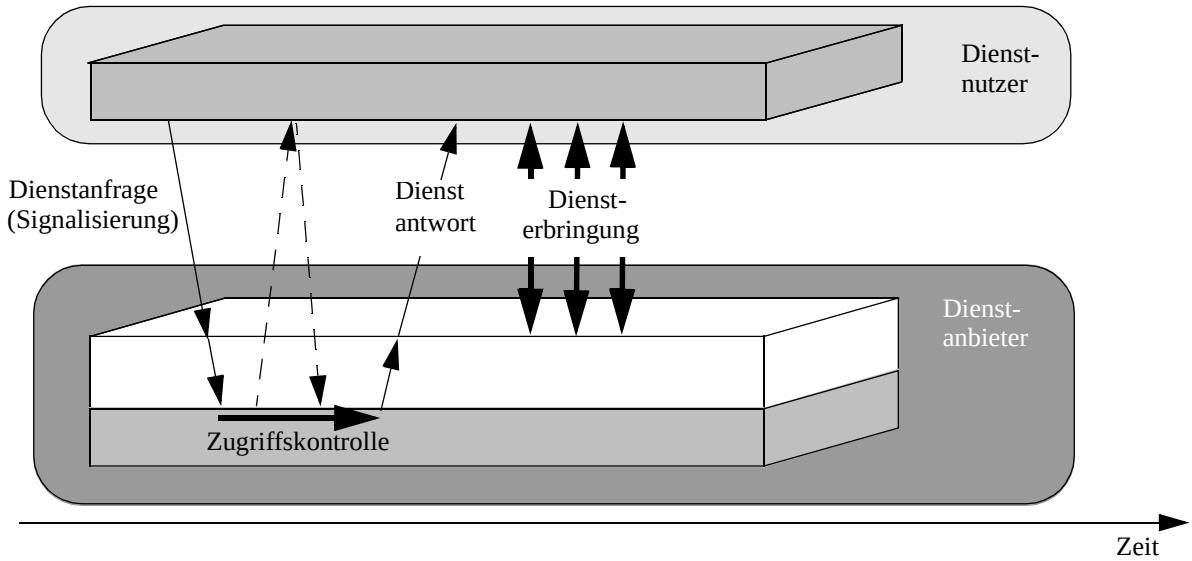

Abbildung 9: Zeitliche Lage der Zugriffskontrolle

Dienstnutzung wird zumeist wiederum mit Hilfe von Signalisierungsnachrichten des Dienstanbieters oder Dienstnutzers angezeigt. Über die bestätigte Dienstanfrage und das Dienstende wird eine Session bestimmt.

*Eine **Session** bezeichnet den Zeitraum zwischen Beginn und Ende einer Dienstnutzung.*

Ein Protokoll mit dem ein Internet-Dienst angefragt wird, wird von obigen Beispielen abweichen in vielen Fällen auch für die Diensterbringung selbst genutzt werden. Dies ist oftmals beim Zugriff auf Inhalte der Fall, die keine zeitliche Dauer besitzen. Ein Beispiel ist die Verwendung des Hypertext Transfer Protokolls (HTTP) zum Zugriff auf Objekte im World Wide Web (WWW). Mittels eines Requests wird ein Objekt angefordet, welches mit einem Response übermittelt wird. Jede einzelne Anfrage kann als eigener Internet-Dienst angesehen werden, für den jeweils eine Zugriffskontrolle durchzuführen ist. Das macht aus ökonomischen Gesichtspunkten nur Sinn, wenn dem einzelnen Objekt ein Wert zugewiesen werden kann. Mittels verschiedener technischer Verfahren werden daher in solchen Fällen oftmals mehrere der aufeinanderfolgenden Request- und Response-Nachrichten einander zugeordnet und zu einer Session zusammengefasst. Dann muss die Zugriffskontrolle nur bei Beginn der Session erfolgen. Beide Möglichkeiten sollen im Rahmen der Arbeit betrachtet werden.

Im Internet existieren zudem feingranulare Dienste, die vom Dienstnutzer nicht angefragt sondern unmittelbar genutzt werden. So stellt auch der Transport jedes einzelnen IP-Pakets zwischen zwei Netzknoten einen Dienst dar. Auch in solchen Fällen kann eine Zugriffskontrolle durchgeführt werden, beispielsweise durch Firewalls oder in QoS-Architekturen. Sie werden in der Arbeit nicht betrachtet, da sie zum einen keinen eigenen bezifferbaren Wert darstellen und zum anderen die Zugriffskontrolle sehr schnell und mit geringen Transaktionskosten verbunden erfolgen muss. Nach der gegebenen Definition handelt es sich dabei aber nicht um Internet-Dienste. Gleiches gilt auch für andere mögliche Formen der Interaktion zwischen dem Dienstnutzer und Dienstanbieter,

z.B. wenn der Dienstanbieter den Dienst als Callback-Dienst oder ereignisbasiert an den Dienstnutzer erbringt [FMB01].

Die Zugriffskontrolle auf Internet-Dienste findet an der Schnittstelle zwischen Dienstnutzer und Dienstanbieter statt, an der Dienste angeboten und genutzt werden, wie anhand des Internet-Dienstmodells in Abbildung 10 illustriert. Sie ist damit weiterhin abzugrenzen von der Sicherung der Kommunikation, der Sicherung des Systems und der internen Zugriffskontrolle. Mittels der Sicherung der Kommunikation zwischen Dienstnutzer und Dienstanbieter wird eine Nutzung eines Internet-Dienstes in nicht autorisierter Weise durch eine Angreifer im Internet vermieden. Sie stellt im Sinne der allgemeinen ITU-Definition auch eine Form der Zugriffskontrolle dar, ist aber nicht eigentlicher Gegenstand der Arbeit. Auch der Schutz der Dienstinfrastruktur selbst vor Angiffen, oftmals als Systemsicherheit bezeichnet, steht nicht im Fokus der Arbeit. Diese beiden Formen der Sicherheit werden nur konzeptionell bei der Definition der Architektur betrachtet. Die Kontrolle des Zugriffs auf Ressourcen der Dienstinfrastruktur, die der Dienstanbieter für die Erbringung der Dienste benötigt, die aber nicht eigene Internet-Dienste darstellen, ist in Abbildung 10 als interne Zugriffskontrolle bezeichnet. Sie ist ebenfalls nicht Hauptgesichtspunkt der Arbeit.

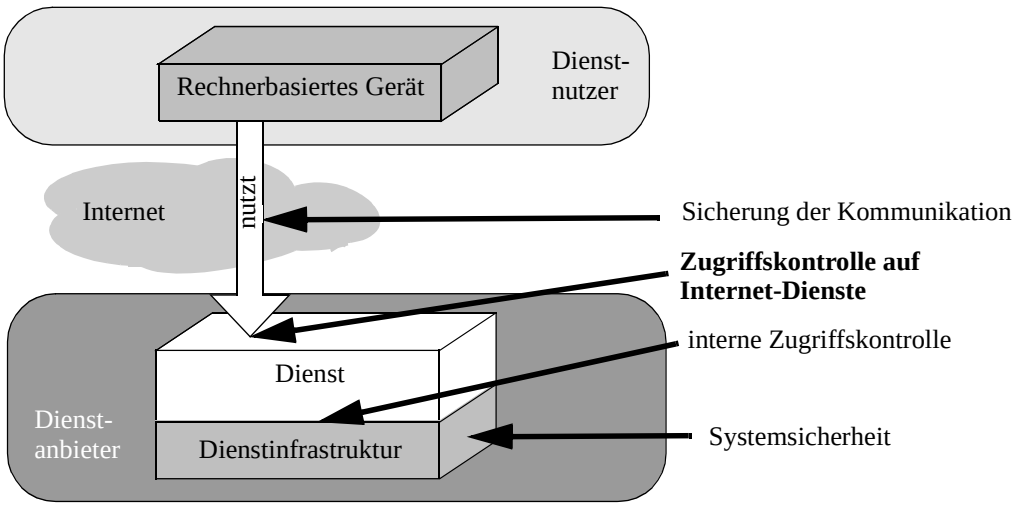

Abbildung 10: Zugriffskontrolle auf Internet-Dienste

Die Zugriffskontrolle auf Internet-Dienste kann sich aus mehreren Teilfunktionen zusammensetzen, verschiedene Verfahren zu ihrer Realisierung verwenden und anhand verschiedener Kriterien erfolgen, wie die Anwendungsfälle dies bereits illustriert haben. Eine Zugriffskontrolle bedeutet, wie aus der Definition hervorgeht, immer eine Autorisierung des Dienstnutzers durch den Anbieter. Grundsätzlich muss die Autorisierung nicht durch den Anbieter des angefragten Dienstes selbst erfolgen, sondern kann auch durch einen vertrauenswürdigen Dritten vorgenommen werden. Der Duden [Dos90] definiert autorisieren wie folgt:

> ***autorisieren:*** *1. jmdn. bevollmächtigen, [als einzigen] zu etwas ermächtigen. 2. etwas genehmigen.*

Überträgt man diese Definition auf die Zugriffskontrolle auf Dienste im Internet, so kann man Autorisierung folgendermaßen definieren:

> ***Autorisierung*** *ist die Verifikation, ob ein Dienstnutzer (Subjekt) einen Dienst nutzen (eine Aktion auf einem Objekt ausführen) darf.*

Wenn der Dienstanbieter die Nutzung eines Dienstes auf einen einzelnen oder eine ihm bekannte Gruppe von Dienstnutzern beschränken will, wird in der Autorisierung geprüft, ob der Dienstnutzer die Berechtigung besitzt, den Dienst zu nutzen. Ein solcher Dienst wird wie folgt definiert:

> ***Private Dienste*** *sind solche Dienste, die nur von einer Gruppe von Dienstnutzern oder sogar nur einzelnen Subjekten genutzt werden können.*

Mehrere Beipiele für private Dienste finden sich in den zu Beginn dieses Kapitels vorgestellten Anwendungsfällen. Die Nutzung des Programms *Rechnungswesen-Deluxe* und der Zugriff auf die Mandantendaten in Anwendungsfall 1 ist nur Mitarbeitern der *Kanzlei Huber & Partner* erlaubt. Der Abruf der E-Mails aus dem Postfach in Anwendungsfall 3 ist nur *Herrn Grimm* erlaubt, ebenso wie die Pflege des Webauftritts der Kanzlei in Anwendungsfall 4 nur dem Administrator von *Huber & Partner.*

> ***Öffentliche Dienste*** *sind solche Dienste, die theoretisch von jedermann, gegebenenfalls auch anonym genutzt werden können.*

Öffentliche Dienste im Beispiel sind die von *Recht&Urteil* (Anwendungsfall 5) im Internet zur Verfügung gestellten Urteilstexte oder auch der Webauftritt der Kanzlei. Auch für öffentliche Dienste kann eine Autorisierung erfolgen. Diese basiert dann nicht auf Berechtigungen des Dienstnutzers, sondern beispielsweise auf dem Kriterium der Zahlungsfähigkeit.

Voraussetzung für die bei einer Anfrage nach privaten Dienste durchzuführende Autorisierung anhand von Benutzerrechten ist eine Verifikation, dass der Dienstnutzer tatsächlich der ist, als der er sich ausgibt. Seine Identität muss verifiziert werden. Das entspricht einer Authentifizierung, die nicht nur für eine Autorisierung privater Dienste, sondern auch in anderen Fällen benötigt werden kann.

Der Duden [Dos90] definiert authentifizieren folgendermaßen:

> ***authentifizieren:*** *beglaubigen, die Echtheit bezeugen.*

Im Zusammenhang von Kommunikationsnetzen unterscheidet man zwei Arten der Authentifizierung. In Anlehnung an [ITU95b] werden folgende Definitionen gegeben:

> ***Authentifizierung einer Entität (entity authentication)*** *ist die Verifikation der Identität eines Antragstellers im Rahmen einer Kommunikationsbeziehung mit dem Authentifzierer.*
>
> ***Authentifizierung der Datenherkunft (data origin authentication)*** *ist die Verifikation der Identität des Absenders von Daten durch den Empfänger.*

Die Authentifizierung einer Entität kann, wie beschrieben, eine Teilfunktion der Autorisierung sein. Die Authentifizierung der Datenherkunft ist ein Mechanismus zur Sicherung der Kommunikation in Netzen. Voraussetzung für eine Authentifizierung von Entitäten ist, dass die zu authentifizierende Entität, also der Dienstnutzer, über eindeutige Identitätsmerkmale verfügt, damit er

identifiziert werden kann. Identifizierung oder auch Identifikation bezeichnet den Vorgang des Identifizierens und ist nach [Dos90] wie folgt definiert:

> ***identifizieren**: 1. genau wiedererkennen; die Identität, Echtheit einer Person oder Sache feststellen. ...*

Diese allgemeine Definition wird für die Arbeit wie folgt konkretisiert:

> ***Identifizierung** ist die Prüfung der Identitätsmerkmale einer Entität, welche die Entität eindeutig kennzeichnen.*

Die Identifizierung wird durch den Dienstanbieter mittels der Identitätsmerkmale des Dienstnutzers vorgenommen. Wenn die Identitätsmerkmale eines Dienstnutzers nicht weltweit einmalig sind, bleibt die Eindeutigkeit der Identitätsmerkmale auf eine Organisation begrenzt. Das ist der Fall, wenn die Identitätsmerkmale, wie z.B. eine Benutzerkennung, von einem Dienstanbieter vergeben werden. Alle Dienstanbieter, die innerhalb einer Organisation eindeutige Identitätsmerkmale verwenden, gehören der identischen Sicherheitsdomäne an. Es handelt sich also um einen Begriff des Organisationsmodells, wie es in Kapitel 2.2 vorgestellt wurde.

> *Die **Heimatdomäne** eines Dienstnutzer ist diejenige Sicherheitsdomäne, in der er vom Dienstanbieter, dem **Heimatdienstanbieter**, ein Identitätsmerkmal erhalten hat, um sich ihm gegenüber zu identifizieren.*

2.5 Kaufmännische Funktionen zur Abrechnung von Internet-Diensten

Viele Internet-Dienste werden kommerziell angeboten und sind gebührenpflichtig. Daher muss der jeweilige Dienstanbieter die Leistungserbringung messen und protokollieren, die der Leistungserbringung entsprechenden Gebühren bestimmen und diese dem Dienstnutzer in Rechnung stellen. Das entspricht kaufmännischen Funktionen, die jeder Kaufmann, auch ein solcher, der keine Internet-Dienste erbringt, in ähnlicher Weise durchführen muss. Die für einen Anbieter von Internet-Diensten bedeutsamsten Funktionen sollen folgendermaßen definiert werden:

> ***Metering (Leistungsmessung)** bezeichnet die vom Dienstanbieter vorgenommene Sammlung von Einzeldaten über die Nutzung von Diensten.*

Im Internet erfolgt das Metering auf einer technischen Ebene durch die Server oder andere Netzwerkkomponenten der Dienstanbieter.

> ***Accounting (Kontenführung)** bezeichnet die Sammlung und Zusammenfassung von Informationen, die mit einer Dienstnutzung eines einzelnen Kunden verbunden sind, auf einem Konto.*

Die Kontenführung erfolgt in Mengeneinheiten entsprechend der Einheit der genutzten Ressourcen. Die Kontenführung kann auf Basis der Ergebnisse der Leistungsmessung vorgenommen werden.

> ***Charge-Calculation (Gebührenerhebung)** ist der Prozess der Berechnung eines Gebührensatzes für eine in Anspruch genommene Leistung.*

Auf Basis eines einzelnen Kontensatzes des Accountings wird der erbrachten Leistung eine Gebühr zugerechnet. Ein Mengeneinheit wird in einen Geldwert überführt.

> *Im **Billing (Rechnungsstellung)** werden die einzelnen Gebührensätze eines Dienstnutzers gesammelt, zusammengefasst und in einer Rechnung an den Dienstnutzer übersandt.*

Daneben kann oder muss der Dienstanbieter ein Auditing durchzuführen.

*Im **Auditing (Überwachung und Prüfung)** wird die Korrektheit jeden Prozesses im Zusammenhang mit der Diensterbringung verifiziert oder es werden Kontrollen durchgeführt, um Sicherheitsbrüche zu entdecken.*

Der Umfang der Überwachung und Prüfung wird in einigen Bereichen, z.B. bei der Gebührenerhebung und Rechnungsstellung, durch rechtliche Vorschriften bestimmt. In anderen Bereichen kann der Dienstanbieter diesen selbst definieren. Ein Auditing wird entweder durch ein unabhängiges Echtzeit-Monitoring oder die Auswertung von protokollierten Systemdaten realisiert.

Die kaufmännischen Funktionen insgesamt werden zumeist während und nach der Diensterbringung mittels der Dienstinfrastruktur des Dienstanbieters durchgeführt, wie in Abbildung 11 dargestellt. Sie schließen sich dann zeitlich an die Zugriffskontrolle an.

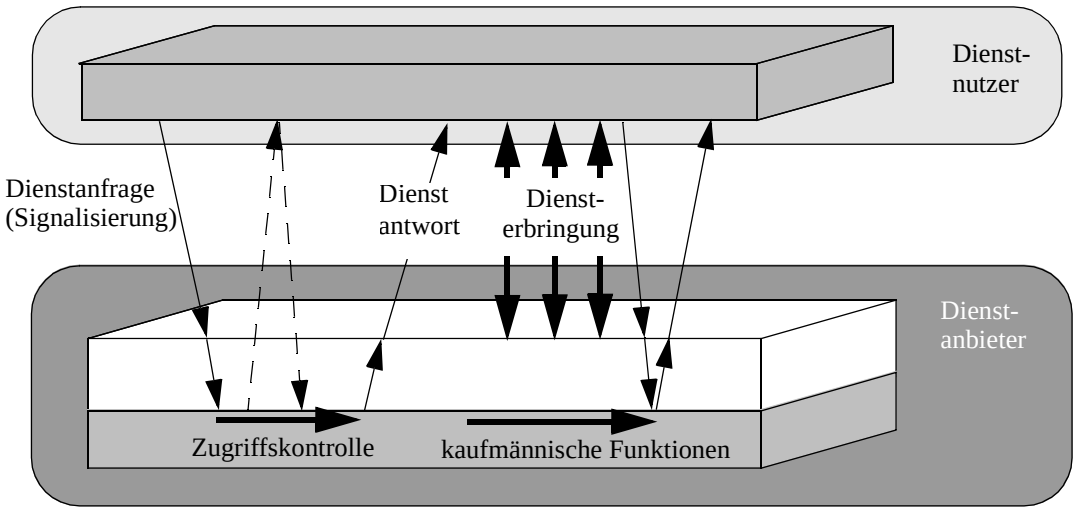

Abbildung 11: Zeitliche Lage der kaufmännischen Funktionen

2.6 Unterstützungsdienste im Internet

Um Endnutzerdienste, wie sie in Kapitel 2.2 definiert wurden, kommerziell, d.h. gegen Gebühr, anbieten zu können, muss ein Dienstanbieter eine Zugriffskontrolle und die kaufmännischen Funktionen ausführen. Das Internet-Dienstmodells erlaubt es, die Durchführung der Zugriffskontrolle und die Erbringung der kaufmännischen Funktionen als eigene Dienste anzusehen, wie in Abbildung 12 illustriert. Dementsprechend sollen sie auch definiert werden:

> ***Zugriffskontrolldienste** sind diejenigen Dienste, die im Rahmen einer Zugriffskontrolle erbracht werden, also insbesondere Authentifizierung und Autorisierung.*

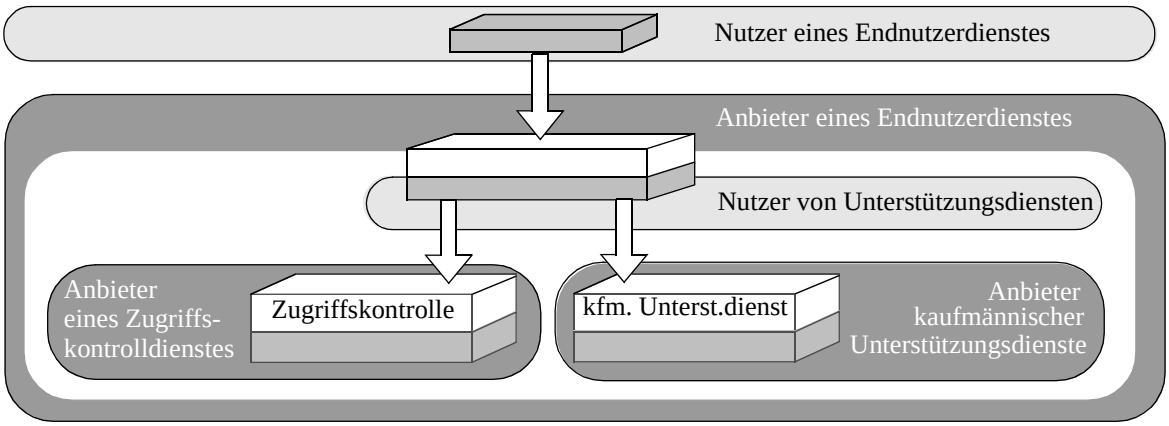

Abbildung 12: Unterstützungsdienste im Internet-Dienstmodell

Kaufmännische Unterstützungsdienste *sind diejenigen Dienste, die kaufmännische Funktionen realisieren, die ein Anbieter von Internet-Diensten ausführen muss.*

All diese Dienste bieten einem Endnutzer im Internet keinen direkten Mehrwert. In Abgrenzung zu den Endnutzerdiensten werden sie daher zusammenfassend als Unterstützungsdienste definiert.

Ein ***Unterstützungsdienst*** *ist ein Dienst, der einem Endnutzer keinen eigenen direkten Mehrwert bietet. Er wird durch einen Dienstanbieter erbracht, um kaufmännische Funktionen oder eine Zugriffskontrolle zu realisieren.*

Es gibt im Internet noch weitere Dienste, die dem Endnutzer keinen eigenen Mehrwert bieten, wie z.B. Managementdienste zum Betrieb und zur Kontrolle der Dienstinfrastruktur. Diese werden in dieser Arbeit nicht betrachtet.

In existierenden Anwendungen und Architekturen im Internet findet man häufig eine Verknüpfung von Zugriffskontrolldiensten mit dem Accounting. Diese werden dann zusammenhängend Authentication, Authorization und Accounting (AAA)-Dienste genannt. Sie bilden eine Teilmenge der Unterstützungsdienste und werden der Vollständigkeit halber wie folgt definiert:

AAA-Dienste *sind solche Dienste, die eine Authentifizierung, Autorisierung und ein Accounting realisieren.*

A^x-Dienste *sind solche Dienste, die Zugriffskontrolldienste und kaufmännische Unterstützungsdienste zur Verfügung stellen.*

2.7 Policybasiertes Netzwerkmanagement

Die im Rahmen der Arbeit entwickelte generische Zugriffskontrollarchitektur basiert auf einer Umsetzung des Paradigmas des policybasierten Netzwerkmanagements [RHKS02]. Sie verwendet Policies zur Konfiguration des Systems. Die wichtigsten Begriffe und die grundlegende Funktionsweise des policybasierten Netzwerkmanagements stellen daher eine weitere Grundlage zum Verständnis der Arbeit dar und werden nachfolgend erläutert.

2.7.1 Policies

Die Systeme als Teil der Dienstinfrastruktur eines Anbieters von Internet-Diensten müssen konfiguriert und verwaltet werden, um ihre Funktionen in der geforderten Weise durchzuführen. Das Management von Netzwerksystemen ist auf verschiedene Arten möglich. Policies stellen einen weitverbreiteten Ansatz dar, um Kommunikationsnetze zu verwalten. Sie werden zunehmend auch zur Konfiguration von Systemen im Internet verwendet [RHKS01].

> *Eine **Netzwerk-Policy** beschreibt das von Netzwerkelementen, Dienstnutzern und Dienstanbietern erwartete Verhalten mit Hilfe von Aktionen, die von den Netzwerkelementen, Dienstnutzern und Dienstanbietern ausgeführt werden sollen.*

Policies definieren also das Verhalten von Entitäten. Sie beschreiben Aktionen, welche die Entitäten ausführen müssen bzw. dürfen. Diese Aktionen können abhängig sein von Bedingungen, die ebenfalls in der Policy spezifiziert sind. Die Verwendung von Policies erlaubt somit eine Trennung der Beschreibung des Verhaltens von Entitäten, d.h. der Konfiguration der Entitäten, von den Entitäten selbst. Die Policies müssen von den Entitäten durchgesetzt werden, indem die spezifizierten Aktionen ausgeführt werden. Die Trennung einer Policy von der sie umsetzenden Entität ermöglicht dynamische Änderungen in der Konfiguration und Verwaltung von Entitäten, ohne dass die Entitäten selbst geändert werden müssen. Sie erlaubt weiterhin die Wiederverwendung von Policies in verschiedenen heterogenen Umgebungen und unterschiedlichen administrativen Domänen [Slo94].

Die Policies von Internet-Dienstanbietern beschreiben zunächst Managementziele. Sie sind dann oftmals abstrakt in Form von Geschäftsmodellen definiert. Aus solchen abstrakten Policies lassen sich die Netzwerk-Policies ableiten, die das Verhalten der Systeme, d.h. der von den Dienstanbietern verwalteten Anwendungen und Netzwerkelemente, bestimmen.

Innerhalb einer Policy werden, entsprechend der Definition von Policy, Aktionen spezifiziert, die das mittels der Policy konfigurierte System ausführen muss. Diese Aktionen werden in einer Policy-Beschreibungssprache formuliert. In den unterschiedlichen Anwendungsbereichen existieren verschiedene Sprachen zur Spezifikation von Policies. Sie verwenden entweder einen natürlichsprachlichen, einen syntaktischen oder einen formal logischen Ansatz. Eine zusammenfassende Darstellung von verschiedenen Policy-Sprachen findet sich in [SX01].

2.7.2 Komponenten einer policybasierte Managementarchitektur

Eine Architektur zur Realisierung eines policybasierten Managements besteht zumeist aus den in Abbildung 13 dargestellten Komponenten [MBHS00][LMS99][BP01]. Policies werden mittels eines Policy Management Werkzeugs erstellt. Dabei kann es sich um einen einfachen Editor handeln oder auch ein vollständiges Werkzeug, welches die Policies auf Inkonsistenzen und Konflikte überprüft. Policies werden in einem Policy-Repository gespeichert.

Muss die mittels der Policy zu konfigurierende Entität, die Dienstinfrastruktur, eine Funktion erbringen, so weiß sie nicht, welche Aktionen dazu auszuführen sind. Diese Aktionen sind als Teil der Policy gerade im Policy Repository gespeichert. Die Dienstinfrastruktur stellt daher eine

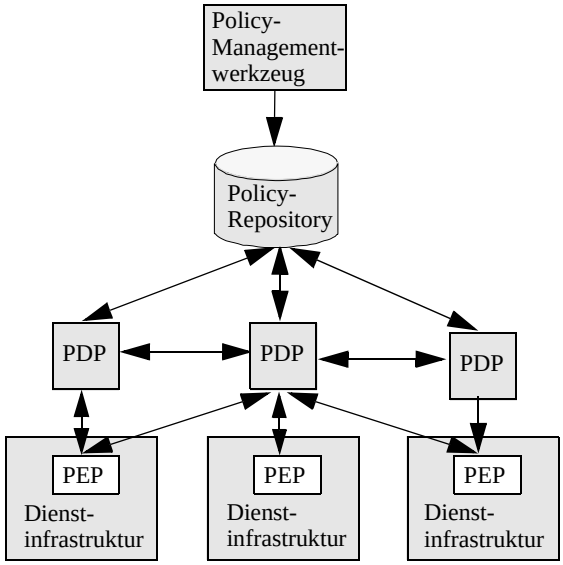

Abbildung 13: Policybasierte Managementarchitektur

Anfrage an einen sogenannten Policy Decision Point (PDP). Dieser Policy Decision Point fragt die gültige Policy aus dem Policy-Repository ab und wertet, falls die auszuführenden Aktionen von einer Bedingung abhängig sind, die Bedingungen aus. Damit bestimmt der Policy Decision Point, welche Aktionen durchzuführen sind. Diese Aktionen übermittelt er in Form von Konfigurationsparametern an das zu konfigurierende System, welches die Aktionen nachfolgend durchführt. Es setzt also die Policy durch und wird daher im Allgemeinen als Policy Enforcement Point (PEP) bezeichnet.

Auf die Zugriffskontrollarchitektur übertragen bedeutet dies, dass das Zugriffskontrollsystem bei einer Dienstanforderung eines Dienstnutzers zunächst eine Anfrage an einen Policy Decision Point und dieser an ein Policy-Repository stellt. Darüber wird bestimmt, welche Aktionen bzw. Teilfunktionen der Zugriffskontrolle auszuführen sind, die dann vom Zugriffskontrollsystem vollzogen werden.

2.8 Zusammenfassung

In einem realistischen Anwendungsszenario und seinen Anwendungsfällen wurden verschiedene Internet-Dienste, deren Nutzung auf unterschiedliche Weise kontrolliert und teilweise auch abgerechnet wird, beschrieben. Sie stellen einen Ausschnitt aus dem Problemfeld der Arbeit dar und werden in den folgenden Kapiteln zu Illustration und als Referenzszenario wiederverwendet. Die in den Anwendungsfällen beschriebenen Dienste lassen sich fünf verschiedenen, in diesem Kapitel definierten Klassen von Diensten, den Verbindungsdiensten, Internet-Zugangsdiensten, QoS-Transportdiensten sowie Anwendungs- und Inhaltsdiensten zuordnen.

Zur Abgrenzung der Arbeit wurde definiert, was unter einem Internet-Dienst und unter einer Zugriffskontrolle auf Internet-Dienste verstanden wird. Ein Internet-Dienst muss für den Dienstnutzer einen ökonomischen Wert darstellen und mittels einer Dienstanfrage vom Anbieter angefordert werden. Die Definitionen wurden anhand eines eigenen Internet-Dienstmodells

vorgenommen, welches eine ökonomische Sicht auf die Dienste erlaubt. Neben den Diensten, die für den Dienstnutzer einen ökonomischen Wert erbringen, muss der Dienstanbieter Funktionen der Zugriffskontrolle und der Abrechnung realisieren. Die einzelnen Teilfunktionen zur Zugriffskontrolle und die kaufmännischen Funktionen wurden definiert. Das Internet-Dienstmodell erlaubt es, diese Funktionen als eigene Dienste, die sogenannten Unterstützungsdienste anzusehen. Dieses Verständnis von Zugriffskontrolle und Abrechnung als eigene Dienste bildet die konzeptionelle Basis für die im Rahmen der Arbeit entwickelte Architektur. Diese wird ergänzt durch das zum Abschluss des Kapitels vorgestellte Paradigma des policybasierten Managements.

Kapitel 3: Zugriffskontrollsysteme im Internet

Das Internet entstand historisch als Netz verschiedener öffentlicher Einrichtungen. Der Betrieb des Netzes selbst wie auch der darüber angebotenen Dienste war durch den öffentlichen Betreiber finanziert. Eine Zugriffskontrolle fand in der Regel nur zum Schutz privater Ressourcen statt. Mit der zunehmenden Bereitstellung von Diensten im Internet durch private Personen und Institutionen und der wachsenden Anzahl der Nutzer ändert sich dies. Privatwirtschaftliche Dienstanbieter verfolgen mit dem Angebot an Diensten ökonomische Interessen. Zum Schutz privater Ressourcen und zur Kontrolle des Zugriffs auf kostenpflichtige Dienste existieren verschiedene Zugriffskontrollsysteme.

Zum Zwecke der Einordnung der eigenen Arbeit in den Stand der Technik wird in diesem Kapitel ein Überblick über die existierenden Systeme, die darin genutzten Verfahren zur Identifizierung, Authentifizierung und Autorisierung, sowie häufig zur Zugriffskontrolle verwendete Protokolle gegeben.

3.1 Verfahren für die Zugriffskontrolle

Im vorhergehenden Kapitel wurde definiert, was unter einer Zugriffskontrolle auf Internet-Dienste zu verstehen ist, und erläutert, dass Zugriffskontrolle aus den Teilfunktionen der Identifizierung, Authentifizierung und der Autorisierung bestehen kann. Diese Teilfunktionen und grundsätzliche Verfahren zu ihrer Realisierung werden im Folgenden zunächst beschrieben und dienen im Weiteren zur Klassifikation bestehender Zugriffskontrollsysteme.

3.1.1 Identifizierung

Identifizierung wurde zuvor in Kapitel 2.4 als Vorlage von einen Dienstnutzer eindeutig kennzeichnenden Identitätsmerkmalen definiert. Dazu muss der Dienstnutzer über solche Identitätsmerkmale verfügen und sie dem Dienstanbieter übermitteln. Der Dienstanbieter nimmt die Identifizierung vor, indem er überprüft, ob er die Identitätsmerkmale einer ihm bekannten Person oder Sache zuordnen kann.

Wie in der allgemeinen Definition von Identifizierung lassen sich auch im Internet Identitätsmerkmale einer Person und einer Sache unterscheiden. Der Dienstnutzer als Person verwendet den Dienst mittels eines elektronischen Geräts, wie im Internet-Dienstmodell, vgl. Abbildung 2 auf Seite 10, gezeigt. Zu den Geräten im Internet zählen beliebige Endgeräte, wie Rechner, Personal Digital Assistants (PDAs) oder Mobiltelefone, und andere aktive Komponenten, wie Hosts, Router oder Switche innerhalb des Internets.

Eine natürliche Person identifiziert sich im Alltagsleben über ihren Namen. Da dieser nicht eindeutig sein muss, wird er in Kombination mit weiteren Informationen wie dem Geburtsdatum und Geburtsort oder Wohnsitz verwendet. Der Name und die weiteren Informationen sind amtlich dokumentiert. Das gilt äquivalent auch für juristische Personen. Biometrische Merkmale sind solche Merkmale, die einer Person auf natürliche Weise zugeordnet sind. Weiterhin dienen als persönliche Identitätsmerkmale solche Merkmale, die von einer Institution, welche die Person identifizieren will, im Rahmen einer Registrierung an die Person vergeben und dieser zugeordnet werden. Diese Merkmale ermöglichen es dann nur dieser Institution, auf die amtlichen Merkmale der Person zurückschließen. Dazu zählen Benutzerkennungen, Konten- und Kundennummern sowie eindeutig vergebene kryptographische Schlüssel. Zertifikate binden für einen Dritten nachprüfbar einen öffentlichen kryptographischen Schlüssel an eine natürliche oder juristische Person.

Eine Sache wird identifiziert über eine dieser Sache zugeordnete Kennzeichnung, zumeist eine Nummer. Im Internet sind solche Kennzeichnungen beispielsweise die Medium Access Control (MAC)-Adresse einer Netzwerkkarte, die IP-Adresse eines Endgeräts oder Hosts, die in der Subscriber Identity Module (SIM) Karte eines Mobiltelefons gespeicherte International Mobile Subscriber Identity (IMSI) oder die Nummer eines physikalischen Ports an einem Netzwerk-Switch. Tabelle 1 zeigt typische Identitätsmerkmale einer Person und eines Geräts im Überblick.

Person	Biometrische Merkmale Amtliche Merkmale: Name - Geburtsdatum - Geburtsort **Benutzerkennung**, Kunden- oder Kontennummer, Network Access Identifier kryptographische Schlüssel, Zertifikate
Gerät	MAC-Adresse der Netzwerkkarte Nummer des Ports an einem Netzwerk-Switch International Mobile Subscriber Identity **Teilnehmeranschlussnummer** IP-Adresse DNS-Rechnername

Tabelle 1: Überblick über Identitätsmerkmale

Identitätsmerkmale können grundsätzlich gefälscht werden. So kann ein Dritter Eigenschaften eines anderen (d.h. einer Person oder Sache) annehmen und dies zum Zweck der Identifizierung als die seinen vorweisen. Die Identitätsmerkmale unterscheiden sich aber hinsichtlich des Aufwandes, mit dem sie gefälscht werden können. Biometrische Merkmale sind nur mit sehr hohem Aufwand und besonderen Fähigkeiten fälschbar. Eine Benutzerkennung dagegen kann von jedermann genutzt werden, sofern er sie kennt. Bei auf kryptographischen Verfahren basierenden Identitätsmerkmale ist die Sicherheit abhängig vom verwendeten Verfahren. Eine IP-Adresse eines Rechners lässt sich von nahezu jedermann ändern, bei MAC-Adressen ist der Aufwand höher. Einer Teilnehmeranschlussnummer im Telefonnetz oder einem physikalischen Port am Netzwerk-Switch kann ein relativ hohes Vertrauen (Trust by wire) entgegengebracht werden. Letztendlich muss der Dienstanbieter, in Relation zum Wert des angebotenen Dienstes bzw. zum Schaden der

durch einen Mißbrauch entstehen kann, bestimmen, welchen Identitätsmerkmalen er vertraut. Dabei muss er zudem berücksichtigen, auf welchem Weg ihm das Identitätsmerkmal übermittelt wird. Bei Internet-Diensten ist es das Internet, welches grundsätzlich als durch Angriffe gefährdet angesehen werden sollte.

Innerhalb der Arbeit wird im Weiteren als sicheres Identitätsmerkmal nur die Teilnehmeranschlussnummer bei der Kontrolle des Zugriffs auf einen Verbindungsdienst angesehen. Als Identitätsmerkmal einer Person wird grundsätzliche eine Benutzerkennung verwendet, die von einem Dienstanbieter an einen Dienstnutzer vergeben wird.

Beim Vorgang der Identifizierung im Internet muss zwischen einer expliziten und impliziten Identifizierung unterschieden werden. Bei der expliziten Identifizierung wird der Dienstnutzer vom Dienstanbieter aufgefordert, seine Identitätsmerkmale an den Dienstanbieter zu übermitteln. Bei der impliziten Identifizierung hingegen sind die Identitätsmerkmale Bestandteil der Dienstanfrage des Dienstnutzers, so dass der Dienstanbieter die Identitätsmerkmale des Dienstnutzers aus der Dienstanfrage selbständig bestimmen kann. Die eigentliche Identifizierung nimmt der Dienstanbieter vor, indem er überprüft, ob er die Identitätsmerkmale einer ihm bekannten Person oder Sache zuordnen kann.

Die explizite Identifizierung kann eine aktive oder passive Identifizierung sein. Bei der aktiven Identifizierung muss der Dienstnutzer als Person aktiv sein Identitätsmerkmal, z.B. eine Benutzerkennung, eingeben, bevor sie an den Dienstanbieter übermittelt werden kann. Sie kann somit nur bei der Identifizierung von Personen erfolgen. Bei der passiven Identifizierung hingegen ist das Identitätsmerkmal auf dem elektronischen Gerät des Dienstnutzers gespeichert und wird für den Dienstnutzer als Person transparent vom Dienstanbieter abgefragt. Mittels einer passiven Identifizierung können Geräte oder Personen identifiziert werden. Der Identifikator kann ständig oder nur temporär auf dem elektronischen Gerät gespeichert sein. Ständig gespeicherte Identitäten sind beispielsweise die International Mobile Subscriber Identity (IMSI) auf der Smartcard eines Funktelefons oder der ebenfalls auf einer Smartcard gespeicherte kryptographische Schlüssel bzw. das Zertifikat eines Dienstnutzers. Temporär auf dem Gerät gespeichert sind Token, wie sie von Single Sign On- und Identity Management Systemen (vgl. Kapitel 3.3.4) verwendet werden. Sie besitzen in der Regel nur eine zeitlich begrenzte Gültigkeit.

3.1.2 Authentifizierung

Im Rahmen der Zugriffskontrolle auf Internet-Dienste bedeutet Authentifizierung, entsprechend der in Kapitel 2.4 gegebenen Definition, die Verifikation der Identität des Dienstnutzers durch den Dienstanbieter. Damit unterscheidet sich die Authentifizierung von der reinen Identifizierung, auch wenn in der Literatur die Begriffe oftmals synonym verwendet werden oder die Identifizierung als Teil der Authentifizierung angesehen wird [WL92].

An einem einfachen Beispiel soll der Unterschied erläutert werden: Ein Dienstnutzer sendet seine Benutzerkennung an den Dienstanbieter. Dieser überprüft, ob er die Benutzerkennung kennt und sie einer Person zuordnen kann. Er identifiziert den Dienstnutzer, kann aber nicht wissen, ob die Identität nicht von einer anderen Person gefälscht wurde. Daher muss der Dienstnutzer auch

sein Passwort übertragen, das er vertraulich behandeln sollte. Anhand des Passwortes authentifiziert der Dienstanbieter den Dienstnutzer, indem er überprüft, ob die Zuordnung von Benutzerkennung und Passwort korrekt ist. Für die Authentifizierung muss der Dienstanbieter also die Zuordnung des Passwortes zur Benutzerkennung kennen. Die Benutzerkennung ist eine sogenannte Authentifizierungs-Information.

Abhängig vom eingesetzten Verfahren und Protokoll kann der Dienstnutzer die Authentifizierungs-Informationen entweder gemeinsam mit seinen Identitätsmerkmalen an den Dienstanbieter übermitteln oder sie werden vom Dienstanbieter in einem zweiten Schritt angefordert, wie in Abbildung 14 gezeigt. Die eigentliche Authentifizierung besteht in der Verifikation der Authentifizierungs-Informationen durch den Dienstanbieter. Der Dienstanbieter speichert die Authentifizierungs-Informationen in einem sogenannten Authentifizierungs-Repository.

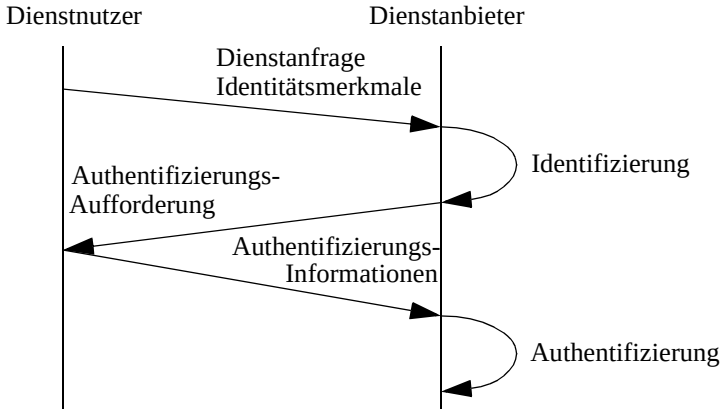

Abbildung 14: Identifizierung und Authentifizierung im Beispiel

Mechanismen zur Authentifizierung lassen sich nach der Art des Identitätsmerkmals bzw. der Authentifizierungs-Informationen in die drei Gruppen unterscheiden: (1) wissensbasierte Verfahren (was man weiß), (2) besitzbasierte Verfahren (was man besitzt) und (3) biometrische Verfahren (was man ist) [Sch98][ITU95b].

In wissensbasierten Verfahren erfolgt die Authentifizierung auf Basis eines gemeinsamen Geheimnisses (Shared Secret) von Dienstnutzer und Dienstanbieter als Authentifizierungs-Information. Der Dienstanbieter ordnet dem Identitätsmerkmal ein Geheimnis zu. Geheimnisse sind z.B. Passwörter oder Persönliche Identifikations Nummern (PINs). Damit die Übertragung des Geheimnisses im Internet nicht von einem Angreifer abgehört werden kann, wird oftmals nicht das Geheimnis selbst übertragen. Stattdessen werden kryptographische Verfahren eingesetzt, die es erlauben zu prüfen, ob der Dienstnutzer das Geheimnis kennt. In sogenannten Challenge Response Verfahren sendet beispielsweise der Dienstanbieter dem Dienstnutzer einen Zufallswert, das Challenge, welches der Dienstnutzer gemeinsam mit dem Geheimnis verwendet, um ein Response zu berechnen. Dabei werden häufig Zero Knowledge Verfahren angewandt, die das Geheimnis nicht preisgeben [Buc01]. Das Response lässt sich nur bei Kenntnis des Geheimnisses berechnen und kann somit genutzt werden, um zu prüfen, ob der Dienstnutzer das Geheimnis kennt.

Bei den besitzbasierten Verfahren sind die Authentifizierungs-Informationen unveränderlich auf einem physikalischen Träger gespeichert. Als Träger dient oft eine Smartcard [CG92][Fan97]. Der Dienstnutzer muss dann diesen Träger besitzen und zum Zeitpunkt der Authentifizierung aktiv nutzen, um die Authentifizierungs-Informationen zu übermitteln bzw. in kryptographischen Verfahren zu verwenden. Die Authentifizierungs-Information ist zumeist ein kryptographischer Schlüssel. Bei der Verwendung von Smartcards als Träger des kryptographischen Schlüssels kann zugleich das kryptographische Verfahren auf der Smartcard implementiert sein, so dass der Schlüssel selbst nicht ausgelesen werden kann und somit zusätzlich geschützt ist [Möl99]. Die heute verbreitetste Smartcard ist die in Mobiltelefonen verwendete SIM-Karte.

Bei den biometrischen Verfahren wird ein physiologisches oder verhaltenstypisches Merkmal einer Person zur Authentifizierung verwendet [DJK+02]. Physiologische Merkmale sind der Fingerabdruck, die Iris, die Retina, das Gesicht oder die Handgeometrie einer Person. Die verhaltenstypischen Merkmale sind zum Beispiel die Sprache oder die Handschrift. Diese Merkmale eines Menschen verändern sich nicht oder nur wenig. Da man annimmt, dass sie nur sehr schwer auf Dritte übertragen und missbraucht werden können, kann man davon ausgehen, dass sie einen sehr hohen Sicherheitsgrad gewährleisten. Zur Authentifizierung werden die biometrischen Merkmale des Nutzers über Sensoren gemessen und mit hinterlegten Merkmalsmustern der vorab identifizierten Personen verglichen. Stimmen gemessene Merkmale und hinterlegte Merkmale bei Berücksichtigung einer definierten Abweichungsschranke überein, ist die Authentifizierung erfolgreich.

Um die Sicherheit zu erhöhen, ist es sinnvoll, Verfahren zweier Gruppen in Kombination zu verwenden. Man spricht in diesem Fall von starker Authentifizierung [Jäg02]. Zum Beispiel muss man bei Verwendung einer Smartcard oftmals zusätzlich eine PIN angeben, mit der sich der Dienstnutzer authentifiziert. Der Besitz der Smartcard ohne Kenntnis der PIN genügt nicht zur Authentifizierung. Wird nur ein Verfahren einer Gruppe zur Authentifizierung verwendet, handelt es sich um eine schwache Authentifizierung.

3.1.3 Autorisierung

Autorisierung bezeichnet, nach der in Kapitel 2.4 gegebenen Definition, die vom Dienstanbieter vorgenommene Verifikation, ob ein Dienstnutzer einen Dienst nutzen darf. Nachfolgend werden verschiedene Verfahren zur Autorisierung klassifiziert. Sie lassen sich grundsätzlich zwei Kategorien zuordnen: Sie können auf einer Authentifizierung oder einem Berechtigungsnachweis basieren. Weiterhin muss hinsichtlich der im Rahmen der Autorisierung zu überprüfenden Aspekte unterschieden werden zwischen statischer und dynamischer Autorisierung.

Beim Zugriff auf private Dienste ist die Identifizierung des Dienstnutzers und dessen Authentifizierung eine Vorbedingung für die Autorisierung, wie in Kapitel 2.4 dargestellt. Die im Rahmen der Authentifizierung verifizierten Identitätsmerkmale des Dienstnutzers verwendet der Dienstanbieter dazu, zu prüfen, ob der Dienstnutzer die Berechtigung besitzt, den angefragten Dienst zu nutzen. Dazu muss der Dienstanbieter Berechtigungen der Dienstnutzer verwalten, in einem sogenannten Rechte-Repository speichern und prüfen. Der Vorgang der Prüfung der Berechtigung ist

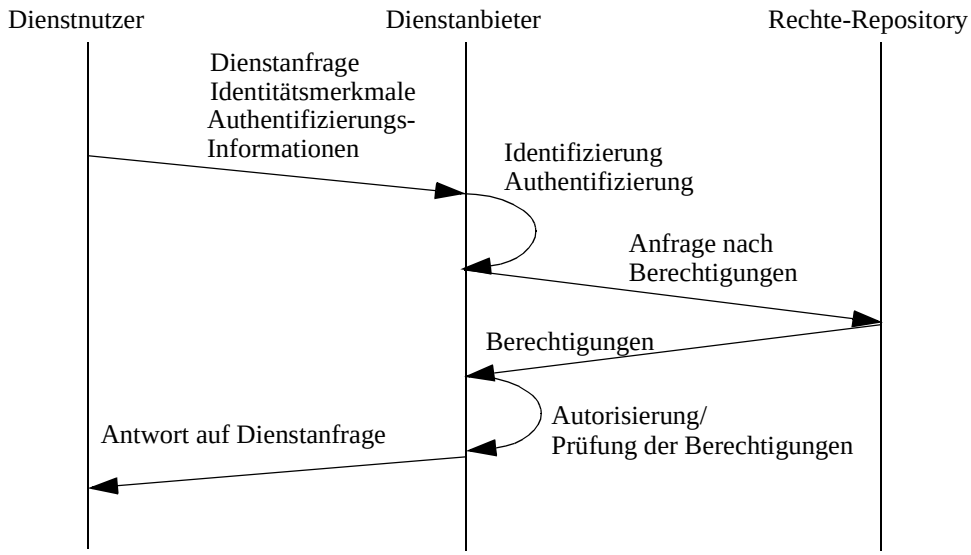

Abbildung 15: Authentifizierungsbasierte Autorisierung mit Prüfung von Berechtigungen

die Autorisierung. Der genaue Ablauf ist abhängig vom verwendeten Protokoll. Im in Abbildung 15 gezeigten Beispiel sendet der Dienstnutzer seine Identitätsmerkmale und Authentifizierungs-Informationnen bereits als Teil der Dienstanfrage an den Dienstanbieter.

Für die Verwaltung und Prüfung von Berechtigungen existieren unterschiedliche Verfahren und Modelle, die auch in Betriebs- und Dateisystemen bei der Kontrolle des Zugriffs auf beliebige Ressourcen genutzt werden. Sie unterscheiden sich insbesondere dahingehend, in welcher Form einzelne Berechtigungen gespeichert werden, durch wen sie vergeben werden dürfen und wie detailliert die Aktionen, die ein Subjekt auf einem Objekt ausführen kann, spezifiziert werden können. Eine wichtige Grundformen der Speicherung von Berechtigungen sind Zugriffskontrollisten. In Zugriffskontrollisten wird dem zu schützenden Objekt, also z.B. einer Datei oder einem Verzeichnis, eine Liste von Subjekten, also Dienstnutzern, und deren Rechte Aktionen mit dem Objekt auszuführen, zugeordnet [Eck98]. Für die im Rahmen der Arbeit zu entwickelnde Architektur, wird von verschiedenen Verfahren der Rechteverwaltung und Prüfung abstrahiert. Es wird davon ausgegangen, dass eine Anfrage an ein Rechte-Repository gestellt werden kann, in welcher der angefragte Dienst spezifiziert wird und das Rechte-Repository zurückmeldet, welche Benutzer den Dienst nutzen dürfen, oder dass zusätzlich auch das authentifizierte Identitätsmerkmal des Dienstnutzers Teil der Anfrage an das Repository ist und dieses selbst die Prüfung vornimmt.

Eine authentifizierungsbasierte Autorisierung muss keine Prüfung von Benutzerberechtigungen umfassen, z.B. wenn die Dienste öffentlich sind. Dann dient die Authentifizierung primär dazu, die Identität des Dienstnutzers zu validieren, um ihm die Dienstnutzung zurechnen zu können und die Dienste abzurechnen.

Bei den auf einem Berechtigungsnachweis basierenden Verfahren legt der Dienstnutzer dem Dienstanbieter einen Berechtigungsnachweis vor. Berechtigungsnachweise werden vom Dienstanbieter selbst oder einem Dritten, dem der Dienstanbieter vertraut, ausgestellt. Die Autorisierung durch den Dienstanbieter erfolgt in mehreren Teilschritten. Zunächst muss der Berechtigungs-

nachweis selbst auf eine Fälschung überprüft werden. Dazu kann er z.B. mit kryptographischen Verfahren oder Hologrammen, falls es sich um einen physikalischen Berechtigungsnachweis handelt, gesichert worden sein. Zum zweiten muss die Vertrauenswürdigkeit des Ausstellers geprüft werden. Zuletzt erfolgt die Kontrolle, ob der Berechtigungsnachweis die Nutzung des angefragten Dienstes erlaubt.

Ein Berechtigungsnachweis kann in verschiedenen Formen existieren. Er kann die Form eines auf kryptographischen Verfahren basierenden Zertifikats haben. In diesem Zertifikat sind die Berechtigungen eines Nutzers spezifiziert und vom Zertifikatsaussteller mittels einer digitalen Signatur unterzeichnet. Solche Berechtigungszertifikate sind beispielsweise die im Kerberos-Protokoll [KN93] genutzten Service-Tickets (vgl. Kapitel 3.3.4) oder die in der Simple Public Key Infrastructure (SPKI) [EFL+99] definierten Zertifikate. In der Autorisierung wird geprüft, ob der angefragte Dienst mit den auf dem Berechtigungsnachweis angegeben übereinstimmt.

Eine andere Form eines Berechtigungsnachweises sind Zahlungsmittel in physikalischer und elektronischer Form. Der Dienstanbieter prüft dann im Rahmen der Autorisierung, ob der Dienstnutzer über genügend Geld verfügt, um den Dienst zu bezahlen. Auch eine Kreditkarte ist ein Berechtigungsnachweis. Sie berechtigt den Inhaber der Karte zur Nutzung von Diensten und Abrechnung dieser Dienste über eine Kreditkartengesellschaft.

Eine Authentifizierung des Dienstnutzers und eine Abfrage und Prüfung seiner Berechtigungen ist bei Vorlage eines Berechtigungsnachweises nicht notwendig. Somit können Berechtigungsnachweise auch zur Kontrolle des Dienstzugriffs durch anonyme Dienstnutzer genutzt werden.

Bei den bisher vorgestellten Beispielen handelt es sich jeweils um eine statische Autorisierung.

> *Eine **statische Autorisierung** ist eine Autorisierung anhand von Berechtigungen bzw. Berechtigungsnachweisen, die vor der Dienstanfrage vergeben werden. Einzig der Zeitpunkt der Diensterbringung kann bei der Prüfung zusätzlich berücksichtigt werden.*

Der Zeitpunkt der Diensterbringung kann einen Einfluss auf die Autorisierungsentscheidung haben, wenn beispielsweise die Berechtigung der Dienstnutzung auf einen Zeitraum beschränkt ist oder ein Berechtigungsnachweis in seiner zeitlichen Gültigkeit beschränkt ist. Im Gegensatz dazu wird eine dynamische Autorisierung definiert:

> *Eine **dynamische Autorisierung** ist eine Autorisierung, in der die Autorisierungsentscheidung auch von der Prüfung eines zum Zeitpunkt der Diensterbringung gültigen Systemzustands abhängig ist.*

Beispiele für solche dynamischen Bedingungen sind vielfältig: Bei einer Anfrage nach einem QoS-unterstützten Transportdient muss der Anbieter eines Transportdienstes überprüfen, ob für den Zeitpunkt der Reservierung ausreichend Ressourcen für die Reservierung verfügbar sind. Bei der Autorisierung mittels Kreditkarten kann die Sperrung der Karte online überprüft werden. Der aktuell gültige Kontenstand eines Guthaben-Kontos des Dienstnutzers kann ebenfalls dynamischer Teil der Autorisierungsentscheidung sein. Abbildung 16 zeigt einen möglichen Protokollablauf. In diesem Beispiel müssen für eine positive Zugriffskontrollentscheidung drei Voraussetzungen gegeben sein: Der Dienstnutzer muss durch den Dienstanbieter authentifiziert

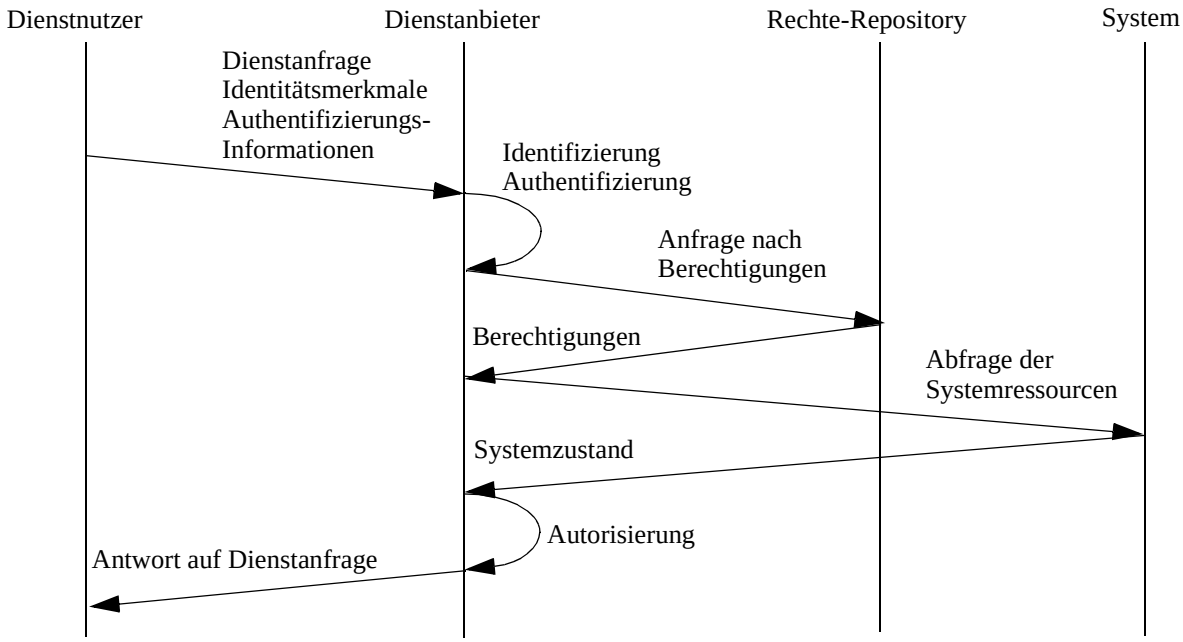

Abbildung 16: Dynamische Authentifizierungsbasierte Autorisierung

sein, er muss die Berechtigungen besitzen den Dienst zu nutzen und die zur Diensterbringung notwendigen Ressourcen müssen verfügbar sein.

Jede Autorisierung und damit jede Form der Zugriffskontrolle lässt sich einer der vier Kombinationen aus authentifizierungs- oder berechtigungsnachweisbasierter bzw. statischer oder dynamischer Autorisierung zuordnen.

3.2 Merkmale zur Klassifikation von Zugriffskontrollsystemen

Um bestehende Zugriffskontrollsysteme untereinander und mit der im Rahmen der Arbeit neu entwickelten Architektur vergleichend beurteilen zu können und einen Überblick über den Stand der Technik zu geben, werden im kommenden Abschnitt weitverbreitete Zugriffskontrollsystem vorgestellt und anhand verschiedener Merkmalskriterien klassifiziert. Die Merkmalskriterien betrachten verschiedene Aspekte. Sie beschreiben den Funktionalitätsumfang der Zugriffskontrollsysteme, die unterstützten Formen bzw. Verfahren der Zugriffskontrolle sowie konzeptionelle Aspekte der zugrundeliegenden Architektur, die einen Einfluss auf die Flexibilität der Systemkonfiguration und die Unterstützung der Mobilitätsanforderungen der Dienstnutzer haben. Damit sind wichtige Merkmale erfasst, um die existierenden Systeme auf Ihre Eignung als generisches Zugriffskontrollsystem, wie es Internet-Dienstanbieter benötigen, zu prüfen. Sie sind in Tabelle 2 zusammengefasst.

Funktionalitätsumfang								Verfahren						Konzeption			
Kontrolle von								Identifizierung anhand		unterstützte Autorisierungsverfahren				Modell		Konfiguration	
Verbindungsdiensten	Internet-Zugangsdiensten	QoS-Transportdiensten	Anwendungsdiensten	Inhaltsdiensten	fremden Dienstnutzern	anonymen Dienstnutzern	privaten Diensten	Merkmal einer Person	Merkmal eines Geräts	authentifizierungsbasiert	berechtigungsbasiert	statisch	dynamisch	integriertes Modell	Third Party Modell	statisch	dynamisch

Tabelle 2: Merkmale zur Klassifikation von Zugriffskontrollsystemen

Zur Beschreibung des Funktionalitätsumfangs werden die Klassen von Internet-Diensten bestimmt, die mittels des Zugriffskontrollsystems kontrolliert werden (vgl. Kapitel 2.2). Weiterhin wird analysiert, ob der Zugriff von Dienstnutzern außerhalb der Heimatdomäne sowie von anonymen Dienstnutzern und der Zugriff auf private Dienste zu kontrollieren ist (vgl. jeweils Kapitel 2.4). Es wird untersucht, welche Identitätsmerkmale in der Authentifizierung genutzt werden und welche Authentifizierungs- und Autorisierungs-Verfahren unterstützt werden. Aus konzeptioneller Sicht wird angegeben, ob eine dynamische Konfiguration des Systems z.B. mittels des policybasierten Managements möglich ist (vgl. Kapitel 2.7) und welches Architekturmodell verwendet wird.

Als Architekturmodelle werden hier das integrierte und das Third Party Modell unterschieden. Sie differieren sich hinsichtlich des Grads der funktionalen Verknüpfung der Authentifizierung und Autorisierung mit den Funktionen zur Erbringung des Internet-Dienstes selbst. Im integrierten Modell erfolgt die Authentifizierung und Autorisierung als Teil der Anwendung, die den Internet-Dienst realisiert. Die Identifizierungs- und Authentifizierungs-Informationen und die Berechtigungen können dann entweder mit den des Betriebs- bzw. Dateisystems übereinstimmen oder in anwendungsspezifischen Benutzer- und Rechtedateien gespeichert sein.

Im Third Party Modell hingegen wird das Client Server Paradigma angewandt. Die Zugriffskontrollfunktionen können gemäß des Dienstemodells als eigene Dienste angesehen werden, wie in Kapitel 2.6 erläutert. Die Authentifizierung und Autorisierung werden dann von einem eigenen Server, einer dritten Partei, vorgenommen. Auch das Authentifizierungs- und Rechte-Repository sind dann Teil des Authentifizierungs- und Autorisierungs-Servers. Der Anbieter des Dienstes fragt als Client vom Server die Autorisierungs-Entscheidung an. Der Server kann dazu selbst den Dienstnutzer authentifizieren. Die Vorteile dieses Modells bestehen darin, dass mehrere Systeme eines Dienstanbieters oder auch mehrere Dienstanbieter einen gemeinsamen Authentifizierungs- und Autorisierungs-Server nutzen können und auf gemeinsame Repositories zugreifen können. Dies vereinfacht die Verwaltung der Nutzerberechtigungen und Identitäts- und Authentifizierungs-Informationen bei der Verwendung mehrerer Systeme zur Erbringung der Dienste. Wei-

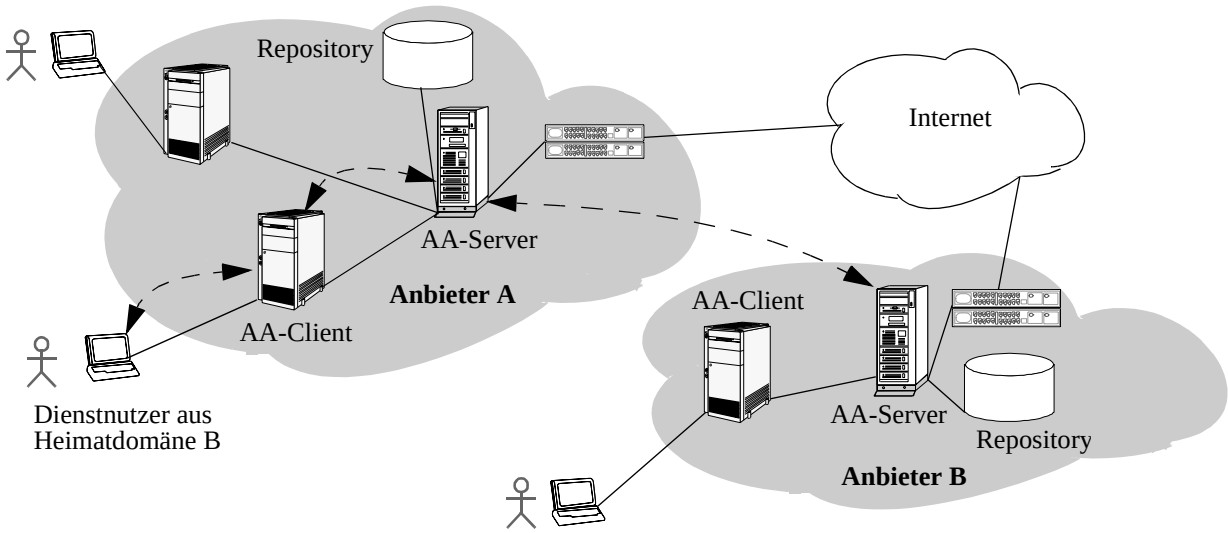

Abbildung 17: Authentifizierung eines fremden Dienstnutzers im Third Party Modell

terhin ist es möglich, eine Kette von Authentifizierungs- und Autorisierungs-Servern zu nutzen, wenn der erste angefragte Server nicht über die notwendigen Informationen verfügt, wie dies in Abbildung 17 gezeigt ist. Damit lässt sich auch eine Zugriffskontrolle für mobile Dienstnutzer realisieren, denn der erste angefragte Server muss den Dienstnutzer nicht authentifizieren können und kann die Anfrage an den Heimatdienstanbieter des Dienstnutzers weiterleiten. Notwendig dazu ist ein zwischen den Dienstanbietern bestehendes Vertrauensverhältnis und dessen Umsetzung in Form eines Vertrages und organisatorischen Regeln.

3.3 Überblick über existierende Zugriffskontrollsysteme im Internet

Im Folgenden wird ein Überblick über existierende Zugriffskontrollsysteme gegeben, die anhand der zuvor definierten Merkmale beschrieben werden. Zur Strukturierung dienen die verschiedenen Klassen von Internet-Diensten. Der Überblick schließt ab mit einer Erläuterung der Zugriffskontrolle in Firewalls als Spezialfall und eines Ansatzes, der Kontrolle von Diensten verschiedener Dienstklassen umfasst.

3.3.1 Kontrolle das Zugriffs auf Verbindungsdienste

Nach der Definition aus Kapitel 2.3 bieten Verbindungsdienste eine physikalische und logische Verbindung zwischen zwei Geräten an. Ein Verbindungsdienstanbieter stellt somit ein aus mehreren Komponenten bestehendes Netz zur Verfügung. Netze lassen sich allgemein unterscheiden hinsichtlich ihrer Struktur (Punkt-zu-Punkt Kanäle, Rundsendekanäle), ihrer räumlichen Ausdehnung und hinsichtlich der verwendeten Technik (Kabelnetze, Funknetze, Satellitenverbindungen) [Tan02].

Verbindungsdienste werden zum einen vom Endnutzer benötigt, um auf einen Internet-Zugangsdienst zugreifen zu können, zum anderen von Anbietern von Internet-Diensten, um ihre

Systeme mit anderen Teilen des Internets zu vernetzen. Ein Endnutzer verwendet zumeist eine der folgenden Formen von Netzen zum Aufbau einer Verbindung zu einem Internet-Zugangsrechner:

- Wählverbindungen im Festnetz
- Wählverbindungen im Funktelefonnetz
- Lokale kabelbasierte Netze / Wired Local Area Network (Wired-LAN)
- Lokale Funknetze / Wireless Local Area Network (Wave-LAN)
- Punkt-zu-Punkt Festverbindungen im regionalen Bereich

Beim Zugriff auf Kabelnetze erfolgt die Zugriffskontrolle auf die Verbindungsdienste zumeist nur durch die Kontrolle des räumlichen Zugangs zum Kabel als Übertragungsmedium. Jeder der einen Zugang zum Kabel besitzt, kann auf diesem unbeschränkt Daten übertragen. Das gilt auch bei der Verwendung von lokalen, kabelbasierten Rundsendekanälen im LAN. Dort kontrollieren die Dienstnutzer selber auf der Data Link Ebene ihren Zugang auf das Übertragungsmedium. Dies dient der Optimierung der Auslastung des Verbindungskanals. Es werden verschiedene Media Access Control (MAC)-Protokolle verwendet, wie z.B. Ethernet 802.3 [IEE02]. In Anwendungsfall 1 aus Kapitel 2.1 erfolgt also keine explizite Zugriffskontrolle auf den Verbindungsdienst. Eine Ausnahme innerhalb der Kabelnetze stellen öffentlich zugängliche Telefone als Zugangspunkt dar. Dort wird die Dienstnutzung direkt bezahlt oder der Dienstnutzer identifiziert sich mit einer speziellen ihm zugewiesenen Vorwahlnummer.

Muss die Dienstnutzung einem Kunden als Basis für eine Abrechnung zugeordnet werden, verwendet der Dienstanbieter hierzu die Identitätsmerkmale der Geräte. Dies sind insbesondere die Teilnehmeranschlussnummern bei Wählverbindungen.

Im Gegensatz zu Kabelnetzen, ist in lokalen Funknetzen eine räumliche Kontrolle des Zugangs zum Übertragunsmedium kaum möglich. Niemanden kann technisch verwehrt werden, auf der entsprechenden Frequenz Daten zu senden. Daher ist eine explizite Zugriffskontrolle notwendig. Diese kann frühestens am Endpunkt der Funkverbindung erfolgen. Das IEEE-Protokoll 802.11 [IEE99] sieht für lokale Funknetze eine Autorisierung durch die Basisstation vor. Diese sendet nur die Daten eines autorisierten Dienstnutzers weiter. Die Autorisierung erfolgt durch den Nachweis der Kenntnis eines kryptographischen Schlüssels seitens des Dienstnutzers, unter Einsatz eines Challenge Response Verfahrens. Der kryptographische Schlüssel ist ein Gruppenschlüssel, der für alle Stationen innerhalb des lokalen Netzes identisch ist. Damit handelt es sich im Sinne der Definition aus Kapitel 3.1.3 um einen Berechtigungsnachweis. Der Dienstnutzer ist über den Schlüssel nicht identifizierbar und authentifizierbar. Weiterhin ist es möglich, dass der Access-Point eine Kontrolle der Dienstnutzer über die MAC-Adresse des verwendeten Geräts vornimmt.

In Funktelefon-Netzen ist ebenfalls eine Autorisierung notwendig. Im Global System of Mobile Communication (GSM) Netz erfolgt eine Identifizierung und Authentifizierung der innerhalb des Endgeräts verwendeten SIM-Karte [ATA99]. Außerdem authentifiziert sich die Person gegenüber der SIM-Karte über die Eingabe einer persönlichen Identifikationsnummer (PIN). Die SIM-Karte lässt sich über eine eindeutige IMSI identifizieren. Sie enthält außerdem einen geheimen Schlüssel, der zur Authentifizierung genutzt wird und die Authentifizierungs-Information ist. Während des Verbindungsaufbaus zum sogenannten Visitor Location Register (VLR) identifiziert sich die

Mobile Station, bestehend aus Endgerät und SIM-Karte, mittels der IMSI. Das VLR leitet die IMSI weiter an das Home Location Register (HLR), wie im ersten Teil von Abbildung 74 in Anhang B.3 gezeigt. Beim HLR ist die SIM-Karte registriert. Sie kennt den zugehörigen geheimen Schlüssel, bestimmt einen Satz von Challenges und zugehörigen Responses und sendet diesen an das VLR. Dieses wählt ein Challenge aus, schickt es an die mobile Station weiter und vergleicht das von der SIM-Karte berechnete Response mit dem von der HLR gesendeten. Stimmen beide überein, ist die Authentifizierung erfolgreich. Zusätzlich wird anhand einer sogenannten schwarzen Liste geprüft, ob das Endgerät gesperrt ist, z.B. weil es als gestohlen gemeldet wurde. Die Autorisierung umfasst also neben der Authentifizierung auch die Prüfung eines dynamischen Kriteriums. Da die Authentifizierung vom VLR und HLR gemeinsam durchgeführt wird, handelt es sich weder um die Anwendung des rein integrierten noch des Third Party Modells. Sofern zwischen zwei GSM-Anbietern eine sogenannte Roaming-Vereinbarung besteht, kann auch einem fremden Dienstnutzer die Erlaubnis zur Dienstnutzung erteilt werden.

Die Merkmale der Zugriffskontrolle in lokalen Funknetzen unter Nutzung des 802.11-Protokolls und in GSM sind in Tabelle 3 zusammengefasst.

	Kontrolle von								Ident.		Autorisierung				Modell		Konfig.	
	Verbindungsdiensten	Internet-Zugangsdiensten	QoS-Transportdiensten	Anwendungsdiensten	Inhaltsdiensten	fremden Dienstnutzern	anonymen Dienstnutzern	privaten Diensten	Merkmal einer Person	Merkmal eines Geräts	authentifizierungsbasiert	berechtigungsbasiert	statisch	dynamisch	integriertes Modell	Third Party Modell	statisch	dynamisch
802.11	x									x		x	x		x		x	
GSM	x					x				x	x			x	x		x	

Tabelle 3: Merkmale von Systemen zur Kontrolle von Verbindungsdiensten

3.3.2 Kontrolle des Zugriffs auf Internet-Zugangsdienste

Der Zugriff auf Internet-Zugangsdienste muss grundsätzlich nur dann explizit kontrolliert werden, wenn der Router, welcher den Zugang zum Internet realisiert, über eine öffentlich zugängliche Verbindung erreichbar ist. Dies ist der Fall bei Einwahl-Knoten oder Routern in lokalen Netzen, die mit Funk-Basisstationen bzw. öffentlich zugänglichen Ethernet-Anschlüssen verbunden sind. Wenn der Router nicht über eine öffentliche Verbindung erreichbar ist, so erfolgt in der Regel auch keine explizite Kontrolle des Zugangs zum Internet, sondern es wird davon ausgegangen, dass die räumliche Zugriffskontrolle auf den Verbindungsdienst ausreichend ist. Dies ist z.B. der Fall in kabelbasierten LANs. Der Betreiber der LANs ist zugleich Betreiber des Internet-Zugangsrechners und bietet allen seinen Nutzern einen Zugang zum Internet an. In Anwendungs-

fall 1 (vgl. Kapitel 2.1) kann *Herr Grimm* an seinem lokalen PC ohne eine weitere Kontrolle über einen Zugang zum Internet verfügen.

Eine explizite Zugriffskontrolle kann gegebenenfalls auch bei der Bereitstellung von Einwahl-Knoten entfallen. Wird der Internet-Zugangsrechner von einem Telekommunikationsanbieter betrieben und eine von ihm bereitgestellte Verbindung zum Zugang genutzt, kann er den Dienstnutzer mittel der Anschlussnumer identifizieren und über diese Nummer den Kunden bestimmen. Bei sogenannten Call-by-Call Internet-Zugängen, wie sie vielfach angeboten werden, geben die Anbieter der Verbindungsdienste die Informationen über den Dienstnutzer an den Betreiber des Zugangs weiter. Auch hier muss keine explizite Zugriffskontrolle vorgenommen werden.

In den anderen Fällen muss der Dienstnutzer autorisiert werden. Unabhängig von der verwendeten Verbindungstechnologie wird dazu zumeist ein grundsätzlich einheitliches Verfahren genutzt, wie es in Abbildung 18 gezeigt ist.

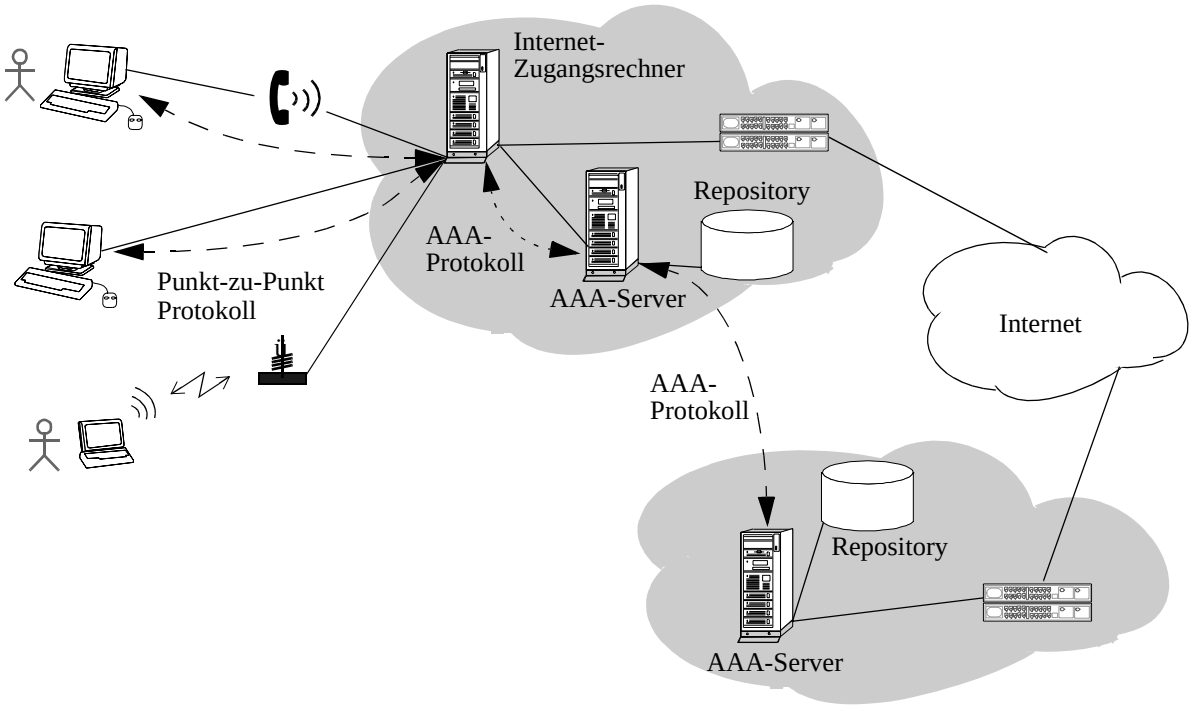

Abbildung 18: Kontrolle des Internet-Zugangs

Der Rechner, der den Internet-Zugang zur Verfügung stellt, bzw. an den eine Dienstanfrage gestellt wird, verlangt vom Dienstnutzer eine Authentifizierung. Er kommuniziert dazu mit dem Endgerät des Dienstnutzers über ein sogenanntes Punkt-zu-Punkt Protokoll. Die Zugriffskontrolle in Form einer Authentifizierung und Autorisierung unter Nutzung des Third Party Modells führt ein sogenannter AAA-Server durch. Befindet sich der Dienstnutzer in der Domäne eines fremden Dienstanbieters und verfügt der lokale AAA-Server nicht über die Authentifizierungs-Informationen des Dienstnutzers, so stellt dieser seinerseits eine Anfrage an den Heimatdienstanbieter des

Dienstnutzers. Dieses Verfahren wird bei der Verwendung mehrere Zugangstechnologien genutzt, wie nachfolgend beschrieben. Die gemeinsamen Merkmale sind in Tabelle 4 aufgeführt.

	Kontrolle von								Ident.		Autorisierung				Modell		Konfig.	
	Verbindungsdiensten	Internet-Zugangsdiensten	QoS-Transportdiensten	Anwendungsdiensten	Inhaltsdiensten	fremden Dienstnutzern	anonymen Dienstnutzern	privaten Diensten	Merkmal einer Person	Merkmal eines Geräts	authentifizierungsbasiert	berechtigungsbasiert	statisch	dynamisch	integriertes Modell	Third Party Modell	statstisch	dynamisch
AAA-Server		x				x			x		x		x			x	x	

Tabelle 4: Merkmale von Systemen zur Kontrolle von Internet-Zugangsdiensten

Verwendung von Wähl- und Festverbindungen. Diese Form des Zugangs wurde erstmalig Anfang der 90er Jahre mit der Einrichtung von Einwahlknoten (Modem Pools) und zugehörigen Internet-Zugangsrechnern in Rechenzentren zur Verfügung gestellt und hat bis heute, insbesondere für private Dienstnutzer, eine hohe Relevanz. Wurde ursprünglich als Zugnagstechnologie nur das analoge Telefonnetz genutzt, so sind es heute alle Arten von Wählverbindungen, über das Integrated Services Digital Network (ISDN), Funktelefonnetze oder verschiedene Ausprägungen von Digital Subscriber Line (DSL).

Der Zugriff auf die Transportdienste erfolgt über den Internet-Zugangsrechner, zu welchem die Wählverbindung aufgebaut wird. Das Endgerät des Dienstnutzers und der Internet-Zugangsrechner verwenden zumeist das Point to Point Protocol (PPP) [Sim94] für ihre Kommunikation. PPP erlaubt die Aushandlung von Authentifizierungs-Protokollen. So ist eine Nutzung des Password Authentication Protocol (PAP) [LS92], des Challenge Handshake Authentication Protocol (CHAP) [Sim96] oder des Extensible Authentication Protocol (EAP) [BV98] vorgesehen. Für die Kommunikation zwischen dem Internet-Zugangsrechner und dem AAA-Server wird dann das Remote Authentication Dial In User Service (RADIUS) Protokoll [RWRS00] oder alternativ Diameter [CLG+02] als AAA-Protokoll verwendet. Der AAA-Server heißt dann dementsprechend RADIUS-Server oder Diameter-Server. Die verschiedenen Protokolle werden in Kapitel 3.4.1 vertieft erläutert.

Kontrolle des Netzzugangs in lokalen Netzen. Auch in lokalen Funknetzen oder lokalen Netzen mit öffentlich zugänglichen Ethernet-Anschlüssen ist eine Kontrolle des Internet-Zugangs notwendig. Diese kann, wie zuvor in Kapitel 3.3.1 beschrieben, mit Hilfe eines Gruppenschlüssels auf der Ebene der Verbindungsdienste erfolgen. Das ist aber nicht möglich, wenn der Zugang grundsätzlich öffentlich verfügbar sein soll und eine Identifizierung der Dienstnutzer zum Zwecke der Abrechnung notwendig ist. Diese Szenario gewinnt mit der Einrichtung sogenannter öffentlicher Hot Spots zunhemend an Bedeutung. Spezialisierte Verbindungsdienstleister, die

lokale Funknetze in hoch frequentierten Bereichen, wie Hotels oder Flughäfen, als Zugangsnetz zum Internet anbieten, wollen den Zugang zum Internet auf registrierte Benutzer beschränken und für die Nutzung Gebühren in Rechnung stellen. Damit ist es notwendig, den Benutzer zu authentifizieren.

Mittels des IEEE-Standards 802.1X [IEE01] lässt sich diese Anforderung erfüllen. Äquivalent zu PPP für Einwahl-Verbindungen ermöglicht 802.1X eine Authentifizierung der Dienstnutzer in lokalen Netzen. Eine sogenannte Port Access Entity (PAE), die dem Internet-Zugangsrechner entspricht, kontrolliert den Zugang zum Netz. Dabei kann es sich um eine Basisstation eines Funknetzes oder auch um einen Switch handeln. Will ein Endgerät einen Netzzugang über die Port Access Entity nutzen, so muss zunächst die Übermittlung der Authentifizierungs-Informationen des Benutzers mittels EAP und eine Authentifizierung und Autorisierung durch einen AAA-Server erfolgen. Erst wenn die Autorisierung durch den AAA-Server erfolgreich ist, leitet die Port Access Entity den IP-Verkehr vom entsprechenden Endgerät oder dem entsprechenden Port weiter; zuvor werden alle einkommenden Pakete verworfen. Dieses Verfahren wird auch in Anwendungsfall 2 genutzt. Der genaue Protokollablauf ist in Abbildung 71 in Anhang B.2 illustriert.

Kontrolle des Netzzugangs bei Verwendung von Mobile IP. Mobile IP [Per96] unterstützt die Mobilität von Geräten, die durch IP-Adressen identifiziert werden, auf der Verbindungsebene. Ein Gerät kann an einer entfernten Lokation seine IP-Adresse weiter verwenden, indem IP-Pakete zu ihm getunnelt werden. Dazu wird jedes mobile Gerät, im allgemeinen mobiler Knoten genannt, unabhängig von seinem aktuellen Netzzugang auf der Verbindungsebene über seine Heimat IP-Adresse identifiziert. Befindet er sich an einem entfernten Ort, so empfängt der sogenannte Home-Agent, in der Regel ein Router in seinem Heimatnetz, die IP-Pakete an seiner Stelle und tunnelt sie an die in der sogenannte Care of Adresse des mobilen Knoten angegebene IP-Adresse. Diese Care of Adresse teilt der mobile Knoten im Rahmen einer Registrierung seinem Home-Agent mit, sobald er einen fremden Netzzugang verwendet. Im fremdem Netz gibt es einen sogenannten Foreign-Agent, der die Pakete für das mobile Gerät entgegennimmt und sie an den mobilen Knoten weiterleitet sowie die Routing-Funktionen für das mobile Gerät übernimmt. Das Mobile IP Protokoll selbst sieht nur eine Authentifizierung der Registration-Message des mobilen Knotens vor. Dazu müssen dieser und sein Home-Agent über ein gemeinsames Geheimnis verfügen. Über einen Security Parameter Index (SPI) identifiziert sich der mobile Knoten und authentifiziert sich über einen unter Nutzung des gemeinsamen Geheimnisses berechneten Hashwert. Optional ist auch eine Authentifizierung beim Foreign-Agent möglich. Es findet aber tatsächlich keine Zugriffskontrolle durch den Anbieter des Zugangs zum Internet in der fremden Domäne statt. Dieser sendet alle Pakete von und an den mobilen Knoten weiter. Eine solche Zugriffskontrolle durch die Foreign-Domain ist aber in vielen Fällen notwendig, wenn z.B. eine gegenseitige Abrechnung der genutzten Dienste erfolgen soll.

Es existieren eine Reihe von Vorschlägen zur Kontrolle des Zugriffs durch einen mobilen Knotens unter Verwendung des Third Party Modells und von AAA-Servern[Per00]. In [GHJP00] ist ein Basismodell erläutert. In diesem Modell interpretiert der Foreign-Agent die Registrierungsanfrage des mobilen Knoten als eine Dienstanfrage nach einem Internet-Zugang. Der Foreign-Agent

stellt nun anstelle des mobilen Knotens als AAA-Client eine Autorisierungs-Anfrage an seinen lokalen AAA-Server. Dieser leitet die Anfrage an den AAA Home Server des mobilen Knoten weiter. Über den Network Access Identifier (NAI), den der mobile Knoten in seiner Registrierungsanfrage angibt [CP00], kann die Heimatdomäne des mobilen Knoten bestimmt werden und somit ermittelt werden, an welchen AAA-Server die Anfrage zu stellen ist.

3.3.3 Kontrolle des Zugriffs auf QoS-Transportdienste

Eine Zugriffskontrolle auf QoS-Transportdienste im Internet findet sich z.B. in der IETF Integrated Services (IntServ) Architektur [BZB+97] [YPG00]. Die IntServ Architektur verfolgt die Zielsetzung, QoS-Anforderungen, die Anwendungen an die Güte der Transportdienste stellen, mittels verschiedener Mechanismen der Transportebene sicherzustellen. Dazu stellen die Anwendungen eine Reservierungsanfrage an die Anbieter von Transportdiensten bzw. deren Router. In Anwendungsfall 6, vgl. Kapitel 2.1, stellt das Videokonferenzsystem eine solche Anfrage. Der Anwendungsdatenstrom (Flow) wird dann über einen reservierten virtuellen Kanal übertragen.

Die Zugriffskontrolle durch die einzelnen Router entlang des virtuellen Kanals erfolgt in zwei Formen: Die Policy-Control prüft, ob der Benutzer autorisiert ist, eine Reservierungsanfrage zu stellen. Dazu muss der Benutzer authentifiziert werden. Zu diesem Zweck können innerhalb der Reservierungsanfrage nach dem Resource ReSerVation Protocol (RSVP) [BZB+97] Authentifizierungs-Informationen übertragen werden. Diese sind in RSVP nicht genauer definiert, die Verwendung von Zertifikaten ist vorgesehen. Die Authentifizierung und Autorisierung wird nicht von den Routern selbst vorgenommen, sondern durch einen Policy-Server [YPG00]. Zur Kommunikation zwischen den Routern und dem Policy-Server kann das COPS-Protokoll [DBC+00] (vgl. Kapitel 3.4.3) verwendet werden, wie in Abbildung 19 gezeigt. Die Router werden dann über die Policies des Policy-Servers konfiguriert.

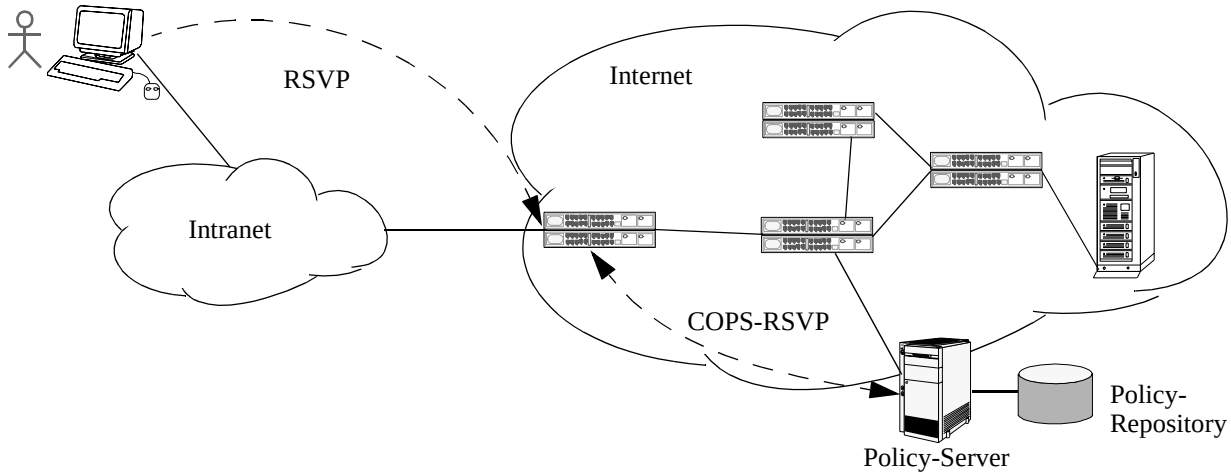

Abbildung 19: Zugriffskontrolle in der Integrated Services Architektur

In der Admission-Control wird überprüft, ob die zu reservierenden Ressourcen für den Reservierungszeitpunkt zur Verfügung stehen und somit die geforderte Qualität vom Dienstanbieter

garantiert werden kann. Es handelt sich somit um eine dynamische Komponente der Zugriffskontrolle.

Während der Übertragung des Datenstroms (Flows) erfolgt eine fortlaufende Zugriffskontrolle in Form des Policing und Traffic-Shaping. Dabei werden einzelne Pakete überprüft. Diese Form der Kontrolle geht über den Rahmen der Arbeit hinaus, da die Übertragung der einzelnen Pakete für den Dienstnutzer keinen Wert darstellt. Die Merkmale der Integrated Services Architektur sind in Tabelle 5 aufgeführt.

	Kontrolle von								Ident.		Autorisierung				Modell		Konfig.	
	Verbindungsdiensten	Internet-Zugangsdiensten	QoS-Transportdiensten	Anwendungsdiensten	Inhaltsdiensten	fremden Dienstnutzern	anonymen Dienstnutzern	privaten Diensten	Merkmal einer Person	Merkmal eines Geräts	authentifizierungsbasiert	berechtigungsbasiert	statisch	dynamisch	integriertes Modell	Third Party Modell	statisch	dynamisch
IntServ			x			x			x		x			x		x		x

Tabelle 5: Merkmale von Systemen zur Kontrolle von QoS-Transportdiensten

3.3.4 Kontrolle des Zugriffs auf Anwendungs- und Inhaltsdienste

Für eine Kontrolle des Zugriffs auf Anwendungs- und Inhaltsdienste existieren verschiedene Verfahren, die sich in Abhängigkeit von der Anwendung, der Gruppe der Dienstnutzer und der Form der Kontrolle entwickelt haben. Sie werden nachfolgend in drei Gruppen zusammengefasst. Die Vorstellung zweier Spezialformen der Zugriffskontrolle, die eine Erhöhung der Benutzerfreundlichkeit bzw. Sicherheit zum Ziel haben, nämlich die Single Sign On Systeme und Identity Management Systeme, ergänzen den Überblick.

Direkte Zugriffskontrolle unter Nutzung der Betriebssysteme. Die Kontrolle des Zugriffs auf Anwendungs- und Inhaltsdienste erfolgte vor der einsetzenden Kommerzialisierung des Internets pimär zum Zweck des Schutzes privater Daten und Ressourcen. Das aus zentralisierten Großrechnersystemen bekannte Verfahren der Zugriffskontrolle durch eine Authentifizierung des Benutzers mittels Benutzerkennung und Passwort und eine Autorisierung auf Basis von Zugriffskontrollisten wird im Internet zunächst weiter verwendet. Der einzige Unterschied besteht nun darin, dass die Kommunikation zwischen dem Benutzer und dem entferntem Rechner keine direkte ist, wie bei Großrechnern, sondern über verschiedene Systeme im Internet erfolgt. Für diese Kommunikation werden Protokolle der Anwendungsebene, wie beispielsweise Telnet [PR83], File Transfer Protocol (FTP) [PR85], Remote Login (rlogin)[Kan91] oder Remote Procedure Call (RPC) [Mic88] genutzt. Diese Protokolle übermitteln die Dienstanforderung, tauschen die zur Identifizierung und Authentifizierung notwendigen Informationen aus und übertragen die

Nutzdaten zur eigentlichen Diensterbringung. Die Zugriffskontrolle erfolgt also weiterhin durch den Rechner, der die Daten und Ressourcen zur Verfügung stellt. Er verwendet dazu die jeweils bestehenden Funktionen des Betriebs- bzw. Dateisystems.

Direkte authentifizierungsbasierte Zugriffskontrolle durch eine Server-Anwendung. Mit der fortschreitenden Entwicklung der Internet-Dienste erbringen spezialisierte Rechner, die Server, spezialisierte Dienste, die sie vielen Clienten, zunächst zumeist ohne Beschränkung des Zugriffs anbieten. Beispiele sind Web-Server im WWW (World Wide Web), Streaming-Server oder Verzeichnis-Server. Bei Nutzung des Client-Server-Paradigmas gibt es nun zwei Klassen von Personen, die mit unterschiedlichen Zielsetzungen auf die Server zugreifen. Dies ist zum einen die Gruppe der Personen, welche die Server-Dienste konfiguriert oder Inhalte bereitstellt, und zum anderen sind es die Nutzer der Dienste. Die erste Gruppe beschränkt sich auf einen bekannten Personenkreis des Dienstanbieters. Sie werden zunächst weiterhin mittels der Mechanismen des Betriebs- und Dateisystems authentifiziert und autorisiert. Die Gruppe der Dienstnutzer hingegen kann wesentlich größer sein. Ihr Zugriff wird vom bisherigen Modell abweichend durch die Server-Anwendung selbst kontrolliert. Basis dazu ist nicht mehr die Benutzer- und Rechteverwaltung des Betriebssystems, sondern eigene Repositories oder Anwendungsdateien, welche die zur Autorisierung notwendigen Informationen speichern.

Die Übermittlung der Identifizierungs- und Authentifizierungs-Informationen vom Client zum Server kann in zwei Formen erfolgen: Das Anwendungsprotokoll sieht in der Signalisierungsphase den Austausch vor. Falls die Autorisierung durch den Server nicht erfolgreich ist, wird das Protokoll mit einer entsprechenden Meldung abgebrochen und der Dienst nicht erbracht. Zu diesen Protokollen zählen beispielsweise das Post Office Protocol (POP) [Mye94] und unter Verwendung der entsprechenden Option das Hypertext Transfer Protocol (HTTP) [FGM+99][FHBH+99]. Alternativ werden die Identifizierungs- und Authentifizierungs-Informationen als Teil der Nutzdaten der Anwendungs-Protokolle übertragen und von der Anwendung selbst geprüft. Dieses Verfahren wird sehr häufig in nicht standardisierten Anwendungen genutzt oder z.B. auch wenn der Dienstnutzer die Benutzerkennung und sein Passwort in ein HTML-Formular eingibt, welches per HTTP zum Server übertragen wird. Die Server-Anwendung nimmt nun die Autorisierung vor und sendet erst im erfolgreichen Fall weitere HTML-Objekte zum Client. Dieses Verfahren wird beispielsweise in Anwendungsfall 4 eingesetzt. Der Protokollablauf ist in Abbildung 79 in Anhang B.4 dargestellt. Neben der Authentifizierung kann auch noch eine dynamische Autorisierung Teil der Zugriffskontrollentscheidung sein, z.B. kann das aktuelle Kreditlimit eines registrierten Kunden zum Zeitpunkt der Dienstanfrage geprüft werden.

Die Authentifizierung erfolgt in den beschriebenen Fällen immer auf Basis einer Authentifizierung vom Dienstanbieter registrierter Dienstnutzer. Ist der Dienst kostenpflichtig, kann anhand der Registrierungsinformationen der Dienstnutzer als Person bestimmt werden und eine Abrechnung erfolgen.

Direkte berechtigungsnachweisbasierte Zugriffskontrolle durch eine Server-Anwendung. Wenn der Dienstanbieter keine privaten Dienste erbringt, sondern mit Hilfe der Zugriffskontrolle nur sicherstellen will, dass seine Dienste bezahlt werden, ist eine Authentifizierung des Dienstnutzers nicht unbedingt notwendig. In diesen Fällen können Autorisierungs-Verfahren auf Basis eines Berechtigungsnachweises zum Einsatz kommen. Der Dienstnutzer muss dazu seine Berechtigung an den Dienstanbieter übermitteln. Der Berechtigungsnachweis wird dabei zumeist ebenfalls als Teil der Nutzdaten der Anwendungs-Protokolle übertragen und von der Anwendung selbst geprüft. Eines der meist verwendeten Beispiele dafür ist die Eingabe der Kreditkarteninformationen in ein HTML-Formular, wie es z.B. in Anwendungsfall 5 geschieht (vgl. Abbildung 80 in Anhang B.5). Bei Nutzung einer Kreditkarte kann als dynamisches Kriterium zusätzlich eine Online-Überprüfung der Kartengültigkeit bei der die Karte ausstellenden Gesellschaft vorgenommen werden.

Weiterhin kann die Zugriffskontrolle auch mit dem Bezahlvorgang verknüpft werden. Dabei wird zumeist unmittelbar vor der Diensterbringung geprüft, ob der Dienstnutzer über ausreichende Zahlungsmittel verfügt. Unmittelbar nach der Diensterbringung erfolgt die Bezahlung in Form einer elektronischen Transaktion.

Single Sign On Systeme. Bei den zuvor vorgestellten Formen der direkten Zugriffskontrolle auf Basis einer Authentifizierung durch den Dienstanbieter ist es notwendig, dass der Dienstnutzer durch verschiedene Dienstanbieter bzw. Rechner und für jede Dienstanfrage neu aktiv identifiziert und authentifiziert wird. Dies ist bei der Verwendung einer Vielzahl von Diensten und in verteilten Systemen sehr benutzerunfreundlich. Single Sign On (SSO) Systeme verfolgen die Zielsetzung, dass der Nutzer sich nur einmalig an einem speziellen Single Sign On Dienst anmeldet und von diesem authentifiziert wird. Die Authentifizierung erfolgt also nicht durch den Dienstanbieter selbst, sondern durch einen vertrauenswürdigen Dritten. Die Autorisierung durch alle innerhalb der Sicherheitsdomäne zur Verfügung stehenden Rechner und Dienste kann dann ohne mehrfache aktive Authentifizierung transparent für den Dienstnutzer vorgenommen werden. Zusätzlich zum Authentifizierungs-Server kann in Single Sign On Systemen auch eine zentrale Rechteverwaltung in Form eines Berechtigungs-Servers existieren. Dieser speichert die Berechtigungen der Benutzer und nimmt direkt die Autorisierung vor oder stellt einen Berechtigungsnachweis aus.

Hinsichtlich ihrer Realisierung lassen sich zwei grundsätzliche Formen von Single Sign On Systemen unterscheiden: Es sind dies die ticketbasierten Systeme und die proxybasierten Systeme. Das Kerberos-Protokoll [KN93] kann zur Realisierung ticketbasierter Systeme genutzt werden. Dessen Funktionsweise soll hier exemplarisch beschrieben werden. Es ist in Abbildung 20 dargestellt. Der Benutzer meldet sich zunächst beim Authentifizierungs-Server (Key Distribution Server) an. Dieser authentifiziert den Benutzer und sendet ein, für den befristeten Zeitraum einer Session gültiges, Session Ticket (Ticket Granting Ticket) zurück an den Dienstnutzer. Fordert der Dienstnutzer einen Dienst an, so sendet er zunächst eine Dienstanfrage zusammen mit dem Session Ticket an einen Berechtigungs-Server (Ticket Granting Server). Dieser Berechtigungs-Server nimmt nun die Prüfung der Berechtigung der Dienstnutzung auf Basis

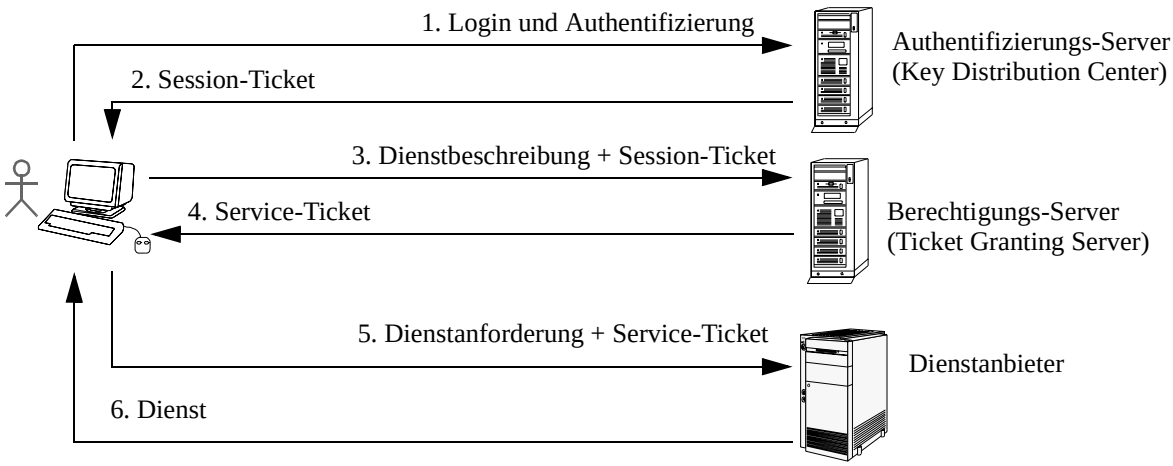

Abbildung 20: Zugriffskontrolle bei ticketbasierten SSO-Systemen am Beispiel Kerberos

der durch das Session-Ticket nachgewiesen authentifizierten Identität vor und stellt ein Service-Ticket aus. Dieses Service-Ticket ist ein Berechtigungsnachweis, wie er in Kapitel 3.1.3 vorgestellt wurde. Es kann vom Dienstnutzer nun bei mehrfach aufeinanderfolgenden Dienstanfragen genutzt werden, um seine Berechtigung gegenüber dem Dienstanbieter nachzuweisen. Die Autorisierung durch den Dienstanbieter erfolgt dann auf Basis des Berechtigungsnachweises. Bei tikketbasierten Systemen werden somit die Authentifizierung und die Überprüfung der Berechtigungen vom Dienstanbieter an einen vertrauenswürdigen Dritten übertragen.

Das Kerberos Protokoll wird vielfach verwendet, beispielsweise im Advanced File System (AFS) [FRR00] oder in Microsofts Active Directory. Eine weitere ticketbasierte Architektur ist in der Secure European Systems for Applications in a Multivendor Environment (SESAME) Architektur [KPP94] spezifiziert.

Proxybasierte Systeme setzen auf existierenden Anwendungen und Protokollen auf und greifen über einen sogenannten SSO-Client als Proxy direkt in die Kommunikation zwischen Dienstnutzer und Dienstanbieter ein. Der SSO-Client ist ein auf dem System des Dienstnutzers ablaufender Prozess. Der Benutzer meldet sich einmalig über den SSO-Client bei einem Authentifizierungs-Server an und wird von diesem authentifiziert. Nachfolgende Authentifizierungs-Anfragen der verschiedenen Dienstanbieter werden vom SSO-Client abgefangen und an den SSO-Server umgeleitet, wie in Abbildung 21 gezeigt. Der SSO-Server verfügt über die anwendungsspezifischen Authentifizierung-Informationen, die er an den SSO-Client sendet, der sie für den Benutzer transparent dem Anwendungs-Server zur Verfügung stellt. Die Authentifizierung und Autorisierung wird unabhängig vom Einsatz des SSO-Clients weiterhin durch den Dienstanbieter vorgenom-

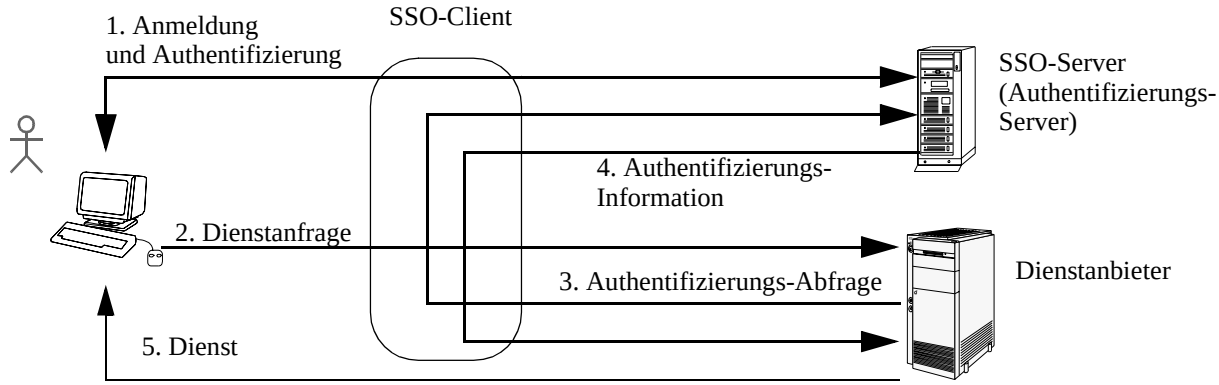

Abbildung 21: Zugriffskontrolle bei proxybasierten SSO-Systemen

men. Beispiele für proxybasierte Systeme sind zumeist herstellerspezifische Produkte wie IBM Tivoli Global Sign On [IBM01] oder Netegrity SiteMinder [Net03].

Die zentrale Benutzeradministration, die in SSO-Systemen durch den Authentifizierungs-Server realisiert ist, vereinfacht den Einsatz von sicheren bzw. starken Authentifizierungs-Verfahren, wie z.B. eine Authentifizierung mittels eines kryptographischen Schlüssels [Jäg02]. Es muss nicht jede einzelne Anwendung dahingehend modifiziert werden, dass sie die entsprechenden Authentifizierungs-Verfahren unterstützt. Bei Herstellerlösungen sind SSO-Systeme oftmals auch mit Systemen zum Aufbau und Betrieb einer Public Key Infrastruktur (PKI) zur Abfrage der Schlüssel bzw. Prüfung der Zertifikate kombiniert.

Identity Management. Die Realisierung der Single Sign On Systeme ist auf eine Sicherheitsdomäne, ein geschlossenes System, beschränkt. Sogenannte Identity Management Systeme unterstützen die Anwendung von SSO in offenen Systemen, d.h. über Domänengrenzen hinweg. Ein vertrauenswürdiger Dritter, der Identity-Provider, speichert die Authentifizierungs-Informationen und Benutzerprofile der Dienstnutzer und nimmt die Authentifizierung vor. Diesen Dienst stellt er mehreren Dienstanbietern zur Verfügung. Identity Management Systeme werden in verschiedenen Bereichen verwendet. Die durch den Identity-Provider gespeicherten Profilinformationen können entsprechend vielfältig sein. Im Bereich des Arbeitslebens sind es Rollen, Berechtigungen und Zuständigkeiten der Person. In der öffentlichen Verwaltung können dazu der Wohnsitz, die Ausweis- oder die Führerscheinnummer zählen. Zuletzt sind es Informationen über die Person als Verbraucher, wie seine Kreditkartennummer oder Einkaufspräferenzen. Das derzeit aktivste Anwendungsfeld von Identity Management Systemen ist das WWW. Die Systeme übernehmen die Authentifizierung von Dienstnutzern für Anbieter von Web-Diensten auf Basis von HTTP oder SSL. Bedeutende Beispiele sind Micosofts .NET Passport [MS02b] und das Liberty Alliance Project [HW03].

Microsofts .NET Passport Service stellt einen vollständig zentralisierten Dienst zur Verfügung. Als Identitätsmerkmal der Dienstnutzer wird eine von Microsoft eindeutig vergebene *Passport* UserID verwendet. Als Authentifizierungs-Information dient ein Passwort. Der zentrale Passport

Identity Server führt einmalig die Authentifizierung durch. Dazu wird ein zu autorisierender HTTP-Request an einen Web-Server an den Identity-Server umgeleitet. Dieser fragt vom Benutzer dessen Benutzerkennung und Passwort ab. Im Falle der erfolgreichen Authentifizierung wird ein Ticket Granting Cookie auf dem Benutzersystem abgelegt. Ein Authentication-Ticket und die Profilinformationen des Nutzers werden per HTTP-Redirect an den Web-Server geschickt. Fragt der Benutzer erneut einen Web-Dienst an, wird diese Abfrage wiederum an den Passport Identity Server umgeleitet, der zunächst überprüft, ob der Browser des Benutzer ein gültiges Ticket Granting Cookie gespeichert hat. Ist dies der Fall, kann die Authentifizierung entfallen und die Autorisierung läuft für den Dienstnutzer transparent ab.

Das Liberty Alliance Project [HW03] verfolgt als Gegenbewegung zum zentralisierten Konzept von .NET Passport eine verteilte, auf einem sogenannten Circle of Trust basierende Variante des Identity Managements. In einem Circle of Trust sind verschiedene Identity-Provider und Anbieter von Endnutzerdiensten zusammengeschlossen. Innerhalb dieses Circle of Trust erfolgt eine gegenseitige Zuordnung der bei verschiedenen Providern bestehenden Benutzerkennungen eines Dienstnutzers und darauf basierend ein SSO. Der Dienstnutzer kann darin selbst steuern, ob eine solche Zuordnung seiner Benutzerkennungen erfolgen soll. Als Protokollmechanismus werden ebenfalls Redirects von HTTP-Requests benutzt.

Die Merkmale der drei Gruppen der direkten Zugriffskontrolle sowie von SSO- und Identity Management Systemen sind in Tabelle 6 zusammengefasst.

	Kontrolle von								Ident.		Autorisierung				Modell		Konfig.	
	Verbindungsdienste	Internet-Zugangsdienst	QoS-Transportdienste	Anwendungsdienste	Inhaltsdiensten	fremden Dienstnutzern	anonymen Dienstnutzern	privaten Diensten	Merkmal einer Person	Merkmal eines Geräts	authentifizierungsbasiert	berechtigungsbasiert	statisch	dynamisch	integriertes Modell	Third-Party-Modell	statstisch	dynamisch
Betriebssystem bas.				x	x			x	x		x		x		x		x	
Authentifizier. bas.				x	x			x	x		x			x	x		x	
Berechtigungs. bas.				x	x	x	x					x		x	x		x	
Ticketbasiertes SSO				x	x			x	x			x	x			x	x	
Proxybasiertes SSO				x	x			x	x		x		x		x		x	
Identity-Mgmnt.				Web		x			x			x	x			x	x	

Tabelle 6: Merkmale von Systemen zur Kontrolle von Anwendungs- und Inhaltsdiensten

3.3.5 Zugriffskontrolle durch Firewalls

Die Zugriffskontrolle durch Firewalls stellt einen Spezialfall der Zugriffskontrolle dar. Firewalls dienen im Allgemeinen zur Sicherstellung der Vertraulichkeit und Integrität der Ressourcen innerhalb eines privaten Intranets und zum Schutz vor verschiedensten Angriffen aus dem Internet. Dazu ist es notwendig, den Zugriff zwischen dem Intranet und dem Internet zu beschränken. Firewalls führen eine solche Zugriffskontrolle aus, indem sie alle über eine Netzgrenze fließenden Daten kontrollieren und gegebenenfalls auch modifizieren [Roe02]. Sie sind also oftmals zwischen einem Intranet und dem öffentlichen Internet lokalisiert.

Die einfachste Form von Firewalls sind Paketfilter. Ein Paketfilter ist eine Teilkomponente eines Routers, der zum Verbinden der Netzwerke und zur Weiterleitung der Pakete zwischen den Netzen verwendet wird. Er kontrolliert jedes einzelne weiterzuleitende IP-Paket anhand verschiedener Informationen, wie z.B. der IP-Absender- und Zieladresse, der Art des Pakets (TCP oder UDP) und der TCP-Portnummer. Diese Form der Zugriffskontrolle entspricht damit nicht dem Gegenstand der Arbeit. Gleiches gilt auch für die zweite Ausprägung der Firewalls, die Stateful-Filter. Sie erweitern die Funktionalität von Paketfiltern, indem sie die Zustände von Verbindungen überwachen können. Dazu wird nicht jedes einkommende Paket isoliert betrachtet, sondern in die Entscheidung über eine Weiterleitung können Informationen über vorhergehende Pakete mit einbezogen werden. Dennoch wird weiterhin für jedes einzelne Paket eine Kontrollentscheidung getroffen.

Proxies sind die dritte wichtige Form von Firewalls. Sie trennen die Verbindung zwischen den Netzen komplett auf und stellen für beide Seiten den Endpunkt der Kommunikation dar, d.h. sie implementieren client- und serverseitig die kompletten Anwendungsprotokolle. Auf die Informationen der einzelnen Pakete können sie nur eingeschränkt zugreifen. Als Entscheidungskriterium für die Zugriffskontrolle verwenden sie daher aussschließlich benutzerbezogenen Informationen [Roe02]. Ein Proxy kann die Authentifizierung des Benutzers einer Anwendung verlangen, bevor er die Verarbeitung der Protokoll fortführt. Die Benutzerinformationen werden zum Zeitpunkt der Signalisierung ausgewertet und erlauben nachfolgenden Paketen, die Dienstnutzung in Form der Weiterleitung der Pakete. Damit können Proxies die Nutzung von Anwendungen durch Dienstnutzer einschränken. Die Konfiguration der Proxies erfolgt dabei nicht durch den Dienstanbieter selbst, sondern durch die das Intranet betreibende Organisation. Bei Dienstanfragen aus dem Intranet an einen Internet-Dienst kann also der Zugriff auf den Dienst nicht grundsätzlich kontrolliert werden. Bei entgegengerichteten Dienstanfragen können nur bekannte Nutzer, die sich außerhalb des Intranets befinden, kontrolliert werden. Hier ist die Kontrollmöglichkeit also stark eingeschränkt. Tabelle 7 zeigt die Merkmale von Proxies im Überblick

	Kontrolle von								Ident.		Autorisierung				Modell		Konfig.	
	Verbindungsdienste	Internet-Zugangsdienst	QoS-Transportdienste	Anwendungsdienste	Inhaltsdiensten	fremden Dienstnutzern	anonymen Dienstnutzern	privaten Diensten	Merkmal einer Person	Merkmal eines Geräts	authentifizierungsbasiert	berechtigungsbasiert	statisch	dynamisch	integriertes Modell	Third-Party-Modell	statisch	dynamisch
Firewall-Proxies				x	x			x	x		x		x			x	x	

Tabelle 7: Merkmale eine Zugriffskontrolle durch Firewall-Proxies

3.3.6 IRTF AAA-Architektur

Bei der IRTF AAA-Architektur handelt es sich im Gegensatz zu den bisher vorgestellten Lösungen nicht um ein existierendes System, sondern zunächst um ein Konzept. Während die IETF Authenticiation Authorization Accounting Arbeitsgruppe [AM03] ein AAA-Protokoll für die Kontrolle von Internet-Zugangsdiensten in verschiedenen Szenarien entwickeln will (vgl. Kapitel 3.4.1), verfolgt die aus dieser Arbeitsgruppe hervorgegangene IRTF Forschungsgruppe AAA-Architecture [VdL01] ein anderes Ziel. Es besteht darin, eine Architektur für ein Authentifizierungs-, Autorisierungs- und Accounting-System zu definieren, welches über die Grenzen einer Organisation nutzbar und für verschiedene Internet-Dienste erweiterbar ist. Die Zielsetzung stimmt damit in vielen Teilen mit der dieser Arbeit überein. Die Arbeit der Gruppe basiert auf einem Autorisierungs-Framework [VCF+00], welches grundlegende Autorisierungs-Modelle beschreibt. Als Basisparadigma wird das Policy-Modell (vgl. Kapitel 2.7) angewandt. Die Architektur ist im wesentlichen in [dLGG+00] beschrieben. Sie basiert auf der Annahme, dass in jeder Sicherheitsdomäne mindestens ein AAA-Server existiert, der Autorisierungs- und Accounting-Dienste erbringt. Die wesentlichen Komponenten der Architektur, ihre Dienste und Protokolle sollen nachfolgend erläutert werden.

AAA-Komponenten. Ein AAA-Server besteht aus einer sogenannten Rule based Engine (RBE) als zentraler Komponente. Sie fragt die in einem Policy-Repository (PR) gespeicherten Policies ab, evaluiert Policy-Bedingungen, nimmt Policy-Entscheidungen vor und führt entsprechende Policy-Aktionen aus. Die Durchsetzung (Enforcement) von Policy-Aktionen erfolgt in Abhängigkeit von der Art des Dienstes durch verschiedene Komponenten. Die meisten Aktionen im Zusammenhang mit der Policy-Durchsetzung müssen durch das Service Equipment (SE) der Dienanbieter und nicht durch das AAA-System ausgeführt werden. Andere Aktionen, z.B. das Accounting erfolgen durch die AAA-Server selbst [dLGG+00]. Innerhalb der IRTF Research Gruppe liegt der primäre Fokus auf Definition und Anwendung von Autorisierungs-Policies und Accounting-Policies. Die Authentifizierung wird nicht betrachtet.

AAA-Dienste. Die AAA-Server bieten den Dienstanbietern Autorisierungs- und Accounting-Dienste an. Der Autorisierungs-Dienst umfasst dabei das Treffen der Entscheidung, ob die Anfrage eines Dienstnutzers erfüllt oder zurückgewiesen wird. Eine zu erfüllende Dienstanfrage wird innerhalb der Architektur in eine autorisierte Session transformiert. Das Service Equipment wird für die Diensterbringung konfiguriert und der Status der Session gespeichert. Die Accounting-Dienste zeichnen die notwendigen Accounting-Informationen aus der Autorisierungs-Entscheidung und den fortlaufenden Ressourcenverbrauch der Session auf. Die Authentifizierung ist nicht Teil dieser Dienste. Um AAA-Dienste domänenübergreifend zu erbringen, sind Vertrauensverhältnisse zwischen verschiedenen AAA-Serven notwendig. Mittels eines Vertrages baut der Dienstnutzer ein Vertrauensverhältnis zu seinem Heimat-Dienstanbieter, der sogenannten User Home Organization (UHO) auf. Dieser Heimat-Dienstanbieter betreibt wie alle anderen Dienstanbieter auch einen AAA-Server. Besteht eine Kette von Vertrauensverhältnissen zwischen dem Heimat-Dienstanbieter und einem fremden Dienstanbieter, bzw. zwischen ihren AAA-Servern, so kann der fremde Dienstanbieter einem Dienstanfrager vertrauen.

AAA-Architektur und -Protokolle. Die zuvor vorgestellten Komponenten sind in einer AAA-Architektur, wie sie in Abbildung 22 dargestellt ist, strukturiert. Die Rule Based Engine ist der zentrale Teil des AAA-Servers. Der AAA-Server empfängt Dienstanfragen vom Service-Equipment des Dienstanbieters über ein sogenanntes Application Specific Module (ASM) oder von anderen AAA-Servern. Er wertet die erhaltene Anfrage unter Berücksichtigung der Policies aus dem Policy-Repository aus. Dazu kann es notwendig sein, eine Anfrage an einen anderen AAA-Server zu stellen oder den Status des Service-Equipments des Dienstanbieters zu berücksichtigen. Um des Status des Service-Equipments abzufragen, wird das ASM genutzt. Die ASMs sind zudem notwendig, um Policy-Aktionen auszuführen. Dazu können entweder die ASMs das Service-Equipment konfigurieren oder die AAA-Server selbst Dienste zur Verfügung stellen. Zu letzterem zählen die Speicherung des Session-Status, die Sammlung von Accounting-Daten und das Aufzeichnen von Aktionen [dLGG+00].

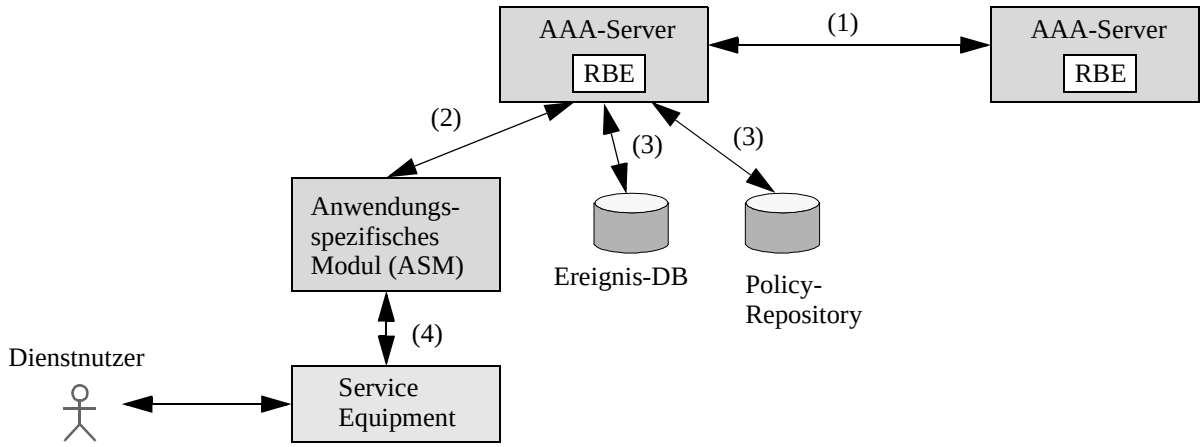

Abbildung 22: IRTF AAA-Architektur

Innerhalb der Architektur sollen folgende Protokolle verwendet, werden, die aber noch nicht vollständig spezifiziert sind:

- ein spezialisiertes AAA-Protokoll (1),
- ein Application Programming Interface (API) oder ebenfalls ein AAA-Protokoll (2),
- abhängig von der Implementierung des Policy-Repositories das Light-weight Directory Access Protokoll (LDAP) oder ebenfalls eine API (3),
- ein anwendungsspezifisches Protokoll (4) .

Die Arbeit der IRTF-Forschungsgruppe beschränkt sich neben den konzeptionellen Grundlagen vertieft auf die Untersuchung eines policybasierten Accountings [ZZC02].

Anwendung der Konzepte der AAA-Architektur im Moby Dick Projekt. Die in der AAA-Architecture Forschungsgruppe entwickelten Konzepte dienen als Basis zur Entwicklung eines Zugriffskontroll- sowie Accounting- und Charging-Systems im Projekt Moby Dick [MDK03]. Der Anwendungsschwerpunkt innerhalb dieses Projekts liegt auf der Unterstützung mobiler Dienstnutzer und der Kontrolle und Abrechnung von QoS-Transportdiensten. Als AAA-Protokoll wird innerhalb der Moby Dick Architektur Diameter (vgl. Kapitel 3.4.1) verwendet. Die Funktionen des Accounting und Charging werden von ASMs ausgeführt. Für die Authentifizierung können verschiedene externe Module genutzt werden[HJZS01].

3.4 Protokolle für die Zugriffskontrolle

Protokolle für eine Zugriffskontrolle auf die verschiedenen Klassen von Internet-Diensten sind teilweise bereits bei der Vorstellung der unterschiedlichen Architekturen angesprochen worden. Sie sollen hier nochmals zusammengefasst, vertieft erläutert und zudem um Weitere ergänzt werden.

3.4.1 Protokolle zur Kontrolle des Zugriffs auf Internet-Zugangsdienste

Wie in Kapitel 3.3.2 beschrieben, wird zur Kommunikation zwischen dem Endgerät und dem Netzwerkzugangsrechner ein sogenanntes Punkt-zu-Punkt Protokoll oder ein lokales Netzwerkprotokoll verwendet. Für Punkt-zu-Punkt Verbindungen wird zumeist PPP [Sim94] genutzt. PPP erlaubt die Aushandlung eines Authentifizierungs-Protokolls.

Zu diesen zählt PPP-PAP [LS92]. Bei einer Nutzung von PAP werden Identifizierung und Authentifizierung anhand der Benutzerkennung und des Passworts des Dienstnutzers vorgenommen. PPP-CHAP [Sim96] realisiert ein Challenge Response Verfahren auf Basis des Passworts als gemeinsames Geheimnis von Dienstnutzer und Dienstanbieter. Des Dienstnutzer identifiziert sich mittels seiner Benutzerkennung, erhält vom Dienstanbieter ein Challenge und antwortet mit einem Response. Der Vorteil von CHAP gegenüber PAP liegt darin, dass das Passwort nicht im Klartext übertragen werden muss. PPP-EAP [BV98] sieht die Aushandlung verschiedener Authentifizierungs-Verfahren vor. Über EAP kann derjenige, der die Authentifizierung vornimmt, mehrfach beliebige, für die Identifizierung und Authentifizierung notwendige Informationen vom

Dienstnutzer anfordern. In EAP vorgesehen sind die Verwendung von Challenge Response Verfahren, Einmalpasswörtern oder Tokencards.

Neben den Authentifizierungs-Protokollen wird zur Kontrolle des Zugriffs auf Internet-Zugangsdienste, nach dem in Kapitel 3.3.2 vorgestellten allgemeinen Modell, ein AAA-Protokoll zur Kommunikation zwischen dem Netzwerkzugangsrechner und dem AAA-Server benötigt. Innerhalb der IETF diskutiert die Authentication Authorization Accounting Arbeitsgruppe [AM03] geeignete Protokolle. Ihre Aufgabe besteht in der Definition von Anforderungen an AAA-Protokolle [ACG+00], der Beurteilung existierender Protokolle [MJB+01] und der Spezifikation eines neuen Protokolls.

Die wichtigsten AAA-Protokollvertreter sind das Remote Authentication Dial In User Service (RADIUS) Protokoll und dessen Weiterentwicklung Diameter. Weiterhin lassen sich dazu das Common Open Policy Service (COPS) Protokoll und das Simple Network Management Protocol (SNMP) zählen.

RADIUS [RWRS00] wird verwendet, um Authentifizierungs-, Autorisierungs- und Konfigurationsinformationen zwischen dem Internet-Zugangsrechner und einem RADIUS-Server auszutauschen. Der Zugangsrechner agiert als Client gegenüber dem RADIUS-Server, der die Authentifizierung und Autorisierung und als Erweiterung [Rig00] das Accounting vornimmt. Jeder RADIUS-Sever kann selbst wiederum als Client gegenüber einem anderen RADIUS-Server auftreten. RADIUS sieht die Nutzung von PAP, CHAP und EAP als Authentifizierungs-Protokoll vor. Wurde RADIUS ursprünglich für eine Zugriffskontrolle für Einwahl-Server entwickelt, so wird es heute in vielen weiteren, beispielsweise Mobilität unterstützenden Szenarien eingesetzt. Allerdings besitzt RADIUS eine Reihe von dem Protokoll inhärenten Schwächen [CLG+02]. Dazu zählt die Verwendung von UDP als Transportprotokoll, was beispielsweise die Erkennung von Ausfällen von RADIUS-Servern erschwert und die Anzahl der Retransmissions auf Anwendungsebene erhöht. Die Unterstützung von RADIUS-Servern als Proxies ist unzureichend. Ausserdem existieren in RADIUS Sicherheitslücken. So gibt es z.B. keinen Mechanismus zum Schutz vor Replay-Angriffen und keine Unterstützung der Ende-zu-Ende Sicherheit.

Um die Schwächen von RADIUS zu beseitigen, spezifiziert die IETF AAA Arbeitsgruppe Diameter als Nachfolger des RADIUS-Protokolls. Es besteht aus einem Basisprotokoll [CLG+02], welches Header-Formate, Sicherheitserweiterungen und sogenannte Attribute-Werte Paare (AVPs) definiert, die dem Austausch der eigentlichen Authentifizierungs-, Autorisierungs- und Accounting-Informationen dienen. Daher können verschiedene Authentifizierungs-Verfahren realisiert werden. Vorgesehen sind derzeit die in EAP verwendeten Verfahren [HZ03].

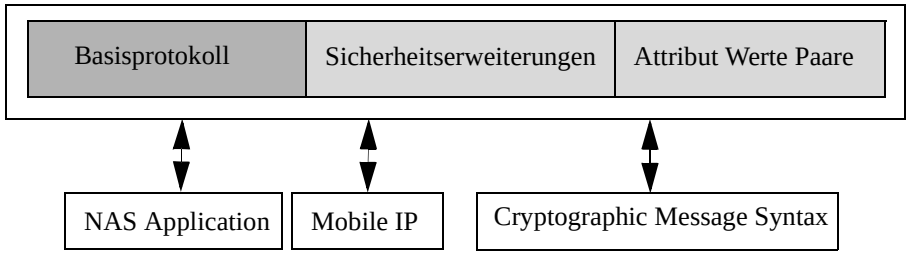

Abbildung 23: Diameter Protokollarchitektur

Die speziellen Protokollelemente zur Verwendung von Diameter bei Nutzung verschiedener Zugangstechnologien sind als Ergänzung zum Basisprotokoll in Erweiterungen definiert. Die Network Access Server Application [CZSM03] beschreiben die Kontrolle des allgemeinen Zugriffs auf einen Internet-Zugangsrechner. Die Mobile IP Erweiterungen [CJP03] dienen zur Unterstützung der Zugriffskontrolle von Mobile IP Knoten über verschiedene administrative Domänen hinweg. Als Identitätsmerkmal verwendet der mobile Knoten dabei seinen Network Access Identifier (NAI) [AB99]. Eine weitere Ergänzung umfasst Verfahren zur Ende-zu-Ende Sicherung einzelner Attribute Werte Paare innerhalb der Diameter-Nachrichten mittels der Cryptographic Message Syntax (CMS) [CFB02][Hou02].

3.4.2 Protokolle zur Kontrolle des Zugriffs auf QoS-Transportdienste

Bei der Kontrolle des Zugriffs auf QoS-Transportdienste innerhalb der Integrated-Services Architecture wird, wie in Kapitel 3.3.3 erläutert, das COPS-Protokoll [DBC+00] für die Kommunikation zwischen Router und Policy Decision Point verwendet. Das Protokolldesign von COPS erlaubt darüberhinaus eine Anwendung in einem breiteren Kontext. COPS und seine Ergänzung [CSD+01] stellen ein grundsätzliches Protokoll zum Austausch von Informationen innerhalb einer Policy-Architektur, wie sie Kapitel 2.7.2 vorgestellt wurde, dar. Es handelt sich um ein einfaches Frage Antwort Protokoll, mit welchem Informationen zwischen einem Policy Enforcement Point (PEP) als Client und einem Policy Decision Point (PDP) als Server ausgetauscht werden können. COPS verwendet eine allgemeine PEP-ID zur Identifizierung des Clients. Dieser muss innerhalb der Sicherheitsdomäne eindeutig sein. Die IP-Adresse oder der DNS-Rechnername des Clients werden zumeist genutzt. Die gegenseitige Authentifizierung von Client und Server erfolgt unter Einsatz eines gemeinsamen Geheimnisses.

3.4.3 Protokolle zur Kontrolle des Zugriffs auf Anwendungs- und Inhaltsdienste

In der Klasse der Anwendungs- und Inhaltsdienste können, wie in Kapitel 3.3.4 dargestellt, die Anwendungsprotokolle oftmals auch zur Zugriffskontrolle genutzt werden. Beispielsweise unterstützen POP [Mye94] oder FTP [PR85] eine Identifizierung und Authentifizierung mittels Benutzerkennung und Passwort. HTTP/1.1 [FGM+99] ermöglicht als Option ebenfalls eine Zugriffskontrolle mittels Benutzerkennung und Passwort. Als Erweiterung ist für HTTP die Möglichkeit spezifiziert, das Passwort nicht im Klartext zu übertragen, sondern ein Challenge

Response Verfahren zu nutzen [FHBH+99]. Die HTTP-Authentifizierung wird in der Regel nur dann genutzt, wenn private Inhalte nur einer begrenzten Gruppe zugänglich sein sollen. Diese erhalten dann auf einem sicheren Wege die Benutzerkennung und das Passwort.

Daneben erfolgt die Zugriffskontrolle zumeist über anwendungsspezifische Protokolle, die nicht standardisiert sind, oder durch die Anwendung selbst. Wie in Kapitel 3.3.4 erläutert, können auch elektronische Bezahlverfahren als Zugriffskontrollprotokolle angesehen werden. Drei wesentliche Methoden sollen hier kurz vorgestellt werden. Auf andere, wie Paybox [pA02] oder die Bezahlung mit der Geldkarte über das Internet [fc03], sei nur verwiesen.

Bei den kontenbasierten Verfahren wird die Bezahlung und Leistungsbereitstellung über einen spezialisierten Anwendungsdienstanbieter als vertrauenswürdigen Dritten abgewickelt [CSW97]. Will der Dienstnutzer über HTTP auf einen gebührenpflichtigen Inhaltsdienst zugreifen, so wird seine Anfrage zu diesem Dienstanbieter umgeleitet. Dort muss der Dienstnutzer vorab registriert sein oder sich aktuell registrieren. Der Benutzer gibt dann in ein HTML-Formular seine Benutzerkennung und sein Passwort ein und wird damit authentifiziert. Vom Anwendungsdienstanbieter und nicht vom eigentlichen Inhaltsdienstanbieter erhält er dann den angefragten Inhaltsdienst. Der Anwendungsdienstanbieter verbucht die Leistungserbringung auf dem Kundenkonto und stellt dem Kunden periodisch eine Rechnung. Die Gebühren erhält der Inhaltsanbieter unter Abzug einer Provision. Ein erfolgreiches Beispiel für einen solchen Anwendungsdienstanbieters ist in Deutschland die Firstgate AG mit “Click and Buy” [FIA03].

Bei geldartigen Verfahren, wie z.B. Millicent [GMAG95], besitzt der Dienstnutzer digitales von einer spezialisierten Bank ausgegebenes und auf Basis von blinden Signaturen signiertes Geld [Cha83]. Dieses transferiert er im Rahmen der Dienstanfrage und -erbringung an den Dienstanbieter. Die geldartigen Verfahren ermöglichen eine anonyme oder zumindest pseudonyme Dienstnutzung; es erfolgt keine Identifizierung und Authentifizierung [CSW97].

Sogenannte Dialer, wie z.B. Net900 bauen, beim Zugriff auf einen gebührenpflichtigen Inhalt die bestehende Wählverbindung des Dienstnutzers ab und stattdessen eine höher tarifierte neue Verbindung auf [Mül02]. Über diese Verbindung wird dann der Inhalt übertragen und zugleich damit auch der eigentlich angefragte Dienst über den Anbieter des Verbindungsdienstes abgerechnet. Zwischen den zwei Dienstanbietern erfolgt dann wiederum eine Verrechnung der Gebühren.

3.4.4 Weiter Protokolle für die Zugriffskontrolle

Die verschiedenen im Rahmen der Zugriffskontrolle eingesetzten Protokolle sind in Abbildung 24 zusammenfassend dargestellt. Neben den zuvor beschriebenen sind hier noch weitere aufgenommen, mittels derer auch eine Zugriffskontrolle realisiert werden kann, die aber eigentlich zu anderen Zwecken spezifiziert wurden.

Bedeutsam im Zusammenhang der Kontrolle des Zugriffs auf Anwendungs- und Inhaltsdienste ist das Secure Socket Layer (SSL) Protokoll. Es ist anwendungsunabhängig und dient zur Sicherung von Protokollen der Anwendungsebene, die keine eigenen Mechanismen zur Realisierung der Kommunikationssicherheit vorsehen. Das SSL-Protokoll ist technisch zwischen der Transport- und der Anwendungsschicht angesiedelt. Es ermöglicht eine anwendungsunabhängige

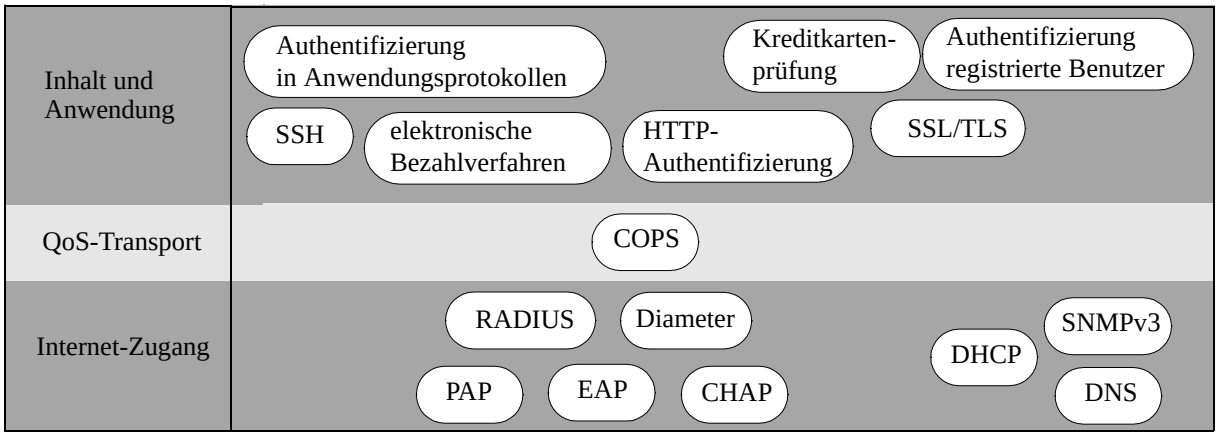

Abbildung 24: Überblick über Protokolle für die Zugriffskontrolle

Authentifizierung des Dienstanbieters oder beidseitig von Dienstanbieter und Dienstnutzer. Dazu werden als Identitätsnachweis Zertifikate nach dem Standard X.509 [ITU97] ausgetauscht und verifiziert. Weiterhin wird die Transportverbindung durch einen Schlüsselaustausch abgesichert, d.h. die Vertraulichkeit, Integrität und Authentizität der ausgetauschten Nachrichten wird hergestellt. SSL 3.0 ist als Teil von Transport Layer Security (TLS) in [DA99] spezifiziert.

SNMP ist in der Version 3 [HPW02] wie auch COPS innerhalb der IETF als AAA-Protokoll diskutiert worden. Beide werden aber dazu in der Regel nicht eingesetzt. Das erweiterte Dynamic Host Configuration Protocol (DHCP) [DA01] und das Domain Name System (DNS) [Eas99] können in Spezialfällen auch als Zugriffskontrollprotokolle angesehen werden [RHKS01].

Secure Shell (SSH) definiert eine Client-Server-Anwendung, mit deren Hilfe eine einfache sichere Transportverbindung zwischen Client und Server auf Basis von TCP aufgebaut wird [FRR00]. Voraussetzung für den Verbindungsaufbau ist eine erfolgreiche Authentifizierung des Servers während der Schlüsselaustauschphase. Dazu verwendet er einen öffentlichen Schlüssel, der mit dem auf dem Client-System gespeicherten verglichen wird. Über die gesicherte SSH-Transportverbindung kann im zweiten Schritt eine Authentifizierung des Dienstnutzers erfolgen. Dieser identifiziert sich durch seine Benutzerkennung. Der Server kann nun dienstspezifisch verschiedene Authentifizierungs-Informationen anfordern. SSH wird ursprünglich anstelle von Telnet für sichere Remote-Logins verwendet. Aufgrund der Möglichkeit des Aufbaus einer allgemeinen sicheren Transportverbindung kann es auch andere Anwendungen sichern.

3.5 Zusammenfassung

In diesem Kapitel wurden zunächst verschiedene Verfahren für die Identifizierung, Authentifizierung und Autorisierung als Teilfunktionen der Zugriffskontrolle vorgestellt. Eine Autorisierung basiert grundsätzlich entweder auf einer erfolgreichen Authentifizierung oder der erfolgreichen Prüfung eines Berechtigungsnachweises. Zusätzlich zu dieser jeweiligen Prüfung kann noch die Validierung verschiedener dynamischer Kriterien in die Zugriffskontrollentscheidung einbezogen werden.

Die verschiedenen existierenden Zugriffskontrollsysteme, die in diesem Kapitel vorgestellte wurden, unterscheiden sich neben den eingesetzten Verfahren auch in ihrem Funktionalitätsumfang und in der ihnen zugrundeliegenden Architektur, insbesondere ob die Zugriffskontrolle als Teil der Diensterbringung erfolgt oder im Third Party Modell ausgelagert ist.

	Kontrolle von								Ident.		Autorisierung				Modell		Konfig.	
	Verbindungsdiensten	Internet-Zugangsdiensten	QoS-Transportdiensten	Anwendungsdiensten	Inhaltsdiensten	fremden Dienstnutzern	anonymen Dienstnutzern	privaten Diensten	Merkmal einer Person	Merkmal eines Geräts	authentifizierungsbasiert	berechtigungsbasiert	statisch	dynamisch	integriertes Modell	Third Party Modell	statisch	dynamisch
802.11	x									x		x	x		x		x	
GSM	x					x				x	x			x	x		x	
allg. AAA-Server		x				x			x		x		x			x	x	
IntServ			x			x			x		x			x		x		x
Betriebssystem bas.				x	x			x	x		x		x		x		x	
Authentifizier. bas.				x	x			x	x		x			x	x		x	
Berechtigungs. bas.				x	x	x	x					x		x	x		x	
Ticketbasiertes SSO				x	x			x	x			x	x			x	x	
Proxybasiertes SSO				x	x			x	x		x		x		x		x	
Identity-Mgmnt.				Web		x			x			x	x			x	x	
Firewall-Proxies				x	x			x	x		x		x			x	x	

Tabelle 8: Überblick über die Merkmale existierender Zugriffskontrollsysteme

Betrachtet man die Merkmale der verschiedenen erläuterten Systeme, wie sie Tabelle 8 zusammengefasst sind, im Überblick, so fällt dabei auf, dass für eine Kontrolle der Dienste der verschiedenen Klassen jeweils unterschiedliche Systeme genutzt werden und sich auch die zugrundeliegenden Architekturen unterscheiden. Die Kontrolle von Anwendungs- und Inhaltsdiensten weist die größte Vielfalt auf, wobei nicht jedes System in allen Szenarien und für alle Dienste zu nutzen ist. Es ist zudem abhängig vom Bezahlverfahren und damit teilweise vom Geschäftsmodell des Dienstanbieters. Es erfolgt in den vorgestellten Systemen immer eine direkte Zugriffskontrolle durch die Anwendung selbst, wobei sowohl eine authentifizierungs- als auch eine berechtigungsbasierte Autorisierung verwendet werden. Die Kontrolle des Zugriffs auf Internet-Zugänge erfolgt letztendlich unabhängig von der genutzten Zugangstechnologie immer mit-

tels eines einheitlichen Basismodells auf Basis einer Authentifizierung, die von einem ausgelagerten Server übernommen wird. Sie unterstützt auch die Mobilität der Dienstnutzer.

Die AAA-Architecture Forschungsgruppe der IRTF verfolgt eine ähnliche Zielsetzung wie diese Arbeit. Sie basiert mit der Realisierung des policybasierten Managements auf einem Ansatz, der die flexible Konfiguration des Systems ermöglicht. Dieser wird aber nicht konsequent umgesetzt. Die AAA-Architektur besitzt deutliche Schwächen: Die Funktion der Policy-Auswertung und der Policy-Durchsetzung sind nicht klar voneinander getrennt. So ist es möglich, dass zur Policy-Auswertung bereits ein Authentifizierungs-Ergebnis vorliegen muss. Die Authentifizierung wird aber per Funktionsaufruf innerhalb der Policy spezifiziert [TSdL03]. Konsequent mittels einer Policy gesteuert wird nur das Accounting. Eine Erweiterbarkeit der Funktionen eines AAA-Servers über Autorisierung und Accounting hinaus ist nur schwer möglich, da die einzelnen Komponenten nicht generisch definiert sind. Viele Funktionen der Policy-Durchsetzung sind Teil des AAA-Servers selbst oder des Application Specific Modules. Das Accounting wird bisher nur für Internet-Zugangs und Transportdienste definiert. Eine generelle Nutzung der AAA-Dienste für Inhalts- und Anwendungsdienste ist daher noch nicht möglich. Die Funktion des Application Specific Modules ist nicht eindeutig definiert. Es hat den Anschein, dass es als Platzhalter für diejenigen Aufgaben dient, die nicht anderen Komponenten zugeordnet werden können. Die AAA-Architecture scheint daher nicht die geeignete Grundlage für ein generisches AAA-System, wie es von Internet-Dienstanbietern benötigt wird, darzustellen.

Kapitel 4: Policies für Geschäftsmodelle und Unterstützungsdienste

Das Angebot an Internet-Diensten wächst aufgrund der Kürze der Innovationszyklen und aufgrund des Markteintritts immer neuer Anbieter sehr schnell. Die Anbieter sind häufig gezwungen, auf Marktentwicklungen und Konkurrenzverhalten zu reagieren. Sie müssen fortlaufend ihre Geschäftsmodelle überprüfen und gegebenenfalls an neue Gegebenheiten anpassen. Ein Teilziel der Arbeit besteht aus diesem Grund darin, dass die zu entwickelnde Architektur in dem Sinne generisch ist, dass sie unabhängig von den Geschäftsmodellen der Internet-Dienstanbieter verwendbar ist und der Dienstanbieter das Zugriffskontroll- und Abrechnungssystem in Abhängigkeit von seinem Geschäftsmodell flexibel konfigurieren kann. Um der ersten Teilforderung nachzukommen, möglichst alle verschiedenen Ausprägungen von Geschäftsmodellen unterstützen zu können, muss das Zugriffskontroll- und Abrechnungssystem möglichst viele Formen der Zugriffskontrolle und Abrechnung realisieren, wie sie im vorhergehenden Kapitel vorgestellt wurden. Um zweitens das System flexibel konfigurieren zu können, wird in der zu definierenden Architektur das Paradigma des policybasierten Managements angewandt. Dazu muss der Dienstanbieter Policies definieren, die spezifizieren, wie die Unterstützungsdienste in Abhängigkeit vom durch den Dienstnutzer angefragten Dienst durchzuführen sind.

Bevor die Umsetzung des policybasierten Managements innerhalb der Architektur beschrieben wird, werden in diesem Kapitel die Zusammenhänge zwischen dem Geschäftsmodell und den Unterstützungsdiensten untersucht. Beide müssen in integrierter Weise betrachtet werden, denn das Geschäftsmodell beeinflusst die Gestaltung der Unterstützungsdienste und die Realisierung der Unterstützungsdienste bilden eine Vorraussetzung für die Umsetzung von Geschäftsmodellen. Die existierenden Zusammenhänge werden anhand eines eigenen Policy-Modells illustriert, welches nachfolgend vorgestellt wird. Elemente des Policy-Modells sind zum einen Policies der einzelnen typischen Bestandteile eines Geschäftsmodells und zum anderen Policies der Teilfunktionen der jeweiligen Unterstützungsdienste.

Das Geschäftsmodell und die Policies der Unterstützungsdienste werden jeweils in einer auf der eXtensible Markup Language (XML) basierenden Policy-Sprache beschrieben. Zur Definition der Syntax der Sprache wird XML-Schema verwendet. Die Sprachen und ihre Syntaxdefinition werden in Kapitel 4.3 und 4.4 detailliert vorgestellt. Die Sprache für die Unterstützungsdienste dient dann im Weiteren auch dazu, das Zugriffskontroll- und Abrechnungssystem zu konfigurieren. Im Rahmen der Diskussion der Sprachen werden die Zusammenhänge zwischen den einzelnen Policies erörtert, die dann zum Abschluß des Kapitels erneut im Policy-Modell dargestellt werden. Zu Beginn des Kapitels wird vorab kurz definiert, was unter einem Geschäftsmodell zu

verstehen ist und warum existierende Ansätze zur formalen Beschreibung von Geschäftsmodellen nicht verwendet werden konnten.

4.1 Ansätze zur formalen Beschreibung von Geschäftsmodellen

Der Begriff *Geschäftsmodell* oder *Business-Modell* wird insbesondere innerhalb der Internet-Ökonomie sehr häufig verwendet. Was ein Geschäftsmodell ist, bleibt dabei oftmals sehr unscharf. So existieren viele verschiedene Interpretationen. Ihnen gemein ist, dass ein Geschäftsmodell sich auf die Aspekte Unternehmensstrategie, Unternehmensressourcen, unternehmerisches Umfeld und Wettbewerbsvorteile bezieht und eine Umsetzung der Strategie des Unternehmens unter Berücksichtigung dieser Aspekte beschreibt [Sch02]. Geschäftsmodelle werden also auf Basis einer langfristigen Strategie definiert, mit dem Ziel diese umzusetzen. In dieser Arbeit wird unter einem Geschäftsmodell eine Beschreibung von drei wesentlichen Aspekten der Geschäftstätigkeit eines Internet-Dienstanbieters verstanden, nämlich eine Beschreibung der angebotenen Dienste selbst, der potentiellen Dienstnutzer und der kaufmännischen Regeln, die der Anbieter für die Nutzung der Dienste durch den Kunden bestimmt.

Die Definition von Geschäftsmodellen beeinflusst die Gestaltung der operativen Prozesse eines Unternehmens. Zum einen bestimmt das Geschäftsmodell direkt die Ausgestaltung der angebotenen Internet-Dienste selbst, zum anderen indirekt auch die Unterstützungsdienste, wie sie in Kapitel 2 definiert wurden. Eine Anpassung des Geschäftsmodells geht zumeist mit einer Änderung der kaufmännischen Funktionen und der Zugriffskontrolle einher. Aus dem Geschäftsmodell resultieren Anforderungen an die operative Ebene, wohingegen die technologischen Fähigkeiten die Strategie und das Geschäftsmodell beeinflussen, wie in Abbildung 25 in Anlehnung an [SBJT02] illustriert. Strategische Fragen und operative Aspekte wurden von Dienstanbietern dennoch bisher zumeist getrennt betrachtet [STB02]. Eine integrierte Betrachtung, welche die Zusammenhänge berücksichtigt, ist aber sinnvoll.

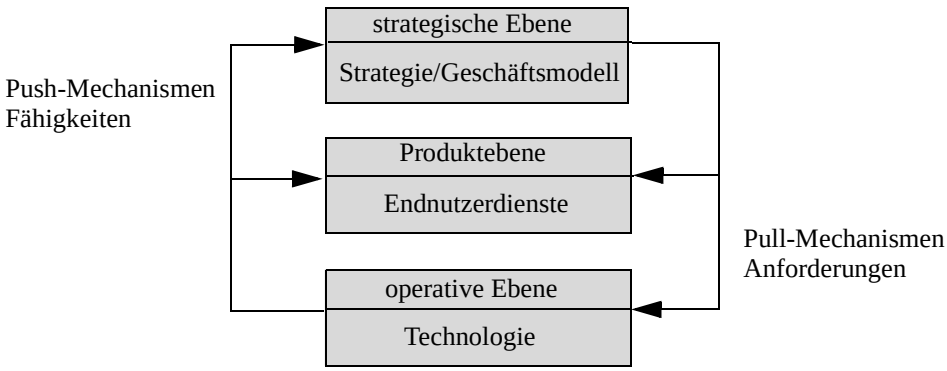

Abbildung 25: Zusammenhänge zwischen Geschäftsmodell und operativer Ebene

Da Internet-Dienstanbieter Geschäftsmodelle häufig verändern müssen und im Bereich der Internet-Ökonomie grundsätzlich wenig Erfahrungen mit der Umsetzung von Geschäftsmodellen besteht, ist es hilfreich, Geschäftsmodelle formal zu beschreiben, um die Arbeit mit ihnen operationalisieren zu können. Bestehende Ansätze für eine Operationalisierung verfolgen in der Regel

die Zielsetzung, das Geschäftsmodell auf seine ökonomischen Auswirkungen hin zu überprüfen, ohne dass es komplett umgesetzt werden muss. Im Rahmen der Arbeit ist ein zweiter Aspekt bedeutsam. Mittels einer Strukturierung und formalen Beschreibung sollen die Abhängigkeiten zwischen Geschäftsmodell und den Unterstützungsdiensten transparent gemacht werden.

Es existieren verschiedene Arbeiten zur Operationalisierung des Umgangs mit einem Geschäftsmodell. Zumeist wird ein Geschäftsmodell in einem ersten Schritt in verschiedene Bereiche gegliedert. Beispielsweise werden von einem Geschäftsmodell Antworten auf die Fragen nach den Prozessen mit denen Werte geschaffen werden, den Teilnehmern an den Prozessen, den Erlösen und den Transaktionen verlangt [KB01]. Vier Dimensionen eines marktorientierten Geschäftsmodells werden in [BS02] unterschieden: die Definition des Kundennutzens und der Mehrwerte, die Definition der Kernaufgaben und Prozesse, die Definition notwendiger Partner zur Realisierung der Aktivitäten und die Definition der Einnahme- und Erlösquellen.

Die Spezifikation einer XML-basierten formalen Beschreibungssprache für Geschäftsmodelle, die E-Business Modelling Language (eBML) [LOP01], reicht über eine solche Gliederung hinaus. Sie basiert auf einer Ontologie für Geschäftsmodelle, die in vergleichbarer Weise die vier Basiselemente Customer-Relationship, Product-Innovation, Infrastructure-Management und Financials definiert [OP02]. Die eBML verwendet wiederum nur umgangssprachliche Attribute, beispielsweise zur Beschreibung der Dienste. Die Zielsetzung der Autoren der eBML ist primär die ökonomische Überprüfung verschiedener Geschäftsmodelle. Das ist auch Ziel des e^3-value Ansatzes, in welchem die ökonomischen Wertschöpfungen und die gegenläufigen Gebührenströme sehr detailliert graphisch modelliert werden [GA03] [GAvV00].

4.2 Das Policy-Modell für Anbieter von Internet-Diensten

Da die Unterstützungsdienste eines Internet-Dienstanbieters in bestehenden Arbeiten zur Operationalisierung nicht oder nur am Rande berücksichtigt werden, sind sie für die hier vorzunehmende Untersuchung der Zusammenhänge zwischen der Definition eines Geschäftsmodells und der Gestaltung der Unterstützungsdienste nicht zu verwenden. Aus diesem Grunde wird ein eigenes theoretisches Modell, das Policy-Modell für Anbieter von Internet-Diensten, definiert.

4.2.1 Elemente des Policy-Modells

Das Policy-Modell für Anbieter von Internet-Diensten besteht aus den zwei Bereichen Geschäftsmodell und operative Dienste. Jeder dieser Bereiche wird strukturiert: Das Geschäftsmodell unterteilt sich in seine drei Basisabschnitte Dienst, Dienstnutzer und kaufmännische Regeln, die operativen Dienste in ihre Teildienste, vgl. Abbildung 26. Die Komponenten beider Bereiche werden jeweils als Policies beschrieben. Die Dienst-Policy beschreibt die vom Dienstanbieter im Geschäftsmodell definierten Dienste, die Dienstnutzer-Policy spezifiziert Eigenschaften der Dienstnutzer. Die kaufmännische Policy beschreibt die strategischen Entscheidungen des Dienstanbieters hinsichtlich der kaufmännischen Regeln, die für die Dienste gelten, wie z.B. Zahlungskonditionen oder Ertrags- und Preismodelle, die er im Geschäftsmodell festlegt. Auf operativer Ebene beschreibt die Policy zum einen die wichtigsten unterscheidbaren Charakteristiken der

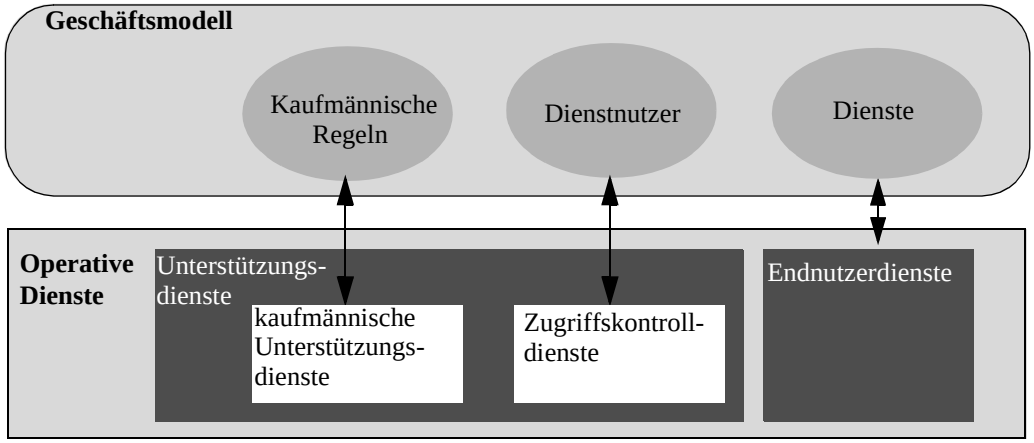

Abbildung 26: Policy-Modell für Internet-Dienste

Teilfunktionen und zum anderen Konfigurationsparameter für ihre tatsächliche technische Realisierung. Die operativen Dienste bestehen aus den Endnutzerdiensten, den Zugriffskontrolldiensten und den kaufmännischen Unterstützungsdiensten, wie sie in Kapitel 2.6 unterschieden wurden.

Erste Zusammenhänge zwischen Geschäftsmodell und operativen Diensten zeigt bereits Abbildung 26. Die im Geschäftsmodell angegeben Dienste sind, wie in Kapitel 2.3 definiert, Endnutzerdienste, die mittels der Endnutzerdienstinfrastruktur realisiert werden. Aus der Beschreibung der potentiellen Kunden lassen sich die Form der auzuführenden Zugriffskontrolle auf der operativen Ebene ableiten. Diese ist zusätzlich auch von den kaufmännischen Regeln abhängig. Die kaufmännischen Regeln werden aber primär mittels der kaufmännischen Unterstützungsdienste umgesetzt.

Dieses erste Model wird verfeinert, indem die drei Basisbereiche des Geschäftsmodells in ihre Einzelaspekte und die operativen Dienste weiter in ihre Teilfunktionen untergliedert werden. Dies ist im erweiterten Policy-Modell in Abbildung 27 illustriert. Die kaufmännischen Regeln des Geschäftsmodells bestehen aus solchen für das Pricing, das Payment und das Billing. Verschiedene Parameter der kaufmännischen Funktionen werden auf strategischer Ebene bestimmt und in Policies festgehalten. Der Dienstnutzer wird über Regeln zu seiner Identifizierung abstrakt beschrieben. Im Geschäftsmodell wir beispielsweise definiert, welche Gruppen von Dienstnutzern unterschieden werden, ob Dienste anonym genutzt werden können und ob sie privat oder öffentlich sind.

Billing und Payment sind nicht nur Bestandteil des Geschäftsmodells, sondern auch der operativen Dienste, da es sich bei Billing und Payment gerade um operative Funktionen handelt. Zusammen mit der Charge-Calculation, dem Accounting und Metering bilden sie die Teilfunktionen der kaufmännischen Unterstützungsdienste. Teilfunktionen der Zugriffskontrolle sind die Autorisierung und Authentifizierung. Zur Autorisierung zählen die Prüfung von Berechtigungsnachweisen, die Prüfung von Nutzerberechtigungen und die dynamische Prüfung von Systemzuständen, wie in Kapitel 3.1.3 erläutert. Die Prüfung der Zahlungsfähigkeit ist entweder eine Prüfung von Berechtigungsnachweisen oder eine dynamische Prüfung. Die Prüfung der Nutzerberechtigungen liegt

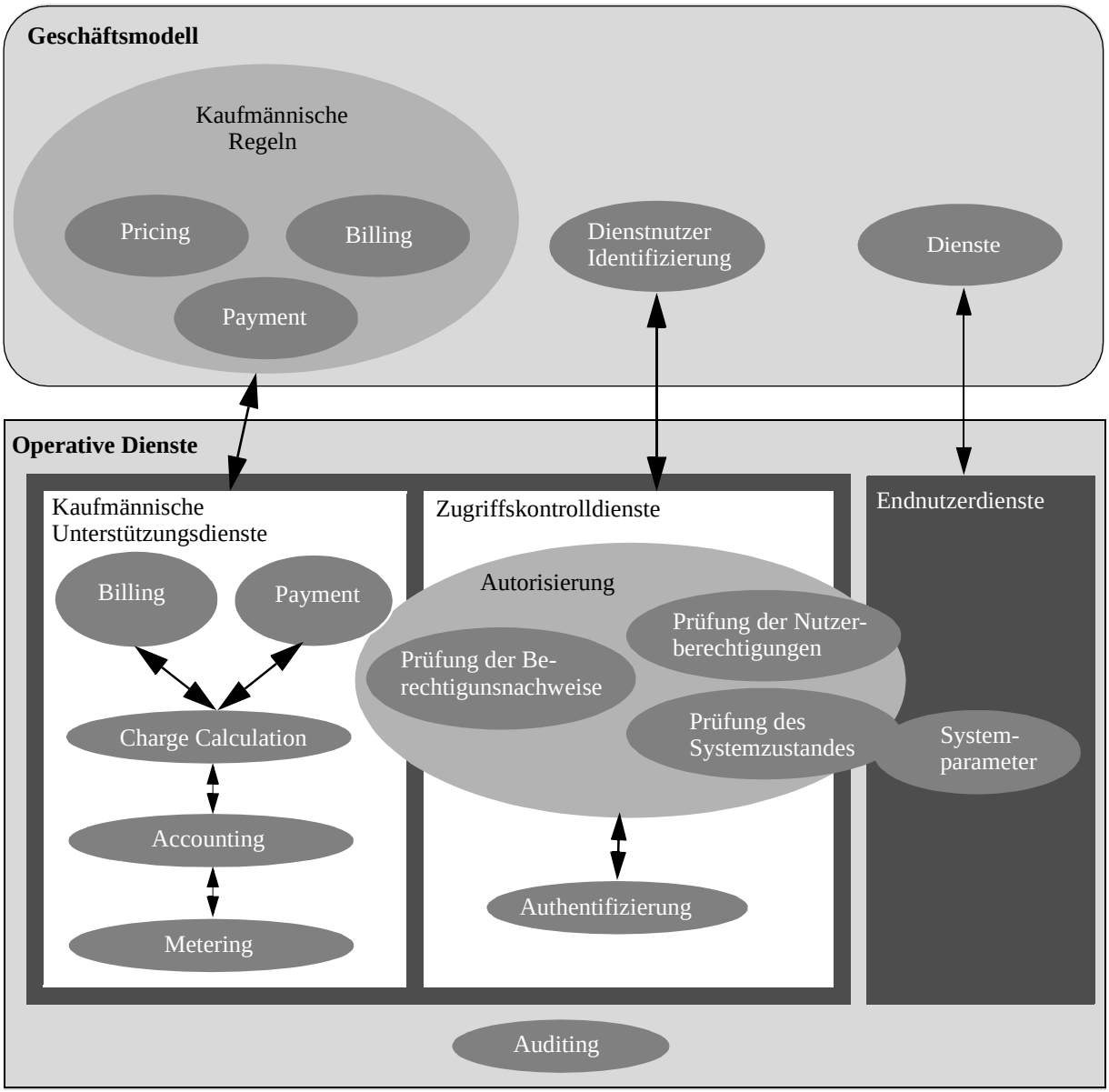

Abbildung 27: Erweitertes Policy-Modell für Internet-Dienste

im Modell an der Schnittstelle zwischen den Zugriffskontroll- und den Endnutzerdiensten, da sie beschreiben, welcher Nutzer welche Dienste verwenden darf. Eine Form der Prüfung des Systemzustands kann die Prüfung von Systemparametern der Endnutzerdienstinfrastruktur sein, so dass sich diese Policies überlappen. Die Funktion des Auditing ist eine Querschnittfunktion über alle anderen Dienste hinweg.

4.2.2 Beziehungen zwischen den Policies des Policy-Modells

Das Policy-Modell ordnet nicht nur die unterschiedlichen Policies in die Bereiche ein, sondern beschreibt zusätzlich Beziehungen zwischen den verschiedenen Policies und damit zwischen den jeweiligen in der Policy spezifizierten Funktionen. Dabei existieren zwei Sichten auf das Modell.

Die erste, systematische Sichtweise betrachtet das Modell von Oben nach Unten. Eine Policy des Geschäftsmodells bzw. einer übergeordneten Ebene innerhalb der operativen Dienste benötigt einen Satz von untergeordneten Mechanismen zu Ihrer Durchsetzung. Beispielsweise werden die kaufmännischen Regeln des Geschäftsmodells mittels Billing- und Payment-Mechanismen durchgesetzt. Die Charge-Calculation-Policy benötigt zu Ihrer Durchsetzung Verfahren des Accountings. Die zweite Sichtweise auf das Modell ist eine operative. Sie erfolgt von Unten nach Oben. Ein Accounting muss z.B. vor einem Charging ausgeführt werden und eine erfolgreiche Authentifizierung ist eine Voraussetzung für eine authentifizierungsbasierte Autorisierung.

Neben diesen Beziehungen, die sich aus den direkten funktionalen Zusammenhängen zwischen den Komponenten ergeben, bestehen auch solche zwischen den einzelnen Policies des Geschäftsmodells und denen der Unterstützungsdienste. Diese werden bei der Darstellung der einzelnen Policies der Unterstützungsdienste genau betrachtet und zum Abschluss des Kapitels zusammenfassend dargestellt.

4.2.3 Eine XML-basierte Policy-Beschreibungssprache für das Policy-Modell

Die einzelnen Policies des Policy-Modells sollen nachfolgend strukturiert und detailliert beschrieben werden, um die Abhängigkeiten genau zu untersuchen. Die Policy der Unterstützungsdienste soll weiterhin innerhalb der zu definierenden Architektur zur Konfiguration des Zugriffskontroll- und Abrechnungssystems verwendet werden. Die Policies werden mittels einer im Rahmen der Arbeit definierten Policy-Sprache beschrieben. Die Verwendung einer formalen Spezifikationssprache ist eine Voraussetzung für die Verarbeitung der Policies in Computersystemen, wie sie zur Konfiguration des Zugriffskontroll- und Abrechnungssystems notwendig ist. Um erstens ein möglichst breites Verständnis für die Spezifikation der Policies zu erreichen und zweitens innerhalb der Architektur existierende Werkzeuge zur Abfrage und Auswertung von Policies verwenden zu können, muss dabei auf existierende Standards zurückgegriffen werden.

Die eXtensible Markup Language (XML) [BPS+00] und XML-Schema [Fal01] stellen solche Standards dar. Die Vorteile von XML liegen in der Erweiterbarkeit, der Trennung von Inhalt und Formatierung, der einfach zu erlernenden Syntax und in der Möglichkeit, die Syntax in einer Document Type Definition (DTD) [BPS+00] oder in einem XML-Schema zu beschreiben. XML wird auch im Bereich des E-Commerce in zunehmenden Maße als Beschreibungsmethode für im Internet auszutauschender Daten genutzt. Es existieren eine Vielzahl von Rahmenwerken zur Spezifikation von strukturierten Nachrichten zum Austausch zwischen Geschäftspartnern, wie Biz Talk [MS03], Electronic Business XML (ebXML) [EN01] und Commerce XML (cXML) [cXM02]. Zum anderen sind Spezifikationen von Dokumentenvorlagen für einzelne Geschäftsfunktionen vorhanden. Die Organization for the Advancement of Structured Information Standards (OASIS) [OAS03] unterstützt die Entwicklung von ebXML und bietet mit xml.org [XML03] eine Plattform zum Austausch von branchenspezifischen Spezifikationen an.

Aufgrund der genannten Vorteile, der weiten Verbreitung von XML im kaufmännischen Bereich verbunden mit dem Know-How und der Existenz von Werkzeugen, wurde die Verwendung von XML zur Spezifikation von Policies und XML-Schema zur Definition der Syntax gewählt. Damit

lassen sich die Beschreibungen der Policies gegen das XML-Schema validieren und auf ihre Gültigkeit hin überprüfen.

Die wichtigsten Elemente der Policy-Beschreibungssprache sind Gegenstand der folgenden Abschnitte. Zunächst erfolgt in Kapitel 4.3 eine Beschreibung der Elemente der Geschäftsmodell-Policy an Hand eines Beispiels. Die Policy-Sprachen zur Definition der Policies der Unterstützungsdienste werden in Kapitel 4.4 erläutert.

4.3 Die Policy-Sprache zur Beschreibung von Geschäftsmodellen

Mittels der im Rahmen der Arbeit definierten und hier vorgestellten Policy-Sprache lässt sich das Geschäftsmodell eines Internet-Dienstanbieters in verschiedenen Ausprägungen umfassend beschreiben. Die wichtigsten Konstrukte der Policy-Sprache werden nachfolgend an Hand eines Beispiels erläutert. Eine Übersicht der übergeordneten Sprachelemente zeigt Abbildung 28. Im Anhang C.1 der Arbeit findet sich ein kompletter Überblick und die Definition der Sprachsyntax als XML-Schema. Das im Folgenden verwendete Beispiel für ein Geschäftsmodell ist in Anhang C.2 aufgeführt.

Geschäftsmodelle werden herkömmlicherweise zumeist umgangssprachlich formuliert. So kann das Geschäftsmodell von *NRW-On*, vgl. Anwendungsfall 6 in Kapitel 2.1, beispielsweise die folgenden Formulierungen enthalten:

Wir bieten Geschäfts- und Privatkunden ein Telefon- und Videokonferenzportal im Internet an. Als Privatkunde können Sie sich direkt anmelden. Nachdem Sie Ihr Kundenkonto über Ihre Kreditkarte aufgeladen haben, können Sie den Konferenzdienst unmittelbar nutzen. Alternativ können Sie sich als Geschäftskunde für den Konferenzdienst freischalten lassen. Sie erhalten dann monatlich eine Gebührenrechnung. Geschäftskunden zahlen eine monatliche Grundgebühr und gegenüber Privatkunden reduzierte Gebühren für einzelne Konferenzen.

Kaufmännische Entscheidungen über Preise, die sich in höheren Frequenzen als Geschäftsmodelle ändern, sind nicht Bestandteil des Geschäftsmodells sondern werden in Form eines Tarifs formuliert:

Als Privatkunde bezahlen Sie lediglich die Konferenzkosten in Höhe von 0,12 EUR für eine Telefon und 0,22 EUR für eine Videokonferenz, die pro Konferenzteilnehmer im Minutentakt berechnet werden. Für Geschäftskunden betragen die Konferenzkosten 0,10 EUR bzw. 0,19 EUR. Die monatliche Grundgebühr für Geschäftskunden beträgt 18,00 EUR. Die Tarife gelten rund um die Uhr an 7 Tagen pro Woche - Vermittlungs- oder Teilnehmerpauschalen werden nicht erhoben.

Das Geschäftsmodell wird in der Policy-Sprache formal mittels eines Kennzeichners und ein Identitätsmerkmal des Dienstanbieters identifiziert. Es besteht aus einer oder mehreren Sequenzen der drei Basiselemente: Dienst, Dienstnutzer und kaufmännische Regeln. Dabei kann das Element Dienst mehrfach auftreten, sofern die kaufmännischen Regeln für mehrere Dienste gelten. Im Beispiel von *NRW-On* ist dies für die zwei Dienste Telefon- und Videokonferenz der Fall. Die Elemente Dienstnutzer und kaufmännische Regeln sind ebenfalls in einer Sequenz gruppiert.

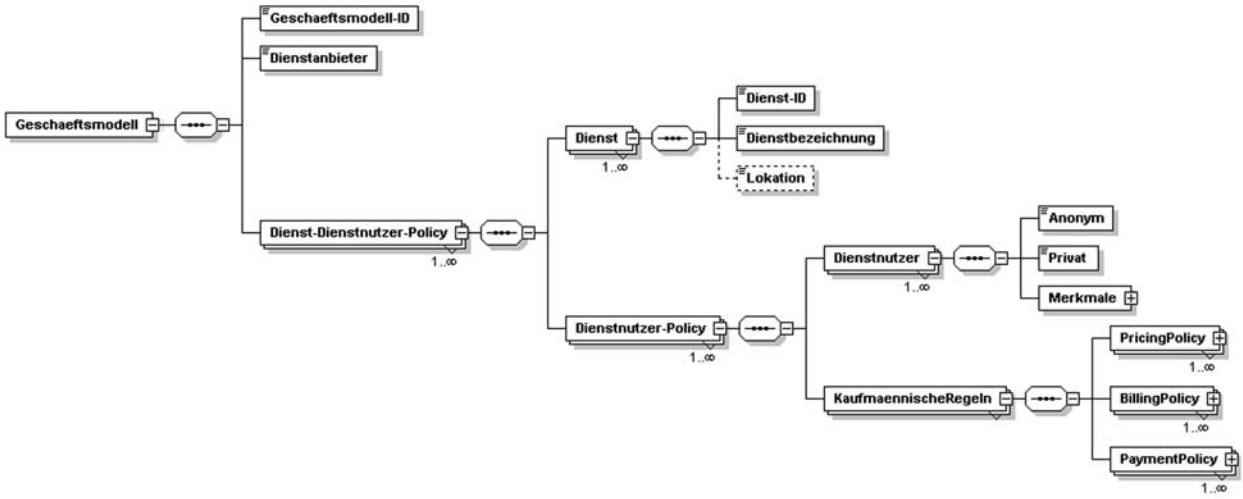

Abbildung 28: Elemente der Policy-Sprache für Geschäftsmodelle

Falls für verschiedene Dienstnutzer, im Beispiel Privat- und Geschäftskunden, die kaufmännischen Regeln unterschieden werden, kann auch diese Sequenz mehrfach genutzt werden.

4.3.1 Dienste

Auf der Abstraktionsebene eines Geschäftsmodells werden die Dienste nur über die Elemente Identifikator und eine umgangssprachliche Bezeichnung beschrieben. Den Identifikator kann der Dienstanbieter, der das Geschäftsmodell definiert, selbst bestimmen. Zum Zwecke der Austauschbarkeit zwischen verschiedenen Dienstanbietern ist es aber sinnvoll, eine einheitliche Identifikation zu verwenden. Dabei kann beispielsweise auf die innerhalb der Internet Assigned Number Authority (IANA) definierten Portnummern zurückgegriffen werden [Aut03], soweit sie für die genutzten Anwendungsprotokolle spezifiziert sind.

```
<xs:element name="Dienst" maxOccurs="unbounded">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="Dienst-ID" type="xs:string"/>
         <xs:element name="Dienstbezeichnung" type="xs:string"/>
         <xs:element name="Lokation" type="xs:anyURI" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Insbesondere bei Inhaltsdiensten ist zudem die zur Verfügung gesetllte Ressource genauer zu spezifizieren. Dazu dient das Element Lokation. Das gilt für öffentliche Inhaltsdienste aber auch für privaten, bei welchem die spezifizierte Ressource auch zur Kontrolle der Nutzerberechtigungen benötigt wird. Die Ressource wird über einen Uniform Ressoucre Locator (URL) [BLFM98] spezifiziert.

Der Videokonferenzdienst aus dem obigen Beispiel lässt sich also in folgender Weise beschreiben.

```
<Dienst>
   <Dienst-ID>555</Dienst-ID>
   <Dienstbezeichnung>Videokonferenz</Dienstbezeichnung>
</Dienst>
```

4.3.2 Dienstnutzer

Eine Beschreibung der vorgesehenen Dienstnutzer ist ebenfalls Bestandteil des Geschäftsmodells. Dabei wird nicht jeder Dienstnutzer einzeln angegeben, sondern es wird im Rahmen der Identifizierungs-Policy bestimmt, ob es verschiedene Gruppen von Dienstnutzern gibt und ob sich Dienstnutzer identifizieren müssen oder nicht. Im Beispiel *NRW-On* werden Privatkunden und Geschäftskunden unterschieden, die sich beide vor einer Dienstnutzung identifizieren müssen, um ihnen ein Kundenkonto zuordnen zu können.

Identifizierungs-Policy. Als Elemente der Identifizierungs-Policy werden angegeben, für welche Gruppen von Dienstnutzern oder individuelle Kunden die in der Policy nachfolgend aufgeführten kaufmännischen Regeln gültig sind. Damit lassen sich Gruppen von Nutzern mit differenzierten Tarifen bzw. differenzierten Dienste-Portfolios unterscheiden. Sind keine Kundengruppen oder individuellen Kunden in der Policy angegeben, gelten die kaufmännischen Regeln für alle Kunden. Weiterhin wird definiert, ob die Dienste auch anonymen Dienstnutzern angeboten werden und ob es sich um private oder öffentliche Dienste handelt.

```
<xs:element name="Dienstnutzer" maxOccurs="unbounded">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="Anonym" type="xs:boolean"/>
         <xs:element name="Privat" type="xs:boolean"/>
         <xs:element name="Merkmale">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="PersonenID" type="xs:string" minOccurs="0"
                        maxOccurs="unbounded"/>
                  <xs:element name="GruppenID" type="xs:string" minOccurs="0"
                     maxOccurs="unbounded"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

4.3.3 Kaufmännische Regeln

Jeder wirtschaftlich handelnde Dienstanbieter will Erträge erzielen. Dazu muss er festlegen, welche Dienste kostenpflichtig sind, für diese Dienste Preise definieren (Pricing), sie in Rechnung stellen (Billing) und den Zeitpunkt der Zahlung (Payment) festlegen. Diese drei Elemente sind als einzelne Policies Bestandteil der kaufmännischen Regeln innerhalb der Geschäftsmodell-Policy.[1]

```
<xs:element name="KaufmaennischeRegeln">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="PricingPolicy" maxOccurs="unbounded"> ... </xs:element>
         <xs:element name="BillingPolicy" maxOccurs="unbounded"> ... </xs:element>
         <xs:element name="PaymentPolicy" maxOccurs="unbounded"> ... </xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Pricing-Policy. Die Funktion des Pricing umfasst den gesamten Prozess der Preisbestimmung. Ein Internet-Dienstanbieter muss dazu zunächst in einem Ertragsmodell festlegen, welche Dienste kostenpflichtig sind und in welcher Form sie berechnet werden. Das Internet bietet die Möglichkeit, verschiedenste Dienste mit verschiedensten Ertragsmodellen zu verwirklichen. Das einfachste Modell nutzt transaktionsbezogene Erträge. Aber auch über Abonnements, Eintrittsgebühren oder Werbung lassen sich Erträge realisieren [Bir02].

Neben den Ertragsmodellen muss der Dienstanbieter die Preise für die einzelnen Dienste festlegen. Diese müssen nicht für jeden Dienstnutzer identisch sein. Sie können für unterschiedliche Kunden oder Kundengruppen und in unterschiedlichen Situationen flexibel gestaltet werden. Man spricht dann von Preisdifferenzierung, die von Dienstanbietern im Internet als Mittel der Differenzierung von Mitbewerbern genutzt werden sollte [SS02]. Die Beschreibung der Preise für die entsprechenden Dienste ist Teil eines Vertrages zwischen Dienstanbieter und Dienstnutzer. Werden sie für alle Kunden oder Kundengruppen auf Basis einer Managemententscheidung festgesetzt, so werden sie für eine Periode in Form eines Tarifs festgeschrieben. Alternativ können Preise für einzelne Dienste individuell zwischen Anbieter und Nutzer ausgehandelt werden. Der Preis wird dann in einem Kontrakt festgelegt.

Eine Pricing Policy besteht aus einer Sequenz von zwei Elementen. Das erste Element sind eine oder mehrere Preiskomponenten aus denen sich der Gesamtpreis für einen Dienst zusammensetzt. Anschliessend wird spezifiziert, ob es sich um die Preise eines Kontrakts oder eines Tarifs handelt. Beim Kontrakt werden eine Kontrakt-ID und eine Kunden-ID als Elemente angegeben. Elemente des Tarif sind wiederum seine ID und die Gültigkeitsperiode. Der Tarif selbst wird außerhalb des Geschäftsmodells spezifiziert, da er kurzfristig geändert werden kann.

1. Die drei Punkte "..." in den XML-Dokumenten bezeichnen eine Auslassung.

```
<xs:element name="PricingPolicy" maxOccurs="unbounded">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="Preiskomponente" maxOccurs="unbounded"> ... </xs:element>
         <xs:element name="Kontrakt-Tarif">
            <xs:complexType>
               <xs:choice>
                  <xs:element name="Kontrakt"> ... <xs:sequence>
                     <xs:element name="KontraktID" type="xs:string"/>
                     <xs:element name="KundenID" type="xs:string"/> </xs:sequence> ...
                  </xs:element>
                     <xs:element name="Tarif"> ... <xs:sequence>
                     <xs:element name="TarifID" type="xs:string"/>
                     <xs:element name="Periode"> ... <xs:sequence>
                        <xs:element name="Beginn" type="xs:dateTime"/>
                        <xs:element name="Ende" type="xs:dateTime"/> </xs:sequence> ...
                     </xs:element>
                  </xs:element>
               </xs:choice>
            <xs:ComplexType>
         </xs:element>
      </xs:sequence>
   <xs:ComplexType>
</xs:element>
```

Preiskomponenten werden unterschieden in eine Grundgebühr (Access Price) sowie leistungs- (Connection Price) und mengenbezogene (Usage Based Price) Gebühren [SFPW98]. Sie werden durch unterschiedliche Elemente beschrieben. Eine Grundgebühr wird dem Kunden periodisch auf Basis eines Abonnements bzw. Vertrages zugeordnet. Innerhalb der Policy muss spezifiziert werden für welchen Zeitraum sie erhoben wird.

```
<xs:element name="Preiskomponente" maxOccurs="unbounded">
   <xs:complexType>
      <xs:choice>
         <xs:element name="Grundgebuehr"> ... </xs:element>
         <xs:element name="LeistungsbezogeneGebuehr"> ... </xs:element>
         <xs:element name="MengenbezogeneGebuehr"> ...</xs:element name>
      </xs:choice>
   </xs:complexType>
</xs:element>
```

Leistungsabhängige Preiskomponenten können im Internet vielfältig genutzt werden [RLS99]. Es gibt viele Beispiele. Eine Gebühr kann je Darstellung, z.B. eines Videos, je Sitzung, z.B. für jeden Aufbau einer Videokonferenz, je Download, z.B. von MP3-Audio-Dateien, je Transaktion, z.B. bei einer e-Bay Kontraktvermittlung oder einer Auktion, oder je Reservierung in einer QoS-Klasse fällig werden. Mittels der leistungsbezogenen Preiskomponente können auch Erlösmodelle auf Basis von Anzeigen abgebildet werden. Die Berechnungsbasis ist dann beispielsweise je Klick.

Von den leistungsbezogenen Preiskomponenten zu unterscheiden ist eine mengenabhängige Komponente. In dieser bestimmt sich die Höhe des Preises beispielsweise von der Dauer der Leistungserbringung, vom Übertragungsvolumen, von der Anzahl der Teilnehmer an einem Kommunikationsdienst oder von Qualitätsparametern. Jede einzelne Preiskomponente wird daher innerhalb durch das Element Berechnungsbasis oder bei Grundgebühren durch das Element Berechnungsintervall näher beschrieben.

Im Falle der leistungs- und mengenabhängigen Preiskomponenten ist nicht von vornherein eindeutig, wem die Dienstnutzung zugerechnet werden kann und wem die Gebühren in Rechnung zu stellen sind. Informations- und Datenflüsse auf technischer Ebene müssen nicht mit dem Fluss der Wertschöpfung übereinstimmen. Einige Beispiele veranschaulichen dies. In einer Videokonferenz erzielen gegebenenfalls alle Teilnehmer einen Mehrwert. Bei einem E-Mail Versand können es Versender und Empfänger der E-Mails sein. Bei einem Video on Demand Dienst ist es derjenige, der den Videostrom abruft. Dabei handelt es sich auf der technischen Transportebene um den Empfänger der Daten. Bei einem kombinierten Dienst aus Web-Hosting und Transportdiensten zahlt derjenige der die Daten hostet, verkehrsabhängig Gebühren für den Abruf der Seiten durch andere. Bei Online-Anzeigen gewinnt derjenige, der die Anzeige schaltet, einen Mehrwert, obwohl er auf Transportebebe gar nicht an der Diensterbringung beteiligt ist. Bei mengen- und leistungsbezogeneng Gebühren müssen daher als weitere Elemente ein oder mehrere Dienstnutzer, dem die Dienste zugerechnet werden können, angegeben sein.

Die im Beispiel definierte Pricing Policy für einen Geschäftskunden entspricht folgender Darstellung im XML.

```
<PricingPolicy>
   <Preiskomponente>
      <Grundgebuehr>
         <Berechnungsintervall>monatlich</Berechnungsintervall>
      </Grundgebuehr>
   </Preiskomponente>
   <Preiskomponente>
      <MengenbezogeneGebuehr>
         <Berechnungsbasis-Menge>Teilnehmer</Berechnungsbasis-Menge>
         <Berechnungsbasis-Menge>Dauer</Berechnungsbasis-Menge>
         <Dienstzurechnung>Anfrager</Dienstzurechnung>
      </MengenbezogeneGebuehr>
   </Preiskomponente>
   <Kontrakt-Tarif>
      <Tarif>
         <TarifID>Geschaeftskunde</TarifID>
            <Periode>
               <Beginn>2003-01-01T00:00:00</Beginn>
               <Ende>2003-06-30T24:00:00</Ende>
            </Periode>
         </Tarif>
   </Kontrakt-Tarif>
</PricingPolicy>
```

Für einen Geschäftskunden wird der bis zum 30.06.2003 gültige Geschäftskundentarif verwendet. Der Preis für die Dienstnutzung setzt sich aus zwei Komponenten zusammen, einer monatlichen Grundgebühr und einer mengenbezogenen Gebühr, die abhängig ist von der Anzahl der Teilnehmer und der Dauer der Konferenz. Der Dienst und die Gebühren werden demjenigen der den Dienst anfragt, also demjenigen der die Konferenz aufbaut zugerechnet.

Billing-Policy. Billing bezeichnet den umfassenden Prozess der Rechnungslegung und Rechnungsstellung. Dabei handelt es sich grundsätzlich um rein operative Funktionen. Ein Dienst kann unmittelbar nach der Diensterbringung einzeln berechnet werden, oder es werden mehrere in einer Periode angefallenen Dienste kumuliert in Rechnung gestellt. Die Entscheidung darüber ist eine strategische. Sie ist abhängig vom Preis einzelner Dienste, der über die Periode anfallenden Summe und den Transaktionskosten für die Rechnungsstellung und Zahlung sowie den finanziellen und anderen Rahmenbedingungen des Dienstanbieters. In der Billing Policy wird daher die Rechnungsperiode angegeben, die von täglich bis zu jährlich reichen kann.

```
<xs:element name="BillingPolicy" maxOccurs="unbounded">
   <xs:complexType name="Rechnungsstellung">
      <xs:choice>
         <xs:element name="Periodisch" type="Rechnungsperiode"/>
         <xs:element name="Einzeln"/>
      </xs:choice>
   </xs:complexType>
</xs:element>
```

Payment-Policy. Payment entspricht dem Vorgang der Zahlung einer Rechnung, das heißt dem Austausch von monetären Werten zwischen dem Dienstnutzer und Anbieter, gegebenenfalls unter Einbeziehung eines vertrauenswürdigen Dritten. Eine Zahlung für einen Dienst erfolgt historisch zumeist nach der Diensterbringung und Rechnungsstellung. Damit trägt der Dienstanbieter das Risiko, dass die Zahlung durch den Dienstnutzer nicht erfolgt. Dieses Risiko kann der Dienstanbieter mindern, indem er die Kreditwürdigkeit des Dienstnutzers vor der Diensterbringung prüft oder einen Dritten einbezieht, z.B. eine Kreditkartengesellschaft, die für die Zahlungsfähigkeit des Kunden bürgt. Alternativ kann die Zahlung auch vor der eigentlichen Diensterbringung erfolgen. Dazu werden Pre Paid Verfahren genutzt. Der Dienstnutzer erwirbt dann quasi ein Recht auf die Nutzung von unter Umständen nicht genau spezifizierten Diensten. Dieses Recht wird dem Dienstnutzer in Rechnung gestellt. Bei den Pre Paid Verfahren trägt der Kunde das Risiko, dass der Dienstanbieter einen Dienst nicht mehr erbringen und es sein Recht nicht einlösen kann. Bei anderen Verfahren, z.B. beim Einsatz von geldartigen Micro-Payments oder Geldkarten, fallen der Zeitpunkt von Diensterbringung und Zahlung praktisch zusammen. Der Dienstanbieter prüft dazu unmittelbar vor der Diensterbringung, ob der Nutzer über entsprechende elektronische Zahlungsmittel verfügt. Diese werden unmittelbar nach der erfolgreichen Transaktion bzw. Diensterbringung umgebucht.

Im Rahmen des Geschäftsmodells muss der Dienstanbieter nur definieren, ob die Zahlung vor der Diensterbringung oder gleichzeitig mit ihr erfolgen muss oder nicht. Damit steuert er u.a. sein Risiko, dass die Zahlung durch den Kunden nicht erfolgt.

```
<xs:simpleType name="Zahlungszeitpunkt">
   <xs:restriction base="xs:string">
      <xs:enumeration value="vor Diensterbringung"/>
      <xs:enumeration value="nach Diensterbringung"/>
      <xs:enumeration value="gleichzeitig"/>
   </xs:restriction>
</xs:simpleType>
```

4.4 Die Policy-Sprachen zur Beschreibung von Unterstützungsdiensten

Neben der zuvor vorgestellten Policy-Sprache zur Beschreibung der Geschäftsmodelle werden an dieser Stelle Sprachen zur Beschreibung der Zugriffskontrolldienste und der kaufmännischen Unterstützungsdienste definiert. Diese können vom Dienstanbieter innerhalb der in der Arbeit entwickelten Architektur verwendet werden, um das Zugriffskontroll- und Abrechnungssystem in Abhängigkeit vom angefragten Dienst zu konfigurieren. Die Sprachen umfassen die im erweiterten Policy-Modell aus Abbildung 27 gezeigten Teilfunktionen.

Die Policies für die Zugriffskontrolldienste werden getrennt von den Policies der kaufmännischen Unterstützungsdienste betrachtet und es werden zwei Sprachen definiert, da die Policies zum einen innerhalb des Zugriffskontroll- und Abrechnungssystems zu unterschiedlichen Zeitpunkten abgefragt und ausgewertet werden und zum anderen anhand unterschiedlicher Kriterien selektiert werden. Die hier vorgenommene Trennung erlaubt auch eine unabhängige Speicherung und Verwaltung der Policies in zwei verschiedenen Repositories, falls eine solche aus Gründen der Performanz notwendig erscheint. Die beiden Policy-Sprachen verfügen aber grundsätzlich über eine identische Struktur, so dass sie sich ohne einen größeren Aufwand auch zu einer Sprache zusammenfassen lassen.

4.4.1 Die Policy-Sprache zur Beschreibung von Zugriffskontrolldiensten

Die an dieser Stelle vorgestellte Policy-Sprache erlaubt eine Beschreibung der einzelnen Teilfunktionen der Zugriffskontrolle, die bei der Anfrage eines Dienstes durchzuführen ist. Abbildung 29 zeigt die wichtigsten Sprachelemente zur Beschreibung einer Zugriffskontroll-Policy im Überblick. Im Anhang C.4 sind sie komplett gezeigt. Dort findet sich auch die vollständige Definition der Sprachsyntax als XML-Schema. Die bedeutendsten Elemente werden im Folgenden erläutert.

Die innerhalb der Zugriffskontrolle auszuführenden Funktionen sind abhängig von den zu kontrollierenden Diensten und gegebenenfalls auch von den Heimatdienstanbietern der Dienstnutzer. So kann die Zugriffskontrolle für Dienstnutzer aus verschiedenen Anbieterdomänen unterschiedlich durchzuführen sein. Für Dienstnutzer der eigenen Domäne kann beispielsweise die Authentifizierung selbst durchgeführt werden, wohingegen sie für andere durch deren Heimatdienstanbieter oder Dritte erfolgen muss. Die drei übergeordneten Elemente der Zugriff-

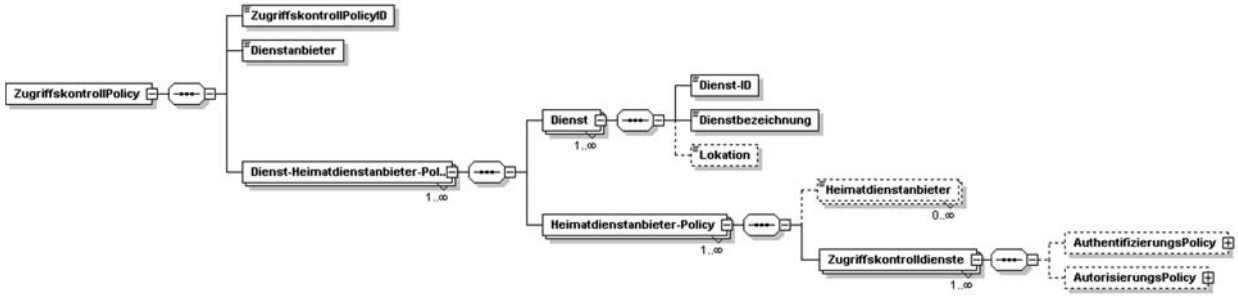

Abbildung 29: Elemente der Policy-Sprache für Zugriffskontroll-Policies

skontroll-Policy sind daher die Dienste, die Information über den Heimatdienstanbieter des Dienstnutzers und die Spezifikation der Zugriffskontrolldienste selbst. Jede Zugriffskontroll-Policy wird mittels eines Bezeichners identifiziert.

Die Dienste werden wie in der Geschäftsmodell-Policy definiert spezifiziert. Der Heimatdienstanbieter wird über den Home-Bestandteil seines Uniform Ressource Identifiers (URI) [BLFM98] gekennzeichnet. Innerhalb des Elements Zugriffskontrolldienste werden die einzelnen Teilfunktionen, wie nachfolgend erläutert, beschrieben.

Authentifizierungs-Policy. Das erste Element der Authentifizierungs-Policy ist die Angabe der Art der Identitätsmerkmale, welche zur Identifizierung der Dienstnutzer verwendet werden kann. Sie muss in Übereinstimmung mit dem Geschäftsmodell des Dienstanbieters stehen. Der Dienstanbieter muss die Identitätsmerkmale im Rahmen des Austauschs der mit der Dienstanfrage zusammenhängenden Signalisierungsnachrichten implizit bestimmen oder explizit beim Nutzer abfragen. Eine solche Abfrage kann auch durch das Zugriffskontrollsystem gestellt werden. Die Endnutzerdienstinfrastruktur leitet diese dann als Proxy an den Dienstnutzer weiter. Dazu ist es notwendig, die vom Endnutzer unterstützen Authentifizierungs-Protokolle, anzugeben.

```
<xs:element name="AuthentifizierungsPolicy" minOccurs="0"> <xs:complexType>
   <xs:sequence>
      <xs:element name="AuthentifizierungsID" type="AuthentifizierungsID"
                  maxOccurs="unbounded"/>
      <xs:element name="Authentifizierungsprotokoll" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="OrtderDurchfuehrung"> ... </xs:element>
   </xs:sequence> </xs:complexType>
</xs:element>
```

Die möglichen Arten der Identitätsmerkmale stimmen mit den in Kapitel 3.1.1 in Tabelle 1 aufgeführten überein.

```
<xs:simpleType name="AuthentifizierungsID">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Teilnehmeranschlussnummer"/>
      <xs:enumeration value="Benutzerkennung"/>
   </xs:restriction>
</xs:simpleType>
```

Das letzte Element der Authentifizierungs-Policy ist der Ort der Durchführung. Über dieses kann spezifiziert werden, ob die Authentifizierung lokal, vom fremden Heimatdienstanbieter des Endnutzers oder von einer dritten vertrauenswürdigen Parteie, also z.B. einem Broker durchgeführt werden soll.

```
<xs:element name="OrtderDurchführung">
   <xs:complexType>
      <xs:choice>
         <xs:element name="lokal"/>
         <xs:element name="fremderHeimatdienstanbieter"/>
         <xs:element name="ThirdParty" type="xs:anyURI" maxOccurs="unbounded"/>
      </xs:choice>
   </xs:complexType>
</xs:element>
```

Autorisierungs-Policy.

Die Autorisierungs-Policy beschreibt die Teilfunktionen, die im Rahmen der Autorisierung durchgeführt werden müssen. Dabei kann es sich um die Prüfung von Berechtigungsnachweisen, die der Dienstnutzer vorlegt, verschiedene dynamische Autorisierungsfunktionen und die Prüfung von Nutzerberechtigungen handeln. Ist keines der Elemente gefüllt, so ist entweder nur eine reine Authentifizierung vorzunehmen oder keine Zugriffskontrolle notwendig. Somit gilt:

```
<xs:element name="AutorisierungsPolicy">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="PruefungBerechtigungsnachweis" minOccurs="0"> ...
         </xs:element>
         <xs:element name="DynamischeAutorisierung" minOccurs="0" > ...
         </xs:element>
         <xs:element name="PruefungNutzerberechtigungen" minOccurs="0"> ...
         </xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Jedes der übergeordneten Elemente wird über das Element Ort der Durchführung, wie es zuvor bereits erläutert wurde, und einen jeweiligen Typ beschrieben. Für Berechtigungsnachweise können beispielsweise die folgenden Typen verwendet werden:

```
<xs:simpleType name="Berechtigungsnachweis">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Kreditkarte"/>
      <xs:enumeration value="GuthabenkontoHardware"/>
      <xs:enumeration value="MicropaymentGeldartig"/>
      <xs:enumeration value="Ticket"/>
      <xs.enumertaion value="SPKI Zertifikat"/>
   </xs:restriction>
</xs:simpleType>
```

4.4.2 Die Policy-Sprache zur Beschreibung von kaufmännischen Unterstützungsdiensten

Die kaufmännischen Unterstützungsdienste werden mittels einer eigenen Policy-Sprache beschrieben. Die Policy gliedert sich in die im Policy-Modell angegebene Teilfunktionen. Einen Ausschnitt aus der Übersicht über die Sprachelemente zeigt Abbildung 30. Sie findet sich ebenso wie die Definition der Sprachsyntax komplett im Anhang C.3.

Der Dienstanbieter spezifiziert die kaufmännischen Unterstützungsdienste für eine Gruppe von Diensten und Dienstnutzern oder alle Dienstnutzer. Diese beiden Elemente sind daher neben der Beschreibung der einzelnen kaufmännischen Funktionen Bestandteil der Policy. Die Beschreibung der Dienste erfolgt wiederum wie in der kaufmännischen Policy spezifiziert (vgl. Kapitel 4.3.1). Der Dienstnutzer, für den die kaufmännischen Unterstützungsdienste gelten, wird über eine Gruppen-ID oder Personen-ID oder als anonymer Dienstnutzer gekennzeichnet. Sind die Gruppen- und Personen-ID nicht gefüllt, gilt die Policy für alle Dienstnutzer. Als zusätzliches Element muss wie in der Zugriffskontroll-Policy der Heimatdienstanbieter der Dienstnutzer angegeben werden, da die Realisierung der kaufmännischen Funktionen für Nutzer unterschiedlicher Domänen differieren kann. So kann beispielsweise die Rechnungsstellung direkt durch den Dienstanbieter, durch einen Broker oder durch den Heimatdienstanbieter erfolgen. Die Spezifikation der eigentlichen kaufmännischen Unterstützungsdienste gliedert sich in Beschreibungen der operativen Funktionen Metering, Accounting und Charge-Calculation sowie Billing und Payment, wie im Policy-Modell dargestellt.

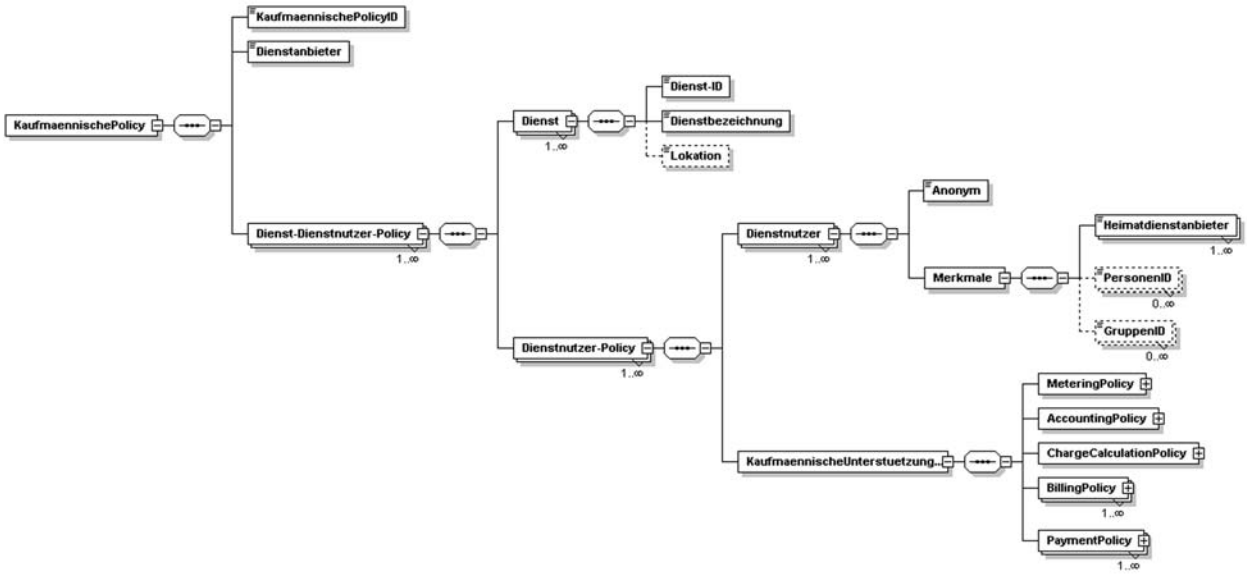

Abbildung 30: Elemente der Policy-Sprache für kaufmännische Unterstützungsdienste

Metering-Policy. Metering bezeichnet das Sammeln von Informationen über eine Dienstnutzung auf technischer Ebene. Es ist Voraussetzung zur Berechnung mengenbezogener Gebühren für paketvermittelte Dienste. Für verbindungs- oder sessionorientierte Dienste, ist in der Regel kein explizites Metering notwendig, da sich die Informationen über die Dienstnutzung im Accounting bestimmen lassen. Welche Informationen gesammelt werden müssen, lässt sich aus der Pricing-

Policy ableiten. Sie gibt an, welche Parameter zu messen sind und wem die Dienstnutzung zuzurechnen ist. Dazu muss der Dienstnutzer im Metering implizit anhand der im Rahmen der Diensterbringung übertragenen Identitätsmerkmale identifiziert werden. Das Metering erzeugt sogenannte Metering-Datensätze (Metering-Records), die Angaben über das Verkehrsvolumen, die Dauer der Diensterbringung und gemessene QoS-Parameter enthalten können.

Eine Metering-Policy besteht aus drei Elementen, den Parametern zur Konfiguration der Metering-Komponenten, der Spezifikation eines Formats für die Metering-Records und dem Typ des Identitätsmerkmals, mittels dessen der Dienstnutzer bestimmt werden soll. Die Parameter zur Konfiguration der Metering-Komponenten und das Record-Format sind abhängig von der technischen Realisierung des Metering und sollen hier nicht weiter spezifiziert werden.

```
<xs:element name="MeteringPolicy">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="Metering-Konfigurationsparameter"/>
         <xs:element name="Metering-Record-Format"/>
         <xs:element name="MeteringID-Typ"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Für Internet-Transportdienste bietet die Realtime Traffic Flow Management Architektur (RTFM) eine generische Methode zur Messung verschiedener Parameter von Flows an [BMR99][BB00]. Der Dienstnutzer wird mit Hilfe von aus dem Flow erkennbaren Attributen identifiziert. Das sind die IP-Adresse des Senders und Empfängers, das verwendete Interface und der Protokoll Typ. Optional ist auch die Verwendung nicht aus dem Flow erkennbare Attribute vorgesehen, nämlich die Subscriber-ID, also eine Benutzerkennung.

Accounting-Policy. Mit Accounting wird die Sammlung und Zusammenfassung von Informationen, die mit der Dienstnutzung durch einen Endnutzer in Zusammenhang stehen, in sogenannten Accounting-Records bezeichnet. Welche Informationen gesammelt werden müssen, lässt sich wiederum aus der Pricing-Policy ableiten, denn die Accounting-Records bilden die Grundlage für die Berechnung von mengen- und leistungsbezogenen Gebühren. Ein Accounting-Record muss daher ebenfalls einem Dienstnutzer zugeordnet werden. Die Erzeugung von Accounting-Records erfolgt durch ein Accounting-Module auf Basis von Nachrichten, die das System, welches den Endnutzerdienst erbringt, dem Accounting-Modul direkt oder indirekt sendet. Diese Transaktionen finden zu Beginn und Ende aber auch während der Diensterbringung statt. Beispielsweise sendet ein Internet-Zugangsrechner zum Zeitpunkt des Aufbaus und Abbaus der Transportverbindung des Endnutzers eine Nachricht an das Accounting Modul, in welchem unter anderem der Endnutzer und der erbrachte Dienst identifiziert werden. Das Accounting-Modul kann, falls notwendig, auf Metering-Records zurückgreifen und diese auswerten.

Eine Accounting Policy besteht ebenfalls aus drei Elementen, den Parametern zur Konfiguration der Accounting-Module, der Spezifikation des Formats der Accounting-Records und dem Typ des Identitätsmerkmals, zur Identifizierung der Dienstnutzer.

```
<xs:element name="AccountingPolicy">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="Accounting-Konfigurationsparameter"/>
         <xs:element name="Accounting-Record-Format"/>
         <xs:element name="AccountingIDTyp" type="AccountingIDTyp"/>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

Für das Accounting existieren mehrere Standards zur Spezifikation des Formats von Accounting-Records und Protokolle für Accounting-Transaktionen. Im verbindungsorientierten PSTN wird keine Trennung zwischen Accounting und Metering vorgenommen. Vielmehr umfasst nach [ITU95a] Accounting die Funktionen Usage-Metering, Charging und Billing. Jede Ressourcen Nutzung im PSTN lässt sich einem sogenannten Call zuordnen, der in Form von Accounting-Datensätzen sogenannten Call Detail Records (CDR) [ITU98] beschrieben wird. CDRs fassen verschiedene Informationen über einen Call zusammen. Dazu gehören die Dauer des Calls und eine Identifizierung der Dienstnutzer in Form der Teilnehmeranschlussnummern des Anrufers und bzw. oder des Gerufenen.

Für Internet-Zugangsdienste wird das Accounting in der Regel durch den ausgegliederten Server, also den RADIUS- oder AAA-Server, der auch die Zugriffskontrolle durchführt, realisiert. Sowohl in RADIUS als auch in Diameter (vgl. Kapitel 3.4.1) sind Accounting-Attribute und Nachrichtentypen definiert, mittels derer der Internet-Zugangsrechner die notwendigen Informationen an den Server übermittelt [ACZ01][Rig00]. Eine Trennung in Accounting und Metering erfolgt auch hier nicht. Der Client selbst misst verschiedene Dienstparameter, wie die Dauer der Verbindung oder das Volumen und sendet diese Informationen nach Ablauf der Verbindung an den Server. Zur Identifikation des Dienstnutzers wird die Benutzerkennung verwendet.

Auf der Anwendungs- und Inhaltsebene erfolgt ein Accounting zumeist integriert durch die Anwendung selbst. Es gibt keine oder nur proprietäre Formatdefinitionen für Accounting-Records. Eine Ausnahme bilden die Bemühungen der Internet Protocol Data Records (IPDR) Organisation. Sie verfolgt die Zielsetzung, äquivalent zu den CDRs ein Format für die Beschreibung von Nutzungsparametern von Internet-Diensten zu spezifizieren und damit einen Austausch von Accounting-Records zwischen Anbietern und zwischen verschiedenen Systemen zu ermöglichen. Dazu wird ein IPDR Master Schema [Cot02c] definiert, das für alle Dienste einheitlich ist, und eine Vielzahl von dienstspezifischen Erweiterungen, z.B. für VoIP-Dienste [Cot02a] oder für E-Mail Dienste [Cot02b].

Charge-Calculation-Policy. Die Gebührenerhebung (Charge-Calculation) bezeichnet einen komplexen Vorgang, der Accounting, Pricing und Billing zusammenführt. Für einen erbrachten Dienst wird ein Preis ermittelt, der in Charging-Records gespeichert wird und einem Kunden in Rechnung gestellt werden kann [KSSW00]. Dazu müssen die in den Accounting-Records vorliegenden technischen Werte in monetäre Werte überführt werden. Weiterhin muss dem Dienstnut-

zer aus einem Accounting-Record, der über eine Accounting-ID spezifiziert ist, ein Kunde zugeordnet werden, dem eine Rechnung gestellt werden kann. Die Erzeugung von Charging-Records besteht aus mehreren Teilaufgaben:

- Selektion derjenigen Kontensätze des Accounting, die ein Gebühr im Sinne der Pricing-Policy zur Folge haben. Nicht jeder Accounting-Record muss zu einem Charging-Record führen.
- Zusammenfassung verschiedener zusammengehöriger Accounting-Records. Dabei sind bei mengenbezogenen Preiskomponenten die verschiedenen Zeitskalen zu berücksichtigen. Während ein Metering technisch auf Basis von Millisekunden erfolgen kann und auch ein Accounting mit kleinen Zeitskalen arbeitet, sind die Preise innerhalb eines Tarifs oftmals in größeren Zeitskalen, z.B. Sekunden oder Minuten, beschrieben.
- Überführung der technische Werte, der Metering- und Accounting-Records in einen monetären Wert. Dazu werden die Tarife oder Kontrakte, die in der Pricing-Policy spezifiziert sind, verwendet.
- Zusammenfassung verschiedener Preiskomponenten für einen einzelnen Dienst.
- Erzeugung von Charging-Record für Grundgebühren. Für diese existieren keine Accounting-Records sondern sie werden direkt aus der Pricing-Policy abgeleitet und periodisch gebildet.
- Zuordnung der Kunden zu den Diensten innerhalb der Charging-Records. Der Kunde, der für einen Dienst zahlen muss, muss innerhalb eines Charging-Records angegeben werden. Wenn gemäß der Pricing-Policy die Dienstnutzung mehreren Kunden zugerechnet werden kann, die für den Dienst zahlen müssen. so können aus einem Accounting-Record mehrere Charging-Records erzeugt werden. Grundsätzlich müssen den innerhalb des Accountings verwendeten Identitäten Identitäten von Kunden zugeordnet werden, sofern sie sich unterscheiden. Bei Identitäten von Kunden, die auch in der Pricing-Policy verwendet werden, sind es persönliche Identitäten. Im Charging können auch Identitäten von Geräten genutzt werden. Die Zuordnung von Identitäten von Kunden zu den im Accounting-Record angegebenen erfolgt mit Hilfe von Session-Informationen, die im Rahmen der Zugriffskontrolle gespeichert werden. Alternativ kann ein Charging-Record auch einem anonymen Kunden mit einer nur temporär gültigen ID zugeordnet werden, wenn elektronische Pre Paid Bezahlverfahren genutzt werden.

Die Charge-Calculation-Policy besteht aus zwei Elementen, den Konfigurationsparametern für das Charging-Modul und der Charging-Periode. Die Charging-Periode gibt an, in welchem zeitlichen Intervall, z.B. sekündlich, minütlich oder stündlich, Accounting-Records in Charging-Records überführt werden. Das Charging muss nicht periodisch erfolgen, sondern es kann auch bei jeder einzelnen Generierung von Accounting-Records direkt ein Charging-Record erstellt werden. Ein einheitliches Format für Charging-Records existiert derzeit nicht.

```
<xs:element name="ChargeCalculationPolicy">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Charging-Konfigurationsparameter"/>
            <xs:element name="Chargingperiode" type="Chargingperiode"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Billing-Policy. Billing umfasst den Prozess der Rechnungserstellung. Dazu werden die Charging-Records ausgewertet. Diese bilden zusammen mit den Angaben über den Rechnungssteller, den Rechnungsempfänger und die Zahlungskonditionen die Bestandteile der Rechnung. Die Rechnung muss nicht durch den Dienstanbieter selbst, sondern kann auch über einen vertrauenswürdigen Dritten oder im Falle eines Endnutzers aus einer anderen Heimatdomäne über dessen Heimatdienstanbieter gestellt werden.

Die Policy umfasst als Elemente Konfigurationsparameter für das Billing-System, die Angabe, ob eine Kunden-ID oder eine temporäre ID für die Rechnungsstellung an einen anonymen Kunden verwendet wird, sowie die Angabe wer die Rechnung stellt. Letzteres wird über das Element Ort der Durchführung angegeben, wie es auch in der Authentifizierungs-Policy verwendet wird.

```
<xs:element name="BillingPolicy" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Billing-Konfigurationsparameter"/>
            <xs:element name="BillingIDTyp" type="BillingIDTyp"/>
            <xs:element name="OrtderDurchfuehrung"> ... </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

Payment-Policy. Mit Payment wird der eigentliche Bezahlvorgang, d.h. der Austausch monetärer Werte zwischen Dienstanbieter und Dienstnutzer bezeichnet. Internet-Dienstanbieter können verschiedene Verfahren nutzen. Als rechnungsbasierte Bezahlung wird die einfache Rechnungserstellung nach der Diensterbringung und deren Zahlung per klassischer Bezahlverfahren wie einer Banküberweisung bezeichnet. Die Einrichtung eines Guthabenkontos durch den Dienstanbieter ist ein sogenanntes Pre Paid Verfahren. Es kann software- oder hardwarebasiert realisiert werden. Bei Nutzung eines softwarebasierten Pre Paid Kontos authentifiziert sich der Kunde über seine Benutzerkennung und die Gebühren für einen Dienst werden vom Konto abgebucht. Ein hardwarebasiertes Guthabenkonto lässt sich mittels einer Smartcard realisieren. Weiterhin sind die Zahlung per Kreditkarte und die Verwendung der verschiedenen in Kapitel 3.4.3 vorgestellten elektronischen Bezahlverfahren möglich.

Die Payment-Policy spezifiziert das zu verwendende Bezahlverfahren, die technischen Parameter für ein elektronisches Verfahren und denjenigen, der die Zahlung entgegen nimmt, also wie-

derum der Dienstanbieter, ein vertrauenswürdiger Dritter oder der Heimatdienstanbieter des Kunden.

```
<xs:element name="PaymentPolicy" maxOccurs="unbounded">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="Payment-Konfigurationsparameter"/>
         <xs:element name="Bezahlverfahren" type="Bezahlverfahren"/>
         <xs:element name="OrtderDurchfuehrung"> ... </xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

4.5 Abhängigkeiten zwischen Geschäftsmodell und Unterstützungsdiensten

Ein Internet-Dienstanbieter muss Geschäftsmodell und operative Dienste integriert betrachten, wie in Kapitel 4.1 erläutert. Insbesondere muss er auch bei der Definition des Geschäftsmodells die Auswirkungen auf die Unterstützungsdienste beurteilen können. Die Abhängigkeiten zwischen Geschäftsmodell und der Form der Unterstützungsdienste bzw. zwischen den jeweiligen Policies wurden im vorhergehenden Abschnitt bereits vielfach angesprochen. Sie sind oftmals nicht offensichtlich und sollen daher an dieser Stelle zusammenfassend im Policy-Modell dargestellt werden.

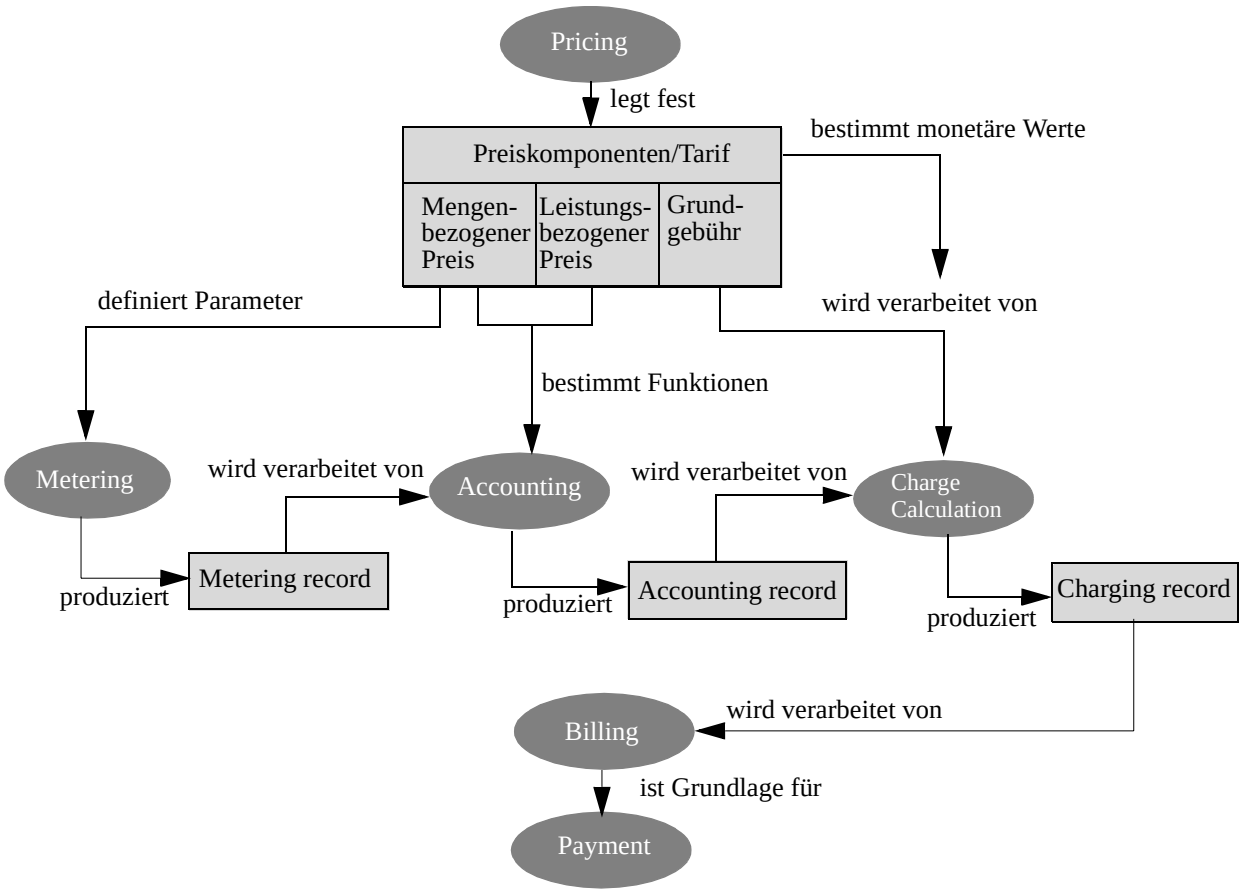

Abbildung 31: Zusammenhänge zwischen kaufmännischen Unterstützungsdiensten

Die Funktionen des Metering, Accounting und Charge-Calculation sind in hohem Maße abhängig von der Pricing-Policy innerhalb des Geschäftsmodells. Sie dienen gemeinschaftlich dazu, den monetären Wert einer Dienstnutzung zu bestimmen, der dann dem Dienstnutzer in Rechnung gestellt wird. Die Zusammenhänge sind in Abbildung 31 veranschaulicht.

Daraus ergeben sich unmittelbar erste Abhängigkeiten zwischen den entsprechenden Policies. Sie sind in Abbildung 32 mit (1), (2) und (3) gekennzeichnet. Die Abbildung zeigt die einzelnen Policies des erweiterten Policy-Modells, wie es in Kapitel 4.2 vorgestellt wurde, und die zwischen den Policies bestehenden Abhängigkeiten, welche nachfolgend kurz erläutert werden.[1]

Pricing-Policy. Die Pricing-Policy bestimmt zusätzlich zu den zuvor genannten Abhängigkeiten auch, ob eine Authentifizierung notwendig ist (4) und ob ein Berechtigungsnachweis geprüft werden muss (5). Wenn nämlich innerhalb der Pricing-Policy Preise oder Tarife für einzelne Kunden

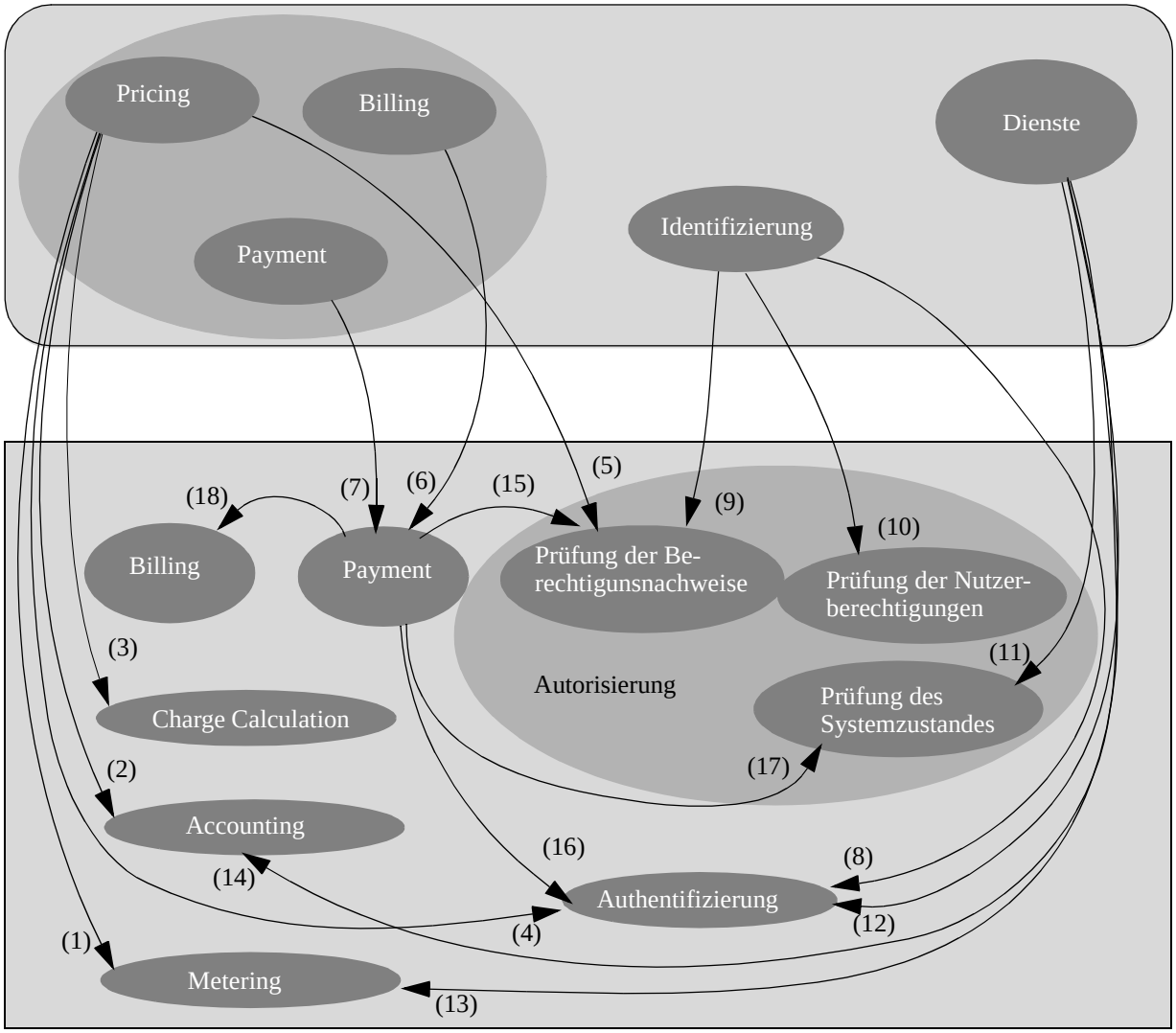

Abbildung 32: Abhängigkeiten zwischen Policies im Policy-Modell

1. Die im Text angegeben Nummer bezieht sich auf die Markierung der Abhängigkeiten in Abbildung 32.

oder Kundengruppen spezifiziert werden, muss der Nutzer sich authentifizieren und es muss dynamisch seine Zugehörigkeit zu einer Kundengruppe bestimmt werden. Alternativ kann er einen Berechtigungsnachweis vorlegen, aus dem seine Zuordnung zu einer Kundengruppe hervorgeht. Mittels der Authentifizierung und einer Session-Beschreibung kann dann in der Charge-Calculation der Nutzer bestimmt werden.

Billing-Policy (strategisch). Die Billing-Policy innerhalb des Geschäftsmodells hat einen Einfluss auf die Wahl des Bezahlverfahrens (6). Soll eine periodische Abrechnung erfolgen, so können nur Post Paid Bezahlverfahren eingesetzt werden.

Payment-Policy (strategisch). Die Payment-Policy auf Ebene des Geschäftsmodells, also die Entscheidung über den Zeitpunkt der Zahlung, bestimmt direkt die Auswahl der auf operativer Ebene zu verwendenden Bezahlverfahren (7).

Identifizierungs-Policy. Von der Identifizierungs-Policy sind die Autorisierungs-Policy und die Authentifizierungs-Policy abhängig. Werden innerhalb der Identifizierungs-Policy Personen oder Gruppen von Nutzern spezifiziert, müssen sich die Nutzer authentifizieren (8) oder einen Berechtigungsnachweis vorlegen, der auf ihre Gruppenzugehörigkeit schließen lässt (9). Sind die Dienste als privat spezifiziert ist wiederum eine Authentifizierung oder Prüfung eines Berechtigungsnachweises notwendig und zudem die Prüfung der Nutzerberechtigungen (10).

Dienst. Die Art des erbrachten Dienst hat direkten Einfluss auf die dynamische Autorisierung, falls eine solche in Abhängigkeit von Systemparametern der Endnutzerdienstinfrastruktur notwendig ist (11) und außerdem auf die Art der Authentifizierungs-ID (12), der Metering-ID (13) und Accountig-ID (14).

Payment-Policy (operativ). Die Wahl des Bezahlverfahrens auf operativer Ebene beeinflusst wiederum die Autorisierungs- und Authentifizierungs-Policy. Werden eine hardwarebasiertes Guthabenkonto oder geldartige elektronische Bezahlverfahren genutzt, so handelt es sich dabei um einen Berechtigungsnachweis, der validiert werden muss (15). Für softwarebasierte Guthabenkonten hingegen ist eine Authentifizierung des Dienstnutzers notwendig (16). Bei der Nutzung von kontenbasierten Verfahren oder Kreditkartenverfahren zur Bezahlung ist zudem eine dynamische Prüfung des Kontenstandes durchzuführen (17). Zusätzlich ist der Typ der Billing ID vom Bezahlverfahren abhängig (18). Bei Kreditkartenverfahren oder geldartigen elektronischen Bezahlverfahren wird eine temporär gültige Kunden-ID genutzt.

4.6 Zusammenfassung

In diesem Kapitel wurde zunächst die Bedeutung von Geschäftsmodellen für Internet-Dienstanbieter erläutert sowie die Notwendigkeit einer formalen Beschreibung von Geschäftsmodellen motiviert. Geschäftsmodell und Policies wurden im Policy-Modell für Internet-Dienstanbieter, welches aus einzelnen Policies für Teilkomponenten bzw. Teilfunktionen besteht, strukturiert. Das Policy-Modell erlaubt gleichzeitig eine systematische und operative Sichtweise auf die Zusammenhänge zwischen den einzelnen Policies. Es wurde zum Abschluss des Kapitels um eine Dar-

stellung der Abhängigkeiten zwischen Geschäftsmodell und den einzelnen Policies der Unterstützungsdienste erweitert. Die Untersuchung der Abhängigkeiten erfolgte im Zusammenhang mit der Erläuterung von XML-basierten Policy-Beschreibungssprachen, deren Syntax mittels XML-Schema definiert wurde. Die Policy-Beschreibungssprache kann ein Internet-Dienstanbieter in der in der Arbeit entwickelten Architektur verwenden, um das die Zugriffskontrolle und Abrechnung durchführende System zu konfigurieren. Die dazu notwendigen Elemente lassen sich in der Policy-Sprache definieren. Soll sie darüberhinaus in anderen Bereichen Verwendung finden, ist ihre Syntaxdefinition gegebenenfalls geeignet zu erweitern. Weiterhin muss der Dienstanbieter sicherstellen, dass die kaufmännischen Policies konfliktfrei sind und er beispielsweise nicht zwei sich widersprechende Pricing-Policies für den gleichen Dienst definiert. Bei Zugriffskontroll-Policies soll es hingegen möglich sein, dass der Dienstanbieter mehrer Zugriffskontroll-Verfahren für einen Dienst spezifiziert, um zwischen Dienstanbieter und Dienstnutzer eine Aushandlung alternativer Verfahren zuzulassen.

Mit dem Policy-Modell und den Policy-Beschreibungssprachen hat der Anbieter von Internet-Diensten zwei Hilfsmittel an der Hand, um Geschäftsmodell und Unterstützungsdienste integriert zu betrachten, formal zu spezifizieren und das nachfolgend vorgestellte Zugriffskontroll- und Abrechnungssystem dynamisch zu konfigurieren.

Kapitel 5: Die policybasierte A^x-Architektur

Gegenstand dieses Kapitels ist die im Rahmen der Arbeit entwickelte policybasierte Zugriffskontrollarchitektur und ihre Erweiterung um Komponenten zur Erfüllung der kaufmännischen Unterstützungsdienste. Sie wird, entsprechend der Definition von A^x-Diensten in Kapitel 2.6, A^x-Architektur genannt. Die Beschreibung der Architektur gliedert sich in eine Darstellung, der im Rahmen des Designs entwickelten und verwendeten Konzepte, der Architektur für die Zugriffskontrolle selbst und deren Erweiterung. Zunächst werden aber allgemeine Anforderungen an Software-Systeme erörtert. Diese bestimmen neben den funktionalen Anforderungen das Design der Architektur.

Die drei wichtigsten innerhalb der A^x-Architektur verwendeten Konzepte sind die Separierung von Diensten, auf Basis des in Kapitel 2.2 vorgestellten Dienstemodells, die Modularisierung der Teilfunktionen und das Paradigma des policybasierten Managements. Weitere Konzepte sind eine einheitliche Identifizierung der Dienstnutzer sowie ein Session-Konzept. Die drei erstgenannten Konzepte bestimmen das abstrakte Bild der Architektur. Dieses wird innerhalb der Beschreibung der A^x-Architektur, die sich an die Vorstellung der Konzepte anschließt, verfeinert. Dazu werden die logischen Komponenten und ihrer jeweiligen Funktionen definiert, die dynamischen Zusammenhänge zwischen den Komponenten beschrieben, sowie die räumlichen Zuordnung der logischen Komponenten zu physikalischen Systemen und verschiedene Organisationsmodelle diskutiert. Weiterhin werden die Systemschnittstellen innerhalb der Architektur und zu externen Systemen analysiert und die notwendigen Nachrichtentypen und auszutauschenden Datenobjekte definiert. Die Erweiterung der A^x-Architektur um Komponenten zur Realisierung der kaufmännischen Unterstützungsdienste wird zum Abschluss des Kapitels vorgestellt.

5.1 Allgemeine Anforderungen an Softwaresysteme

Neben den funktionalen Anforderungen an ein Zugriffskontroll- und Abrechnungssystem insgesamt, wie sie Teil der Zielsetzung der Arbeit sind und in den Kapiteln 2.4 und 2.5 diskutiert wurden, müssen beim Design der Architektur Anforderungen berücksichtigt werden, die generell an Softewaresysteme zu stellen sind. Es gibt verschiedene Modelle zur Evaluation von Softwareprodukten [OPR02]. In dieser Arbeit sollen nach dem FURPS-Modell [Gra92] die Kategorien Funktionalität, Leistungsfähigkeit, Benutzerfreundlichkeit (Usability), Zuverlässigkeit und Supportability unterschieden werden. Daneben ist die Sicherheit von Softwaresystemen insbesondere in offenen Kommunikationsnetzen sehr bedeutsam. Bevor das Design der Architektur vorgestellt wird, sollen die Anforderungen anhand dieser Kategorien kurz erläutert werden, da sie beim

Design der Architektur zu berücksichtigen waren. Ihre Überprüfung wird im Kapitel 6.2 vorgenommen.

Funktionalität. Die Funktionalität des Zugriffskontrollsystems besteht gerade in der Zugriffskontrolle und Abrechnung von Internet-Diensten. Weitere sekundäre Funktionalitätsanforderungen liegen in der Unterstützung beliebiger Geschäftsmodelle und der Mobilität der Dienstnutzer. Um der ersten Anforderung nachzukommen, möglichst alle verschiedenen Ausprägungen von Geschäftsmodellen unterstützen zu können, muss das Zugriffskontrollsystem möglichst viele Formen der Zugriffskontrolle realisieren, wie sie im Kapitel 3.1 unterschieden wurden.

Leistungsfähigkeit. Das Gesamtsystem und dessen einzelne Komponenten müssen leistungsfähig sein in der Hinsicht, dass die Antwortzeit auf eine Dienstanfrage an das System und einzelne Komponenten akzeptabel ist. Es ist erstrebenswert, die Antwortzeit zu minimieren. Die Antwortzeit des Gesamtsystems ist abhängig von der Anzahl der Transaktionen zwischen den Komponenten, der Art und Dauer der Transaktionen und der Antwort- bzw. Berabeitungszeit der Komponenten. Daher müssen auch auch die einzelnen Komponenten lesitungsfähig sein.

Die Lesitungsfähigkeit muss auch bei einer grossen Anzahl von Dienstnutzern und damit einer hohen Anzahl von Anfragen sichergestellt sein. Dazu ist es notwendig, dass das Gesamtsystem und die einzelnen Komponenten skalierbar sind.

Benutzerfreundlichkeit. Allgemeine Merkmale der Benutzerfreundlichkeit oder Usability sind die Erlernbarkeit, Effizienz, Merkbarkeit, Fehlerbehandlung und Nutzerzufriedenheit [Nie94]. Ein Dienstnutzer kann ein Zugriffskontrollsystem zusätzlich anhand weiterer Kriterien beurteilen: Eine Dienstnutzung und die damit verbundene Zugriffskontrolle soll unabhängig von seinem Aufenthaltsort in einer Anbieterdomäne möglich sein, d.h. die Mobilität des Dienstnutzers soll unterstützt werden. Die Anzahl der aktiven für die Zugriffskontrolle notwendigen Identifizierungen und Authentifizierungen, die eine Eingabe des Dienstnutzers verlangen, soll minimiert werden. Weiterhin kann der Wunsch bestehen, dass die Anzahl der Identitätsmerkmale und Authentifizierungs-Informationen, die ein Dienstnutzer für die Zugriffskontrolle auf verschiedene Dienste benötigt und sich dementsprechend merken muss, minimiert wird. Außerdem kann der Dienstnutzer fordern, dass die Abrechnung der genutzten Dienste möglichst einfach ist und gegebenenfalls nur über einen oder wenige Dienstanbieter erfolgt.

Zuverlässigkeit. Die Zuverlässigkeit von Systemen bedeutet, dass sie fehlertolerant sein müssen und insbesondere keinen Single Point of Failure besitzen dürfen, der zu einem Ausfall des Systems führt. Weiterhin muss die Integrität aller Daten, die innerhalb des Systems verwendet und erzeugt werden, gewährleistet sein.

Supportability. Das Verhalten eines Softwaresystems sollte durch dessen Nutzer einfach konfigurierbar, änderbar und an neue Anforderungen anpassbar sein, ohne dass es einen hohen Aufwands oder sogar einer Neuimplementierung bedarf. Zudem soll es gegebenenfalls erweiterbar sein. In verteilten Systemen, in denen ein Softwaresystem nicht als Singulär existiert, sollte grundsätzlich eine Kompatibilität zu externen Systemen, die an das Softwaresystem angebunden

werden, geschaffen werden. Das gilt insbesondere für die Systeme und Anwendungen der Nutzer. Für die Schnittstellen müssen, soweit möglich, standardisierte Formate Verwendung finden.

Sicherheit. Ein System sollte in der Hinsicht sicher sein, dass es nicht böswillig von einem Angreifer mißbraucht oder in seiner Funktionalität gestört werden kann. In verteilten Systemen sind die Systeme selbst und die Kommunikationsverbindungen zwischen den Systemen geeignet abzusichern. Die Sicherheitseigenschaften der Vertraulichkeit, Integrität, Nichtabstreitbarkeit und Authentifizierung der Datenherkunft sind für die Kommunikation zwischen Komponenten des Systems und mit externen Systemen herzustellen.

Ein Zugriffskontrollsystem ist ein System, in welchem die Gewährleistung von Aspekten der Sicherheit eine implizite Aufgabe ist. Dazu ist, wie in Kapitel 2.4 beschrieben, insbesondere die Authentifizierung der Dienstnutzer (Authentifizierung von Entitäten) als Sicherheitsaufgabe durchzuführen. Die Güte eines Zugriffskontrollsystems und seiner Sicherheit ist u.a. abhängig von der Güte der verwendeten Authentifizierungs- und Autorisierungs-Protokolle und -mechanismen. Da die Sicherheitsgüte nicht absolut messbar ist und heute noch als sicher geltende kryptographische Verfahren morgen bereits unsicher sein können, ist es notwendig, dass innerhalb von Systemen ein Austausch der sicherheitsgewährleistenden kryptographischen Verfahren und Protokolle möglich ist [BM99].

5.2 Grundlegende Konzepte der A^X-Architektur

Ein wichtiger Schritt zur Realisierung eines Zugriffskontrollsystems besteht im Design der Architektur. Unter Berücksichtigung der Anforderungen können wesentliche Designüberlegungen angestellt, Entscheidungen über anzuwendende Paradigmen oder Konzepte getroffen und eigene Konzepte zur Realisierung definiert werden. Die grundlegenden Konzepte der A^X-Architektur werden nachfolgend vorgestellt. Dabei handelt es sich um die Konzepte der Separierung der Dienste, der Modularisierung der Teilfunktionen und das Paradigma des policybasierten Managements (vgl. Kapitel 2.7). Diese drei Konzepte zusammen bestimmen die grundlegende Struktur der A^X-Architektur. Weitere Konzepte sind eine einheitliche Identifizierung der Dienstnutzer und ein Session-Konzept.

5.2.1 Separierung von Diensten

Endnutzerdienste und Zugriffskontrolldienste sowie kaufmännische Unterstützungsdienste lassen sich funktional klar voneinander trennen. Dies wurde bereits in Kapitel 2.6 dargestellt. Die Separierung der Endnutzerdienste von den Unterstützungsdiensten und ihre Realisierung in getrennten logischen Komponenten wird in der A^X-Architektur konsequent verfolgt. Damit wird es möglich, zum einen ein Zugriffskontrollsystem für beliebige Internet-Dienste zu entwerfen und zum anderen verschiedene Organisationsmodelle zu unterstützen. Endnutzerdienste werden von der Endnutzerdienstinfrastruktur erbracht, Zugriffskontrolldienste und kaufmännische Unterstützungsdienste von der Unterstützungsdienstinfrastruktur. Die einzelnen Dienste werden, wie bereits in Kapitel 2.4 und Kapitel 2.5 erläutert und in Abbildung 33 nochmals in einer vereinfachten Form

dargestellt, zu unterschiedlichen Zeitpunkten erbracht. Der Dienstnutzer fragt in der sogenannten Pre Service Phase einen Dienst beim Dienstanbieter an. Der Dienstanbieter stellt daraufhin vor der Diensterbringung eine Anfrage nach Unterstützungsdiensten, inklusive eines Zugriffskontrolldienstes, an die Unterstützungsdienstinfrastruktur. Diese führt die Zugriffskontrolle für den vom Dienstnutzer angefragten Dienst durch und teilt dem Anbieter der Endnutzerdienste das Ergebnis mit. Ist das Ergebnis der Zugriffskontrolle positiv, wird daraufhin der Endnutzerdienst erbracht. Zeitgleich wird in der Service Delivery Phase von der Unterstützungsdienstinfrastruktur die Dienstnutzung gemessen und ein Accounting vorgenommen. Die Charge-Calculation, das Billing und das Payment werden zumeist nach der Diensterbringung in der After Service Phase durchgeführt. In einzelnen Fällen müssen diese Funktionen bereits parallel zur Diensterbringung realisiert werden.

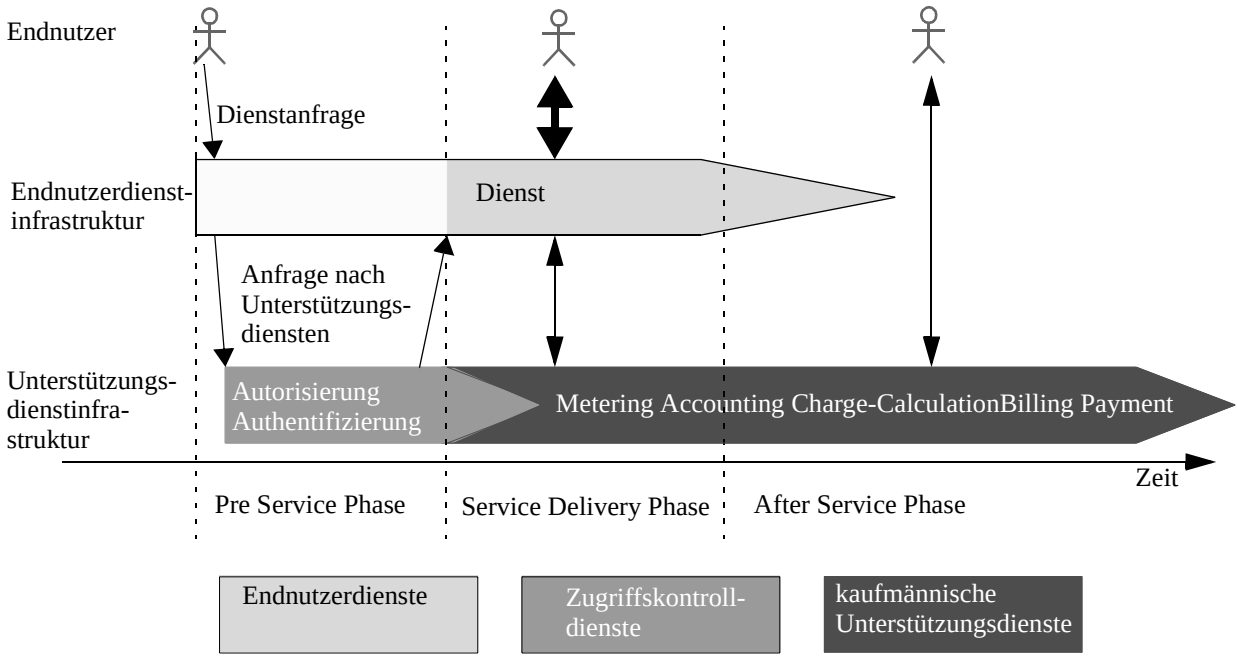

Abbildung 33: Separierung der Dienste

Bei Verwendung einiger Protokolle lassen sich die Phasen nicht unterscheiden. Auf eine Dienstanfrage, z.B. ein HTTP-Request, kann unmittelbar eine Antwort gesendet werden, die der Erfüllung des Dienstes entspricht. Die Zugriffskontrolle kann in diesem Fall grundsätzlich identisch realisiert werden. Es liegt im Ermessen des Dienstanbieters auf welche Anfragen des Dienstnutzers hin er eine Anfrage nach Unterstützungsdiensten stellt.

Ein Zugriffskontrollsystem ist der Teil der Unterstützungsdienstinfrastruktur, der die für die Zugriffskontrolle notwendigen Funktionen in der Pre Service Phase erbringt. Es wird in Kapitel 5.3 beschrieben. Wird es um die kaufmännischen Unterstützungsdienste erweitert, wie dies in Kapitel 5.5 betrachtet wird, handelt es sich um ein vollständiges Zugriffskontroll- und Abrechnungssystem.

5.2.2 Modularisierung der Unterstützungsdienste in Teilfunktionen

Die einzelnen Teilfunktionen der Unterstützungsdienste, wie Authentifizierung und Autorisierung bzw. Accounting, Metering, Charging und Billing werden in der A^X-Architektur als logisch getrennte Komponenten angesehen, wie in Abbildung 34 illustriert. Diese Modularisierung erlaubt die Unterstützung verschiedener Geschäftsmodelle. Die Geschäftsmodelle bestimmen die im Rahmen der Zugriffskontrolldienste und kaufmännischen Dienste auszuführenden Teilfunktionen, wie in Kapitel 4 erörtert. Eine Modularisierung der Funktionen innerhalb der Architektur ermöglicht es dem Dienstanbieter, die notwendige Funktionalität auszuwählen und in Form von Modulen unabhängig voneinander zu nutzen. Weiterhin erleichtert die Modularisierung der Teilfunktionen in verschiedene unabhängige Komponenten den Austausch der zur Realisierung spezieller Funktionen eingesetzten Verfahren. Dies gilt insbesondere auch für die Authentifizierung. In der Authentifizierungs-Komponente können, unabhängig von den anderen Komponenten, beliebige Verfahren eingesetzt werden, falls die Schnittstellen zwischen den einzelnen Komponenten, den Austausch beliebig definierbarer Datenobjekte ermöglichen, und verschiedene Protokolle zur Authentifizierung der Dienstnutzer unterstützt werden. Zusätzlich müssen auch die Anwendungen der Dienstnutzer die Verfahren unterstützen. Ist das der Fall, lässt sich die Forderung nach der Möglichkeit des Austauschs der sicherheitsgewährleistenden kryptographischen Verfahren erfüllen.

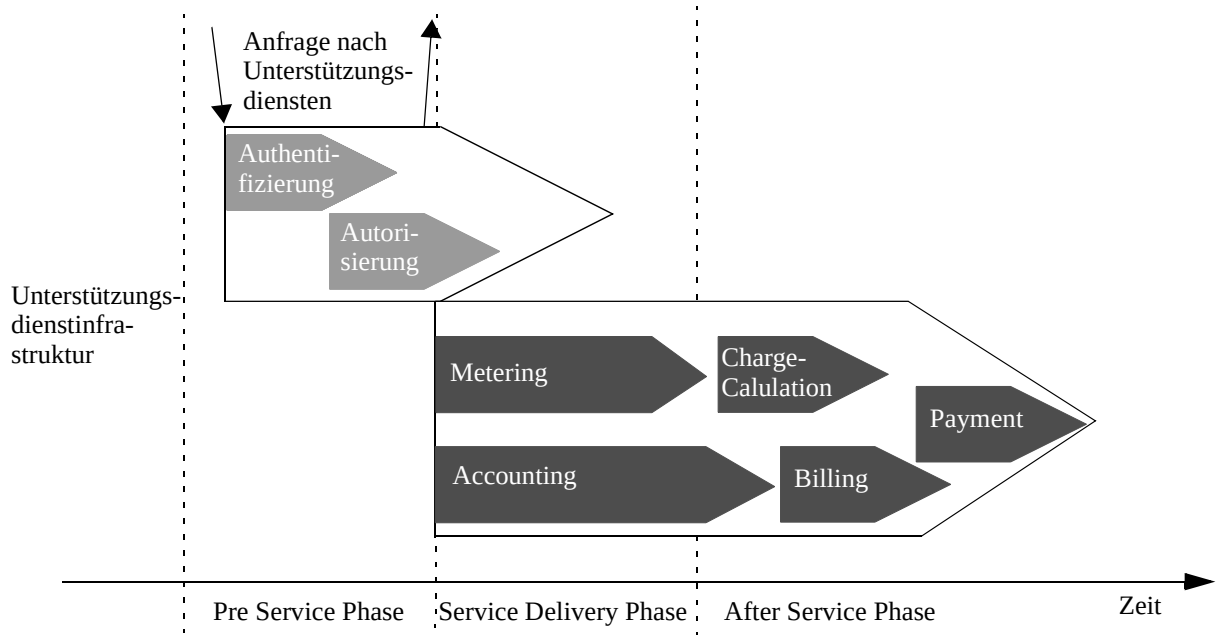

Abbildung 34: Modularisierung der Unterstützungsdienste in Einzelfunktionen

5.2.3 Policybasiertes Management

Als drittes Konzept wird das Paradigma des policybasierten Managements innerhalb der A^X-Architektur angewandt.. Es ermöglicht, wie in Kapitel 2.7 dargestellt, eine Trennung der Beschreibung des Systemverhaltens in Form von Policies von der Implementierung des Systems

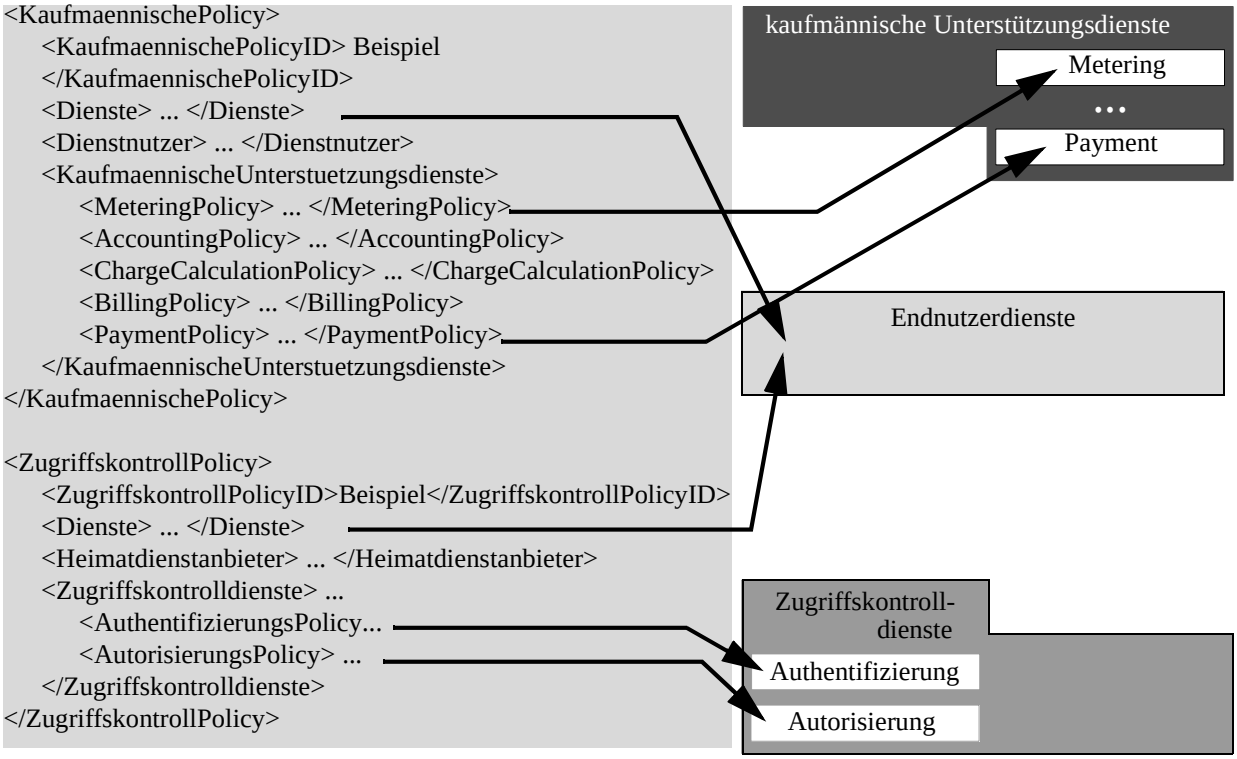

Abbildung 35: Konfiguration der Systemkomponenten durch die operationalen Policies

selbst. Die von der Unterstützungsdienstinfrastruktur zu realisierenden Zugriffskontroll- und kaufmännischen Unterstützungsdienste können in Abhängigkeit vom Geschäftsmodell und für unterschiedliche Dienste, variieren. Daher muss jeder Dienstanbieter in einer Policy spezifizieren können, welche Funktionen, bei der Kontrolle des Zugriffs wie auszuführen sind. Dazu verwendet der Dienstanbieter die in Kapitel 4.4 spezifizierten Policy-Sprachen. Die Policies für die operativen Dienste sind in einzelne Abschnitte unterteilt, die jeweils das Verhalten von Teilkomponenten der Unterstützungsdienstinfrastruktur beschreiben, wie in Abbildung 35 gezeigt. Aufgrund der Modularisierung der Teilfunktionen, können sie dann getrennt voneinander durchgesetzt werden.

5.2.4 Identifizierung der Dienstnutzer und Heimatdienstanbieter

Eine Identifizierung der Dienstnutzer innerhalb der Unterstützungsdienste erfolgt aus zwei Gründen. Zum einen ist sie Voraussetzung zur Authentifizierung, zum anderen ist sie notwendig, um die Ressourcennutzung einem Nutzer als Basis für eine Abrechnung zuordnen zu können. Eine Identifizierung der Dienstnutzer ist nicht notwendig, wenn die Policy eine anonyme Nutzung der Dienste erlaubt und das System anonyme elektronische Bezahlverfahren unterstützt.

Als einheitliches Identitätsmerkmal zum Zwecke der Authentifizierung und Zurechnung der Dienstnutzung zu einem Kunden wird innerhalb der A^X-Architektur eine Benutzerkennung und optional für Internet-Zugangsdienste die Teilnehmeranschlussnummer verwendet. Die Benutzerkennung muss innerhalb der Domäne des Heimatdienstanbieters eindeutig vergeben sein. Zur Realisierung von domänenübergreifenden Szenarien und der Kontrolle des Zugriffs eines fremden Dienstnutzers muss das Identitätsmerkmal global eindeutig sein. Dazu wird die Benutzerken-

nung des Dienstnutzers und die global eindeutige Bezeichnung der Anbieterdomäne kombiniert. So kann innerhalb der A^X-Architektur zugleich auch der Heimatdienstanbieter identifizert werden. Die Teilnehmeranschlussnummer ist global eindeutig und erlaubt in Kombination mit der IMSI auch in Funktelefonnetzen die Bestimmung des Heimatdienstanbieters. In Kapitel 3.1.1 wurde die Teilnehmeranschlussnummer als ausreichend sicheres Identitätsmerkmal angesehen. Stimmt eine Anbieter von Internet-Diensten mit dieser Annahme nicht überein, so kann er auf jeden Fall eine explizite Identifizierung des Dienstnutzers über ein persönliches Merkmal verlangen.

Als gemeinsames Format für Benutzerkennung und Teilnehmeranschlussnummern wird in der A^X-Architektur ein SIP-URI in der Form sip:user@home genutzt. Dieser ist im Session Initiation Protocol (SIP) [RSC+02] definiert. Der User-Teil" enthält die innerhalb der administrativen Domäne eindeutige Benutzerkennung oder eine Telefonnummer, wie in [VS00] spezifiziert. Der Home-Teil enthält den Domänennamen des Heimatdienstanbieters. Beim SIP-URI handelt es sich um einen speziellen Uniform Ressource Identifier (URI) [BLFM98] zur Identifizierung von beliebigen Ressourcen im Internet. Die oben genanten Anforderungen an ein Identitätsmerkmal für die Dienstnutzer sind erfüllt.

```
sip:+49-221-781234@telekom.de
sip:grimm@k.huber_und_partner.de
sip:grimm@rechtundurteil.de
```

Abbildung 36: Beispiele für Identitätsmerkmale der Endnutzer

Eine Person verfügt zumeist über mehrere solcher Identitätsmerkmale, sofern sie bei mehreren Heimatdienstanbietern registriert ist, wie beispielhaft in Abbildung 36 illustriert. Die verschiedenen Identitätsmerkmale einer Person können einander nicht zugeordnet werden. Damit kann der Dienstnutzer, zumindest teilweise, selber steuern, wieweit ein Profil seiner Dienstnutzung erstellt werden kann.

Im Metering und Accounting kann der Ressourcenverbrauch in Zusammenhang mit einem Dienst dem Dienstnutzer auf Basis der IP-Adresse oder einer Session-Nummer zugeordnet werden. Um diesen Ressourcenverbrauch der Benutzerkennung des Dienstnutzer zuordnen zu können, müssen die IP-Adresse oder Session-Nummer mit dieser verbunden werden. Die Verbindung erfolgt innerhalb der A^X-Architektur mit Hilfe einer Session-Beschreibung, die im Rahmen der Zugriffskontrolle angelegt wird.

Als Anforderung der Benutzerfreundlichkeit wurde zuvor in Kapitel 5.1 eine Minimierung der aktiven Identifizierungen und Authentifizierungen genannt. Eine mehrfache Identifizierung kann nur dann vermieden werden, wenn eine implizite Identifizierung oder eine passive explizite Identifizierung möglich ist. Eine implizite Authentifizierung kann nur für Internet-Zugangsdienste bei Nutzung von Wählverbindungen erfolgen, da andere implizit übertragene Identitätsmerkmal wie insbesondere die IP-Adresse als unsicher angesehen werden. Eine passive explizite Identifizierung verlangt, dass die Anwendung des Dienstnutzers und die Anwendungsprotokolle diese unterstützen. Dieses ist nur in seltenen Fällen vorgesehen. Eine Modifikation der Anwendung zum Zwecke der expliziten Identifizierung steht in direktem Widerspruch zur Forderung nach Kompatibilität des Zugriffskontrollsystems zu existierenden Anwendungen (vgl. Supportability-

Anforderungen). Hier existiert also ein unlösbarer Zielkonflikt. Da der Anforderung der Supportability in dieser Arbeit eine höhere Bedeutung zugeordnet wird, muss innerhalb der A^x-Architektur auf eine vollständige Vermeidung der mehrfachen Identifizierung und Authentifizierung verzichtet werden. Eine weitere Möglichkeit eine solche mehrfache aktive Identifizierung und Authentifizierung zu vermeiden wird aber in der A^x-Architektur genutzt. Dazu werden verschiedene von einem Dienstnutzer angeforderte Dienste durch den Dienstanbieter einander zugeordnet. Es wird eine Session gebildet.

5.2.5 Session-Modell

Das Session-Konzept wird als technisches Konzept vielfach dann verwendet, wenn ein Dienst einen bestimmbaren Beginn und ein bestimmbares Ende besitzt, der über Signalisierungsnachrichten angezeigt wird, wie in Kapitel 2.4 beschrieben. Beispiele für Sessions finden sich insbesondere bei Verbindungs- und Kommunikationsdiensten oder datenstromorientierten Diensten. Eine Session entspricht dann dem Zeitraum des Bestehens einer Verbindung, eines Gesprächs oder der Wiedergabe des Datenstroms. Sessions werden z.B. definiert in SIP [RSC+02] zur Signalisierung von VoIP-Telefongesprächen oder Videokonferenzen und im Real Time Streaming Protokoll (RTSP) [SRL98]. Weiterhin wird das Session-Konzept auch verwendet in TLS/SSL [DA99] oder bei QoS-Transportdiensten. In RSVP [BZB+97] [YPG00] entspricht eine Session dem Zeitraum, für den eine Reservierung gemacht wird. In HTTP werden Sessions bei Verwendung des Status Managements [KM00] oder der Nutzung von Proxies bzw. URL-Parametern gebildet, um mehrere aufeinanderfolgende HTTP-Anfragen eines Nutzers einander zuzuordnen.

Das Session-Konzept dient außerdem oftmals dazu die im Metering und Accounting bestimmten Ressourcennutzungen, die einem Identitätsmerkmal eines Geräts zugeordnet sind, im Rahmen des Charging einem persönlichen Identitätsmerkmal zuzuordnen. In einer Session-Beschreibung sind deshalb die verschiedenen Identitätsmerkmale des Dienstnutzers angegeben. Sie wird zum Zeitpunkt der Zugriffskontrolle, zu dem die verschiedenen Identitätmerkmale des Dienstnutzers bekannt sind, angelegt. In den einzelnen Signalisierungsnachrichten wird über die Angabe der Session-Nummer darauf Bezug genommen.

In diesem Fall beschreibt eine Session einen ökonomischen Blickwinkel. Eine Session ist dann eine Zusammenfassung verschiedener zeitgleich oder aufeinanderfolgend erbrachter Einzeldienste, die von einem Dienstnutzer innerhalb einer gewissen Zeitspanne genutzt werden und aus ökonomischer Perspektive einen zusammengehörigen Dienst bilden.

Eine Session innerhalb eines Protokolls muss nicht mit einer Session aus ökonomischer Sicht übereinstimmen. Innerhalb einer Protokoll-Session können aus ökonomischer Sicht mehrere Einzeldienste oder wiederum Sessions genutzt werden. Beispielsweise kann in Anwendungsfall 5, vgl. Kapitel 2.1, Herr Grimm während einer bestehenden SSL-Session auf mehrere verschiedene Texte zugreifen, wie in Abbildung 37 gezeigt. Diese Texte stellen alle einen eigenen ökonomischen Wert mit einem eigenen Preis dar und werden einzeln abgerechnet.

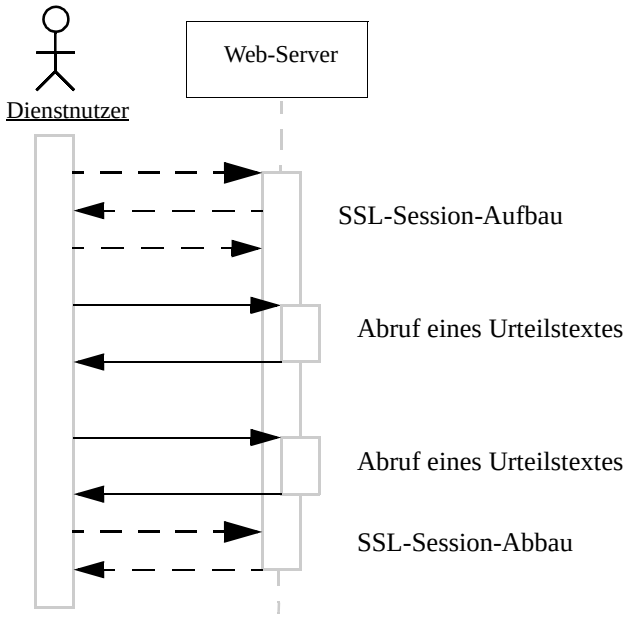

Abbildung 37: Nutzung mehrerer Dienste in einer SSL-Session

Häufig werden auch während einer Protokoll-Session eines Internet-Zugangsdienstes, z.B. während einer RADIUS-Session, mehrere davon völlig unabhängige Anwendungs- und Inhaltsdienste genutzt, wie in Abbildung 38 illustriert.

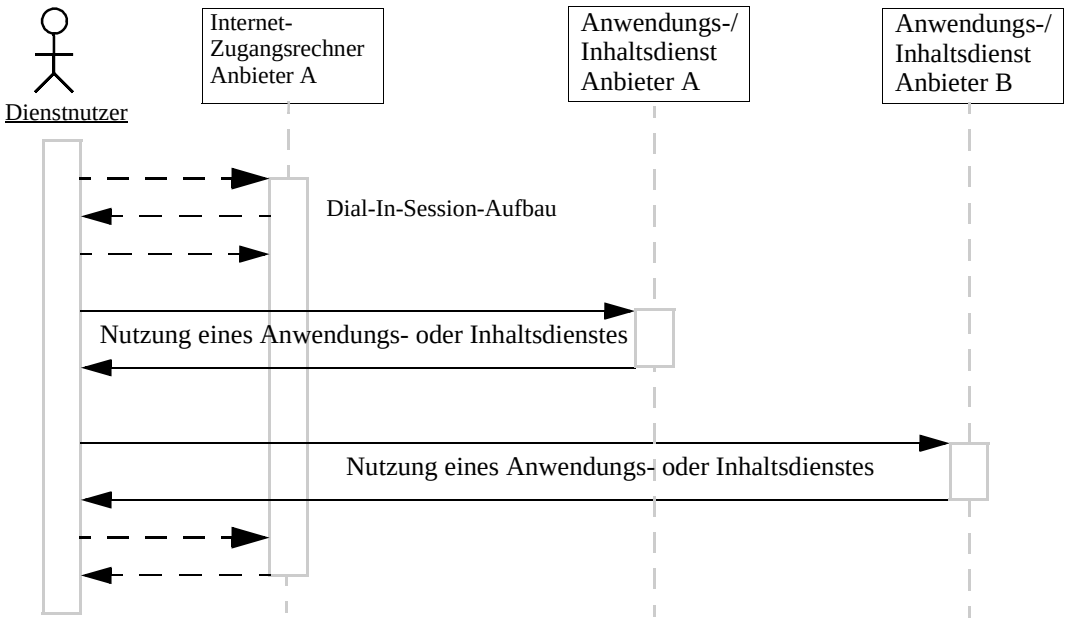

Abbildung 38: Nutzung von Diensten in einer Dial In Session

In der A^X-Architektur werden die beiden Sichten auf die Session miteinander verbunden. Die Informationen über eine Session werden vom Zugriffskontrollsystem in einem fest definierten Format gespeichert. In Anhang C.5 findet sich das XML-Schema für eine Session-Beschreibung.

```
<?xml version="1.0" encoding="UTF-8"?>
<Sessionbeschreibung xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespace-
SchemaLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontroll-
dienste.xsd">
    <SessionID>35-117.52.75.14</SessionID>
    <Session-Beginn>2003-07-15T17:30:00</Session-Beginn>
    <Session-Ende>2003-07-15T17:58:10</Session-Ende>
    <Authentifizierungs-Status>ok-EAP</Authentifizierungs-Status>
    <UserID>grimm@huberpartner.de</UserID>
    <GeraeteID-Typ>IP-Adresse</GeraeteID-Typ>
    <GeraeteID>117.52.139.23</GeraeteID>
    <Session-Kind> 17-117.52.54.103</Session-Kind>
    <Dienstspezifika>
        <Dienstanbieter>117.52.75.14</Dienstanbieter>
        <DienstID>555</DienstID>
        <Dienst-Bezeichnung>Videokonferenz</Dienst-Bezeichnung>
        <MeteringIDTyp>IP-Adresse</MeteringIDTyp>
        <MeteringID>117.52.139.23</MeteringID>
    </Dienstspezifika>
</Sessionbeschreibung>
```

Abbildung 39: Beispiel einer Session-Beschreibung

Jede Session-Beschreibung besteht aus einem allgemeinen und einem dienstspezifischen Teil, wie in Abbildung 39 beispielhaft dargestellt.

Die folgenden Informationen werden in einer Session-Beschreibung gespeichert:

- **Session-Beginn** und **Session-Ende**: Datum und Uhrzeit des Beginns und Endes einer Session werden für ein Auditing und die Charge-Calculation benötigt.
- **Authentifizierungs-Informationen**: Der Authentifizierungs-Status gibt an, ob der Dienstnutzer innerhalb der Session authentifiziert wurde. Erhält das Zugriffskontrollsystem eine Zugriffskontrolldienstanfrage für einen zu einer bestehenden Session gehörenden Dienst, so kann es anhand des Authentifizierungs-Status prüfen, ob bereits eine Authentifizierung erfolgte. Dadurch wird eine mehrfache Identifizierung und Authentifizierung des Dienstnutzers innerhalb einer Session vermieden.
- **User-ID, Geräte-ID**: Die User-ID kennzeichnet den Dienstnutzer. Dieses Feld wird während der Zugriffskontrolle gefüllt. Als Format wird das zuvor beschriebene SIP-URI Format verwendet. Aus der User-ID kann dann der Heimatdienstanbieter bestimmt werden, um ein Billing und Payment über diesen zu realisieren, falls die Policy der kaufmännischen Unterstützungsdienste dies vorsieht. Die Geräte-ID ist diejenige, über die der Dienstanbieter den Dienstnutzer implizit identifizieren kann und die im Metering Verwendung finden. Dies kann bei Internet-Zugangsdiensten die Teilnehmernanschlussnummern sein. Ansonsten handelt es sich zumeist um die IP-Adresse des Dienstnutzers.
- **Session-Kind und Session-Eltern**: Diese Felder dienen der Verknüpfung verschiedener zusammengehöriger Sessions.

- **Dienstspezifische Informationen:** Die dienstspezifischen Informationen sind zur Realisierung der kaufmännischen Unterstützungsdienste notwendig. Sie können in einer Session-Beschreibung für unterschiedliche Dienste mehrfach aufgenommen werden. Zunächst wird mittels des Attributs Dienstanbieter angegeben, wer den Dienst erbringt. Weiterhin wird der Dienst spezifiziert und es werden weitere Merkmale aus der Policy für die kaufmännischen Unterstützungsdienste, wie sie in Kapitel 4.4.2 beschrieben wurde, aufgenommen.
- **Signatur:** Über die gesamte Sessionbeschreibung kann optional eine Signatur gebildet werden. Diese ist zum einen notwendig, falls die Kommunikation innerhalb des verteilten Zugiffskontrollsystems nicht anderweitig gesichert wird oder die Verbindlichkeit einer Sessionbeschreibung als notwendig angesehen wird.

Bindung von Diensten zu einer ökonomischen Session. Aus ökonomischer Sicht dient das Session-Konzept dazu mehrere einzelne Dienste, die als zusammengesetzter Dienst erbracht werden, aneinander zu binden. Sie können dann gemeinsam abgerechnet werden. Dieses ist im Anwendungsfall 6 sinnvoll, wie er in Abbildung 40 vereinfacht dargestellt ist.

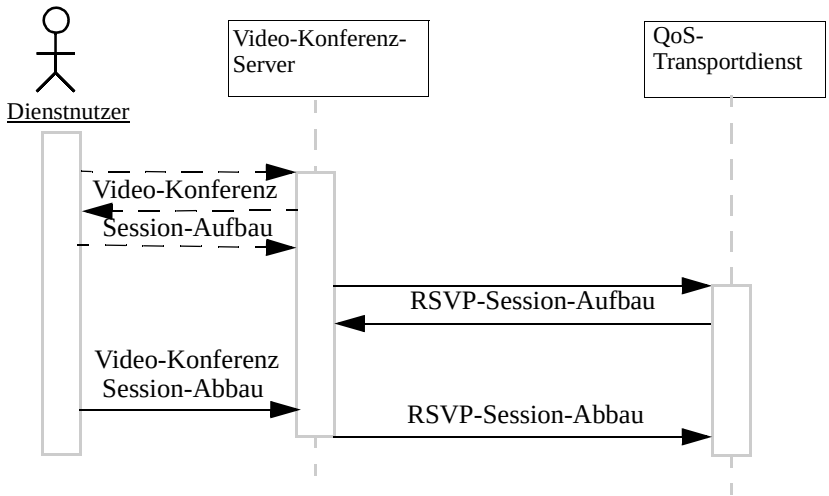

Abbildung 40: Dienste und Sessions im Anwendungsfall 6

Das Video-Konferenzsystem von NRW-On hat eine Session zu einem authentifizierten Dienstnutzer eingerichtet. Der entsprechende Eintrag im Session-Repository des Zugriffskontrollsystems ist oben rechts in Abbildung 41 ausschnittsweise dargestellt. Um eine ausreichende Übertragungsqualität für das Video zu gewährleisten, stellt der Video Konferenz Server eine Reservierungsanfrage für eine QoS-Transportverbindung mit einer hohen Qualität an seinen Backbone-Anbieter. Dieser stellt nun wiederum eine Zugriffskontrolldienstanfrage mit einer eigenen Session-ID an sein Zugriffskontrollsystem und authentifiziert NRW-On als Dienstnutzer. Der entsprechende Eintrag im Session-Repository ist in der Abbildung ebenfalls angegeben. Der Backbone-Anbieter sendet in seiner Dienstantwort an den Video Konferenz Server die für den QoS-Transportdienst gültige Session-ID mit. Der Video Konferenz Server kann nun die beiden Sessions einander zuordnen und mittels einer Zugriffskontrollnachricht an sein Zugriffskontroll-

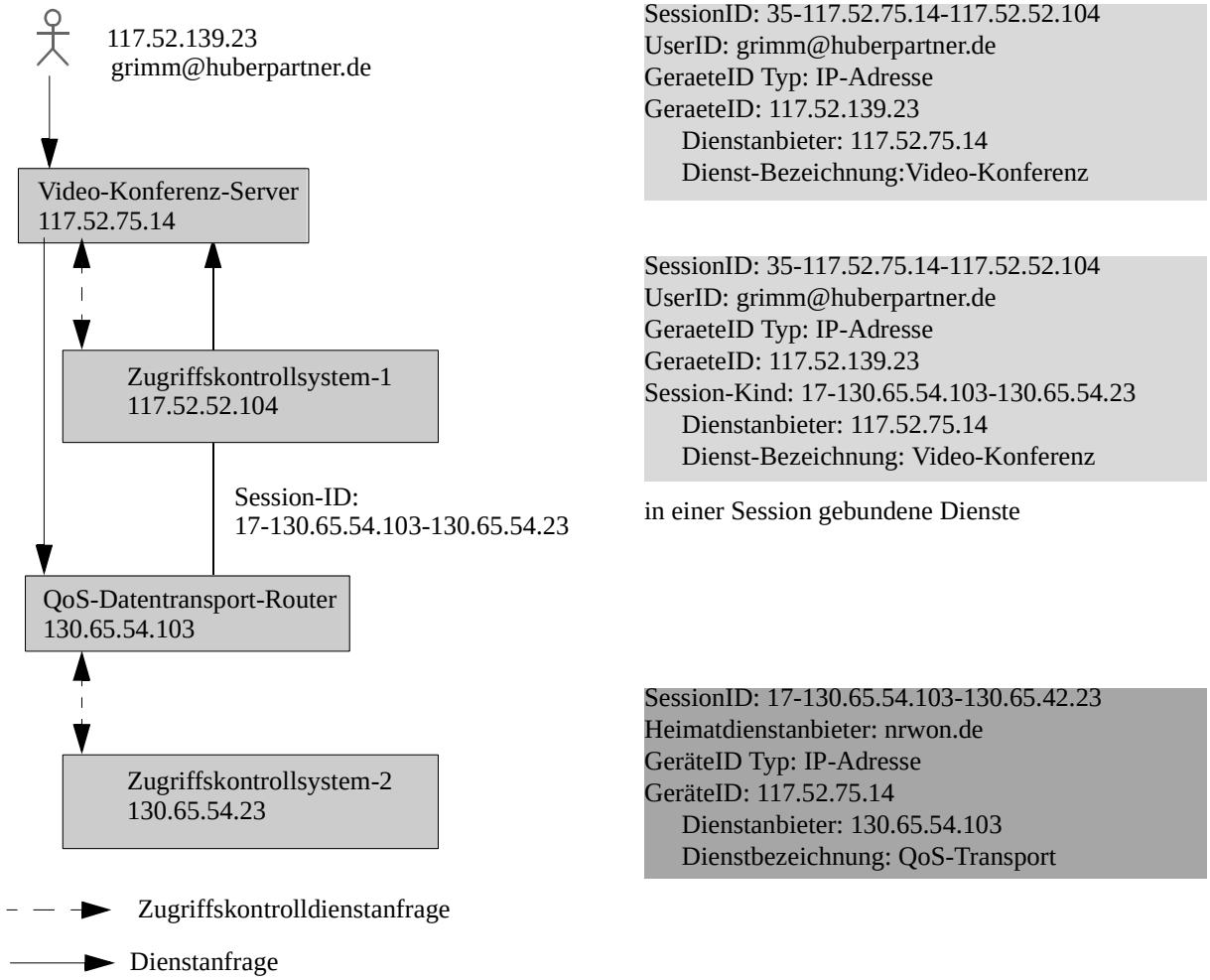

Abbildung 41: Bindung von Diensten zu einer ökonomischen Session

system eine Bindung veranlassen. Damit ist NRW-On in der Lage den QoS-Transportdienst Hernn Grimm als eigentlichem Dienstnutzer zuzuordnen.

Vermeidung der mehrfachen Authentifizierung auf Basis einer technischen Session. Das Session-Konzept wird innerhalb des Zugriffskontrollsystems neben dem Zweck der Bindung von Diensten weiterhin dazu verwendet, eine mehrfache Identifizierung und Authentifizierung des Dienstnutzers bei der Anfrage mehrerer Dienste innerhalb einer Protokoll-Session zu vermeiden. Die entsprechende Zuordnung von Einzeldiensten zu einer Session muss durch die Endnutzerdienstinfrastruktur erfolgen. Sie verwaltet die Session auf Ebene eines Protokolls und nur sie verfügt somit über die Informationen, um verschiedene Dienste einer Session zuzuordnen. Bei der ersten Anfrage des Dienstanbieters an das Zugriffskontrollsystem innerhalb einer technischen Session, vergibt dieser eine Session-ID. Diese besteht aus drei Bestandteilen, einer lokal eindeutigen Nummer und einem global eindeutigen Identifikator der Endnutzerdienstinfrastruktur sowie des Zugriffskontrollsystems, wie z.B. ihrer statischen IP-Adressen. Die Session-ID wird als Attribut der Zugriffskontrolldienstanfrage an das Zugriffskontrollsystem übermittelt. Nachfolgende Anfragen an das Zugriffskontrollsystem, die zur gleichen Session gehören und damit den identi-

schen Dienstnutzer haben, werden unter Verwendung der gleichen Session-ID an das Zugriffskontrollsystem geschickt. Dieses kann, mittels einer Anfrage an ein Session-Repository, den Status der Session prüfen und so feststellen, wer der Dienstnutzer ist und ob er identifiziert und authentifiziert ist. Dieses Verfahren wird immer dann genutzt, wenn der mit der Session verbundene Dienst und der während der Session angefragte Dienst vom gleichen Dienstanbieter erbracht werden. Das gilt z.B. bei mehreren HTTP-Anfragen innerhalb einer SSL-Session. In Anwendungsfall 5, der in Abbildung 37 gezeigt ist, wird dieses Verfahren genutzt. Der Web-Server verwendet bei seinen Zugriffskontrolldienstanfragen an das Zugriffskontrollsystem jeweils die identische Session-Nummer, wenn Herr Grimm die Urteilstexte anfordert.

Eine Verwendung der Informationen über eine existierende Session kann in einzelnen Fällen auch zur Vermeidung der mehrfachen Authentifizierung genutzt werden, wenn die Session nicht vom Dienstanbieter dessen Dienstnutzung zu autorisieren ist verwaltet wird, beide aber das identische Zugriffskontrollsystem nutzen. Dies soll an dem in Abbildung 38 illustrierten Fall erläutert werden. Zur Nutzung des Dial In Internet-Zugangs muss sich der Dienstnutzer über seine persönlichen Identitätsmerkmale authentifizieren. Er erhält vom Dienstanbieter eine dynamisch vergebene IP-Adresse zugewiesen. Das Zugriffskontrollsystem speichert die Session-Beschreibung, wie in Abbildung 42 oben rechts dargestellt. Teil dieser Beschreibung ist auch die IP-Adresse als Geräte-ID.

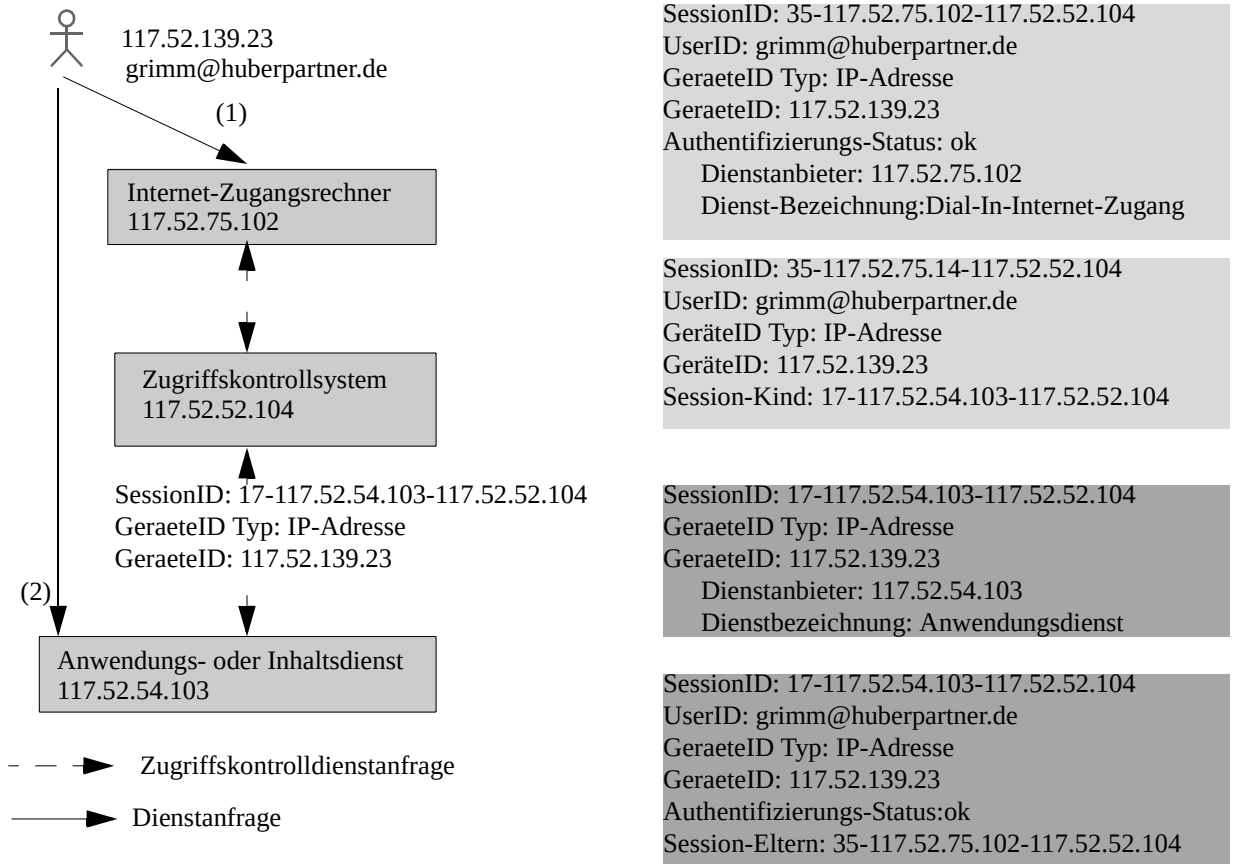

Abbildung 42: Bindung von Sessions zur Vermeidung einer mehrfachen Authentifizierung

Im Anschluss fragt der Dienstnutzer einen Anwendungsdienst von einem anderen Dienstanbieter innerhalb der selben administrativen Domäne an. Dieser Dienstanbieter kann anhand der IP-Absenderadresse erkennen, dass ein anderer Dienstanbieter seiner eigenen Domäne den Internet-Zugang als Dienst bereitstellt. Kann er davon ausgehen, dass die Kommunikation zwischen dem Internet-Zugangsrechner und ihm sicher ist, so kann er die IP-Adresse als einziges Identifizierungsmerkmal innerhalb der Zugriffskontrolldienstanfrage angeben. Das Zugriffskontrollsystem wiederum kann die bestehende Session bestimmen und die beiden Sessions einander zuordnen. Er vererbt dann den Authentifizierungs-Status der bestehenden Session auf die neue Session, wie ebenfalls in Abbildung 42 dargestellt. Für den angefragten Anwendungsdienst ist dann keine Authentifizierung des Dienstnutzers notwendig, auch wenn die Zugriffskontroll-Policy diese vorsieht.

5.3 Überblick über die A^x-Architektur

Die vorgestellten Konzepte bestimmen ein abstraktes Modell der A^x-Architektur, welches im wesentlichen dem in Kapitel 2.7.2 Modell entspricht. Dieses abstrakte Modell soll an dieser Stelle verfeinert werden. Dazu wird die A^x-Architektur im Folgenden textuell und anhand verschiedener Abbildungen vorgestellt. Dabei beschränkt sich die Darstellung in diesem und dem folgenden Abschnitt auf die Funktionalität der Zugriffskontrolle auf Internet-Dienste in der Pre-Service-Phase. Die Erweiterung der Architektur zur Realisierung der kaufmännischen Funktionen wird in Kapitel 5.5 dargestellt. Die Beschreibung der A^x-Architektur gliedert sich in folgende Abschnitte:

- Beschreibung der Gesamtfunktionalität zur Zugriffskontrolle,
- Beschreibung der einzelnen logischen Komponenten und ihrer Funktionalitäten,
- Beschreibung der dynamischen Zusammenhänge zwischen den einzelnen logischen Komponenten,
- Beschreibung der unterstützten organisatorischen Modelle,
- Beschreibung der physikalischen Lokalisierung der logischen Komponenten,
- Beschreibung der notwendigen Vertrauensverhältnisse.

Ein nach der Architektur A^x-Architektur zu realisierendes System wird A^x-System genannt. Es handelt sich dabei somit um eine spezielle Form eines Zugriffskontrollsystems.

5.3.1 Gesamtfunktionalität der A^x-Architektur zur Zugriffskontrolle

Die Aufgabe des A^x-Systems, als Zugriffskontrollsystem besteht, entsprechend der funktionalen Anforderungen, in der Erbringung der Zugriffskontrolldienste. Wie im Anwendungsfalldiagramm in Abbildung 43 gezeigt, trifft das A^x-System die Entscheidung, ob der angefragte Dienst erbracht wird oder nicht. Der Dienstanbieter stellt dazu eine sogenannte Zugriffskontrolldienstanfrage an das A^x-System. Für die Zugriffskontrollentscheidung muss das A^x-System verschiedene Teilfunktionen realisieren. Welche das sind wird in den Zugriffskontroll-Policies des Dienstanbieters spezifiziert, wie in Kapitel 4.4.1 beschrieben. Die Policy muss in Abhängigkeit vom angefragten Dienst aus dem Policy-Repository abgefragt werden. Entsprechend der Zugriffskontroll-Policy,

kann es notwendig sein, dass das AX-System den Dienstnutzer authentifiziert, den vorgelegten Berechtigungsnachweis prüft oder eine dynamische Autorisierung vornimmt. Für die Authentifizierung ist eine Identifizierung Voraussetzung. Weiterhin sind bei privaten Diensten die Nutzerberechtigungen zu prüfen.

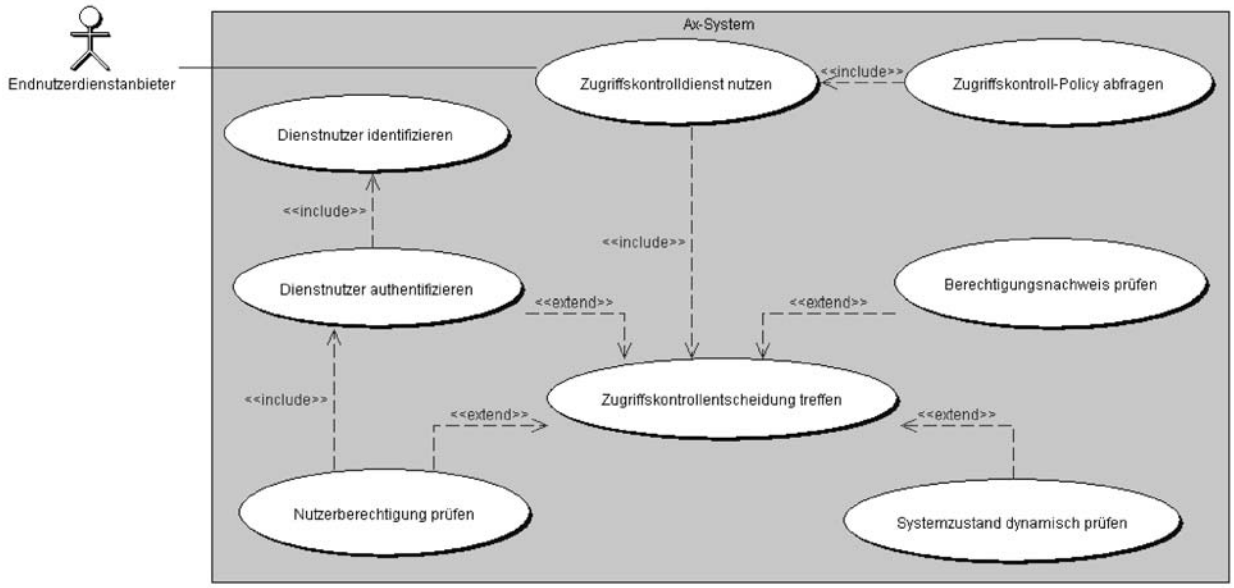

Abbildung 43: Anwendungsfall: Zugriffskontrolldienst nutzen

Zusätzlich zu diesen Funktionen, die für die Nutzung des Zugriffskontrolldienstes benötigt werden, muss das AX-System auch Funktionen zur Konfiguration durch den Anbieter von Endnutzerdiensten bereitstellen, wie sie in Abbildung 44 aufgeführt sind. Die Konfiguration des AX-Systems erfolgt über die Policies des Anbieters der Internet-Dienste. Diese müssen an das AX-System übermittelt und von diesem gespeichert werden. Weiterhin müssen die Identifizierungs- und Authentifizierungs-Informationen sowie die Berechtigungsnachweise in Repositories des AX-Systems hinterlegt werden können.

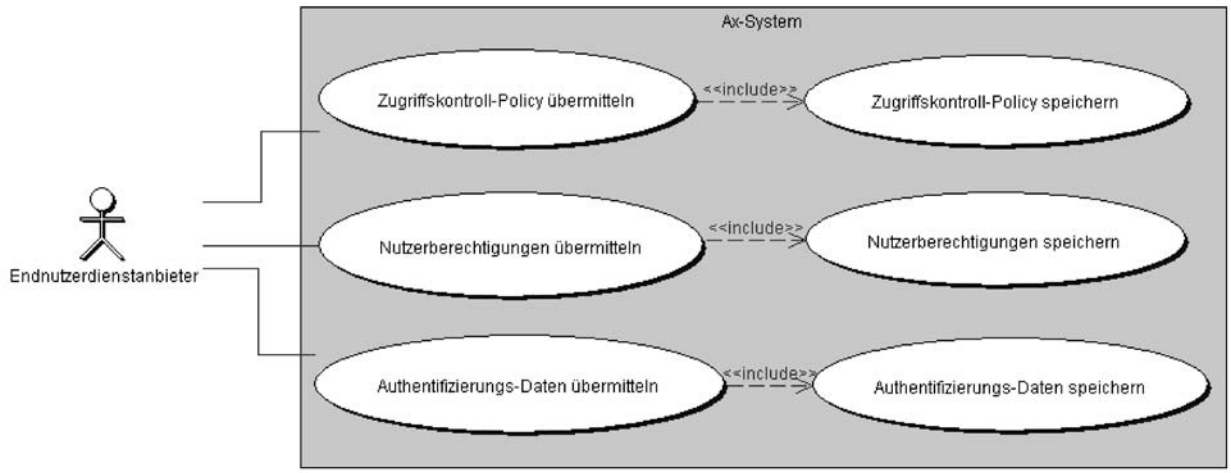

Abbildung 44: Anwendungsfall: AX-System konfigurieren

Der Ablauf der Nutzung des A^X-Systems in einer globalen, vereinfachten Sicht ist in Abbildung 45 illustriert. Für die Erbringung der einzelnen Funktionen der Zugriffskontrolle, insbesondere für die Authentifizierung, kann der Austausch weiterer Nachrichten bis hin zum Dienstnutzer notwendig sein. Dies wird nachfolgend detaillierter betrachtet werden. Abgesehen von diesen Nachrichten werden die Zugriffskontrolldienste ohne Beteiligung des Dienstnutzers für diesen transparent erbracht.

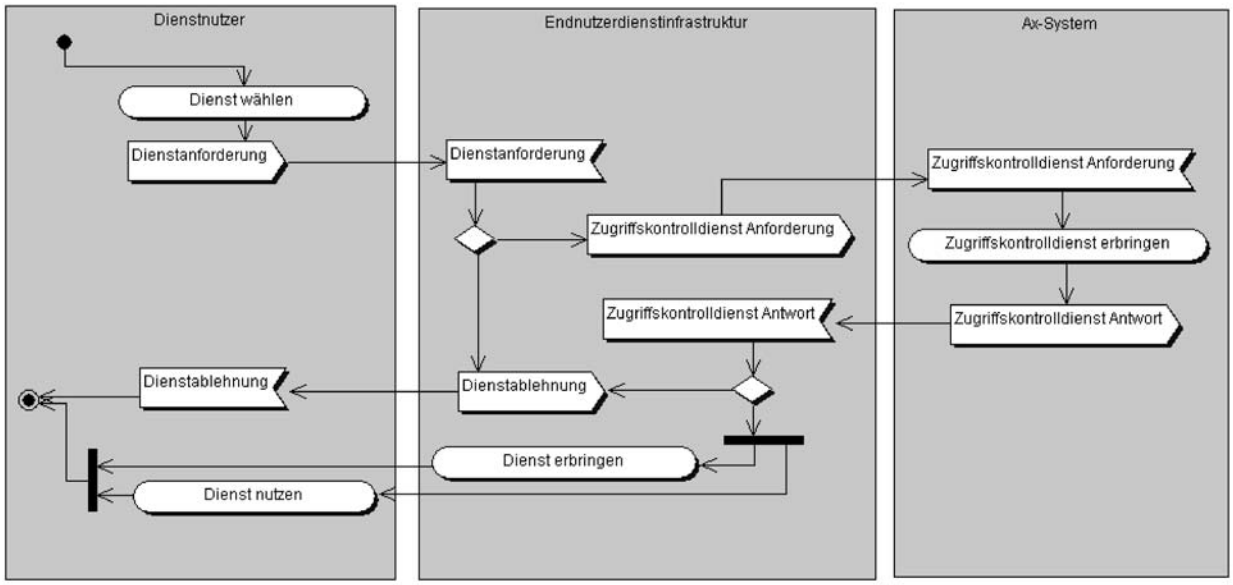

Abbildung 45: Nutzung der Zugriffskontrolldienste

5.3.2 Logische Komponenten der A^x-Architektur

Die Komponenten der A^X-Architektur, wie sie in Abbildung 46 gezeigt sind, lassen sich aus dem in Kapitel 2.7.2 vorgestellten Basis Schema von Policy-Architekturen ableiten. Die Zugriffskontroll-Policies werden in einem Policy-Repository (PR) gespeichert. Die Auswertung der Policies, also die Bestimmung der auszuführenden Zugriffskontrollfunktionen, erfolgt durch einen Policy Decision Point (PDP). Dieser ist Teil des A^X-Servers. Die Durchsetzung der Policies, also die Durchführung der einzelnen Funktionen der Zugriffskontrolle, erfolgt durch die Policy Enforcement Points (PEPs). Dabei existiert für jede Teilfunktion mindestens ein einzelner Policy Enforcement Point, d.h. ein Autorisierungs-PEP und ein Authentifizierungs-PEP. Ein Autorisierungs-PEP kann entsprechend der drei Formen der Autorisierung in drei Formen existieren.

Weitere Komponenten der A^X-Architektur sind ein Rechte-Repository und ein Authentifizierungs-Repository. Diese speichern die Nutzerberechtigungen der Dienstnutzer bzw. die zur Identifizierung und Authentifizierung notwendigen Informationen.

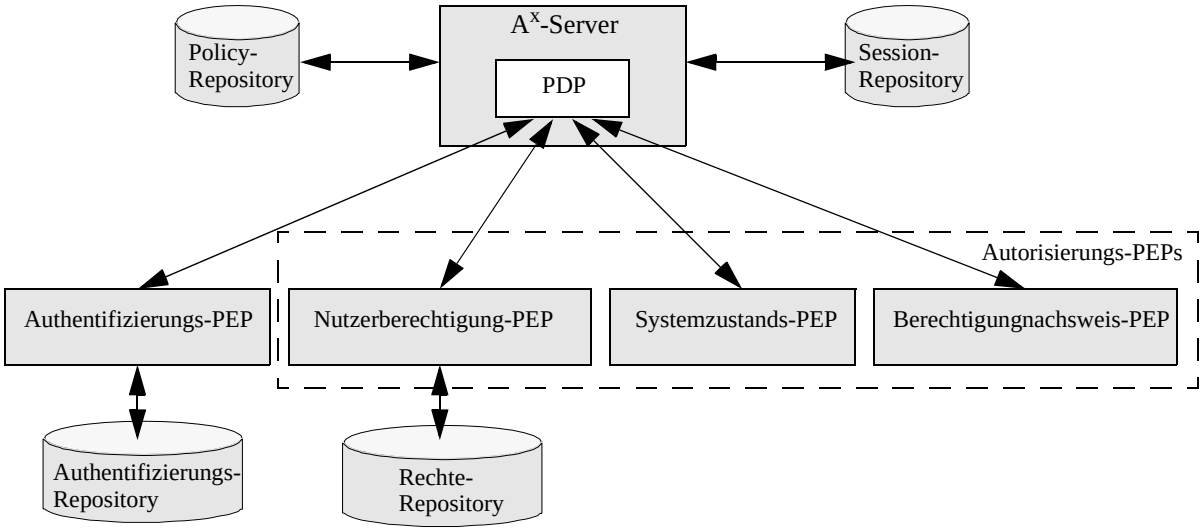

Abbildung 46: Komponenten der A^X-Architektur

A^X-Server. Der A^X-Server mit seinem Policy Decision Point ist die zentrale Komponente eines A^X-Systems. Er nimmt die Zugriffskontrolldienstanfragen eines Anbieters von Endnutzerdiensten entgegen und fragt die dem angefragten Dienst entsprechende Policy aus dem Policy-Repository und den Session-Status aus dem Session-Repository ab. Der Policy Decision Point wertet sie gemeinsam aus und konfiguriert dementsprechend die Authentifizierungs- und Autorisierungs-PEPs. Diese melden ihr Ergebnis an den A^X-Server zurück, der die letztendliche Zugriffskontroll-entscheidung trifft, ob der Endnutzerdienst erbracht wird, und an den Anbieter der Endnutzerdienste zurückmeldet. Dazu führt er die Ergebnisse der einzelnen Teilfunktionen zusammen.

Policy Decision Point. Der Policy Decision Point ist Teil des A^X-Servers. Er wertet die Zugriffskontroll-Policy aus und konfiguriert die notwendigen Policy Enforcement Points.

Session-Repository. Innerhalb des Session-Repositories speichert der A^X-Server den Status einer Session, wie in Kapitel 5.2.5 beschrieben.

Policy-Repository. Im Policy-Repository werden die Zugriffskontroll-Policies gespeichert. Diese müssen vom Dienstanbieter geeignet erstellt und im Policy-Repository gespeichert werden (vgl. Abbildung 44). In der Anfrage des Dienstanbieters an den A^X-Server muss der Endnutzerdienst und gegebenenfalls der Heimatdienstanbieter des Dienstnutzers spezifiziert werden, damit die zugehörige Policy bestimmt werden kann. Da die Erstellung der Policies und auch die Zugriffskontrolldienstanfrage durch den Anbieter der Endnutzerdienste erfolgt, sind hierfür nicht notwendigerweise standardisierte Wertebereiche zu verwenden. Vielmehr genügt dafür ein Identifikator, wie er in Kapitel 4.3.1 beschrieben wurde.

Rechte-Repository. Im Rechte-Repository sind die zur Autorisierung privater Dienste notwendigen Nutzerberechtigungen gespeichert. Sie müssen vom Heimatdienstanbieter des Nutzers angelegt werden, können aber an verschiedenen Orten, z.B. auch bei einem Broker, repliziert

gespeichert werden, sofern die dafür notwendigen Vertrauensverhältnisse und Vereinbarungen bestehen.

Authentifizierungs-Repository. Das Authentifizierungs-Repository enthält die Identifizierungs- und Authentifizierungs-Informationen der Dienstnutzer. Sie werden ebenfalls vom Heimat Dienstanbieter des Dienstnutzers erstellt, da nur dieser den Dienstnutzer und seine Authentifizierungs-Informationen kennt. Eine replizierte Speicherung ist in gleicher Weise möglich.

Autorisierungs-PEP. Der Autorisierungs-PEP führt innerhalb des A^X-Systems die Teilfunktion der Autorisierung durch. So kann ein vorgelegter Berechtigungsnachweis überprüft werden, indem z.B. ein Nutzerzertifikat auf seine Gültigkeit geprüft wird. Es kann die Nutzerberechtigung geprüft werden, indem der vom Dienstnutzer angefragte Dienst mit seinen im Rechte-Repository gespeicherten Nutzerberechtigungen geeignet verglichen wird. Zuletzt können zur Realisierung der dynamischen Autorisierung verschiedene Systemzustände geprüft werden.

Um diese Funktionen zu verwirklichen, muss der Autorisierungs-PEP oftmals das Rechte-Repository und den Systemzustand externer Systeme abfragen. Dabei kann es sich sowohl um Systemzustände anderer zentraler Unterstützungssysteme, wie z.B. kaufmännischer Systeme, aber auch um dezentrale Systemzustände, wie z.B. die Ressourcenverfügbarkeit der Endnutzerdienstinfrastruktur, handeln. Letzteres ist unter dem Gesichtspunkt der Performance in der Regel nicht sinnvoll. Daher kann der Dienstanbieter alternativ diese Prüfung selbst durchführen und nur, wenn die Prüfung zu einem positiven Ergebnis kommt, eine Zugriffskontrolldienstanfrage an den A^X-Server stellen.

Bei den Ergebnissen alle drei Formen der Prüfung handelt es sich grundsätzlich um eine logische Entscheidung, welche die PEPs an den A^X-Server zurückmelden. Bei einer Ablehnung ist es notwendig, auch den Grund für die Ablehnung zurückzumelden, damit dieser an den Dienstanbieter und den Dienstnutzer weitergeleitet werden kann und so zwischen Dienstnutzer und Dienstanbieter Dienste ausgehandelt werden können.

Authentifizierungs-PEP. Der Authentifizierungs-PEP führt die Authentifizierung des Dienstnutzers durch. Die Identitätsmerkmale des Dienstnutzers und notwendige Authentifizierungs-Informationen müssen als Teil der Zugriffskontrolldienstanfrage an den A^X-Server übermittelt und an den Authentfizierungs-PEP weitergeleitet werden. Der Authentifizierungs-PEP fragt, falls die notwendigen Informationen nicht Bestandteil der ersten Zugriffskontrolldienstanfrage sind, und, falls dies das in der Policy angegebene Authentifizierungs-Verfahren und -Protokoll vorsieht, Identitätsmerkmale und Authentifizierungs-Informationen vom Dienstnutzer ab. Dazu kommuniziert der Authentifizierungs-PEP über den A^X-Server und die Endnutzerdienstinfrastruktur als Proxies mit dem Dienstnutzer selbst. A^X-Server und Endnutzerdienstinfrastruktur müssen in diesem Fall das Authentifizierungs-Protokoll nicht aktiv unterstützen sondern nur Mechanismen zur Weiterleitung der Nachrichten des Authentifizierungs-Protokolls anbieten. Zur Authentifizierung selbst greift der Policy Enforcement Point auf die im Authentifizierungs-Repository gespeicherten Informationen zu.

A^X-Client. Eine weitere Komponente, die nur eingeschränkt der Architektur zuzurechnen ist, ist der A^X-Client. Es handelt sich um die Komponente innerhalb der Endnutzerdienstinfrastruktur, welche ausgelöst durch eine Dienstanfrage des Dienstnutzers eine Zugriffskontrolldienstanfrage an den A^X-Server stellt und die weitere Kommunikation mit dem A^X-Server übernimmt.

Komponente		Funktionalitäten
A^X-Client		Zugriffskontrolldienstanfrage stellen und Antwort entgegennehmen
A^X-Server		Zugriffskontrolldienstanfrage entgegennehmen Zugriffskontroll-Policy abfragen Session-Beschreibung anlegen, aktualisieren und abfragen Ergebnis der Zugriffskontroll-PEPs entgegennehmen Zugriffskontrollentscheidung treffen Zugriffskontrolldienstanfrage beantworten
	PDP	Zugriffskontroll-Policy auswerten Zugriffskontroll-PEPs konfigurieren
Policy-Repository		Zugriffskontroll-Policy speichern / zur Verfügung stellen
Session-Repository		Session-Beschreibung speichern / zur Verfügung stellen
Rechte-Repository		Nutzerberechtigungen speichern / zur Verfügung stellen
Authentifizierungs-Repository		Identifizierungs- und Authentifizierungs-Informationen speichern / zur Verfügung stellen
Zugriffskontroll-PEPs		Konfiguration entgegennehmen Ergebnis zurückmelden
	Nutzerberechtigungs-PEP	Nutzerberechtigung abfragen Nutzerberechtigung überprüfen
	Systemzustands-PEP	Systemzustand externer Systeme abfragen Systemzustand überprüfen
	Berechtigungsnachweis-PEP	Berechtigungsnachweis überprüfen
	Authentifizierungs-PEP	Authentifizierungs-Informationen abfragen Authentifizierung durchführen

Tabelle 9: Funktionalitäten der Komponenten der A^X-Architektur

In Tabelle 9 sind die Funktionalitäten der einzelnen Komponenten in übersichtlicher Form zusammengefasst.

5.3.3 Dynamische Zusammenhänge zwischen den logischen Komponenten

Die Gesamtfunktionalität des A^X-Systems wie es in Abbildung 45 in der Aktivität “Zugriffskontrolldienst erbringen” zusammengefasst wurde, ergibt sich aus dem Zusammenspiel der einzelnen Komponenten. Die dynamischen Zusammenhänge zwischen den Komponenten lassen sich in

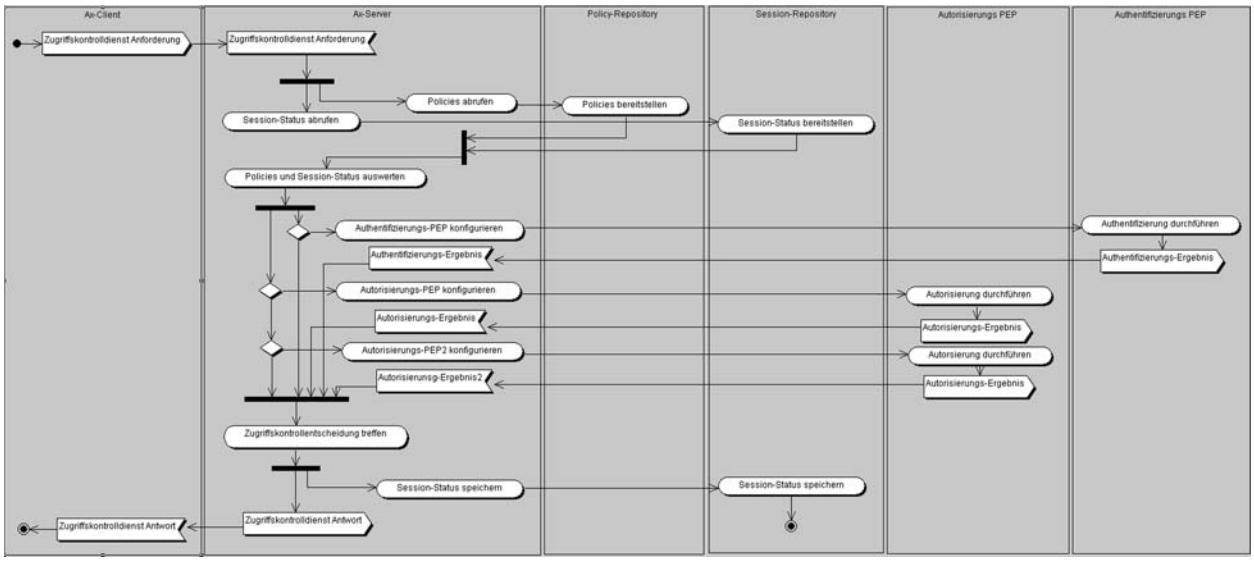

Abbildung 47: Dynamische Zusammenhänge zwischen Komponenten

einem Aktivitätendiagramm (vgl. Abbildung 47) illustrieren. An diesem Diagramm lässt sich zusammenfassend erläutern, wie ein Zugriffskontrolldienst realisiert wird:

Empfängt der A^X-Server eine Zugriffskontrollanfrage vom A^X-Client, so ruft er die dem Dienst zugeordnete Policy aus dem Policy-Repository ab. Weiterhin bestimmt er den Session-Status aus der Session-Datenbank, sofern der Dienst innerhalb der Zugriffskontrolldienstanfrage vom Dienstanbieter einer bestehenden Session zugeordnet wurde. Der Policy Decision Point wertet die Policy aus und bestimmt dadurch, welche Teilfunktionen der Zugriffskontrolle, von wem und in welcher Form auszuführen sind. Der Authentifizierungs-PEP, falls die Policy eine Authentifizierung vorsieht, und der oder die Autorisierungs-PEPs werden entsprechend konfiguriert. Die PEPs führen jeweils ihre Teilfunktion der Zugriffskontrolle aus und senden das Ergebnis zurück an den A^X-Server. Dieser kann auf Basis der ihm vorliegenden Informationen die endgültige Zugriffskontrollentscheidung treffen und das Ergebnis an den A^X-Client melden.

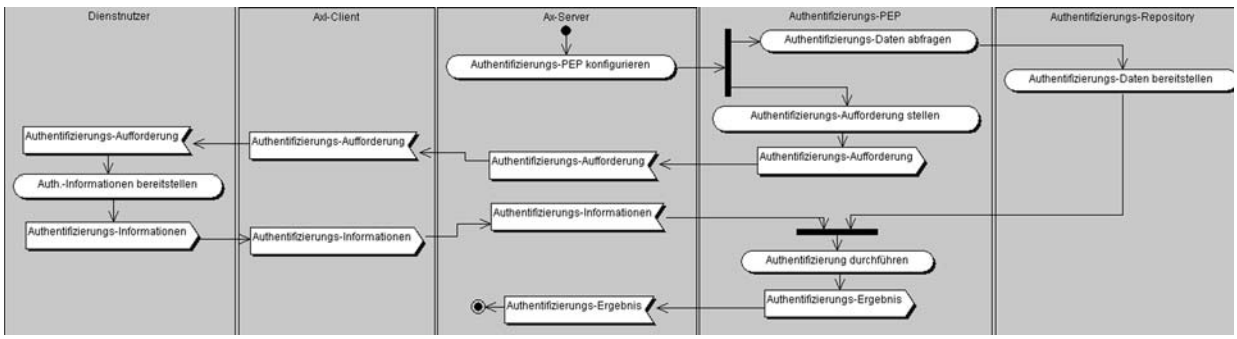

Abbildung 48: Authentifizierung der Dienstnutzer

Die Aktivität “Authentifizierung durchführen” kann in Abhängigkeit vom gewählten Authentifizierungs-Verfahren und -protokoll eine Kommunikation mit dem Dienstnutzer beinhalten. Beispielsweise müssen bei der Verwendung von Challenge Response Verfahren Authentifizierungs-

Informationen ausgetauscht werden. Diese Kommunikation findet, wie in Abbildung 48 gezeigt, nicht direkt statt, sondern über den AX-Server und die Endnutzerdienstinfrastruktur als Proxy.

5.3.4 Aufbau des AX-Servers

Der AX-Server ist die zentrale Komponente der AX-Architektur. Abbildung 49 zeigt seinen Aufbau. Innerhalb des AX-Servers kontrolliert ein zentrales Steuerungsmodul den gesamten Ablauf der Zugriffskontrolle. Es besteht aus einer Status-Maschine, die den Fortschritt der internen Bearbeitung überwacht, einem zentralen Entscheidungsmodul und dem PDP. Die Status-Maschine wird initialisiert über eine Zugriffskontrolldienstanfrage und in ihrem Ablauf gesteuert vom PDP. Erst wenn die endgültige Zugriffskontrollentscheidung durch das zentrale Entscheidungsmodul getroffen ist, kann eine Rückmeldung an die Endnutzerdienstinfrastruktur erfolgen. Zur Kommunikation mit der Endnutzerdienstinfrastruktur und den Policy Enforcement Points werden Protokolle, deren Überwachung entsprechende Protokollmaschinen übernehmen, verwendet. Zur Verwaltung der Sessions und zur Abfrage der Zugriffskontroll-Policies dienen korrespondierende Module innerhalb des AX-Servers.

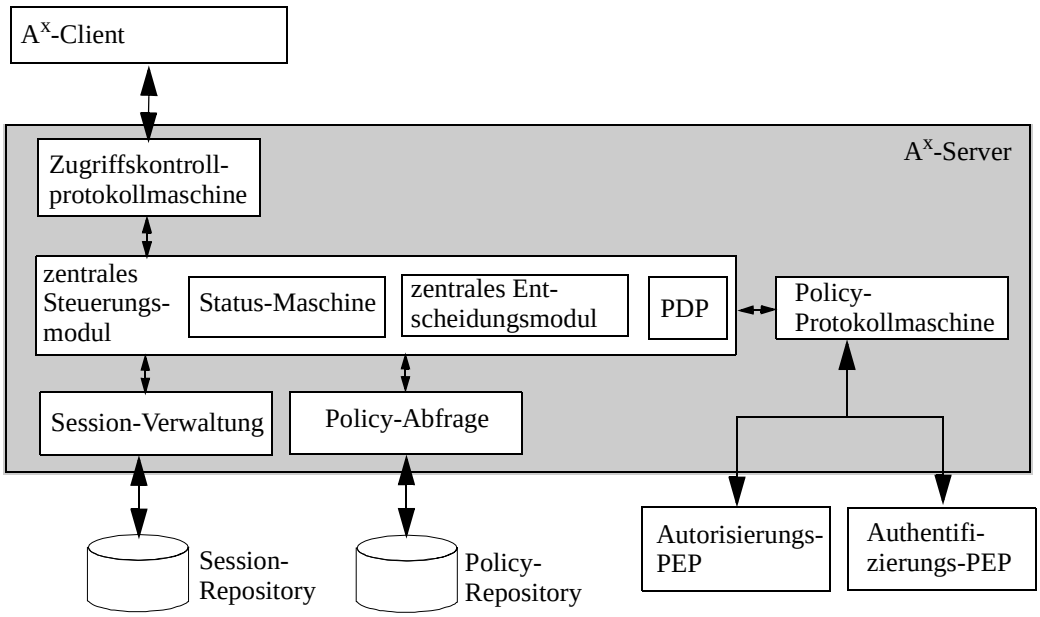

Abbildung 49: Aufbau des AX-Servers

5.3.5 Aufbau eines Policy Enforcement Points

Ein Policy Enforcement Point erhält seine Konfigurationsinformationen in Form einer Protokollnachricht vom AX-Server. Diese Kommunikation wird von einer Policy-Protokollmaschine abgewickelt, wie in Abbildung 50 dargestellt. Zusätzlich muss der Policy Enforcement Point oftmals Informationen aus Repositories und von unabhängig existierenden externen Systemen abfragen, um seine Funktionalität zu erbringen, wie bei der Beschreibung seiner Funktionalität erörtert. Ein Policy Enforcement Point muss daher eine entsprechende API oder ein anwendungsspezifisches

Protokoll unterstützen. Dazu dient das anwendungsabhängige Kommunikationsmodul. Dieses transferiert die Policy-Protokoll Nachricht in eine anwendungsspezifische Anfrage. Das Ergebnis der Anfrage gibt es, wiederum in ein einheitliches Format übertragen, an das zentrale Entscheidungsmodul weiter. Dieses trifft funktionsabhängig die endgültige Entscheidung über die Autorisierung oder Authentifizierung.

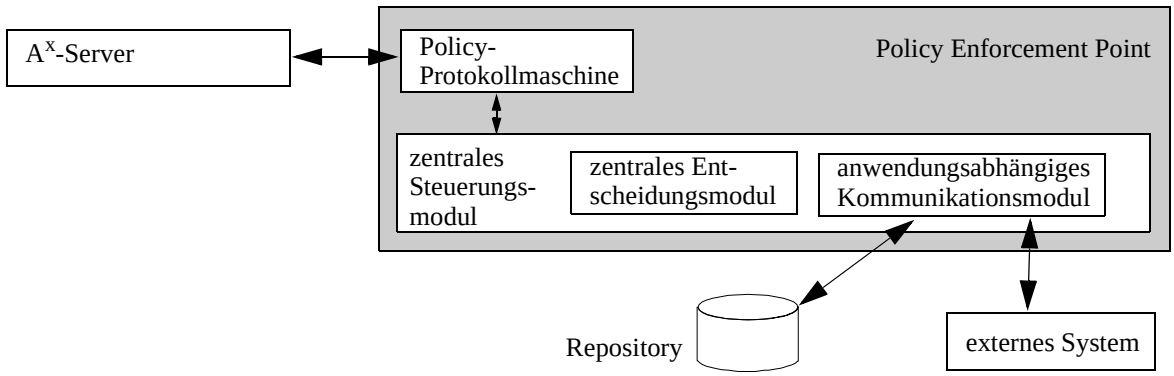

Abbildung 50: Aufbau eines Policy Enforcement Points

5.3.6 Organisationsmodelle

Ein Organisationsmodell beschreibt die Zuordnung der Komponenten der A^x-Architektur zu verschiedenen Domänen. Aufgrund der Separierung der Unterstützungsdienste von den Endnutzerdiensten in Form der A^x-Server und der Modularisierung der einzelnen Teilfunktionen der Dienste in Form der verschiedenen Policy Enforcement Points, lassen sich drei verschiedene Organisationsmodelle verwirklichen.

Integriertes Modell. Im integrierten Modell werden die Endnutzerdienste und alle Teilfunktionen der Zugriffskontrolle innerhalb einer Domäne realisiert. Das A^x-System gehört komplett der Domäne des Dienstanbieters an. Abbildung 51 zeigt das integrierte Modell in der Darstellungsform des Internet-Dienstmodells (vgl. Kapitel 2.2).

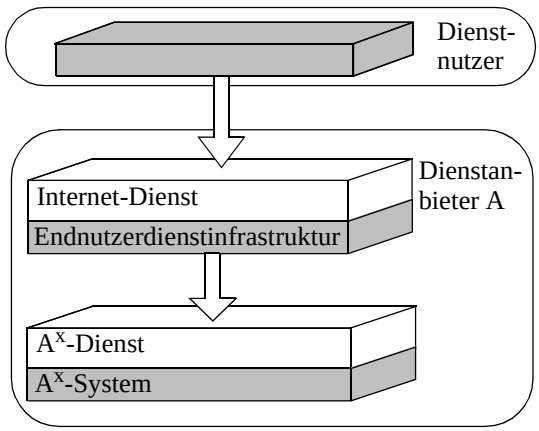

Abbildung 51: Integriertes Organsiationsmodell

Ausgelagertes Modell. Im ausgelagerten Modell (vgl. Abbildung 52) werden Teilfunktionen der Zugriffskontrolle, von einem System erbracht, welches in einer anderen administrativen Domäne lokalisiert ist, d.h. es werden vom A^X-Server PEPs eines A^X-Servers aus einer fremden administrativen Domäne verwendet . Eine Verwendung dieses Modells ist insbesondere notwendig, wenn der Dienstnutzer Dienste unabhängig von seinem Aufenthaltsort in einer fremden Domäne nutzen will und die Zugriffskontroll-Policy des Dienstanbieters eine Authentifizierung vorsieht. Die Authentifizierungs-Informationen liegen aber in der Regel nur beim Heimatdienstanbieter des Dienstnutzers vor, so dass dieser die Authentifizierung vornehmen muss. Das Szenario der ausgelagerten Authentifizierung durch den Heimatdienstanbieter des Dienstnutzers ist in Abbildung 53 gezeigt: Ein Dienstnutzer, dessen Heimatdienstanbieter Anbieter B ist, fragt einen Dienst bei Anbieter A an. Dieser stellt eine Zugriffskontrolldienstanfrage an ein A^X-Server innerhalb seiner Domäne. Als Bestandteil der Anfrage wird der Heimatdienstanbieter des Dienstnutzers angegeben. Innerhalb der Authentifizierungs-Policy hat der Anbieter A spezifiziert, wer die Authentifizierung für Nutzer aus der Heimatdomäne B ausführt. Dies ist Anbieter B, da dieser über die Authentifizierungs-Informationen des Dienstnutzers B verfügt. Dementsprechend erfolgt die Authentifizierung durch den Heimatdienstanbieter B, wozu dessen Authentifizierungs-PEP dirket vom PDP des A^X-Servers von A konfiguriert wird.

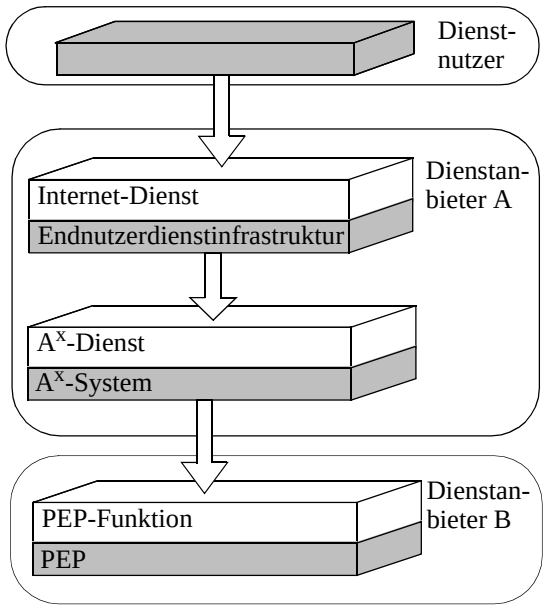

Abbildung 52: Ausgelagertes Organsiationsmodell

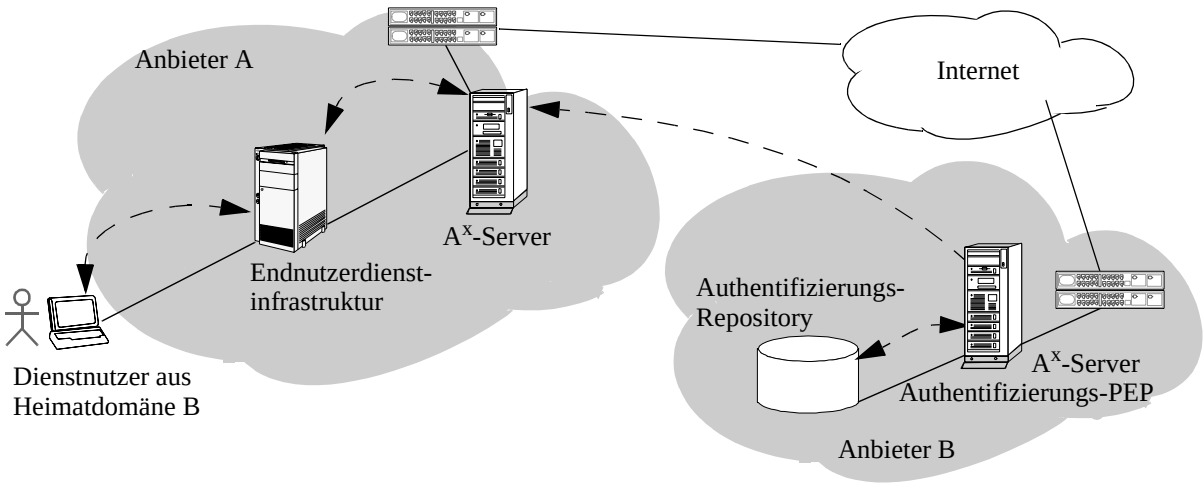

Abbildung 53: Authentifizierung des Dienstnutzers einer fremden Anbieterdomäne

Broker-Modell. Das Broker-Modell (vgl. Abbildung 54) unterscheidet sich vom ausgelagerten Modell darin, dass nicht nur eine Teilfunktion der Zugriffskontrolle in eine andere Domäne ausgelagert wird, sondern die gesamte Zugriffskontrolle. Der A^X-Server wird dann von einem vertrauenswürdigen Dritten, einem sogenannten Broker, betrieben. In diesem Fall stellt der Dienstanbieter die Zugriffskontrolldienstanfrage direkt an einen A^X-Server in einer fremden Domäne, der des Brokers. Dazu ist es notwendig, dass die Zugriffskontroll-Policies des Dienstanbieters und gegebenenfalls die Authentifizierungs-Informationen in dessen Repositories gespeichert sind. Die Verwednung des Broker-Modells ist in Abbildung 55 dargestellt.

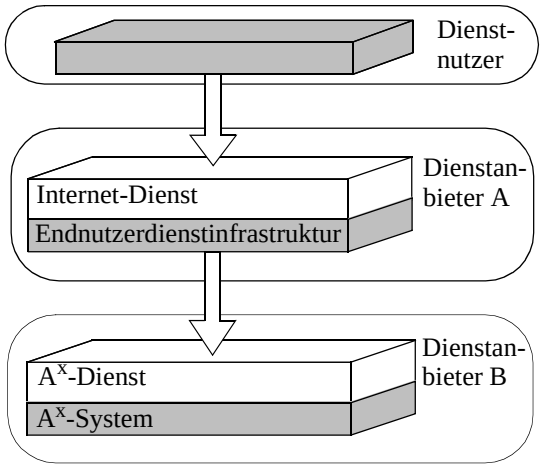

Abbildung 54: Broker Organsiationsmodell

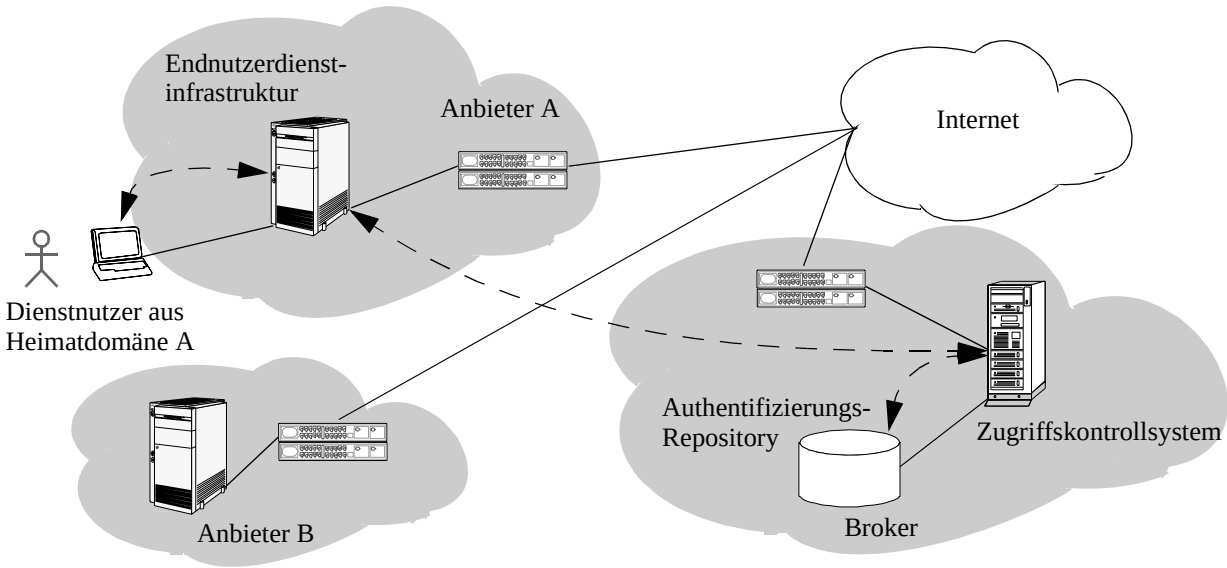

Abbildung 55: Nutzung eines Zugriffskontroll-Brokers

5.3.7 Lokalisierung der logischen Komponenten

Zur Realisierung eines A^X-Systems müssen die verschiedenen logischen Komponenten auf Rechnersystemen realisiert werden. Die Entscheidung über die Lokalisierung ist abhängig vom organisatorischen Modell, den genutzten externen Systemen und der technischen Realisierung der Authentifizierungs- und Autorisierungs-Funktionen. Folgende Rechnersysteme existieren als potentielle Kandidaten zur Lokalisierung der logischen Komponenten:

- Endnutzerdienstinfrastruktur
- A^X-System in eigener Anbieterdomäne

- A^X-System in fremder Anbieterdomäne
- A^X-System in Broker-Domäne
- Repository-Server
- externe Systeme

Bei der Entscheidung über die Lokalisierung ist zu berücksichtigen, dass sie einen Einfluss auf die Antwortzeit auf eine Zugriffskontrollanfrage und damit die Performanz des Zugriffskontrollsystems besitzt. Auch die Zuverlässigkeit des A^X-Systems wird durch die Lokalisierung beeinflusst.

A^X-Server. Zentrales Element der A^X-Architektur ist der A^X-Server. Je administrativer Domäne sollten mehrere A^X-Server eingerichtet werden, um die Ausfallsicherheit des Gesamtsystems zu erhöhen und die Last verteilen zu können. Welcher A^X-Server vom A^X-Client angefragt wird, kann bei diesem als Parameter hinterlegt werden. Kann der primäre A^X-Server nicht erreicht werden, stellt der A^X-Client nach Ablauf einer Zeitspanne seine Zugriffskontrolldienstanfrage erneut an einen sekundären A^X-Server. Bei Verwendung des Broker-Modells als organisatorischem Modell ist der A^X-Server dementsprechend auf einem Zugriffkontrollsystem in der Domäne des Brokers angeordnet.

Policy-Repository. Das Policy-Repository sollte möglichst in unmittelbarer Nähe des A^X-Servers, also im identischen lokalen Netz oder auf dem identischen Rechnersystem, angeordnet sein, um die Antwortzeit auf Repository-Anfragen möglichst gering zu halten. In Abhängigkeit von der technischen Realisierung kann es auf einem speziellen Repository-Server implementiert sein. Wenn mehrere A^X-Server innerhalb einer Anbieterdomäne existieren, bedeutet das zugleich, dass die Zugriffskontroll-Policy in mehreren Repositories repliziert gespeichert werden müssen. Da diese nur in größeren Zeitabständen geändert werden, z.B. bei einer Modifikation der Geschäftsmodelle oder Einrichtung neuer Dienste, kann die Konsistenz der Policies mit bestehenden Mechanismen gewährleistet werden.

Session-Repository. Für das Session-Repository gilt das für Policy-Repositories gesagte. Existieren mehrere A^X-Server und Session-Repositories ist zu beachten, dass ein A^X-Client alle zu einer Session gehörigen Anfragen an den identischen A^X-Server stellt. Ist dieser temporär nicht erreichbar, muss für einen neuen Dienst eine neue Session errichtet werden.

Zugriffskontroll-PEPs. Für die PEPs gilt, dass diese grundsätzlich auf beliebigen A^X-Systemen lokalisiert sein können. Ob ein PEP auf dem Zugriffskontrollsystem in der eigenen Domäne, in der Domäne des Heimatdienstanbieters des Dienstnutzers oder in der eines Brokers genutzt wird, spezifiziert die Zugriffskontroll-Policy über das Policy-Element “Ort der Durchführung”. Im integrierten Modell sollten alle PEPs auf dem lokalen A^X-System lokalisiert sein. Das gilt auch für die Systemzustands- und Berechtigungsnachweis-PEPs, die externe Systeme verwenden. Der lokal angeordnete PEP stellt dann eine Anfrage an das externe System.

Die einzige Ausnahme ist ein Systemzustands-PEP, welcher den Systemzustand der Endnutzerdienstinfrastruktur abfragt, also z.B. die verfügbaren Ressourcen. Dieser PEP muss auf der End-

nutzerdienstinfrastruktur angeordnet sein. Im ausgelagerten Modell sind die PEPs der ausgelagerten Funktionen auf dem A^X-System in einer fremden Anbieterdomäne plaziert.

Rechte- und Authentifizierungs-Repository.

Das Authentifizierungs- und Rechte-Repository sollten dort angeordnet sein, wo die Authentifizierung bzw. Prüfung der Benutzerrechte vorgenommen wird, also dort, wo sich die entsprechenden PEPs befinden. Abhängig von der technischen Realisierung können sie ebenfalls auf spezialisierten Repository-Servern eingerichtet werden.

Eine beispielhafte Lokalisierung der logischen Komponenten für das ausgelagerte Modell ist in Abbildung 56 gezeigt.

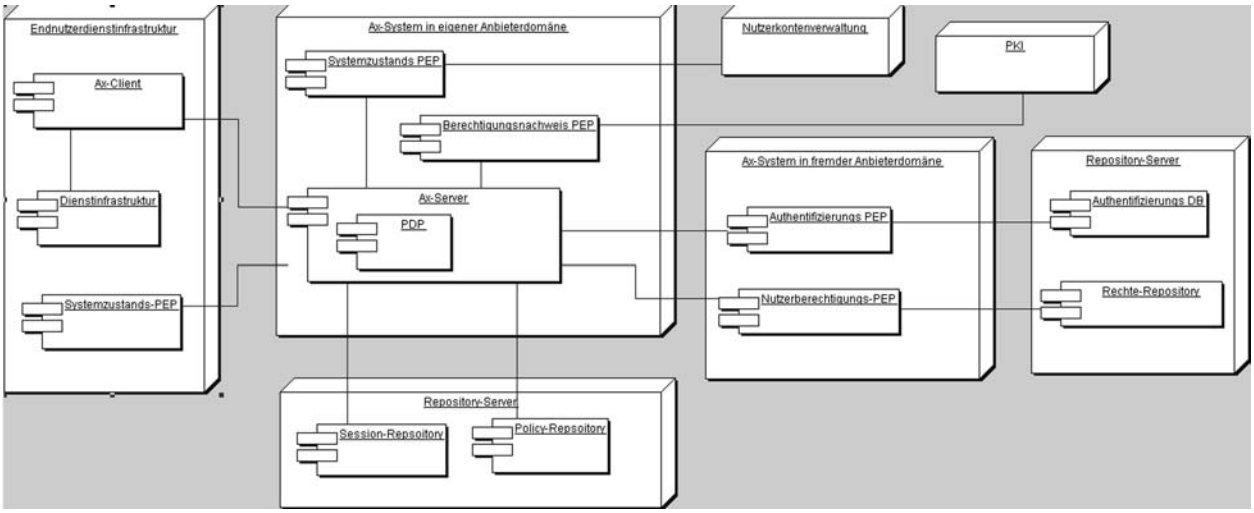

Abbildung 56: Beispiel für eine Lokalisierung der logischen Komponenten

5.3.8 Sicherheit der A^X-Architektur

Die an der Realisierung der A^X-Architektur beteiligten Systeme und die Kommunikationsverbindungen zwischen diesen Systemen sind potentiellen aktiven und passiven Angriffen ausgesetzt. Ein Angreifer will sich Vorteile, wie zum Beispiel den Zugriff auf private Dienste, für die er keine Berechtigung besitzt, oder den kostenfreien Zugriff auf kostenpflichtige Dienste, verschaffen. Oder er will einzelne System in ihrer Funktionserbringung, z.B. durch einen Denial of Service Angriff, behindern. Um die verschiedenen Angriffe zu unterbinden, sind zum einen die einzelnen Systeme und zum anderen die Kommunikationsverbindungen zwischen den Systemen abzusichern. Die Absicherung der Kommunikationsverbindungen soll in diesem Abschnitt betrachtet werden.

Die zwischen den Systemen existierenden Kommunikationsverbindungen müssen gegen mögliche Angriffe gesichert werden. Dies kann mittels einer Nutzung von Basismechanismen der Kommunikationssicherheit zur Gewährleistung der Sicherheitseigenschaften der Vertraulichkeit, Integrität und der Authentifizierung der Datenherkunft sowie von Verfahren zum Schutz vor Replay-Angriffen erfolgen. Dazu können entweder existierende Standardprotokolle auf den Kommunikationsebene, wie besipielsweise das IP Security (IPsec) Protokoll [KA98], verwendet wer-

den, oder es werden innerhalb der AX-Architektur eingene Mechanismen realisiert. Im zweiten Fall müssen die auszutauschenden Nachrichten digital signiert und auch verschlüsselt werden, um die Vertraulichkeit zu gewährleisten. Die dazu notwendigen Datenobjekte sind innerhalb der AX-Architektur vorgesehen.

Voraussetzung für den Einsatz von Protokollen wie IPSec ist das Bestehen von Vertrauensverhältnissen zwischen den einzelnen Systemen der Architektur. Diese werden in einem Vertrauensmodell beschrieben, welches in Abbildung 57 gezeigt ist.

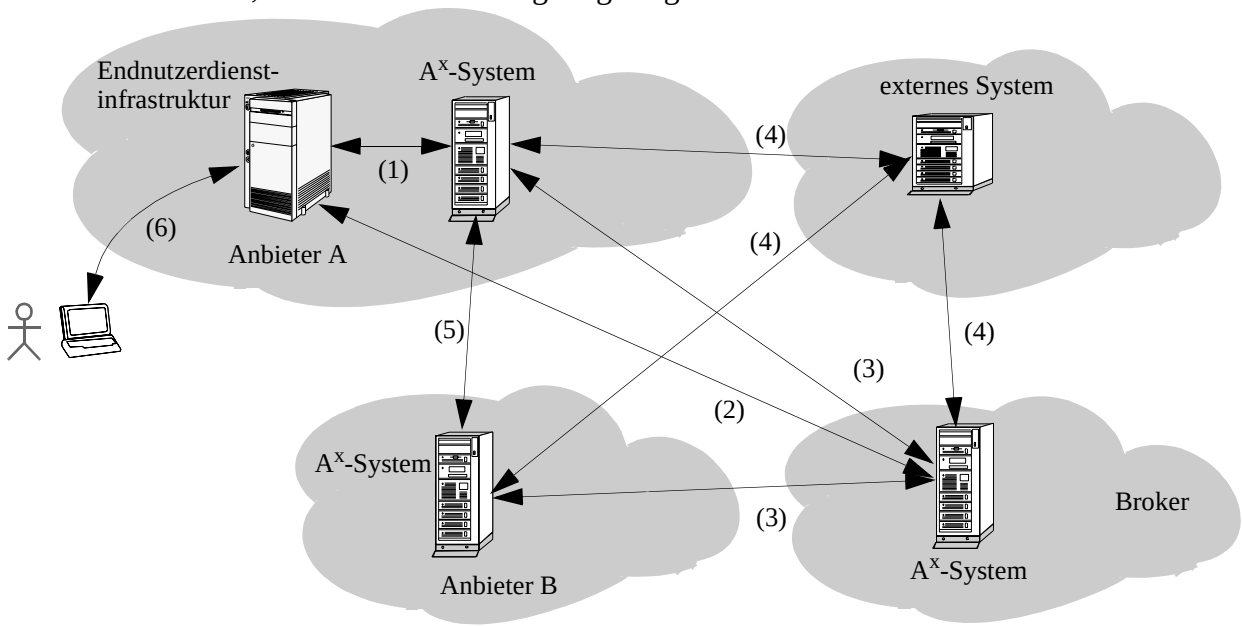

Abbildung 57: Vertrauensmodell zwischen AX-Systemen

Zwischen der Endnutzerdienstinfrastruktur und dem lokalen AX-System kann ein statisches Vertrauensverhältnis (1) eingerichtet werden, da beide der selben administrativen Domäne angehören. Bei Nutzung eines Brokers muss zwischen diesem und den jeweiligen lokalen AX-Systemen ein Vertrauensverhältnis (3) statisch etabliert werden. Bei Verwendung des Broker-Modells muss zwischen der Endnutzerdienstinfrastruktur und dem AX-System des Brokers ebenfalls eine Vertrauensverhältnis (2) bestehen. Dieses kann entweder statisch existieren oder unter Nutzung der statischen Vertrauensverhältnisse (1) und (3) dynamisch eingerichtet werden. Für die Nutzung des ausgelagerten Modells muss zwischen den AX-Systemen in zwei unterschiedlichen Domänen ebenfalls ein Vertrauensverhältnis (5) eingerichtet sein. Dies kann, sofern ein Broker vorhanden ist, dynamisch unter Nutzung der statischen Vertrauensverhältnisse (3) erfolgen. Zuletzt müssen zwischen AX-Systemen und externen Systemen ebenfalls Vertrauensverhältnisse (4) bestehen. Wie diese errichtet werden ist abhängig von der jeweiligen Anwendung und dem verwendeten Protokoll. Zur Absicherung der Kommunikation zwischen dem Dienstnutzer und der Endnutzerdienstinfrastruktur sind ebenfalls geeignete Verfahren einzusetzen. Die Realisierung ist abhängig von der Anwendung, der Art des Endgeräts, der Art der genutzten Verbindung und davon, ob der Dienstnutzer sich in seiner Heimatdomäne befindet oder nicht.

Über diese Sicherheitseigenschaften hinaus ist für die Dienstnutzung durch den Dienstnutzer insbesondere für kostenpflichtige Dienste gegebenenfalls ein Verbindlichkeitsnachweis einzuführen, damit der Dienstnutzer seine Dienstnutzung nicht abstreiten kann. Ein solcher Verbindlichkeitsnachweis kann mittels einer digitalen Signatur hergestellt werden. Ob nur die Dienstanfrage oder jedes einzelne Paket digital zu signieren ist, ist abhängig von der Art der Anwendung und der kaufmännischen Policy.

5.4 Beschreibung des A^x-Systems

Das A^x-System wird im Wesentlichen bestimmt über die Funktionalität der einzelnen Komponenten und die zwischen den Komponenten verwendeten Schnittstellen. Die Funktionen der Komponenten wurde zuvor in Tabelle 9 zusammengefasst. Die einzelnen Schnittstellen sind in Tabelle 10 mit ihren Funktionen und wichtigsten ausgetauschten Daten aufgeführt.

Schnittstelle	Funktionalität	wichtigste ausgetauschte Daten
A^x-Client - A^x-Server	Zugriffskontrolldienstanfragen austauschen	-> Dienstanbieter, Dienstnutzer, Dienstbeschreibung, Auth.-Informationen, Session-ID, Berechtigungsnachweis <- Ergebnis
A^x-Server - Policy-Repsoitory	Abfrage einer Zugriffskontroll-Policy	-> Dienstbeschreibung, Heimatdienstanbieter <- Zugriffskontroll-Policy
A^x-Server - Session-Repository	Verwaltung der Sessions	<-> Session-Informationen
A^x-Server - PEP	Konfiguration des PEPs und Ergebnismitteilung	-> Ausschnitt aus Policy, Datenobjekte aus Zugriffskontrolldienstanfrage <- Ergebnis
Authentifiz. - PEP - Authentifiz.-Reposit.	Abfrage der Authentifizierungs-Informationen	-> Nutzer-ID <- Auth.-Informationen
Nutzerberecht.-PEP - Rechte-Respository	Abfrage der Nutzerberechtigungen	-> Nutzer-ID <- Nutzerberechtigungen
Systemzustands-PEP - Dienstinfrastruktur	Abfrage des Systemzustands	-> Beschreibung des Zustandparameters <- Systemzustand
Systemzustands-PEP - externe Systeme	Abfrage des Systemzustands	-> Beschreibung des Zustandparameters <- Systemzustand

Tabelle 10: Schnittstellen zwischen den Komponenten der A^x-Architektur

Zusätzlich zu diesen Schnittstellen existieren noch solche zum Management der Informationen und Policies in den verschiedenen Repositories, wie sie im Anwendungsfall "A^x-System konfigurieren" in Abbildung 44 dargestellt wurden. Dieses Daten-Management erfolgt mittels eines geeigneten Werkzeugs, welches hier nicht näher betrachtet werden soll.

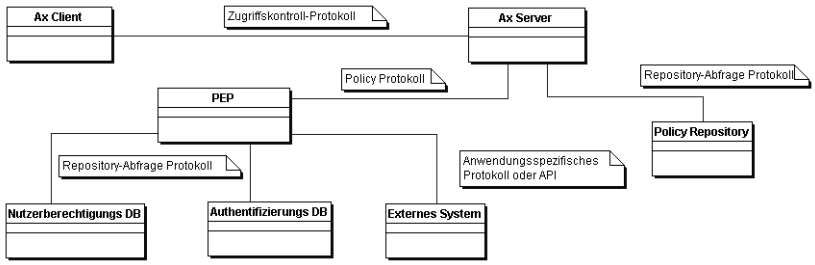

Abbildung 58: Protokollinteraktionen innerhalb der AX-Architektur

Abbildung 58 zeigt die zur Realisierung der Architektur notwendigen Kommunikationsverbindungen zwischen den Komponenten für welche jeweils Protokolle, ihre Nachrichtentypen und Datenobjekte zu spezifizieren sind.

Für die Kommunikation zwischen AX-Client und AX-Server wird in existierenden Architekturen typischerweise ein sogenanntes AAA-Protokoll, wie RADIUS oder Diameter (vgl. Kapitel 3.4.1) verwendet. Innerhalb der AX-Architektur wird es als AX-Protokoll bezeichnet. Die Kommunikation zwischen dem AX-Server mit seinem Policy Decision Point und den verschiedenen Policy Enforcement Points wird über das Policy-Protokoll abgewickelt. Für die Abfrage der Policies, der Nutzerberechtigungen und der Authentifizierungs-Daten aus den jeweiligen Repositories sind sogenannte Abfrageprotokolle zu verwenden. Je nach technischer Realisierung der Repositories können hier Protokolle zur Verzeichnisabfrage, wie LDAP [HM02], oder spezielle Protokolle zur Datenbankabfrage genutzt werden. Die Abfrageprotokolle werden im Gegensatz zum AX-Protokoll und Policy-Protokoll nicht weiter betrachtet werden.

5.4.1 AX-Protokoll

Das AX-Protokoll dient dazu, zwischen AX-Client und AX-Server eine Dienstanfrage zu stellen, die für die Durchführung der Zugriffskontrolle notwendigen Informationen auszutauschen und das Ergebnis auf die Dienstanfrage zurückzumelden.

Datenobjekte. Die folgenden Datenobjekte müssen mittels des AX-Protokolls ausgetauscht werden können:

- **Anfrage-ID:** Eine Anfrage-ID wird dazu verwendet, einzelne Protokollnachrichten einander zuzuordnen. Sie setzt sich zusammen aus einer fortlaufenden Nummer sowie den IP-Adressen als Identitätsmerkmalen des AX-Client und des genutzten AX-Servers. Sie wird vom Client bei der Anfrage initial gesetzt und bei folgenden Antwortnachrichten übernommen.
- **Client-Domäne:** Der Domänenname des Dienstanbieters wird benötigt, um die korrekte Zugriffskontroll-Policy zu ermitteln, wenn die Anfrage an einen Broker gestellt wird, der die Policies mehrerer Dienstanbieter verwaltet.
- **Dienstnutzer-ID:** Mittels der Dienstnutzer-ID wird der Dienstnutzer, der die Dienstanfrage gestellt hat, die die Nutzung des Zugriffskontrolldienstes auslöst, identifiziert. Dazu werden, wie in Kapitel 5.2.4 definiert, SIP-URIs verwendet. Mittels dieses SIP-URIs wird

zugleich der Heimatdienstanbieter des Nutzers identifiziert. Die Information über den Heimatdienstanbieter ist notwendig, um die gültige Zugriffskontroll-Policy zu selektieren und gegebenenfalls den Ort der Funktionsdurchführung für einzelne PEPs zu bestimmen (vgl. Kapitel 4.4.1). Wird keine Dienstnutzer-ID angegeben, ist diese dem Client nicht bekannt und es können nur anonym bereitgestellte Dienste autorisiert werden. Falls die Policy eine Identifizierung oder Authentifizierung verlangt, muss der Server in nachfolgenden Nachrichten die Informationen zur Identifizierung und Authentifizierung des Dienstnutzers anfordern.

- **Dienstbeschreibung:** Dieses Objekt enthält Informationen zur Spezifikation des angefragten Dienstes, entsprechend der Definition in Kapitel 4.3.1. Die Dienstbeschreibung dient ebenfalls als Selektionskriterium für die Zugriffskontroll-Policy.
- **Authentifizierungs-Informationen:** Die Authentifizierungs-Informationen sind die zur Authentifizierung des Nutzers notwendigen Daten, Berechtigungsnachweise, sofern solche verwendet werden, sowie die Information um welchen Typ von Authentifizierungs-Daten es sich handelt. Die Authentifizierungs-Informationen können bereits während der ersten Anfrage übermittelt werden, sofern sie dem Client vorliegen. Alternativ werden sie in nachfolgenden Nachrichten gefüllt, z.B. im Rahmen von Challenge Response Verfahren oder wenn eine Authentifizierung explizit vom Server angefragt wird. In diesem Objekt können auch Nachrichten des Authentifizierungs-Protokolls getunnelt werden.
- **Authentifizierungs-Challenge:** Liegen dem Client nicht alle Informationen zur Identifizierung und Authentifizierung des Dienstnutzers vor, so kann er im diesem Objekt spezifizieren, welche Informationen er über den Nutzer benötigt. Es kann eine Authentifizierungs-Challenge angegeben und Nachrichten des Authentifizierungs-Protokolls in diesem Objekt getunnelt werden. Der Client wertet die Daten aus und stellt entsprechende Anfragen an den Dienstnutzer oder leitet sie als Proxy nur an den Nutzer weiter.
- **Session-Informationen:** Das Objekt Session-Informationen enthält eine Session-ID und weitere Status-Informationen. Mittels der Session-ID werden verschiedene Dienste einer Session zugeordnet, wie in Kapitel 5.2.5 beschrieben. Die Status-Information wird dazu verwendet das Ende einer Session anzuzeigen oder Sessions aneinander zu binden.
- **Ergebnis:** Das Ergebnis Objekt dient dazu, das Ergebnis der Zugriffskontrolldienstanfrage vom Server an den Client zu übertragen. Es muss sich nicht auf eine Ja- oder Nein-Antwort beschränken, sondern kann auch Informationen über den Grund der Dienstablehnung enthalten. Dies ermöglicht eine Aushandlung von Verfahren zur Zugriffskontrolle, wenn die Zugriffskontroll-Policy für einen Dienst mehrere alternative Zugriffskontroll-Policies spezifiziert.
- **Fehler:** Das Fehler-Objekt kann in beliebigen Nachrichten genutzt werden, um Fehler zu spezifizieren. Dies ist beispielsweise der Fall, wenn für den angefragten Dienst keine Policy im Policy-Repository gespeichert ist.

- **Signatur:** Kann keine sichere Kommunikation zwischen Zugriffskontroll-Client und Zugriffskontrol-Server vorausgesetzt werden, wird dieses Objekt für die digitale Signatur der gesamten Nachricht verwendet.

Nachrichtentypen.

Das A^X-Protokoll verwendet folgende Nachrichtentypen:

- **Ax-Dienstanfrage:** Mittels der Ax-Dienstanfrage fordert der Client einen Zugriffskontrolldienst beim Server an. Dabei sind die folgenden Datenobjekte zu verwenden: Anfrage-ID, Client-Domäne, Dienstnutzer-ID (optional), Dienstbeschreibung, Authentifizierungs-Informationen (optional), Session-ID.
- **Ax-Dienstantwort:** Nach kompletter Durchführung der Zugriffskontrolle sendet der A^X-Server eine Zugriffskontrollantwort mit folgenden Objekten: Anfrage-ID, Session-ID, Ergebnis, Fehler (optional)
- **Authentifizierungs-Anfrage:** Sind in der A^X-Dienstanfrage keine Dienstnutzer-ID und Authentifizierungs-Informationen übermittelt worden und verlangt die für den angefragten Dienst gültige Zugriffskontroll-Policy die Bestimmung des Heimatdienstanbieters des Nutzers, dessen Identifizierung bzw. Authentifizierung, so stellt der Server eine entsprechende Anfrage an den Client. Diese verwendet minmal die Objekte Anfrage-ID und Authentifizierungs-Challenge.
- **Authentifizierungs-Antwort:** Die Authentifizierungs-Antwort sendet der Client als Reaktion auf eine Authentifizierungs-Anfrage. Die vom Server abgefragten Informationen finden sich in den Objekten Dienstnutzer-ID und Authentifizierungs-Informationen.
- **Session-Status:** Die Nachricht Session-Status dient dazu, das Ende einer Session anzuzeigen oder zwei Sessions aneinander zu binden, wie in Kapitel 5.2.5 beschrieben. Sie wird vom Client an den Server geschickt.

Zur Realisierung dieser Nachrichtentypen und Datenobjekte ist ein geeignetes Zugriffskontrollprotokoll zu definieren oder ein bestehendes auszuwählen. Das Diameter Basisprotokoll, wie es in Kapitel 3.4.1 vorgestellt wurde, beschreibt die notwendigen Basismechanismen für das Zugriffskontrollprotokoll und bietet mit der Möglichkeit in Ergänzungen eigene Attribut Wert Paare zu definieren. Es ist damit ein möglicher Kandidat zur Verwendung innerhalb der A^X-Architektur.

5.4.2 Policy-Protokoll

Das Policy-Protokoll hat die wesentlichen Aufgaben, die Policy-Abschnitte zur Konfiguration der Policy Enforcement Points an diese zu senden und damit die PDPs zu konfigurieren sowie das Ergebnis der Policy-Durchsetzung an den PDP und damit den A^X-Server zurück zu melden.

Datenobjekte. Die folgenden Datenobjekte müssen mittels des Policy-Protokolls ausgetauscht werden:

- **Anfrage-ID:** Die Anfrage-ID wird verwendet, um einzelne Protokollnachrichten einander zuzuordnen. Sie setzt sich zusammen aus einer fortlaufenden Nummer sowie den IP-Adres-

sen des A^x-Servers und des Systems auf welchem der PDP lokalisiert ist. Sie wird vom PDP initial gesetzt und bei jeweiligen Antworten übernommen.

- **Policy-Informationen:** Das Objekt Policy-Informationen enthält den Ausschnitt der Zugriffskontroll Policy, der zur Konfiguration des jeweiligen PEPs notwendig ist. Für einen Authentifizierungs PEP ist dies die Authentifizierungs Policy, für einen Autorisierungs-PEP die Autorisierungs-Policy, wie in Kapitel 4.4.1 vorgestellt.
- **Policy-Daten:** Neben den Policy-Informationen müssen auch die zur Durchsetzung der Policy vom Policy Enforcement Point benötigten Daten, angegeben werden. Dabei handelt es sich um eine Teilmenge der im A^x-Protokoll verwendeten Objekte, nämlich:
  - Dienstnutzer-ID für Authentifzierungs-, Nutzerberechtigungs-PEP und Systemzustands-PEP
  - Authentifizierungs-Informationen für Authentifizierungs- und Berechtigungsnachweis-PEP
  - Authentifizierungs-Challenge für Authentifizierungs-PEP
  - Dienstbeschreibung für Nutzerberechtigungs- und Systemzustands-PEP
- **Ergebnis:** Im Objekt Ergebnis wird die Antwort des Policy Enforcement Points übermittelt. Darin wird angegeben, ob die Authentifizierung bzw. Autorisierung erfolgreich war oder nicht und warum sie nicht erfolgreich oder nicht möglich war.
- **Fehler:** Das Fehler-Objekt kann in beliebigen Nachrichten genutzt werden, um Fehler zu spezifizieren.
- **Signatur:** Kann keine sichere Kommunikation zwischen PDP und dem PEP, insbesondere bei entfernten PEPs vorausgesetzt werden, wird dieses Objekt für die digitale Signatur der gesamten Nachricht verwendet.

Nachrichtentypen.

Das Policy-Protokoll benötigt die folgenden Nachrichtentypen:

- **PDP-Dienstanfrage:** Die PDP-Dienstanfrage wird vom PDP verwendet, um den Ausschnitt der Zugriffskontroll-Policy und die Policy-Daten zu übermitteln. Alle Datenobjekte außer dem Ergebnis werden verwendet.
- **PEP-Dienstantwort:** Mit Hilfe der PDP-Dienstantwort meldet der PEP das Ergebnis der Authentifizierung bzw. Autorisierung zurück.
- **PEP-Authentifizierungs-Anfrage:** Über diese Nachricht kann der Authentifizierungs-PEP Authentifizierungs-Informationen vom PDP abfragen oder auch ein Challenge an diesen senden. Dazu wird das Objekt Policy-Daten genutzt.
- **PDP-Authentifizierungs-Antwort:** Zur Übermittlung der Antwort auf eine Nachricht PEP-Authentifizierungs-Challenge wird diese Nachricht genutzt.

Als Policy-Protokoll wird oftmals COPS [DBC+00] eingesetzt (vgl. Kapitel 3.4.1). Bei der Verwendung von COPS wird die Kommunikation vom Policy Enforcement Point initiiert. Dieser sendet eine Anfrage an den Policy Decision Point und erhält als Antwort von diesem eine Policy-Entscheidung und Konfigurationsinformationen. Die A^x-Architektur weicht davon ab. Der A^x-

Server mit seinem PDP wird auf eine Anfrage des Dienstanbieters selbst hin aktiv und schickt die Policy-Informationen an den Policy Enforcement Point, der wiederum nach erfolgter Policy-Durchsetzung eine Rückmeldung an den A^X-Server sendet. Diese Strukturen widersprechen den in COPS verwendeten Konzepten der Session und Handles. Daher ist zu prüfen, ob es für eine Verwendung in der generischen A^X-Architektur modifiziert werden kann oder ein eigenes Protokoll auf Basis der oben angegebene Objekte und Nachrichtentypen zu spezifizieren ist.

5.4.3 Protokollverwendung

Die Verwendung der Protokolle wird an einem Beispiel Im Sequenzdiagramm in Abbildung 59 illustriert. Dazu werden die Nachrichtentypen und die wichtigsten jeweils ausgetauschten Datenobjekte und die Kriterien zur Selektion der Policy angegeben. Die Zugriffskontroll-Policy sieht in diesem Beispiel eine Authentifizierung des Dienstnutzers mit einem Challenge-Response-Verfah-

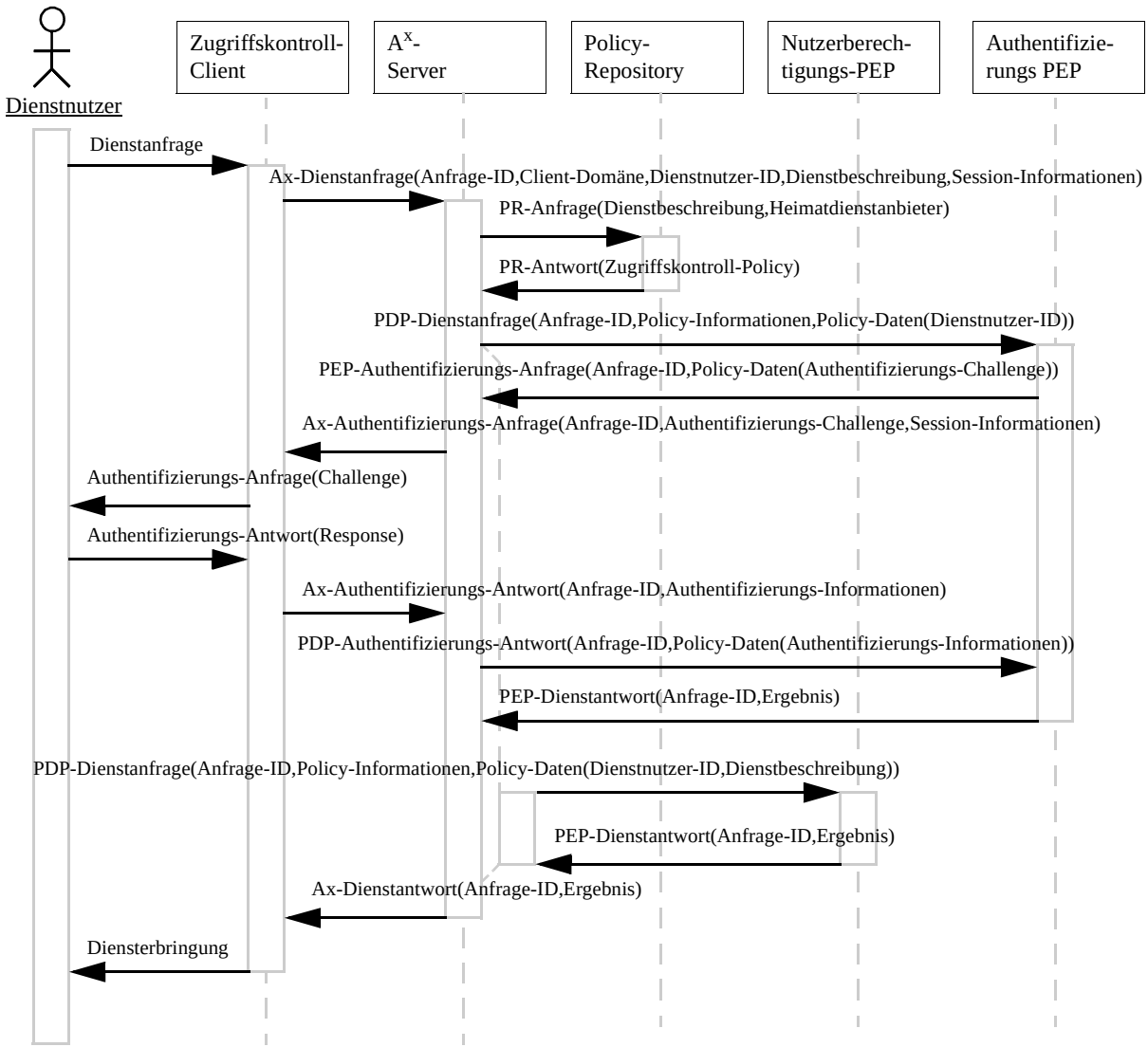

Abbildung 59: Beispiel für den Protokollablauf innerhalb der A^X-Architektur

ren und die Autorisierung mittels der Überprüfung der Nutzerberechtigungen vor. Der Ablauf ist grundsätzlich bereits in Kapitel 5.3.3 erläutert.

5.5 Erweiterung der A^x-Architektur um kaufmännische Unterstützungsdienste

Ein ökonomisch handelnder Anbieter von Internet-Diensten muss nicht nur den Zugriff auf die Dienste kontrollieren, sondern auch kaufmännische Funktionen erbringen, wie sie in Kapitel 2.5 definiert wurden. Die einzelnen Funktionen der kaufmännischen Unterstützungsdienste können nicht unabhängig von den Zugriffskontrollfunktionen realisiert werden. Es gibt eine Vielzahl von Abhängigkeiten und Informationsflüssen zwischen den einzelnen Teilfunktionen. Die im Policy-Modell in Kapitel 4.5 dargestellten Zusammenhänge zwischen den Zugriffskontoll-Policies und den Policies für die kaufmännischen Unterstützungsdienste illustrieren dies deutlich.

Daher ist eine integrierte Sichtweise auf die Zugriffskontrolle und die kaufmännischen Unterstützungsdienste sinnvoll, um einen Internet-Dienstanbieter in der Erbringung der Dienste optimal zu unterstützen. Die vorgestellte A^x-Architektur ist dazu um solche Module zu erweitern, welche die kaufmännischen Funktionen realisieren. Sie bietet aufgrund der verwendeten Konzepte, insbesondere der Modularisierung der Teilfunktionen und des policybasierten Managements, die Voraussetzungen dazu.

5.5.1 Erweiterte Funktionalität des A^x-Systems und zusätzliche Komponenten

Die Erweiterungen der A^x-Architektur muss um solche Module erfolgen, welche die Aufgaben des Meterings, des Accounting, der Charge-Calculation, gegebenenfalls des Billings und Payments sowie des Auditings übernehmen. Ein Metering ist nur bei eine Nutzung mengenbezogener Preiskomponenten notwendig (vgl. dazu die Erläuterung der Metering-Policy in Kapitel 4.4.2). Komponenten zum Billing und Payment sind dann sinnvolle Erweiterungen eines A^x-Systems, wenn elektronische Bezahlverfahren, wie beispielsweise geldartige Micropayments, genutzt werden, die zeitlich in unmittelbaren Zusammenhang mit der Kontrolle des Zugriffs und der Diensterbringung stehen. In den Fällen einer periodischen Abrechnung sind die Rechnungsstellung und Zahlung unabhängig von den übrigen Funktionen und müssen nur auf vom A^x-System erzeugte Daten zugreifen, wie dies in Kapitel 4.5 erörtert wurde. Dann handelt es sich nicht um Funktionen, die ein A^x-System erbringen muss.

Erweiterte Gesamtfunktionalität der A^x-Systems. Die Gesamtfunktionalität des erweiterten A^x-Systems besteht nicht mehr nur in der Erbringung der Zugriffskontrolldienste, sondern auch der kaufmännischen Unterstützungsdienste. Diese sind, wie in Kapitel 5.2.1 beschrieben, der Zugriffskontrolle zeitlich nachgelagert und müssen parallel zur Diensterbringung oder nach der Diensterbringung erfolgen. Abbildung 60 zeigt die Nutzung der A^x-Dienste durch den Dienstanbieter in einem vereinfachten Überblick, der nur die Kontrolle von Diensten betrachtet, die mit einer Session einhergehen, d.h. die einen zeitlichen Verlauf besitzen, wie in Kapitel 2.4 erläutert. Ausgelöst durch eine A^x-Dienstanfrage werden zunächst die Zugriffskontrolldienste erbracht. Liefern diese ein positives Ergebnis, wird dies an den Dienstanbieter zurückgemeldet und gleich-

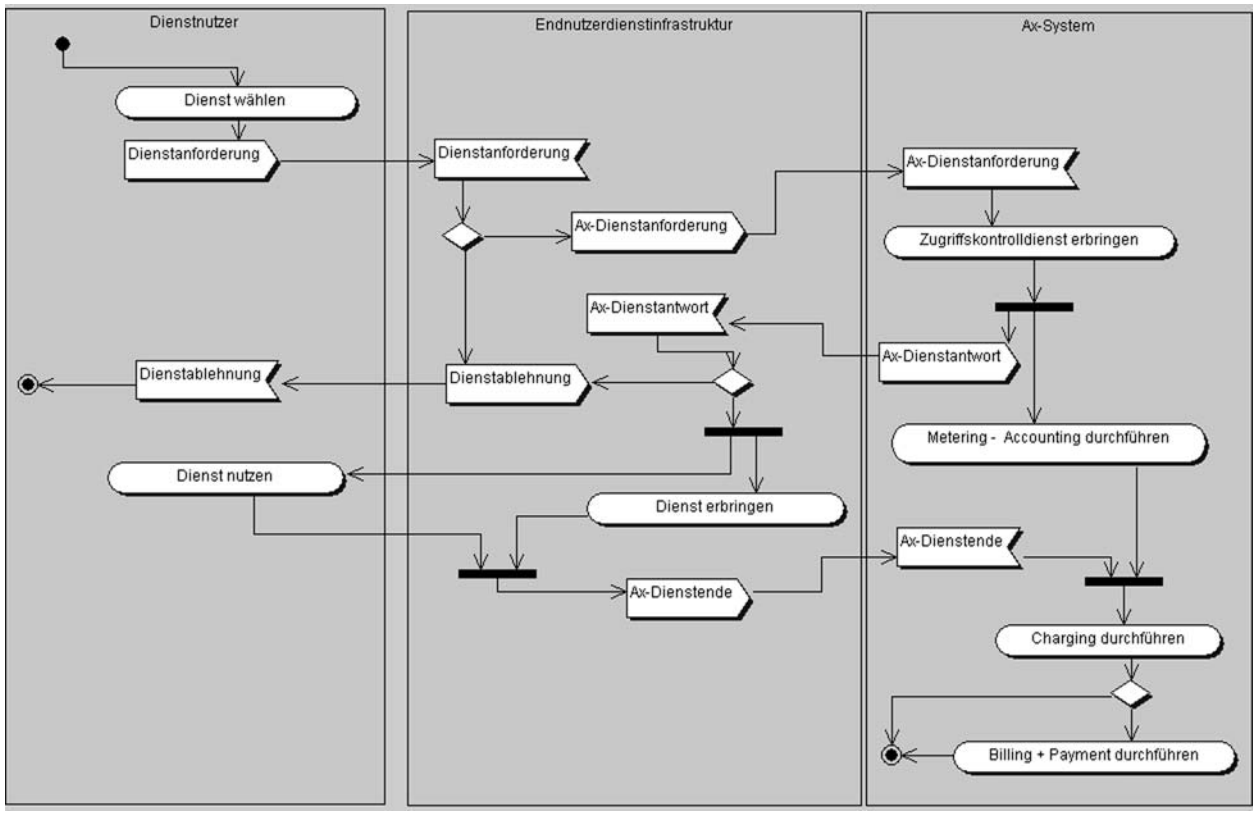

Abbildung 60: Nutzung der erweiterten AX-Dienste

zeitig wird die Funktion des Metering gestartet und mit dem Accounting begonnen. Nachdem der Endnutzerdienst komplett erbracht ist, bzw. die Session durch den Dienstnutzer oder Dienstanbieter beendet ist, wird dieses dem AX System mit einer eigenen Nachricht angezeigt. Daraufhin wird das Charging durchgeführt und gegebenenfalls der Dienst dem Dienstnutzer direkt in Rechnung gestellt und die elektronische Zahlung angestoßen. Um die AX-Dienstnachrichten einander zuordnen zu können, wird zusammenhängenden Nachrichten vom AX-Client die identische Anfrage-ID vergeben.

Die Konfiguration der kaufmännischen AX-Dienste erfolgt in gleicher Weise wie die der Zugriffskontrolldienste, unabhängig von der Nutzung der Dienste. Dazu sind durch den Dienstanbieter die Policy der kaufmännischen Unterstützungsdienste zu spezifizieren und weiter Informationen, z.B. über Tarife, in Repositories zu hinterlegen.

Zusätzliche logische Komponenten der AX-Architektur. Die einzelnen Teilfunktionen der kaufmännischen Dienste werden wiederum mittels eigener Komponenten den Policy Enforcement Points erbracht, die mittels der Policy der kaufmännischen Unterstützungsdienste konfiguriert werden. Außerdem sind weitere Repositories notwendig. Diese enthalten entweder statische Informationen, die für die Erbringung der kaufmännischen Funktionen notwendig sind, oder sie speichern die Ergebnisse der einzelnen Teilfunktionen für eine Weiterverarbeitung durch andere Teilfunktionen oder externe Systeme. Die AX-Architektur ist im Vergleich zu Kapitel 5.3.2 um die folgenden logischen Komponenten zu erweitern:

Policy-Repository. Das Policy-Repository speichert zusätzlich zu den Zugriffskontroll-Policies auch die Policies der kaufmännischen Dienste, wie sie in Kapitel 4.4.2 beschrieben wurden. Die Selektion der gültigen Policy erfolgt über die Objekte Dienstbeschreibung und Dienstnutzer. Die zwei Typen von Policies können auch in zwei getrennte Policy-Repositories gehalten werden. Der Zugriff auf die Policies der kaufmännischen Dienste muss zeitlich nach der durchgeführten Zugriffskontrolle erfolgen, wenn der Dienstanbieter Gruppen von Dienstnutzern unterscheidet. Um die Gruppe zu bestimmen, zu der ein Dienstnutzer gehört, muss dieser zunächst identifiziert und authentifiziert werden.

Metering-PEP und Metering-Repository. Der Metering-PEP misst die Ressourcennutzung, sofern die Policy dies verlangt. Die Ergebnisse des Meterings werden in Form von Metering-Records in einem Metering-Repository gespeichert.

Accounting-PEP und Accounting-Repository. Der Accounting-PEP nimmt das Accounting für einen Dienst vor. Dazu verarbeitet er Nachrichten über die Dienstnutzung und Metering-Records. Nach der erfolgreichen Zugriffskontrolle aufgrund einer A^x-Dienstanforderung wird das Accounting gestartet indem ein erster Accounting-Record geschrieben wird. Ist der Dienst beendet, wird dem Accounting-PEP das Ende der Dienstnutzung angezeigt und der Accounting-Record ergänzt. Dabei kann auch die Auswertung von Metering-Records, die der Dienstnutzung zugeordnet sind, erfolgen. Ändern sich während der Dienstnutzung Dienstparameter, beispielsweise wenn sich während einer Videokonferenz Teilnehmer an- oder abmelden, so muss der Accounting-PEP darüber informiert werden. Die Accounting-Records werden im Accounting-Repository gespeichert.

Charging-PEP und Charging-Repository. Aufgabe des Charging-PEPs ist die Charge-Calculation, d.h. die Bestimmung des Preises für eine Dienstnutzung und die Zuordnung der Dienstnutzung zu einem über ein persönliches Identitätsmerkmal zu charakterisierenden Kunden, wie dies ausführlich innerhalb von Kapitel 4.4.2 auf Seite 77 erläutert wurde. Die Charge-Calculation erfolgt, von Ausnahmen abgesehen, nach der Diensterbringung. Der Charging-PEP greift dazu auf Accounting-Records, Session-Beschreibungen, Tarif- und Kontraktdaten zu und speichert die Ergebnisse in Charging-Records. Diese werden entweder vom Billing- und Payment-PEP oder von externen Rechnungssystemen genutzt.

Billing und Payment-PEP. Billing- und Payment-PEP führen eine elektronische Rechnungsstellung und Zahlungsabwicklung unmittelbar nach dem Ende der Diensterbringung aus. Sie müssen dazu direkt mit dem Dienstnutzer, der über die Zahlungsmittel verfügt, oder einem vertrauenswürdigen Dritten kommunizieren. Die Realisierung der Billing- und Payment-PEPs ist abhängig vom verwendeten Bezahlverfahren. Die Funktion des PEPs kann auch nur darin bestehen, ein externes Rechnungs- und Zahlungssystem anzustoßen und die dazu notwendigen Informationen zu übermitteln.

Auditing-Komponenten und Protokoll-Datenbanken. Die Aufgabe der Auditing-Komponenten besteht in der Protokollierung von Ereignissen und Aktivitäten der einzelnen Komponenten

des Gesamtsystems. Ein Auditing erfolgt unter anderem zum Zwecke der Prüfung der Korrektheit der Arbeitsweise des Gesamtsystem und der vom A^X-System erzeugten Daten [RHKS01]. Es muss unabhängig von den Policies immer erfolgen. Bei den Komponenten, die ein Auditing durchführen, handelt es sich also nicht um einen Policy Enforcement Point im innerhalb der Arbeit verwendeten Sinne. Eine Auditing-Komponente muss auf jedem System, welches einen Dienst erbringt, lokalisiert sein. Insbesondere gilt dies für die Endnutzerdienstinfrastruktur. Zur eigentlichen Prüfung der Korrektheit muss dann von einem Prüfungsmodul auf die, von den Auditing-Komponenten in den Protokoll-Datenbanken protokollierten, Daten zugegriffen werden. Die Daten verschiedener Auditing-Komponenten können direkt miteinander verglichen werden oder in kummulierter Form z.B. mit Charging-Records. Auf die Funktion des Auditings wird nachfolgend nicht mehr näher eingegangen, da es sich um eine statische, d.h. nicht policyabhängige Funktion handelt.

Kontrakt und Tarif-Repository. Die Preise für eine Dienstnutzung sind in Tarifen oder, sofern sie für einzelne Kunden gelten, in Kontrakten festgehalten (vgl. die Beschreibung der Pricing-Policy auf Seite 81). Die Preisinformationen werden im Charging benötigt.

5.5.2 Organisationsmodelle

Bei einer Beschränkung des Funktionsumfangs des A^X-Systems auf die Zugriffskontrolle sind die in Kapitel 5.3.6 beschriebenen Organisationsmodelle verhältnismäßig einfach zu realisieren. Innerhalb der Zugriffskontroll verwendet das A^X-System nur solche Komponenten innerhalb einer anderen Domäne, die auf existierende Informationen, wie Identifizierungs- und Authentifizierungs-Informationen zugreifen. Für die kaufmännischen Unterstützungsdienste gilt dies nicht mehr. Im Metering, Accounting und der Charge-Calculation werden Daten erzeugt, die von anderen Teilfunktionen und externen Systemen weiterverwendet werden. Dies muss bei einer Umsetzung der Organisationsmodelle und der Definition der Policies für die kaufmännischen Unterstützungsdienste geeignet berücksichtigt werden, was an einem Beispiel erläutert werden soll.

Der Ressourcenverbrauch eines Dienstes, der von einem fremden Dienstnutzer genutzt wird, wird in der fremden Domäne gemessen. Auch das Accounting und Charging werden beispielsweise dort durchgeführt. Wenn die Rechnungsstellung und Zahlung über den Heimatdienstanbieter erfolgt, muss dieser die mit der Dienstnutzung verbundenen Charging-Records erhalten. In diesem Szenario ist außerdem zu bestimmen, welcher Tarif für den Dienstnutzer gültig ist, und auf welche Tarif-Daten der Charging-PEP zugreift. Für den Dienstnutzer können die Tarife des fremden Dienstanbieters gelten oder die seines Heimatdienstanbieters, wie es im Roaming innerhalb von Funktelefonnetzen der Fall ist. Wird der Dienst vom Heimatdienstanbieter abgerechnet, muss zusätzlich die Diensterbringung intern zwischen den Dienstanbietern verrechnet werden.

Die organisatorischen Modelle und die zu ihrer Realisierung notwendigen Vereinbarungen zwischen den Dienstanbietern, können, wie anhand des Beispiel erläutert, sehr komplex sein.

5.5.3 Systemaspekte: Zusätzliche Datenobjekte und Protokollnachrichten

Mit der Erweiterung der A^x-Architektur um Komponenten und Funktionen, erhöht sich auch die Zahl der Schnittstellen zwischen den Komponenten und erweitern sich teilweise deren Funktionalitäten. In Tabelle 11 sind im Vergleich zu Tabelle 10 auf Seite 112 nur die zusätzlichen oder modifizierten Einträge aufgenommen.

Schnittstelle	Funktionalität	wichtigste ausgetauschte Daten
A^x-Client - A^x-Server	A^x-Dienstanfragen austauschen A^x-Dienstende anzeigen A^x-Dienständerungen anzeigen	-> Dienstanbieter, Dienstnutzer, Dienstbeschreibung, Auth.-Informationen, Session-ID, Berechtigungsnachweis, Accounting-Informationen <- Ergebnis
A^x-Server - Policy Repsoitory	Abfrage einer Zugriffskontroll-Policy Abfrage einer kaufmännischen Policy	-> Dienstbeschreibung, Heimatdienstanbieter, Dienstnutzer <- Policy
A^x-Server - PEP	Konfiguration des PEPs und Ergebnismitteilung	-> Ausschnitt aus Policy, Datenobjekte aus Zugriffskontrolldienstanfrage <- Ergebnis
Metering-PEP - Metering-Repository	Speicherung der Metering-Informationen	Metering-Records
Accounting-PEP - Accounting-Repository	Speicherung der Accounting-Records	Accounting-Records
Accounting-PEP - Metering-Repository	Abfrage von Metering-Records	Metering-Records
Charging-PEP - Charging-Repository	Speicherung der Charging-Records	Charging-Records
Charging-PEP - Accounting-Repository	Abfrage der Accounting-Informationen	Accounting-Records
Charging-PEP - Tarif/Kontrakt-Repository	Abfrage der Tarif-/Kontraktdaten	Tarif-/Kontraktdaten
Billing-PEP - Charging-Repository	Abfrage der Charging-Records	Charging-Records
Charging-Repository externe Systeme	Abfrage der Charging-Records	Charging-Records

Tabelle 11: Zusätzliche Schnittstellen zwischen den Komponenten der erweiterten A^x-Architektur

Der Überblick zeigt, dass keine weiteren Protokolle notwendig sind. Zur Kommunikation mit den verschiedenen Repositories sind wiederum allgemein Abfrageprotokolle zu verwenden. Zur

Realisierung der zusätzlichen Funktionalitäten sind aber zusätzliche Nachrichtentypen für das A^X-Protokoll und zusätzliche Datenobjekte für das A^X-Protokoll zu definieren.

A^X-Protokoll. Zur Verwendung im A^X-Protokoll ist ein weiteres Datenobjekt **Accounting-Informationen** notwendig. In diesem werden verschiedene Informationen, die mit der Dienstnutzung in Verbindung stehen und über die Dienstbeschreibung und Dienstnutzer-ID hinausgehen, zusammengefasst. So können darin beispielsweise über den Dienstnutzer, der den Dienst anfragt, weitere Dienstnutzer am Dienst partizipierende Dienstnutzer angegeben werden (Vgl. dazu die Erläuterung der Pricing-Policy auf Seite 81.).

Außerdem sind, wie bereits erläutert und in Abbildung 60 gezeigt, zwei weitere Nachrichtentypen zu definieren:

- **A^X-Dienstende:** Das Ende der Diensterbringung zeigt der A^X-Client dem A^X-Server mit Hilfe einer eigenen Nachricht an. Damit wird zum einen das Accounting beendet und das Charging gestartet. Weiterhin muss der Session-Eintrag im Session-Repository gelöscht werden. Verwendete Datenobjekte sind die Anfrage-ID und die Session-Informationen.
- **A^X-Dienständerung:** Ändern sich während der Diensterbringung Parameter der Dienstbeschreibung, so müssen diese dem A^X-Server mit einer A^X-Dienständerungsnachricht angezeigt werden. Dazu können neben dem Datenobjekt Accounting-Informationen auch die Datenobjekte Dienstnutzer-ID, Dienstbeschreibung und Session-Informationen genutzt werden.

Policy-Protokoll. Mittels des Policy-Protokolls werden die verschiedenen PEPs konfiguriert und senden sie eine Meldung an den PDP zurück. Wie am Beispiel der Accountings erläutert, kann dies innerhalb einer Session mehrfach erfolgen. Auch für das Policy-Protokoll sind zusätzliche Nachrichten notwendig, bzw. übermittelte Datenobjekte zu modifizieren.

PDP-Dienstende: Im Gegensatz zu den anderen PEPs sind der Metering-PEP und der Accounting-PEP über einen längeren Zeitraum aktiv, während dessen sie ihre Aktivität erbringen. Das Ende der Dienstes muss durch eine PDP-Dienstende Nachricht übermittelt werden.

PDP-Dienstantwort: Authentifizierungs- und Autorisierungs-PEPs übermitteln mit Hilfe dieser Nachricht ihr jeweiliges Ergebnis, also ob die Authentifizierung bzw. Autorisierung des Dienstnutzers erfolgreich war. Die PEPs der kaufmännischen Funktionen speichern die von ihnen erzeugten Daten hingegen in eigenen Repositories. Mittels der PDP-Dienstantwort zeigen sie am. ob sie die geforderte Funktion erbringen können.

5.6 Zusammenfassung

Basierend auf drei existierenden Konzepten, der Separierung der Endnutzerdienste von den Ax-Diensten, der Modularisierung der Teilfunktionen der A^X-Dienste und dem policybasierten Management wurde die A^X-Architektur entworfen. Die Architektur nutzt also das Third Party Modell als Basis und erlaubt eine dynamische Konfiguration der A^X-Systems mittels der Policies.

Die Funktionsweise des A^X-Servers lässt sich knapp in folgender Form zusammenfassen. Er erhält vom Dienstanbieter eine A^X-Dienstanfrage unter Verwendung des A^X-Protokolls, fragt die

dem Dienst zugehörige Policy ab, konfiguriert die Policy Enforcement Points, wie in der Policy spezifiziert, erwartet die Rückmeldung der Authentifizierungs- und Autorisierungs-PEPs, verkettet diese und trifft daraufhin die Zugriffskontrollentscheidung. Die Funktionsweise ist also völlig unabhängig von den Diensten, die kontrolliert werden sollen. Weiterhin wird über die Policy spezifiziert, ob und in welchen Formen die Authentifizierung und Autorisierung durchgeführt werden sollen, wozu die einzelnen PEPs verwendet werden. Existieren PEPs für die in Kapitel 3.1 beschriebenen Grundformen der Authentifizierung und Autorisierung, so werden diese Formen und damit auch verschiedenste Geschäftsmodelle eines Dienstanbieters unterstützt. Eine anonyme Dienstnutzung kann kontrolliert werden, sofern die Policy des Dienstanbieters das vorsieht und ein PEP zur Prüfung der Berechtigungsnachweise existiert. Ein Nutzerberechtigungs-PEP prüft die Rechte des Dienstnutzers und ermöglicht damit die Kontrolle privater Dienst. Zuletzt kann, mittels der Realisierung verschiedener Organisationsmodelle und der global eindeutigen Identifizierung der Dienstnutzer, auch die Kontrolle von Dienstnutzern ausserhalb ihrer Heimatdomäne durchgeführt werden.

	Kontrolle von								Ident.		Autorisierung				Modell		Konfig.	
	Verbindungsdiensten	Internet-Zugangsdiensten	QoS-Transportdiensten	Anwendungsdiensten	Inhaltsdiensten	fremden Dienstnutzern	anonymen Dienstnutzern	privaten Diensten	Merkmal einer Person	Merkmal eines Geräts	authentifizierungsbasiert	berechtigungsbasiert	statisch	dynamisch	integriertes Modell	Third Party Modell	statisch	dynamisch
A^x-Architektur		x	x	x	x	x	x	x	x	x	x	x		x		x		x

Tabelle 12: Merkmale der A^x-Architektur

Insgesamt zeichnet sich die A^x-Architektur also über die in Tabelle 12 gezeigten Merkmale aus. Sie ist damit wesentlich generischer als existierende Systeme, wie sie in Tabelle 8 auf Seite 57 in Kapitel 3.5 zusammengefasst wurden.

Eine fortlaufende Zugriffskontrolle während der Diensterbringung auf dem Datenpfad, wie sie von auf Paketfiltern basierenden Firewalls oder im Policing und Traffic-Shaping genutzt wird, wäre augrund der Separierung der Dienste theoretisch auch abbildbar. Die A^x-Dienste müssen dann mehrfach genutzt werden. Ein solches Vorgehen wird aber den Anforderungen an die Performanz nicht genügen können. Daher muss es grundsätzlich dem Dienstanbieter überlassen bleiben, welche Dienste er kontrollieren will und in welcher Form.

Kapitel 6: Bewertung der A^x-Architektur

Damit ein Anbieter von Internet-Diensten schnell auf die technologischen und ökonomischen Entwicklungen reagieren kann und dazu seine Geschäftsmodelle ändern und neue Dienste bereitstellen kann, benötigt er ein flexibel konfigurierbares und für verschiedene Dienste und Nutzungsszenarien einsetzbares Zugriffskontroll- und Abrechnungssystem. Weiterhin muss er und damit das Design des Zugriffskontrollsystem die zunehmende Mobilität der Dienstnutzer berücksichtigen, welche die Dienste unabhängig von ihrem Aufenthaltsort in gewohnter Weise nutzen wollen. Die Bewertung der A^x-Architektur erfolgt daher zunächst an diesem Ziel und den sich daraus ergebenden Teilzielen, wie sie in der Einleitung formuliert wurden. Es wird die Erfüllung der funktionalen Anforderungen überprüft. Weiterhin müssen zur Bewertung Kriterien berücksichtigt werden, die grundsätzlich für Softwaresysteme gelten. Das sind die Leistungsfähigkeit, Benutzerfreundlichkeit, Zuverlässigkeit, Supportability und nicht zuletzt die Sicherheit des Systems. Diese Kriterien werden im zweiten Abschnitt betrachtet. Der Aspekt der Leistungsfähigkeit wird im dritten Abschnitt detailliert untersucht, indem die Zugriffskontrolle mittels eines A^x-Systems mit der in existierenden Systemen verglichen wird.

6.1 Überprüfung der funktionalen Anforderungen

Mit einem A^x-System soll dem Internet-Dienstanbieter ein Zugriffskontrollsystem zur Verfügung stehen, welches er als einheitliches System zur Kontrolle der von ihm angebotenen Dienste nutzen kann. Dazu muss das System (1) den Zugriff auf existierende und neu zu entwickelnde Dienste aller Dienstklassen kontrollieren können und (2) beliebige Geschäftsmodelle des Dienstanbieters abbilden können. Weiterhin soll die, aufgrund der neuen Zugangstechnologien und der Miniaturisierung der Endgeräte, zunehmende Mobilität der Dienstnutzer unterstützt werden. Dazu muss das System (3) den Zugriff fremder Dienstnutzer, die sich außerhalb ihrer Heimatdomäne befinden, kontrollieren können. Um jederzeit das vom Dienstanbieter geforderte Maß an Sicherheit realisieren zu können, ist es weiterhin notwendig, dass (4) die sicherheitsgewährleistenden kryptographischen Verfahren innerhalb der Architektur ausgetauscht werden können, sobald sie als unsicher angesehen werden. Auf diese vier Anforderungen, die auch als vier Teilziele der Arbeit formuliert wurden, wird nachfolgend im Einzelnen eingegangen.

Kontrolle des Zugriffs auf Internet-Dienste der verschiedenen Dienstklassen. Innerhalb der A^x-Architektur ist die grundsätzliche Funktionalität des A^x-Servers und der PEPs unabhängig von den Diensten, die kontrolliert werden. Dies wurde in Kapitel 5.6 bereits mit der Anwendung des Konzepts der Separierung der Dienste argumentativ begründet.

Anhand verschiedener exemplarischer Anwendungsfälle lässt sich nachweisen, dass die Dienste der verschiedenen Dienstklassen mittels eines A^x-Systems kontrolliert werden können. Dazu wurde die Realisierung der Zugriffskontrolle in den verschiedenen Anwendungsfällen aus Kapitel 2.1 geprüft. Diese Anwendungsfälle sind repräsentativ. Sie umfassen Dienste aller Dienstklassen, wie in Tabelle 15 in Anhang B auf Seite 165 aufgeführt. Die Zugriffskontrolle in den verschiedenen Anwendungsfällen variiert den in Kapitel 5.4.3 vorgestellten Ablauf. In Anhang B sind die einzelnen Protokollabläufe bei Nutzung der A^x-Architektur in den Anwendungsfällen dargestellt. Einzelne Beispiele werden im folgenden Kapitel 6.3 vertieft erläutert.

Die Zugriffskontrolle auf Internet-Zugangsdienste ist zudem unabhängig von der verwendeten Zugangstechnologie. In Kapitel 3.3.1 wurden 5 Arten von Verbindungen unterschieden. Sie sind in den Anwendungsfällen wie folgt vertreten: In Anwendungsfall 1 wird ein kabelbasiertes LAN, in Anwendungsfall 2 ein Funk LAN und in Anwendungsfall 3 eine Wählverbindung im Funknetz genutzt. Die Zugriffskontrolle bei Verwendung einer Wählverbindung im Festnetz und auch von Standleitungen erfolgt auf vergleichbare Weise. Dazu kann beispielsweise statt des Protokolls EAP over Wireless, wie es in Anwendungsfall 3 zur Kommunikation zwischen der Port Access Entity und dem Endgerät des Dienstnutzers verwendet wird, PPP-EAP eingesetzt werden.

Abbildung beliebiger Geschäftsmodelle. Damit der Dienstanbieter in der Lage ist, flexibel auf veränderte Marktbedingungen reagieren zu können, müssen sich im Zugriffskontroll- und Abrechnungssystem nahezu beliebige Geschäftsmodelle eines Internet-Dienstanbieters abbilden lassen. Dazu sind drei Voraussetzungen zu erfüllen: Das System muss flexibel konfigurierbar sein, es müssen vom System alle möglichen Formen der Zugriffskontrolle unterstützt werden und es müssen weiterhin die kaufmännischen Regeln abgebildet werden können, die Bestandteil des Geschäftsmodells sind.

Die flexible Konfigurierbarkeit ist innerhalb der A^x-Architektur aufgrund des policybasierten Managements möglich. Die Form der Zugriffskontrolle ist, wie dies in Kapitel 4 erläutert wird, abhängig vom Geschäftsmodell. Für unterschiedliche Ausprägungen von Geschäftsmodellen sind alle möglichen Grundformen der Zugriffskontrolle notwendig, wie sie in Kapitel 3.1 analysiert wurden. Sollen beliebige Geschäftsmodelle von einer Zugriffskontrollarchitektur abgebildet werden, so müssen auch alle existierenden Formen der Zugriffskontrolle im Zugriffskontrollsystem ausführbar sein. Die Autorisierung kann authentifizierungs- oder berechtigungsnachweisbasiert sowie statisch oder dynamisch erfolgen. Beim Zugriff auf private Dienste müssen die Nutzerberechtigungen geprüft werden. Aus diesem Grund werden in der A^x-Architektur Authentifizierungs-PEPs, Berechtigungsnachweis-PEPs, Systemzustands-PEPs und Nutzerberechtigungs-PEPs unterschieden. Diese führen jeweils die entsprechende Form der Zugriffskontrolle aus, sofern die Zugriffskontroll-Policy dies spezifiziert. Die grundlegenden, für eine Zugriffskontrolle notwendigen Formen der Enforcement Points sind also existent. Die Protokollnachrichten zum Austausch der Datenobjekte wurden für die A^x-Architektur definiert. Ist es aufgrund des Geschäftsmodells notwendig, bisher nicht realisierte dynamische Prüfungen vorzunehmen und ein neues externes System einzubinden, so muss die A^x-Architektur um einen entsprechenden

PEP mit anwendungsspezifischen Kommunikationsschnittstellen zu diesem externen System erweitert werden. Die übrigen Komponenten bleiben unverändert.

Die Abbildung der kaufmännischen Regeln des Geschäftsmodells erfolgt primär mittels der Policy zur Beschreibung der kaufmännischen Unterstützungsdienste (vgl. Kapitel 4.4.2). Diese sieht die Umsetzung einer Vielzahl von kaufmännischen Regeln und Elementen dieser Regeln vor. Werden abweichend von den existierenden Elementen neue entwickelt, müssen die Policies und die Syntax der Policy-Beschreibungssprache um entsprechende Elemente erweitert werden.

Unterstützung der Zugriffskontrolle bei Dienstanfragen fremder Dienstnutzer. Die Mobilität der Dienstnutzer nimmt aufgrund der neuen Zugangstechnologien und der Miniaturisierung der Endgeräte ständig zu. Anbieter von Internet-Diensten müssen dem Rechnung tragen. Das zeigt sich heute primär bei Anbietern von Funktelefonnetzen. In diesem Bereich lassen sich vier kritische Voraussetzungen für eine Mobilitätsunterstützung ermitteln: Es sind die Verwendung einheitlicher Endgerätetechnologien, die Nutzung einheitlicher Verfahren zur Zugriffskontrolle und Abrechnung, die Verwendung global eindeutiger Identitätsmerkmale der Dienstnutzer, mit deren Hilfe sich auch der Heimatdienstanbieter bestimmen lässt, und die Unterstützung entsprechender Organisationsmodelle und Realisierung von sogenannten Roaming-Abkommen zwischen den Dienstanbietern.

Die ersten beiden Voraussetzungen sind aufgrund der offenen im Internet verwendeten Protokolle und bei Einsatz von A^X-Systemen durch alle Dienstanbieter gegeben. Für eine eindeutige Verwendung von Identitätsmerkmalen wurde für die A^X-Architektur ein Konzept zur Identifizierung der Dienstnutzer definiert (vgl. Kapitel 5.2.4). Verschiedene Organisationsmodelle lassen sich aufgrund der Separierung der Dienste und der Modularisierung der Teilfunktionen der Zugriffskontrolle ebenfalls verwirklichen, wie in Kapitel 5.3.6 erläutert. Die A^X-Architektur bietet also grundsätzlich die Voraussetzung für die geforderte Unterstützung der Zugriffskontrolle bei Dienstanfragen fremder Dienstnutzer. Auch die Abrechnung der Dienste durch fremde Dienstanbieter ist vorgesehen. Auf diesen Aspekt wurde in Kapitel 5.5.2 vertieft eingegangen.

Der Nachweis der Funktionalität erfolgt über zwei repräsentative Mobilitätsszenarien: der Mobilität auf Ebene der Internet-Zugangsdienste und mittels einer Verwendung von Mobile-IP. Die Mobilität des Dienstnutzers auf Ebene der Internet-Zugangsdienste lässt sich in Anwendungsfall 2 betrachten. Der Dienstnutzer kann einen Internet-Zugang nutzen, der von einem fremden Dienstanbieter zur Verfügung gestellt wird. Der Protokollablauf ist in Anhang B.2 beschrieben.

Die A^X-Architektur kann außerdem auch für eine Kontrolle eines mobilen Knoten durch den Foreign-Agent in einem Mobile-IP Szenario verwendet werden. Diese Kontrolle wird ausgelöst durch die Mobile-IP-Registration Nachricht des mobilen Knotens, wie in Abbildung 61 gezeigt. Neben der Authentifizierung des Endnutzers durch den Authentifizierungs-PEP in der Heimatdomäne des mobilen Knoten ist zusätzlich die Registrierung beim Home-Agent notwendig. Dazu wird die Authentifizierung mit der Registrierung verknüpft, d.h. der Authentifizierungs-PEP, der die Authentifizierung vornimmt, sendet die Registrierung an den Home-Agent. Alternativ kann

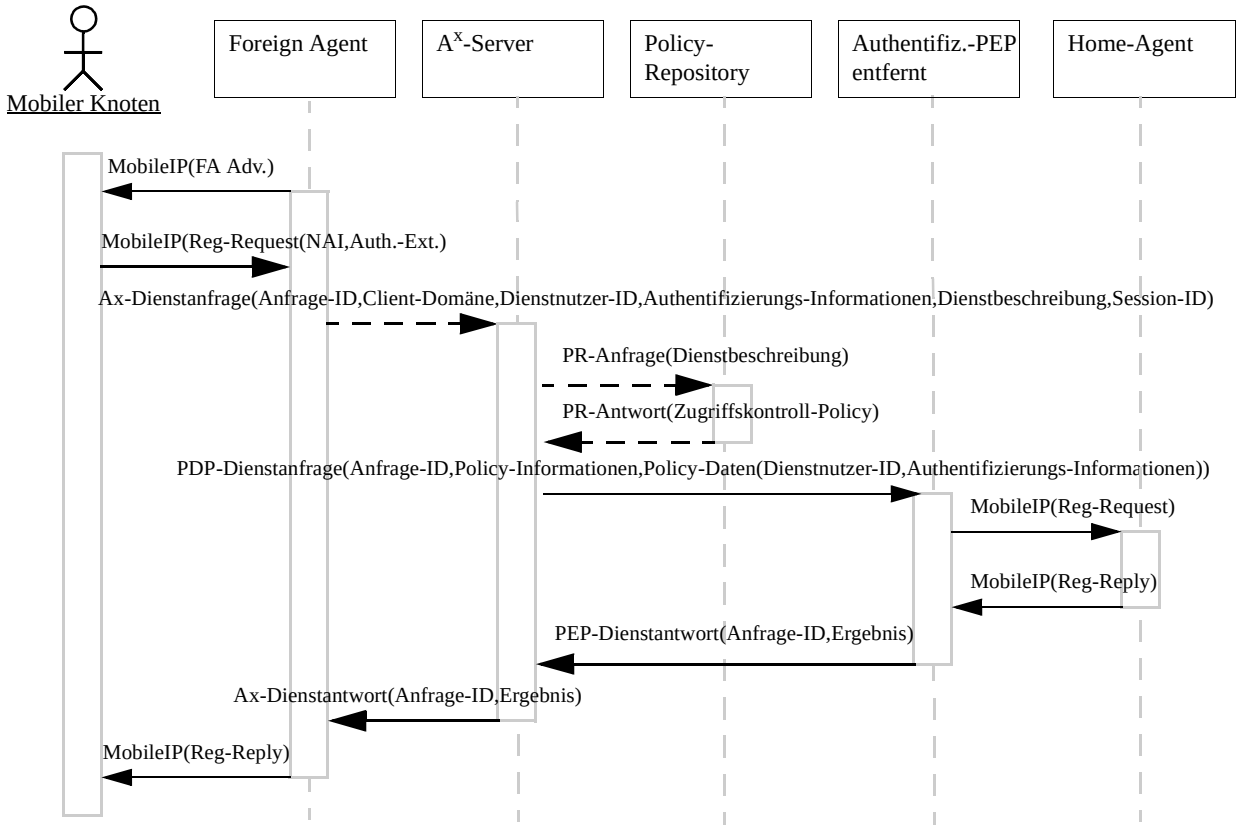

Abbildung 61: Protokollablauf für die Registrierung eines mobilen Knoten

der Foreign-Agent die Registrierung nach erfolgter Authentifizierung auch direkt an den Home-Agent senden.

Die Kontrolle des Zugriffs auf Verbindungsdienste ist nicht primäre Aufgabe des A^X-Systems. Einen Einsatz der Architektur ist aber grundsätzlich möglich, sofern zur Signalisierung der Dienste IP-basierte Protokolle verwendet werden, wie das für die Mobilfunknetze der dritten Generation u.a. für das Multimedia Subsystem diskutiert wird [GPP03]. Dann handelt es sich auch bei den Verbindungsdiensten um Internet-Dienste. Eine Unterstützung der Mobilität des Benutzers auf Ebene der Verbindungsdienste erfolgt außerdem in der Form, dass bei einem Handover auf Verbindungsebene keine erneute aktive Authentifizierung des Dienstnutzers notwendig ist. Die A^X-Architektur verwendet aus diesem Grund das Session-Konzept, wie es in Kapitel 5.2.5 vorgestellt wurde. Dem Internet-Zugangsrechner muss im Rahmen des Handovers signalisiert werden, dass es sich nicht um eine neue Session, sondern um eine bestehende handelt. Ändert sich beim Handover auch der Anbieter des Zugangsdienstes, können die zwei Sessions zum Zwecke der gemeinsamen Abrechnung aneinander gebunden werden.

Die Kontrolle des Zugriffs auf Anwendungs- und Inhaltsdienste ist unabhängig von der physischen Mobilität des Dienstnutzers, da zur Identifizierung der Dienstnutzer in der A^X-Architektur persönliche Identitätsmerkmale verwendet werden. Der Zugriff auf die Dienste erfolgt aus Sicht des Dienstanbieters letztendlich unabhängig vom Aufenthaltsort des Dienstnutzers in identischer Weise über eine Dienstanfrage auf Basis des IP-Protokolls. Eine Unterstützung der Kontrolle des

Zugriffs entfernter Dienstnutzer bedeutet in diesem Falle eher die Verwirklichung eines Single Sign Ons.

Einsatz beliebiger sicherheitsgewährleistender Verfahren in der Zugriffskontrolle. Wenn ein kryptographisches Verfahren als unsicher angesehen wird, was jederzeit der Fall sein kann [BM99], ist es notwendig, die sicherheitsgewährleistenden kryptographischen Verfahren innerhalb des Zugriffskontroll- und Abrechnungssystems schnell auszutauschen. Dieser Anforderung wird zum einen durch die Separierung der Dienste und die Modularisierung der Teilfunktionen der Dienste, zum anderen mittels der für ein A^X-Protokoll definierten Nachrichtentypen und Datenobjekte entsprochen. Sicherheitsgewährleistende kryptographische Verfahren werden nur in der Authentifizierung und der berechtigungsnachweisbasierten Autorisierung genutzt, also in den Policy Enforcement Points und gegebenenfalls in den einbezogenen externen Systemen. Diese lassen sich modifizieren und austauschen, ohne dass andere Komponenten der A^X-Architektur verändert werden müssen. Die definierten Nachrichtentypen und Datenobjekte erlauben den Austausch beliebiger Informationen, die zur Identifizierung, Authentifizierung und Autorisierung notwendig sein können. So lassen sich verschiedene Protokolle und technische Verfahren unterstützen. Damit kann die Anforderung grundsätzlich erfüllt werden.

Diese Aussage lässt sich anhand der in Kapitel 2.1 vorgestellten Anwendungsfälle verifizieren. Sie verwenden verschiedene Authentifizierungs-Verfahren auf Basis von Besitz und Wissen des Dienstnutzers (vgl. wiederum Tabelle 15 in Anhang B auf Seite 165). In Anwendungsfall 1 erfolgt beispielsweise eine Challenge Response Authentifizierung auf Basis des Besitzes einer Smartcard. In den anderen Anwendungsfällen wird der Dienstnutzer über sein Wissen in Form von Benutzerkennung und Passwort ebenfalls mit oder ohne Verwendung von Challenge Response Verfahren authentifiziert. Sie lassen sich alle unter Nutzung der A^X-Architektur umsetzen. Die Verwendung von biometrischen Verfahren ist ebenfalls möglich. Sie sind nicht Bestandteil der Anwendungsfälle, da sie zur Kontrolle von Internet-Diensten noch wenig praktische Relevanz besitzen. Sie unterscheiden sich von den anderen Verfahren primär in der Form der Authentifizierungs-Informationen, was keine Bedeutung für die Anwendung der A^X-Architektur besitzt.

Die Nutzung von auf kryptographischen Verfahren beruhenden Berechtigungsnachweisen ist ebenfalls möglich. Diese lassen sich, unabhängig von ihrer Form, mittels der definierten Nachrichtentypen vom Dienstanbieter, der sie vom Dienstnutzer anfragt, bis zum Authentifizierungs-PEP übermitteln. Es können z.B. digitale Signaturen zum Nachweis der Urheberschaft der Dienstanfrage unter Einbeziehung von Public Key Infrastrukturen genutzt werden.

Die Verwendung digitaler Signaturen innerhalb der A^X-Architektur und die Einbindung einer PKI soll hier detailliert beschrieben werden. Der Dienstnutzer signiert seine Anfrage, die u.a. eine Beschreibung des angefragten Dienstes und seine Identitätsmerkmale enthält, mittels einer digitalen Signatur unter Verwendung seines privaten Schlüssels. Er sendet die signierte Anfrage an den Anwendungs-Server. Dieser stellt daraufhin eine A^X-Dienstanfrage an den A^X-Server, wie in Abbildung 62 dargestellt. Die gesamte signierte Dienstanfrage ist Teil des Objekts Authentifizierungs-Informationen.

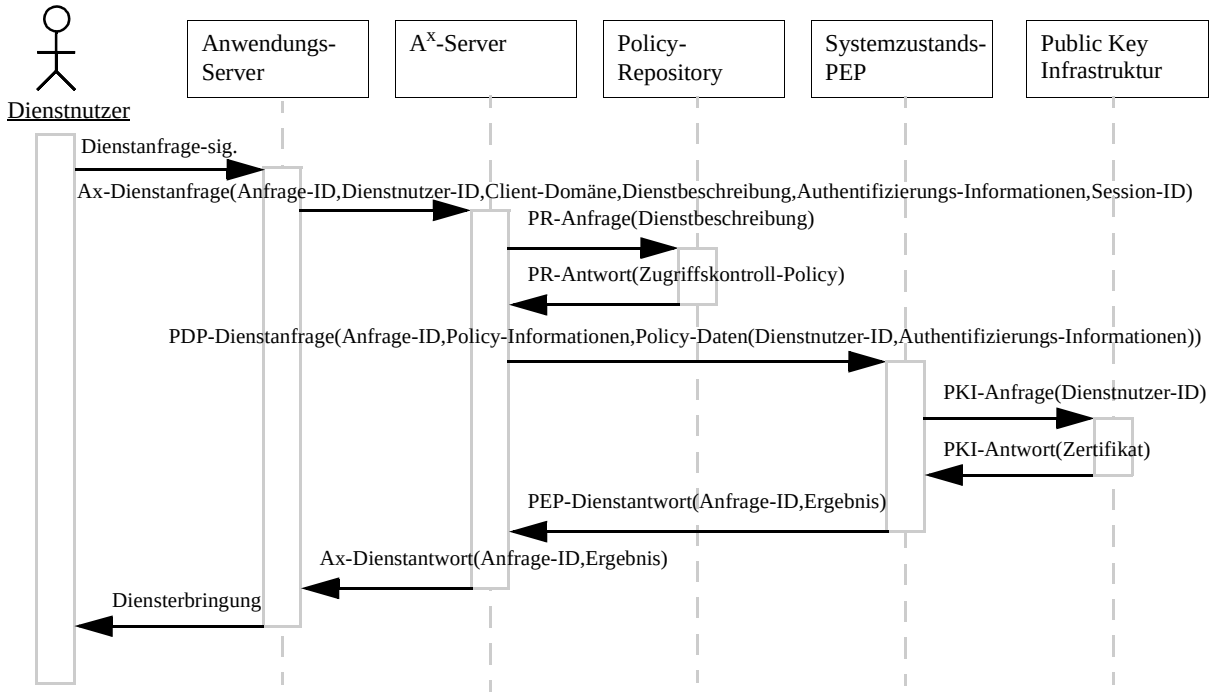

Abbildung 62: Protokollablauf bei einer Verwendung digital signierte Dienstanfragen

Sieht die Zugriffskontroll-Policy für den angefragten Dienst eine Prüfung der digitalen Signatur durch einen Policy Enforcement Point vor, so wird dieser mittels einer PDP-Dienstanfrage konfiguriert. Es handelt sich, nach der in der Arbeit verwendeten Definition, um einen Systemzustands-PEP, denn es wird auf eine externe Public Key Infrastruktur zur dynamischen Abfrage oder Prüfung der Gültigkeit des Zertifikats des Dienstnutzers zugegriffen. Der Enforcement Point prüft die digitale Signatur unter Verwendung des öffentlichen Schlüssels des Dienstnutzers, der Bestandteil seines Zertifikats ist, und verifiziert die Gültigkeit des Zertifikats unter Nutzung der PKI oder er fragt es aktuell von dieser ab. Ist die Prüfung erfolgreich, sendet er eine positive Ergebnisnachricht an den A^X-Server zurück. Dieser trifft die Zugriffskontrollentscheidung, gegebenenfalls unter Berücksichtigung von weiteren Autorisierungs-Ergebnissen, und meldet sie an den Anwendungs-Server.

Die A^X-Architektur unterstützt grundsätzlich beliebige kryptographische Verfahren in der Zugriffskontrolle, sofern die PEPs die Verfahren entsprechend realisieren. Diese Forderung muss äquivalent auch für die Benutzeranwendungen und Endgeräte der Dienstnutzer gelten, weshalb auch diese modular realisiert sein sollten. Die Benutzeranwendungen und Endgeräte sind nicht Bestandteil der A^X-Architektur und werden daher in dieser Arbeit nicht betrachtet.

Die Absicherung der Kommunikation innerhalb eines A^X-Systems und mit externen Systemen erfolgt entweder unter Verwendung standardisierter Protokolle oder über eine Sicherung mittels kryptographischer Verfahren auf Ebene der Datenobjekte. Um in zweiten Fall einen Austausch der kryptographischen Verfahren zu ermöglichen, muss bei der Implementierung eines A^X-Systems auf eine modulare Entwicklung geachtet werden.

Zusammenfassung. Die funktionalen Anforderungen eines Internet-Dienstanbieter an ein Zugriffskontrollsystem sind von der A^X-Architektur abgedeckt. Ein A^X-System ist in der Lage, verschiedene existierende Systeme durch ein einheitliches System ersetzen. Es kann den Internet-Zugang wie auch die Nutzung von Anwendungs- und Inhaltsdiensten kontrollieren. Bei der Entwicklung neuer und zusammengesetzter Dienste kann der Dienstanbieter auf ein existierendes System zurückgreifen.

6.2 Überprüfung der A^X-Architektur auf allgemeine Anforderungen

Neben der Funktionalität müssen zu einer Bewertung der A^X-Architektur weitere Kriterien herangezogen werden. Ein Softwaresystem muss generell nicht nur die erforderliche Funktionalität erbringen, sondern auch leistungsfähig, benutzerfreundlich, zuverlässig und sicher sein. In Kapitel 5.1 wurden die verschiedenen Anforderungen, die an Softwaresysteme zu stellen sind, erläutert. Ihre Erfüllung wird nachfolgend erörtert.

Benutzerfreundlichkeit und Bedienbarkeit. Die allgemeinen Merkmale der Benutzerfreundlichkeit, die Erlernbarkeit, Effektivität, Merkbarkeit, Fehlerbehandlung und Nutzerzufriedenheit [Nie94] sind für einen Dienstnutzer nur in geringem Maße abhängig von der Realisierung der Zugriffskontrolle. Sie bestimmen sich für ihn primär durch seine lokale Anwendung und den genutzten Dienst. Die Zugriffskontrolle wird für den Nutzer transparent erbracht, abgesehen von der aktiven Identifizierung und Authentifizierung. Die genutzten Authentifizierungs-Verfahren haben einen Einfluss auf die Benutzerfreundlichkeit. Insbesondere unterscheiden sich biometrische Verfahren hinsichtlich der Benutzerfreundlichkeit [SSGS00]. Bei der Auswahl der Verfahren, die sich innerhalb der A^X-Architektur flexibel austauschen lassen, ist also neben den Aspekten Sicherheit und Performanz auch auf die Benutzerfreundlichkeit zu achten.

Anwendungsspezifische mögliche Forderungen eines Dienstnutzers an die Benutzerfreundlichkeit eines Zugriffskontrollsystems werden durch die A^X-Architektur erfüllt. Sie unterstützt die Mobilität der Dienstnutzer. Die Anzahl der aktiven Identifizierungen und Authentifizierungen wird durch die Implementierung des Session-Konzepts gering gehalten (vgl. Kapitel 5.2.5). Innerhalb der A^X-Architektur wird ein global eindeutiges Identitätsmerkmal für die Dienstnutzer verwendet. Es lassen sich zudem verschiedene Organisationsmodelle realisieren, so dass der Dienstnutzer theoretisch nur ein einziges Identitätsmerkmal zum Zugriff auf Dienste beliebiger Dienstanbieter verwenden muss. Dagegen sprechen zum einen die Notwendigkeit, dass sich die Dienstanbieter in ein globales Organisationsmodell einbinden müssen, und zum anderen verschiedene Aspekte des Datenschutzes. Gleiches gilt auch für die Gestaltung der Abrechnung und Zahlung. Die A^X-Architektur erlaubt, aufgrund der Flexibilität der Spezifikation der kaufmännischen Policies, verschiedene Organisationsmodelle und auch eine zentrale Abrechnung der von verschiedenen Anbietern genutzten Dienste.

Zuverlässigkeit. Die Zuverlässigkeit eines Zugriffskontrollsystems ist eine wichtige Anforderung aus Sicht des Anbieters von Internet-Diensten. Ein Zugriffskontrollsystem darf keinen Single Point of Failure besitzen. Die A^X-Architektur erlaubt die Replikation der Komponenten:

Innerhalb einer Anbieterdomäne können mehrere A^X-Server und mehrere PEPs für die einzelnen Teilfunktionen eingerichtet werden. Fällt ein A^X-Server aus, kann der Dienstanbieter die A^X-Dienstanforderung an einen anderen A^X-Server stellen. Die zur Zugriffskontrolle notwendigen Daten können ebenfalls repliziert werden. Die Integrität der Daten, der Authentifizierungs-Informationen und Nutzerberechtigungen aber auch der kaufmännischen Daten kann zum einen über Mechanismen der verwendeten Repositorien hergestellt werden. Zum anderen erlauben die für die A^X-Protokolle und Policy-Protokolle definierten Datenobjekte die Verwendung einer digitalen Signatur über die Datenobjekte, um damit die Integrität und Nachweisbarkeit zu realisieren.

Supportability. Der Anforderung der Supportability wird in der A^X-Architektur in zweierlei Hinsicht Rechnung getragen. Die Nutzung des policybasierten Managements erlaubt dem Dienstanbieter eine flexible Konfiguration der Zugriffskontrolle und Anpassung an Rahmenbedingungen, wie sich ändernde Geschäftsmodelle. Auf diesen Aspekt wurde bereits mehrfach eingegangen. Die Modularisierung der Teilfunktionen der Zugriffskontrolle erlaubt zudem eine relativ einfache Anpassbarkeit an externe Systeme. Dazu müssen nur die Kommunikationsschnittstellen der entsprechende PEPs geändert werden. Das Gesamtsystem kann darüberhinaus unverändert bleiben.

Sicherheit. Das Kriterium der Sicherheit beschreibt zum einen die Sicherheit der Zugriffskontrolle selbst. Diese wird wesentlich bestimmt durch die eingesetzten Authentifizierungs- und Autorisierungs-Verfahren. Die A^X-Architektur erlaubt, wie im letzten Abschnitt dieses Kapitels erläutert, die Nutzung beliebiger Verfahren für die Zugriffskontrolle. Zum anderen muss das Zugriffskontrollsystems selbst gegen Angriffe verschiedenster Arten abgesichert sein. Zur Realisierung der Kommunikationssicherheit können in der Architektur entweder standardisierte Protokolle eingesetzt werden oder es erfolgt eine Sicherung mittels Verschlüsselung und Signaturbildung auf Ebene der Datenobjekte. Dass dazu notwendige Vertrauensmodell lässt sich, wie in Kapitel 5.3.8 erörtert, realisieren.

Leistungsfähigkeit. Die Leistungsfähigkeit der A^X-Architektur wird im folgenden Abschnitt ausführlich anhand des Kriteriums der Anzahl der zur Zugriffskontrolle notwendigen Transaktionen analytisch untersucht. Bei einer realen Anwendung der Architektur ist es darüberhinaus notwendig, dass nicht nur das Gesamtsystem, sondern auch die einzelnen Komponenten und die genutzten externen Systeme leistungsfähig sind und mit der Anzahl der Nutzer und Anfragen skalieren. Sind beispielsweise die Antwortzeit auf eine Repository-Anfrage oder die Bearbeitungszeit innerhalb eines Policy Enforcement Points unzureichend, so ist es auch die Antwortzeit des Gesamtsystems. Diese Faktoren sind nicht abhängig von der A^X-Architektur, die einen Einsatz verschiedenster Komponenten erlaubt. Vielmehr ist jeweils bei der Auswahl der externen Systeme und der Verfahren sowie und bei der Implementierung der jeweiligen Policy Enforcement Points erneut auf die Leistungsfähigkeit zu achten. Um eine Skalierbarkeit mit der Anzahl der Nutzer zu erreichen, erlaubt die A^X-Architektur die Replikation verschiedener Komponenten. So können beispielsweise mehrere A^X-Server je administrativer Domäne eingesetzt werden und diese wiederum mehrere Policy Enforcement Points verwenden.

6.3 Vergleich der Leistungsfähigkeit der A^x-Architektur mit existierenden Systemen

Die Leistungsfähigkeit eines A^X-System wird insgesamt bestimmt über die Leistungsfähigkeit der einzelnen Komponenten und die von der Architektur bestimmte Funktionsweise in Form des Ablaufs der Zugriffskontrolle. Dieser durch die Architektur selbst bestimmte Aspekt soll vertieft untersucht werden. Die Anwendung der A^X-Architektur wird dazu mit der Nutzung existierender Systeme unter dem Gesichtspunkt der Leistungsfähigkeit verglichen.

6.3.1 Die Anzahl der Transaktionen als Kriterium für den Vergleich der Leistungsfähigkeit

Das wesentliche Charakteristikum der Leistungsfähigkeit eines Zugriffskontrollsystems aus Sicht des Dienstnutzers ist die Antwortzeit auf seine Dienstanfrage. Diese korreliert mit der Anzahl der für die Zugriffskontrolle notwendigen Transaktionen, die mit einem Faktor zu gewichten sind, der ihren mittleren Zeitbedarf ausdrückt. Für eine qualitative Aussage ist sie als Kriterium ausreichend und soll daher nachfolgend im Rahmen eines analytischen Vergleichs genutzt werden. Die bei einer Zugriffskontrolle auftretenden Transaktionen lassen sich hinsichtlich ihres Zeitbedarfs in die folgenden Arten unterscheiden:

- **Transaktionen zwischen Dienstnutzer und Endnutzerdienstinfrastruktur (E):** Zu diesen Transaktionen zählen die zwischen Dienstnutzer und Dienstanbieter ausgetauschten Protokollnachrichten, zur Anfrage, Aushandlung sowie Bestätigung oder Ablehnung eines Dienstes und zum Austausch von Identifizierungs- und Authentifizierungs-Informationen.
- **Transaktionen zwischen Zugriffskontroll-Client und Zugriffskontroll-Server (Z):** Diese Transaktionen treten nur auf, wenn die Zugriffskontrolle nicht vom Dienstanbieter selbst vorgenommen wird, also bei einer Verwendung des Third Party Modells (vgl. Kapitel 3.2). Alle Protokollnachrichten zwischen einem Dienstanbieter als Client und einem für die Zugriffskontrolle zuständigen Server zählen dazu. Innerhalb der A^X-Architektur sind es die Nachrichten des A^X-Protokolls, im Allgemeinen Nachrichten eines AAA-Protokolls (vgl. Kapitel 3.3.2) oder auch die COPS-Protokollnachrichten innerhalb der Integrated Services Architektur (vgl. Kapitel 3.3.3).
- **Repository-Anfragen (R)**: Zur Durchführung der Zugriffskontrolle ist ein lesender und auch schreibender Zugriff auf Daten notwendig, die in Datenbanken oder allgemeiner Repositories abgelegt sind. Sie werden mit Hilfe von Repository-Anfragen abgefragt oder modifiziert. Diese Transaktionen finden sich in allen Architekturen In der A^X-Architektur handelt es sich um alle Anfragen an das Policy-, Session-, Authentifizierungs- oder Rechte-Repository.
- **Transaktionen mit externen Systemen (EX):** Externe Systeme sind solche, die außerhalb des Zugriffskontrollsystems realisiert sind und unabhängig von der Zugriffskontrolle existieren. Sie werden auch von anderen Anwendungen genutzt und können innerhalb oder außerhalb der administrativen Domäne des Dienstanbieters lokalisiert sein. Innerhalb der Zugriffskontrolle werden sie für eine dynamische Autorisierung verwendet. Zu den Trans-

aktionen mit externen Systemen zählen in der A^X-Architektur die anwendungsabhängigen Protokollnachrichten zwischen PEPs und externen Systemen.

- **Lokale Transaktionen zwischen Zugriffskontroll-Server und PEP (LP):** Diese Art von Transaktionen treten nur in der A^X-Architektur auf. Zwischen dem A^X-Server und dem PEP werden mittels des Policy-Protokolls Nachrichten zur Konfiguration der PEPs und zur Rückmeldung der Ergebnisse der Policy-Durchsetzung ausgetauscht. Befindet sich der PEP auf dem lokalen A^X-System, handelt es sich um lokale Transaktionen.
- **Transaktionen zwischen Zugriffskontroll-Server und anderen Zugriffskontroll-Systemen (EP):** Wenn die Zugriffskontrolle komplett oder ihre Teilfunktionen in einer anderen Domäne durchgeführt werden, erfolgt eine Kommunikation mit dem entfernten Zugriffskontrollsystem. Dies ist bei einer Unterstützung der Zugriffskontrolle unabhängig vom Aufenthaltsort des Dienstnutzers notwendig. Innerhalb der A^X-Architektur handelt es sich bei dieser Form der Transaktionen um Policy-Protokollnachrichten zwischen dem A^X-Server und einem PEP auf einem entfernten A^X-System. Auch die Anfrage an einen Systemzustands-PEP, der Teil der Endnutzerdienstinfrastruktur ist, wird dieser Klasse zugeordnet. Im Allgemeinen zählen zu dieser Art von Transaktionen AAA-Protokollnachrichten zwischen zwei Zugriffskontroll-Servern.

Die einzelnen Transaktionsarten, wie sie in der A^X-Architektur Verwendung finden, sind in Abbildung 63 markiert. Dazu wird die beispielhafte Lokalisierung aus Kapitel 5.3.7 wiederverwendet.

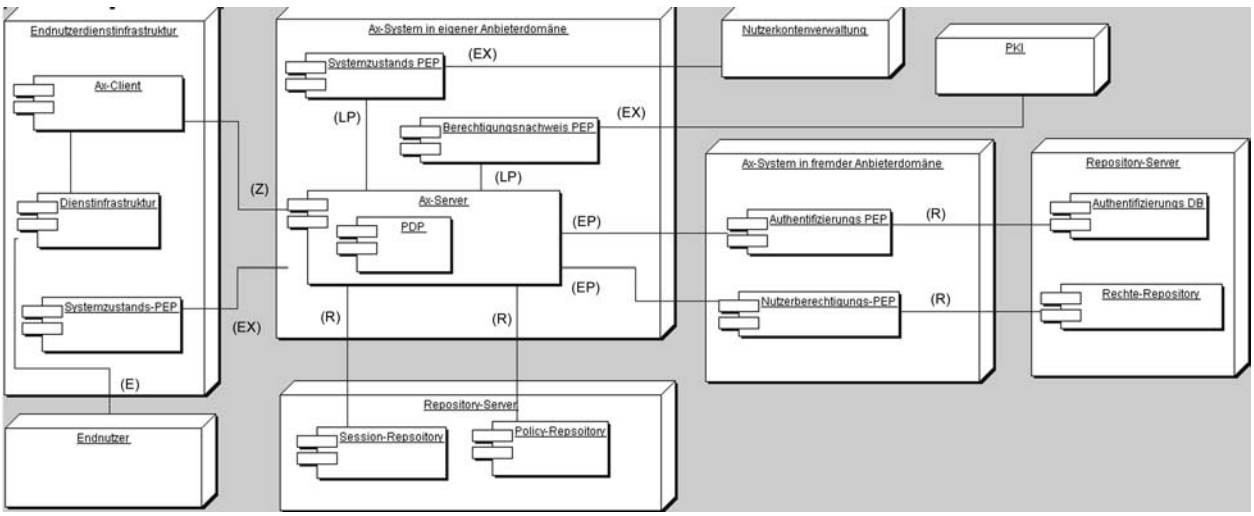

Abbildung 63: Transaktionsarten innerhalb der A^X-Architektur

Für einen Vergleich der A^X-Architektur mit existierenden Systemen sollen die Anzahl der Round-Trips von Protokollnachrichten (*E*, *Z*, *R*, *EX*, *LP*, *EP*), unterschieden in die vorgestellten Arten von Transaktionen, in repräsentativen Anwendungsfällen bestimmt werden. Die einzelnen Transaktionsarten müssen mit unterschiedlichen Faktoren (f_E, f_Z, f_R, f_{EX}, f_{LP} und f_{EP}), die ihren unterschiedlichen Zeitbedarf widerspiegeln, bewertet werden. Der Zeitbedarf einer Transaktion setzt sich zusammen aus der Kommunikationszeit für die Übertragung eines Round-Trips von zwei

Protokollnachrichten und der Bearbeitungszeit zur Bestimmung der Antwort. Dann berechnet sich die Antwortzeit auf eine Dienstanfrage des Dienstnutzers bis zur Dienstererbringung wie folgt:

$$A = f_E \cdot E + f_Z \cdot Z + f_R \cdot R + f_{EX} \cdot EX + f_{LP} \cdot LP + f_{EP} \cdot EP \quad (1)$$

6.3.2 Vergleich anhand exemplarischer Anwendungsfälle

Der Vergleich der A^X-Architektur mit existierenden Systemen erfolgt für die unterschiedlichen Klassen von Internet-Diensten getrennt. Für Internet-Zugangsdienste, QoS-Transportdienste sowie Anwendungs- und Inhaltsdienste werden derzeit Systeme eingesetzt, die auf verschiedenen Architekturen basieren, wie in Kapitel 3.3 erörtert. Für jede Klasse von Internet-Diensten werden daher exemplarische Anwendungsfälle ausgewählt.

Kontrolle des Zugriffs auf Internet-Zugangsdienste. Bei der Kontrolle des Zugriffs auf Internet-Zugangsdienste wird, wie in Kapitel 3.3 vorgestellt, unabhängig von der Zugangstechnologie als einheitliches Architekturmodell das Third Party Modell verwendet. In diesem Modell kommen RADIUS oder Diameter als Protokolle zum Einsatz, der funktionale Ablauf ist aber grundsätzlich identisch. Es müssen daher zwei Szenarien unterschieden werden: Zum einen ist ein Szenario zu vergleichen, in welchem die Dienstanfrage von einem Dienstnutzer aus der eigenen Domäne gestellt wird und zum anderen ein Szenario mit einem Dienstnutzer aus einer fremden Domäne.

Anwendungsfall 2 steht exemplarisch für das Szenario eines Dienstnutzers in einer fremden Domäne. Beim Zugriff auf das Intranet der *Außenstelle der Kanzlei Huber & Partner* erfolgt die Zugriffskontrolle durch einen RADIUS-Server. Da *Herr Grimm* sich außerhalb seiner Heimatdomäne aufhält, muss die Anfrage an seinen Heimatdienstanbieter weitergeleitet werden. Der RADIUS-Server nimmt die Authentifizierung von *Herrn Grimm* anhand seiner Benutzerkennung und seines Passworts vor. Es wird angenommen, dass als Authentifizierungs-Protokoll EAP zusammen mit einem Challenge Response Verfahren genutzt wird, wie es in der Praxis zumeist angewandt wird. Der Protokollablauf bei Verwendung von RADIUS ist komplett in Anhang B.2 illustriert. Mit Hilfe von Repository-Abfragen wird auf die Authentifizierungs-Informationen zugegriffen und die RADIUS-Session verwaltet. Die Antwortzeit beträgt dann:

$$A_{RADIUS}(IntranetZugang) = 3f_E + 2f_Z + 2f_R + 2f_{EP} \quad (2)$$

Den identischen Anwendungsfall bei Verwendung der A^X-Architektur zeigt Abbildung 64. Die Port Access Entity stellt, sobald sie die erste EAP-Nachricht des mobilen Endgeräts erhalten hat, eine Anfrage an den A^X-Server. In dieser kann sie nur den Dienst beschreiben. Der A^X-Server ruft die Policy ab und antwortet mit einer Authentifizierungs-Anfrage, da die Policy eine Authentifizierung vorsieht. Die Port Access Entity leitet diese Anfrage als EAP-Request an den Dienstnutzer weiter. Diese ersten Transaktionen sind nur dann notwendig, wenn die Port Access Entity nicht weiss, welche Art der Zugriffskontrolle auszuführen ist. Muss immer eine Authentifizierung der Dienstnutzer mittels EAP erfolgen, wovon in diesem Beispiel ausgegangen werden kann,

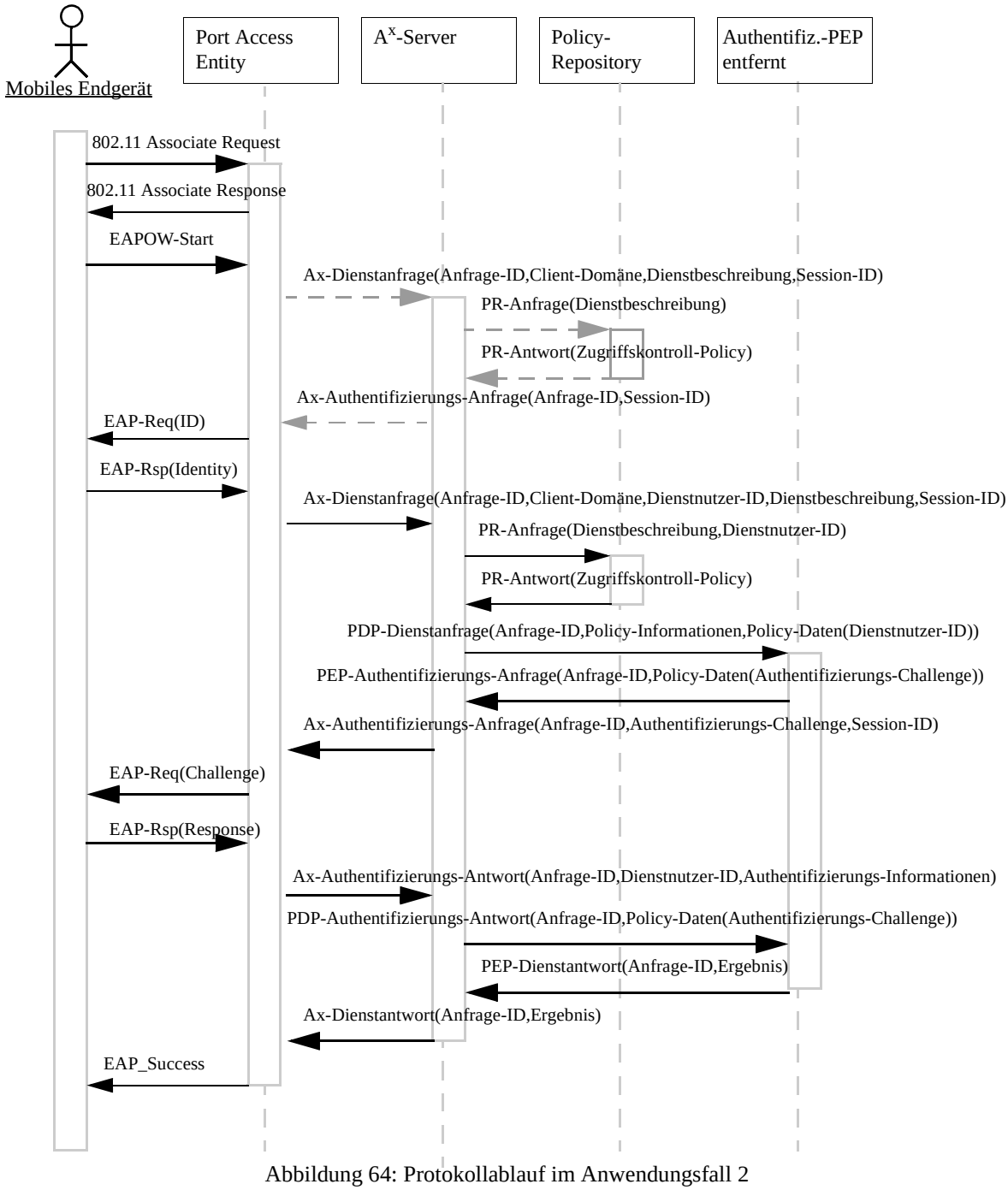

Abbildung 64: Protokollablauf im Anwendungsfall 2

kann sie sofort einen EAP-Request an den Dienstnutzer stellen und nachfolgend in der A^X-Dienstanfrage bereits die Benutzerkennung des Dienstnutzers an den A^X-Server schicken.

Als Authentifizierungs-PEP wird, wie in der Zugriffskontroll-Policy spezifiziert, ein PEP in der Heimatdomäne des Dienstnutzers verwendet. Dieser erhält die PDP-Dienstanfrage. Da ein Challenge Response Verfahren eingesetzt wird, antwortet der Authentifizierungs-PEP mit einer PEP-Authentifizierungs-Anfrage, welche ein Authentifizierungs-Challenge Objekt enthält. Das Challenge wird bis zum Dienstnutzer weitergeleitet wird. Das Response nimmt den gleichen Weg

zurück. Erst dann kann die Authentifizierung erfolgen. Damit beträgt die Antwortzeit in diesem Beispiel:

$$A_{Ax}(IntranetZugang) = 3f_E + 2f_Z + 3f_R + 2f_{EP} \quad (3)$$

Im zweiten exemplarischen Szenario, in welchem die Dienstanfrage durch einen lokalen Dienstnutzer gestellt wird, ist in der RADIUS-Architektur keine Anfrage an den RADIUS-Server in einer fremden Domäne notwendig. Die Authentifizierung kann direkt durch den lokalen RADIUS-Server erfolgen. Damit gilt für die Antwortzeit

$$A_{RADIUS}(IntranetZugangLokal) = 3f_E + 2f_Z + 2f_R \quad (4)$$

Bei Nutzung der A^X-Architektur wird ein lokaler statt eines entfernten PEPs verwendet und dementsprechend gilt:

$$A_{Ax}(IntranetZugangLokal) = 3f_E + 2f_Z + 3f_R + 2f_{LP} \quad (5)$$

Tabelle 13 fasst die Ergebnisse dieser für Internet-Zugangsdienste typischen Beispiele zusammen. Die Differenz besteht zum einen in der zusätzlichen Repository-Anfrage zur Bestimmung der Policy und zum anderen in den Policy-Protokollnachrichten, die an den Authentifizierungs-PEP gestellt werden. Diese beiden Transaktionen existieren in der RADIUS-Architektur nicht, da diese kein policybasiertes Management verwendet. In einem Szenario, in welchem der Dienstnutzer einer fremden Anbieterdomäne angehört, sind die Transaktionen zwischen dem lokalen und dem RADIUS-Server in der Heimatdomäne des Dienstnutzers sowie zwischen dem A^X-Server und dem Authentifizierungs-PEP in der Heimatdomäne des Dienstnutzers vergleichbar. In diesem Fall besteht die Differenz nur in der zusätzlichen Repository-Anfrage.

Anwendungsfall	Vergleichsarchitektur	A^X-Architektur	Differenz
Intranet-Zugang fremder (2.2)	$A_{RADIUS} = 3f_E + 2f_Z + 2f_R + 2f_{EP}$	$A_{Ax} = 3f_E + 2f_Z + 3f_R + 2f_{EP}$	$\Delta = f_R$
Intranet-Zugang lokal	$A_{RADIUS} = 3f_E + 2f_Z + 2f_R$	$A_{Ax} = 3f_E + 2f_Z + 3f_R + 2f_{LP}$	$\Delta = f_R + 2f_{LP}$

Tabelle 13: Vergleich der Antwortzeiten bei Kontrolle von Internet-Zugangsdiensten

Kontrolle des Zugriffs auf QoS-Transportdienste. Bei der Kontrolle des Zugriffs auf Internet-Transportdienste erfolgt die Zugriffskontrolle zumeist ebenfalls durch ein ausgelagertes System, wie in der Integrated Services Architektur beschrieben. Diese Architektur verwendet zudem ein policybasiertes Management. Anwendungsfall 6 steht exemplarisch für diese Klasse von Internet-Diensten.

Bei der Kontrolle der Reservierungsanfrage durch den Backbone-Anbieter in Anwendungsfall 6 wird die Zugriffskontrolle der Integrated Services Architektur verwendet. Der zugehörige Protokollablauf ist in Anhang B.6 dargestellt. Das Video-Konferenzsystem stellt einen RSVP-Reservation-Request an den Backbone-Router. Dieser sendet als Policy Enforcement Point einen COPS-

Request an den Policy Decision Point. Dieser wiederum fragt die Policy aus dem Policy-Repository ab, authentifiziert den Dienstnutzer und verwaltet die Session. Somit werden im Rahmen der Zugriffskontrolle drei Repository-Anfragen gestellt. Die COPS-Nachricht wird zu den Zugriffskontrollnachrichten gezählt, so dass insgesamt gilt:

$$A_{IntServ}(BackboneQoS) = f_E + f_Z + 3f_R \quad (6)$$

Verwendet man in diesem Beispiel die A^x-Architektur, so ist der Ablauf nahezu identisch (vgl. Abbildung 86 in Anhang B.6) mit einer Ausnahme. Die Authentifizierung wird nicht vom PDP selbst vorgenommen, sondern vom einem eigenen Authentifizierungs-PEP. Dieser muss konfiguriert werden und sein Ergebnis zurückmelden. Somit gilt für die Antwortzeit:

$$A_{Ax}(BackboneQoS) = f_E + f_Z + 3f_{LR} + f_{LP} \quad (7)$$

Die Differenz besteht zwischen den beiden Architekturen besteht also nur in einer zusätzlichen lokalen Policy-Protokollnachricht.

Kontrolle des Zugriffs auf Anwendungs- und Inhaltsdienste. Beim Zugriff auf Anwendungs- und Inhaltsdienste verwenden existierende Systeme durchgängig eine direkte Zugriffskontrolle, wie in Kapitel 3.5 (vgl. Tabelle 8 auf Seite 57) gezeigt. Die Zugriffskontrolle unterscheidet sich hinsichtlich der Form der Autorisierung, den verwendeten Authentifizierungs-Protokollen und der Frage, ob eine Prüfung von Nutzerberechtigungen und eine dynamische Autorisierung vorgesehen ist. Der Vergleich der Architektur mit existierenden Systemen für die Kontrolle von Anwendungs- und Inhaltsdiensten erfolgt anhand der exemplarischen Anwendungsfälle 1.3, 3.3, 5.2 und 6.1. Wie Tabelle 15 in Anhang B zeigt, werden dort zur Zugriffskontrolle eine Autorisierung auf Basis einer Authentifizierung mit und ohne Challenge Response Verfahren ebenso verwendet, wie eine berechtigungsnachweis basierte Autorisierung. Es erfolgen statische und dynamische Autorisierungen und Prüfungen von Nutzerberechtigungen oder auch nicht.

In Anwendungsfall 3 muss Herr Grimm bei der Anmeldung am Web Mail Server von WWW-OK seine Benutzerkennung und sein Passwort in ein HTML-Formular eingeben. Die Zugriffskontrolle erfolgt durch den Web Mail Server selbst, der Herrn Grimm authentifiziert und seine Berechtigung prüft, auf das Mail-Verzeichnis zuzugreifen. Dazu sind zwei Repository-Anfragen notwendig. In Anhang B.3 findet sich die Zugriffskontroll-Policy von WWW-OK und der Protokollablauf für die direkte Zugriffskontrolle durch den Web Mail Server. Wird die Zugriffskontrolle mittels der A^x-Architektur realisiert, stellt der Web Mail Server eine Zugriffskontrollanfrage an den A^x-Server. Die Benutzerkennung und das Passwort sendet er als Dienstnutzer-ID und Authentifizierungs-Information in der ersten A^x-Dienstanfrage bereits mit. Aufgrund der Zugriffskontroll-Policy werden ein Nutzerberechtigungs- und ein Authentifizierungs-PEP konfiguriert, wie in Abbildung 65 gezeigt.

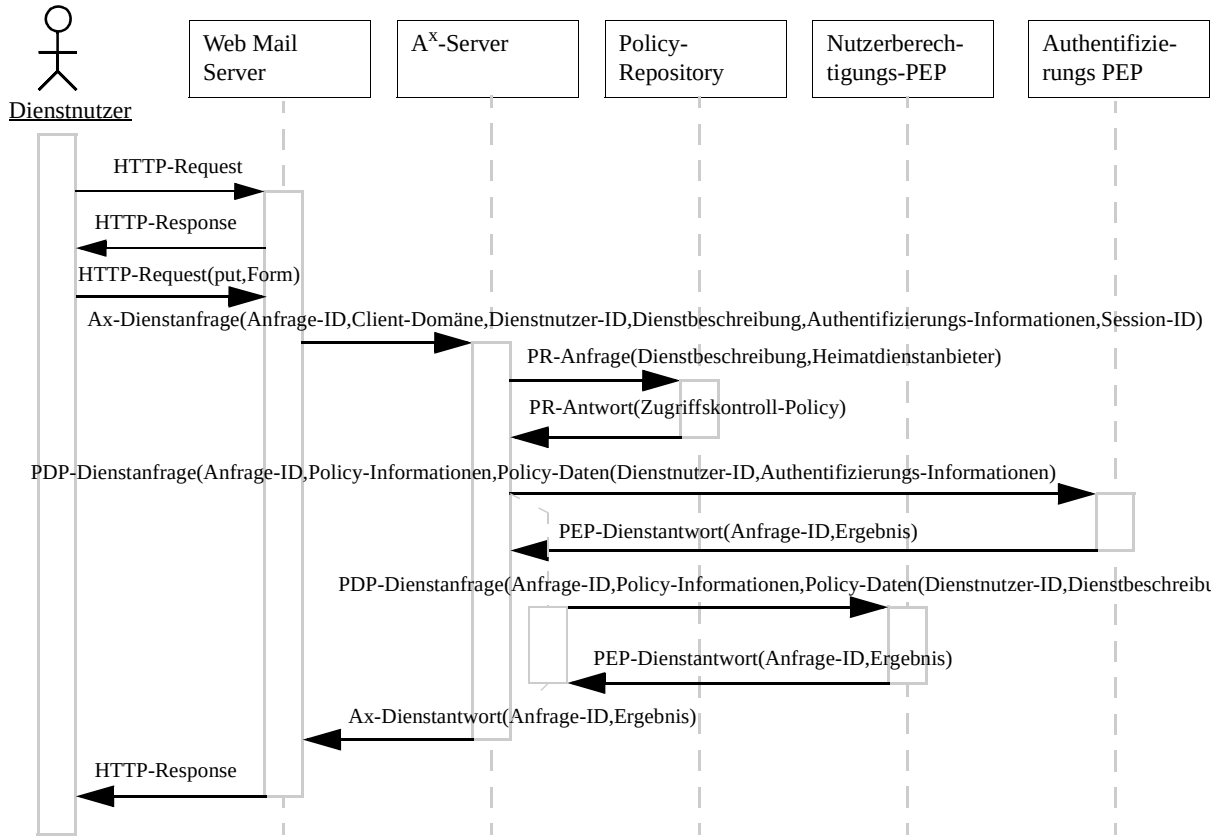

Abbildung 65: Protokollablauf im Anwendungsfall 3.3

Bestimmt man die Transaktionen in beiden alternativen Architekturen, so ergibt sich für diesen Anwendungsfall bei einer direkten Zugriffskontrolle durch den Web Mail Server:

$$A_{direkt}(WebMail) = 2f_E + 2f_R \tag{8}$$

und bei Nutzung der AX-Architektur:

$$A_{Ax}(WebMail) = 2f_E + f_Z + 4f_R + 2f_{LP} \tag{9}$$

Die Anfrage an das Session-Repository sowie Authentifizierungs- und Rechte-Repository sind in Abbildung 65 nicht aufgeführt.

Die Kontrolle des Zugriffs auf die Software *Rechnungswesen Deluxe* in Anwendungsfall 1 gleicht der für den zuvor beschriebenen Anwendungsfall weitgehend. Der Unterschied besteht darin, dass ein Challenge Response Verfahren zur Authentifizierung verwendet wird. Somit sind eine weitere Zugriffskontrollanfrage und eine weitere Transaktion zwischen AX-Server und PEP zum Austausch von Challenge und Response notwendig. Die gesamte Beschreibung des Anwendungsfalls, bestehend aus der Zugriffskontroll-Policy und den Protokolldiagrammen, findet sich in Anhang B.1.

Die Zugriffskontrolle auf das Videokonferenzsystem von *NRW-On* im Anwendungsfall 6 stimmt für einen Geschäftskunden ebenso damit überein (vgl. Anhang B.6). Da *NRW-On* zwei Arten von Kunden unterscheidet, ist neben der Authentifizierung eine dynamische Prüfung not-

wendig, zu welcher Kundengruppe der Kunde gehört. In Anwendungsfall 5 bei einer Zahlung mittels Kreditkarte findet in Abweichung zum zuvor beschriebenen Anwendungsfall 3 keine Authentifizierung und Prüfung der Nutzerberechtigung statt, sondern eine dynamische Prüfung der Gültigkeit der Kreditkarte durch ein externes System.

Über diese vier exemplarischen Anwendungsfälle hinaus sind für einen vollständigen Vergleich noch weitere zu betrachten, in denen andere Kombinationen von Formen der Zugriffskontrolle und eingesetzten Verfahren verwendet werden. Diese wurden, soweit es sich um sinnvolle Kombinationen handelt, vorgenommen, liefern aber keine neuen Erkenntnisse und werden daher hier nicht beschreiben. Tabelle 14 zeigt die Transaktionen für die vier dargestellten Anwendungsfälle im Überblick.

Die Differenz in den Antwortzeiten resultiert immer aus den A^x-Protokollnachrichten, die es bei Verwendung des integrierten Modells nicht gibt, zwei Repository-Anfragen an das Session- und Policy-Repository sowie den lokalen Policy-Transaktionen, die aufgrund des policybasierten Managements notwendig sind.

Anwendungsfall	Vergleichsarchitektur	A^x-Architektur	Differenz
Web-Mailer (3.3)	$A_{direkt} = 2f_E + 2f_R$	$A_{Ax} = 2f_E + f_Z + 4f_R + 2f_{LP}$	$\Delta = f_Z + 2f_R + 2f_{LP}$
Rechnungswesen (1.3)	$A_{direkt} = 2f_E + 2f_R$	$A_{Ax} = 2f_E + 2f_Z + 4f_R + 3f_{LP}$	$\Delta = 2f_Z + 2f_R + 3f_{LP}$
Videokonferenz (6.1)	$A_{direkt} = f_E + 2f_R$	$A_{Ax} = f_E + f_Z + 4f_R + 2f_{LP}$	$\Delta = f_Z + 2f_R + 2f_{LP}$
Download Texte (5.2)	$A_{direkt} = 2f_E + f_{EX} + f_R$	$A_{Ax} = 2f_E + f_Z + f_{EX} + 3f_R + f_{LP}$	$\Delta = f_Z + 2f_R + f_{LP}$

Tabelle 14: Vergleich der Zugriffskontrolle bei Anwendungs-/Inhaltsdiensten

Zusammenfassung. Bei einer Nutzung der im Rahmen der Arbeit entwickelten A^x-Architektur sind im Vergleich zu existierenden Architekturen erwartungsgemäß mehr Transaktionen erforderlich. Damit erhöht sich die Antwortzeit auf eine Dienstanfrage unabhängig davon, mit welchen Faktoren die einzelnen Transaktionsarten gewichtet werden. Dieses resultiert daraus, dass die Zugriffskontrolldienste von den Endnutzerdiensten separiert werden, sie in ihre Teilfunktionen zerlegt werden und das policybasierte Management verwendet wird. Die Differenzen sind bei einem Vergleich mit Systemen zur Kontrolle von Anwendungs- und Inhaltsdiensten am größten, erreicht hier über eine Verwendung der A^x-Architektur aber auch am meisten Flexibilitätszuwachs. Viele verschiedene existierende Systeme, die notwendig sind, um unterschiedliche Formen der Zugriffskontrolle zu realisieren, lassen sich durch einheitliche A^x-Systeme ersetzen. Die A^x-Systeme können zusätzlich auch den Zugriff auf Internet-Zugangsdienste kontrollieren. Die Differenz der Antwortzeit im Vergleich zu einer Verwendung eines RADIUS- oder Diameter-Servers sind relativ gering. Es handelt sich dabei nur um lokal durchzuführende Transaktionen.

Unterstützten RADIUS und Diameter nur eine authentifizierungsbasierte Autorisierung und damit nur eine Grundform der Zugriffskontrolle, kann ein Internet-Dienstanbieter bei Verwendung des A^X-Systems flexibel konfigurieren, welche Form der Prüfung vorzunehmen ist.

6.4 Zusammenfassung

Die A^X-Architektur erlaubt die Kontrolle des Zugriffs auf Internet-Dienste aller Dienstklassen. Mit Ihrer Hilfe lassen sich der Internet-Zugang und die Bereitstellung von Basis- und QoS-Transportdiensten ebenso kontrollieren wie die Nutzung von Anwendungs- und Inhaltsdiensten. Es können nahezu beliebige Geschäftsmodelle der Dienstanbieter berücksichtigt werden. So kann z.B. eine Autorisierung bei der Nutzung privater Dienste ebenso erfolgen wie bei öffentlichen Diensten, in denen die Prüfung der Zahlungsfähigkeit des Dienstnutzers notwendig ist. Auch die Mobilität des Dienstnutzers und die Verwendung verschiedener Zugangstechnologien kann unterstützt werden. Beliebige sicherheitsgewährleistende Funktionen zur Sicherung des Zugriffs auf Internet-Diente können eingesetzt werden. Ein A^X-System bietet einem Internet-Dienstanbieter die notwendige Flexibilität, um auf technologische und ökonomische Änderungen zu reagieren. Um diese Flexibilität zu erreichen, sind nur geringe Einbußen der Leistungsfähigkeit in Form von längeren Antwortzeiten hinzunehmen.

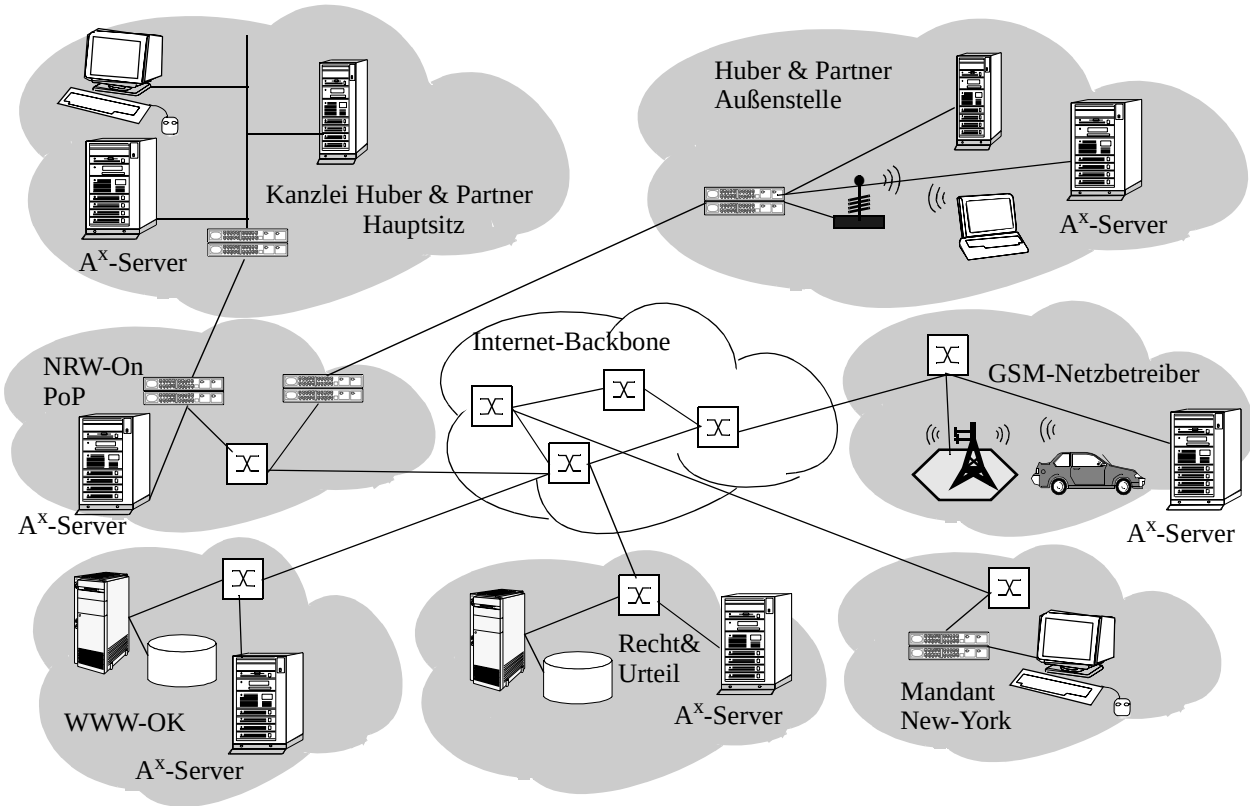

Abbildung 66: Anwendungsszenario mit A^X-Servern

Die Verwendung der A^X-Architektur erlaubt es den Dienstanbietern, eine Vielzahl von bestehenden Systemen zur Zugriffskontrolle und Abrechnung durch einheitliche A^X-Systeme zu ersetzen. Dies gilt auch für das in Kapitel 2.1 vorgestellte realistische Anwendungsszenario. Dazu müssen

in jeder administrativen Domäne der Internet-Dienstanbieter mindesten ein A^x-Server lokalisiert sein oder der Server eines Brokers verwendet werden. Die Anbieter müssen ihre Policies im definierten einheitlichen Format beschreiben. Soll eine Kontrolle des Dienstzugriffs von fremden Dienstnutzern erfolgen, müssen zudem die entsprechenden Vertrauensverhältnisse zwischen den A^x-Server errichtet werden. Ist dies der Fall, wie in Abbildung 66 gezeigt, so können Internet-Dienstanbieter A^x-Systeme zur Kontrolle all ihrer Dienste einsetzen.

Kapitel 7: Zusammenfassung und Ausblick

7.1 Zusammenfassung

Kommerzielle Anbieter von Internet-Diensten müssen in die Lage versetzt werden, über eine Anpassung ihrer Geschäftsmodelle unmittelbar auf Marktsituationen zu reagieren und schnell neue Dienste implementieren zu können. Dazu bedürfen sie u.a. eines flexibel konfigurierbaren generischen Systems, welches zur Kontrolle und Abrechnung verschiedenster Dienste genutzt werden kann. Die in der Arbeit entwickelte A^X-Architektur stellt die Grundlage für ein solches einheitliches System dar.

In der Arbeit wurde untersucht, welche Anforderungen sich aus der Gestaltung eines Geschäftsmodells als übergeordnete Policy eines Dienstanbieters an die Authentifizierung und Autorisierung als Teilfunktionen der Zugriffskontrolle ergeben. Es wurde ein Policy-Modell entwickelt, welches in einer systematischen und einer operativen Sichtweise die Beziehungen zwischen den Aspekten des Geschäftsmodells und den Funktionen der Zugriffskontrolle und Abrechnung beschreibt. Eine eigene Policy-Sprache dient zur Spezifikation der Policies, wie sie zur Konfiguration des Zugriffskontroll- und Abrechnungssystems innerhalb der A^X-Architektur verwendet werden.

Beim Design der Architektur wurden drei existierende Konzepte miteinander kombiniert:

- Zugriffskontrolle und Abrechnung werden als eigene Unterstützungsdienste angesehen und von den Endnutzerdiensten separiert. Sie werden von einem A^X-Server erbracht. Die generische Funktionsweise dieses A^X-Servers ist unabhängig von den zu kontrollierenden Diensten. Somit kann mittels eines A^X-Servers sowohl der Zugriff auf die Zugänge zum Internet als auch der Zugriff auf Inhalte und Anwendungen kontrolliert werden.
- Der Dienstanbieter bestimmt über die Definition einer Policy, welche Form der Zugriffskontrolle und Abrechnung in Abhängigkeit vom Endnutzerdienst auszuführen ist. Im Zugriffskontrollsystem wird das Paradigma des policybasierten Managements umgesetzt.
- Die Funktionen der Zugriffskontrolle und Abrechnung werden modularisiert und innerhalb der Architektur mittels voneinander unabhängiger Policy Enforcement Points ausgeführt. Die strikte Modularisierung erlaubt u.a. einen Austausch der sicherheitsgewährleistenden Verfahren.

Die drei angewandten Konzepte bestimmen das abstrakte Bild der A^X-Architektur und deren Gesamtfunktionalität. Dieses abstrakte Bild wurde im Rahmen der Arbeit konkretisiert. Die einzelnen Komponenten der Architektur und ihre jeweilige Funktionalität wurden definiert und die

Realisierung der wichtigsten Komponenten detailliert betrachtet. Die Systemschnittstellen wurden analysiert und die zwischen den Systemen bzw. Systemkomponenten auszutauschenden Datenobjekte und Nachrichtentypen bestimmt. Eine genaue Beschreibung der Funktionsweise eines A^x-Systems erfolgte mit Hilfe von Nachrichtensequenzdiagrammen. Verschiedene in der A^x-Architektur realisierbare Organisationsmodelle beschreiben die Kontrolle des Zugriffs mobiler Dienstnutzer.

Zur Beurteilung der Architektur wurden verschiedene repräsentative Anwendungsfälle analysiert und in ihnen der Einsatz des A^x-Systems mit dem existierender Systeme verglichen. Derzeit müssen die Dienstanbieter zur Realisierung der Zugriffskontrolle und Abrechnung im zu Beginn der Arbeit vorgestellten realistischen Anwendungsszenario verschiedene Systeme einsetzen. Sie lassen sich alle durch einheitliche A^x-Systeme ersetzen, wobei nur geringe Performanzeinbußen hinzunehmen sind. Auch die Kontrolle und Abrechnung neuer Dienste und die Berücksichtigung verschiedenster Geschäftsmodelle ihrer Anbieter erlauben die A^x-Systeme.

7.2 Ausblick

Die Arbeit legt die konzeptionelle Grundlage für ein generisches A^x-System. Über den argumentativen Nachweis des vollständigen Funktionsumfangs hinaus ist eine Validierung in einem realen Umfeld mit einer Vielzahl von Diensten und Nutzern angestrebt. Der erste Schritt dazu muss in einer Prüfung der Verwendbarkeit und Erweiterbarkeit vorhandener Protokolle wie Diameter oder COPS bestehen.

Die Überprüfung der Leistungsfähigkeit des Systems kann mittels einer Simulation untermauert werden. Ein Dienstanbieter benötigt konkrete Vorschläge zur Lokalisierung und Replikation von A^x-System und deren Komponenten innerhalb seine Domäne und für die Verteilung der A^x-Dienstanfragen. Diese können ebenfalls mit Hilfe einer Simulation bestimmt werden.

Die Arbeit legt ihren Schwerpunkt auf die Zugriffskontrolle anhand kaufmännischer Kriterien und mittels einer Authentifizierung. Bei der Beschreibung der Module zur Überprüfung von Berechtigungsnachweisen wurde von konkreten Realisierungen abstrahiert. Um diese Form der Zugriffskontrolle zu realisieren, sind geeignete Verfahren zur Rechteverwaltung für Internet-Dienstanbieter zu bestimmen und zu integrieren.

Literaturverzeichnis

[AB99] B. Aboba und M. Beadles: The Network Access Identifier. RFC 2486, Internet Engineering Task Force, Januar 1999.

[ACG+00] B. Aboba, P. Calhoun, S. Glass, T. Hiller, P. McCann, H. Shiino, G. Zorn und G. Dommety: Criteria for Evaluating AAA Protocols for Network Access. RFC 2989, Internet Engineering Task Force, November 2000.

[ACZ01] J. Arkko, P. Calhoun und G. Zorn: Diameter Accounting Extensions. Internet Draft draft-ietf-aaa-diameter-accounting-00.txt, Internet Engineering Task Force, Februar 2001. Work in progress.

[AM03] B. Aboba und D. Mitton: Internet Engineering Task Force - Authentication, Authorization and Accounting (aaa) Working Group, Januar 2003. http://www.ietf.org/html.charters/aaa-charter.html.

[ATA99] K. Al-Tawil und A. Akrami: A new authentication protocol for roaming users in GSM networks, In: IEEE International Symposium on Computers and Communications, Seite 93–99. IEEE, 1999.

[Aut03] Internet Assigned Number Authority: Port numbers, Mai 2003. http://www.iana.org/assignments/port-numbers.

[BB00] N. Brownlee und A. Blount: Accounting Attributes and Record Formats. RFC 2924, Internet Engineering Task Force, September 2000.

[Bir02] B. Birkhofer: Ertragsmodelle - Einnahme und Erlösquellen im innovativen Absatzkanal des Electonic Commerce, In: M. Schögel, T. Tomaczak und C. Belz (Hrsg.), Roadm@p to E-Business, Seite 430 – 452. Thexis, St. Gallen, 2002.

[BLFM98] T. Berners-Lee, R. Fielding und L. Masinte:. Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396, Internet Engineering Task Force, August 1998.

[BM99] J. Buchmann und M. Maurer: Wie sicher ist die Public-Key-Kryptographie. Technical Report TI-2/99, TU Darmstadt, http://www.informatik.tu-darmstadt.de/ftp/pub/TI/TR/TI-99-02.PublicKeyKrypto.ps.gz, Januar 1999.

[BMR99] N. Brownlee, C. B. Mills und G. Ruth: Traffic Flow Measurement: Architecture. RFC 2722, Internet Engineering Task Force, Oktober 1999.

[BP01] R. Boutaba und A. Polrakis: Towards Extensible Policy Enforcement Points, In: International Workshop POLICY 2001, Bristol, UK, Lecture Notes on Computer Science 1995, Seite 247–261. Springer-Verlag, Berlin, Heidelberg, Januar 2001.

[BPS+00] T. Bray, J. Paoli, C. M. Sperberg-McQueen und E. Maler (Hrsg.): Extensible Markup Language (XML) 1.0 (Second Edition). Technical report, W3C, http://www.w3.org/TR/2000/REC-xml-20001006, Oktober 2000.

[BS02] B. Birkhofer und M. Schögel: Ansatz, Gestaltung und Umsetzung marktorientierter Geschäftsmodelle im Electronic Commerce, In: Somm-Tomczak Albers, Haßmann (Hrsg.), Digitale Fachbibliothek Verkauf. Symposion Publishing GmbH, Düsseldorf, 2002.

[Buc01] J. Buchmann: Einführung in die Kryptographie. Springer, Berlin Heidelberg, 2001. 2. Auflage.

[BV98] L. Blunk und J. Vollbrecht: PPP Extensible Authentication Protocol (EAP). RFC 2284, Internet Engineering Task Force, März 1998.

[BW01] A. Barua und A. Whinston: Measuring the Internet Economy. Technical report, internetindicators.com, Januar 2001. http://www.internetindicators.com.

[BWY00] A. Barua, A. Whinston und F. Yin: Value and Productivity in the Internet Economy. IEEE Computer, 33(5):102–105, Mai 2000.

[BZB+97] B. Braden, L. Zhang, S. Berson, S. Herzog und S. Jamin: Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205, Internet Engineering Task Force, September 1997.

[CFB02] P. Calhoun, S. Farrell und W. Bulley: Diameter CMS Security Application. Internet Draft draft-ietf-aaa-diameter-cms-sec-04.txt, Internet Engineering Task Force, März 2002. Work in progress.

[CG92] J. Quisquater C. Guillou, M. Ugon: The Smart Card: A Standardized Security Device Dedicated to Public Crytology, In: G. J. Simmons (Hrsg.), Contemorary Cryptology. The Science of Information Integrity, chapter 12, Seite 561–613. IEEE Press, 1992.

[Cha83] D. Chaum: Blind signatures for untraecable payments, In: Advances in Cryptology - Crypto 82 - Santa Barbara, California, USA, Seite 199–203. International Association for Cryptologic Research, Plemum, New York, 1983.

[CJP03] P. Calhoun, T. Johansson und C. E. Perkins: DIAMETER Mobile IP Extensions. Internet Draft draft-ietf-aaa-diameter-mobileip-14.txt, Internet Engineering Task Force, April 2003. Work in progress.

[CLG+02] P. Calhoun, J. Loughney, E. Guttman, G. Zorn und J. Arkko: DIAMETER Base Protocol. Internet Draft draft-ietf-aaa-diameter-17.txt, Internet Engineering Task Force, Dezember 2002. Work in progress.

[Cot02a] S. Cotton (Hrsg.): Network Data Management - Usage (NDM-U) For IP-Based Services - Service Specification Voice over IP (VoIP) Version 3.1.-A.0.2. Technical report, IPDR, August 2002. http://www.ipdr.org/service_specs/VoIP/VoIP3.1-A.0.2.pdf.

[Cot02b] S. Cotton (Hrsg.): Network Data Management - Usage (NDM-U) For IP-Based Services - Service Specification E-mail Version 3.0-A.0. Technical report, IPDR, Februar 2002. http://www.ipdr.org/service_specs/eMail/eMail3.0-A.0.pdf.

[Cot02c] S. Cotton (Hrsg.): Network Data Management - Usage (NDM-U) For IP-Based Services Version 3.1.1. Technical report, IPDR, Oktober 2002. http://www.ipdr.org/documents/NDM-U_3.1.1.pdf.

[CP00] P. Calhoun und C. E. Perkins: Mobile IP Network Access Identifier Extension for IPv4. RFC 2794, Internet Engineering Task Force, März 2000.

[CSD+01] K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer und R. Yavatkar: COPS Usage for Policy Provisioning (COPS-PR). RFC 3084, Internet Engineering Task Force, März 2001.

[CSW97] S. Choi, D. O. Stahl und A. B. Whinston: The Economics of Electronic Commerce. Macmillan Technical Publishing, Indianapolis, 1997.

[CW00] S. Choi und A. B. Whinston: The Internet Economy - Technology and Practice. Smart Econ Publishing, Austin, Texas, 2000.

[cXM02] cXML.org: cXML Users Guide Version 1.2.008. Technical report, cxml.org, November 2002. http://xml.cxml.org/current/cXML.zip.

[CZSM03] P. Calhoun, G. Zorn, D. Spence und D. Mitton: Diameter Network Access Server Application. Internet Draft draft-ietf-aaa-diameter-nasreq-11.txt, Internet Engineering Task Force, Februar 2003. Work in progress.

[DA99] T. Dierks und C. Allen:. The TLS Protocol Version 1.0. RFC 2246, Internet Engineering Task Force, Januar 1999.

[DA01] R. Droms und W. Arbaugh: Authentication for DHCP Messages. RFC 3118, Internet Engineering Task Force, Juni 2001.

[DBC+00] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan und A. Sastry: The COPS (Common Open Policy Service) Protocol. RFC 2748, Internet Engineering Task Force, Januar 2000.

[DJK+02] J.-L Dugelay, J.-C. Junqua, C. Kotropoulos, R. Kuhn, F. Perronnin und I. Pitas: Recent advances in biometric person authentication, In: International Conference

on Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02), Seite 4060–4063. IEEE, 2002.

[dLGG+00] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht und D. Spence: Generic AAA Architecture. RFC 2903, Internet Engineering Task Force, August 2000.

[Dos90] M. Dose (Hrsg.): Duden Fremwörterbuch. Dudenverlag, Mannheim, 1990. 5., neu bearb. u. erw. Auflage.

[Eas99] D. Eastlake: Domain Name System Security Extensions. RFC 2535, Internet Engineering Task Force, März 1999.

[Eck98] C. Eckert: Sichere, verteilte Systeme - Konzepte, Modelle und Systemarchitekturen. Habilitationsschrift, Technische Universität München, 1998.

[EFL+99] C. Ellison, B. Frantz, B. W. Lampson, R. Rivest, B.Thomas und T. Ylonen: SPKI certificate theory. RFC 2693, Internet Engineering Task Force, September 1999.

[EN01] B. Eisenberg und D. Nickull: ebXML Technical Architecture Specification v1.0.4. Technical report, ebXML.org, Februar 2001. http://www.ebxml.org/specs/ebTA.pdf.

[Fal01] D. C. Fallside (Hrsg.): XML Schema Part 0: Primer . Technical report, W3C, Mai 2001. http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/.

[Fan97] C. Fancher: In your pocket: smartcards. IEEE Spectrum, 34(2):47–53, Februar 1997.

[fc03] fun communications: InternetPayment - GeldKarte, 2003. http://www.fun.de/deutsch/produkte/internetpayment/SmartPay.htm.

[FGM+99] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. J. Leach und T. Berners-Lee: Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, Internet Engineering Task Force, Juni 1999.

[FHBH+99] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen und L. Stewart: HTTP Authentication: Basic and Digest Access Authentication. RFC 2617, Internet Engineering Task Force, Juni 1999.

[FIA03] Firstgate Internet AG: Firstgate click&buy, 2003. http://www.firstgate.de.

[FMB01] L. Fiege, G. Mühl und A. Buchmann: An Architectural Framework for Electronic Commerce Applications, In: Informatik 2001: Wirtschaft und Wissenschaft in der Network Economy - Visionen und Wirklichkeit, Tagungsband der GI/OCG-Jahrestagung Band 2, Wien, Österreich, Seite 928–938. GI/OCG, September 2001.

[FRR00] S. Fischer, C. Rensing und U. Roedig: Open Internet Security - Von den Grundlagen zu den Anwendungen. Springer Verlag, Heidelberg, 2000.

[GA03] J. Gordijn und H. Akkermans: Does e-Business Modeling Really Help? In Proceedings of the 36th Hawaii International Conference On System Sciences, Seite 185 –194. IEEE, Januar 2003.

[GAvV00] J. Gordijn, H. Akkermans und H. van Vliet: What s in an electronic business model? In Knowledge Engineering and Knowledge Management - Methods, Models, and Tools - 12th International Conference, EKAW 2000, Juan-les-Prins, France, Lecture Notes in Computer Science 1937, Seite 257–273. Springer Verlag, Berlin, Heidelberg, Oktober 2000.

[GHJP00] S. Glass, T. Hiller, S. Jacobs und C. E. Perkins: Mobile IP Authentication, Authorization, and Accounting Requirements. RFC 2977, Internet Engineering Task Force, Oktober 2000.

[GMAG95] S. Glassman, M. Manasse, M. Abadi und P. Gauthier: The Millicent Protocol for Inexpensive Electronic Commerce, In: World Wide Web Journal: The Fourth International WWW Conference Proceedings, Boston, Massachusetts, USA, Seite 603–618. O'Reilly & Associates, Sebastopol, Cambridge, Dezember 1995.

[GPP03] 3rd Generation Partnership Project: Ip multimedia subsystem (ims). Technical Specification 3GPP TS 23.228 V5.8.0 (2003-03), März 2003. http://www.3gpp.org/ftp/Specs/archive/23_series/23.228/23228-580.zip.

[Gra92] R. B. Grady: Practical Software Metrics for Project Management and Process Improvement. Prentice-Hall, New York, 1992.

[Gri92] G. Grill (Hrsg.): Meyers grosses Taschenlexikon Band 5. BI Taschenbuchverlag, Mannheim, 1992. 4. vollst. überarbeitete Auflage.

[HJZS01] Hasan, J. Jähnert, S. Zander und B. Stiller: Authentication, authorization, accounting, and charging for the mobile internet, In: IST Mobile Communications Summmit 2001, Sitges (Barcelona) Spanien, Seite 923–928. IST, September 2001.

[HM02] J. L. Hodges und R. Morgan: Lightweight Directory Access Protocol (v3): Technical Specification. RFC 3377, Internet Engineering Task Force, September 2002.

[Hou02] R. Housley: Cryptographic Message Syntax (CMS). RFC 3369, Internet Engineering Task Force, August 2002.

[HPW02] D. Harrington, R. Presuhn und B. Wijnen: An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. RFC 3411, Internet Engineering Task Force, Dezember 2002.

[HW03] J. Hodges und T. Wason: Liberty Architecture Overview Version 1.1. Technical report, Liberty Alliance Project, Januar 2003. http://www.projectliberty.org/specs/liberty-architecture-overview-v1.1.pdf.

[HZ03] T. Hiller und G. Zorn: Diameter Extensible Authentication Protocol (EAP) Application. Internet Draft draft-ietf-aaa-eap-01.txt, Internet Engineering Task Force, März 2003. Work in progress.

[IBM01] IBM: Secure single-entry access to your computing resources. Technical report, November 2001. ftp://ftp.software.ibm.com/software/tivoli/whitepapers/gso.pdf.

[IEE99] IEEE: Wireless LAN Medium Access Control (MAC) Sublayer and Physical Layer Specifications. Standard IEEE 802.11, 1999.

[IEE01] IEEE: Local and metropolitan area networks Port-Based Network Access Control. Standard IEEE 802.1x, 2001.

[IEE02] IEEE: CSMA/CD Access Method and Physical Layer Specifications. Standard IEEE 802.3, 2002.

[ISO94] ISO: Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model, 2nd Edition. Standard ISO/IEC 7498-1, November 1994.

[ITU91] International Telecommunication Union: X.800 Security Architecture for Open Systems Interconnection for CCITT Applications. ITU-T Recommendation X.800, 1991.

[ITU95a] International Telecommunication Union: Information Technology - Open Systems Interconnection - Systems Management: Usage Metering Function for Accounting Purposes. ITU-T Recommendation X.742, 1995.

[ITU95b] International Telecommunication Union: X.811 Information Technology - Open Systems Interconnection - Security Framework for Open Systems: Authentication Framework. ITU-T Recommendation X.811, 1995.

[ITU97] International Telecommunication Union: X.509 The Directory: Authentication Framework. ITU-T Recommendation X. 509, 1997.

[ITU98] International Telecommunication Union: Specification of TMN applications at the Q3 interface: Call detail recording. ITU-T Recommendation ITU-T Q.825, 1998.

[Jäg02] T. Jäger: Single Sign-On Software - Einführung starker Authentisierungsmechanismen, In: P. Horster (Hrsg.), Enterprise Security - Grundalgen, Strategien, Anwendungen, Realisierungen, Seite 111–119. it Verlag, Höhenkirchen, 2002.

[KA98] S. Kent und R. Atkinson: Security Architecture for the Internet Protocol. RFC 2401, Internet Engineering Task Force, November 1998.

[Kan91] B. Kantor: BSD rlogin. RFC 1258, Internet Engineering Task Force, September 1991.

[KB01] W. Krüger und N. Bach: Geschäftsmodelle und Wettbewerb im e-Business, In: W. Buchholz (Hrsg.), Suppy Chain Solutions - Best Practices im E-Business, Seite 29–51. Schaeffer-Poeschl, 2001.

[KM00] D. Kristol und L. Montulli: HTTP State Management Mechanism. RFC 2965, Internet Engineering Task Force, Oktober 2000.

[KN93] J. Kohl und C. Neuman: The Kerberos Network Authentication Service (V5). RFC 1510, Internet Engineering Task Force, September 1993.

[KPP94] P. Kajiser, T. Parker und D. Pinkas: SESAME: The Solution To Security for Open Distributed Systems. Computer Communications, 17(7):501–518, 1994.

[Kra00] H. Krafft: Bestandsaufnahme und Entwicklungsperspektiven der Internet-Gründungslandschaft in Deutschland, Discussion paper. Technical report, European Business School, Oestrich-Winkel, Oktober 2000. http://www.e-startup.org/download/grd11_00.pdf.

[KSSW00] M. Karsten, J. Schmitt, B. Stiller und L. Wolf: Charging for Packet-switched Network Communication - Motivation and Overview. Computer Communications, 23(3):290–302, Februar 2000.

[LMS99] D. Levi, P. Meyer und B. Stewart: SNMP Applications. RFC 2573, Internet Engineering Task Force, April 1999.

[LOP01] S. Ben Lagha, A. Osterwalder und Y. Pigneur: Modelling e-Business with eBML, In: 5e Conference International de Management des Reseaux d Entreprises (CIMRE), Mahadia, Tunesien, Oktober 2001.

[LS92] B. Lloyd und W. Simpson: PPP authentication protocols. RFC 1334, Internet Engineering Task Force, Oktober 1992.

[MBHS00] H. Mahon, Y. Bernet, S. Herzog und J. Schnizlein: Requirements for a Policy Management System. Internet Draft draft-ietf-policy-req-02.txt, Internet Engineering Task Force, November 2000. Work in progress.

[MDK03] Moby Dick Konsortium: Moby Dick - Mobility and Differentiated Services in a Future IP Network IST-2000-25394, 2003. http://www-int.berkom.de/ mobydick/.

[Met00] L. B. Methlie: A Business Model for Electronic Commerce. Telektronik, 96(2):8–19, 2000.

[Mic88] Sun Microsystems: RPC: remote procedure call protocol specification: Version 2. RFC 1057, Internet Engineering Task Force, Juni 1988.

[MJB+01] D. Mitton, M. St. Johns, S. Barkley, D. L. Nelson, B. Patil, M. L. Stevens und B. Wolff: Authentication, Authorization, and Accounting: Protocol Evaluation. RFC 3127, Internet Engineering Task Force, Juni 2001.

[Möl99] B. Möller: Benutzer berwachte Erzeugung von DSA-Schl sseln in Chipkarten, In: P. Horster (Hrsg.), Sicherheitsinfrastrukturen - Grundlagen, Realisierungen, Rechtliche Aspekte, Anwendungen, Seite 238–246. Vieweg, Wiesbaden, 1999.

[MS02] A. Meyer und M. Specht: Pioniervorteile für Anbieter von Informationsgütern im Electronic-Commerce, In: M. Schögel, T. Tomaczak und C. Belz (Hrsg.), Roadmap to E-Business, Seite 244– 268. Thexis, St. Gallen, 2002.

[MS02b] Microsoft Corporation: Microsoft .NET Passport Review Gudie. Technical report, November 2002. http://www.microsoft.com/net/downloads/passport_review_guide.doc.

[MS03] Microsoft Corporation: Microsoft Biztalk Server. http://www.microsoft.com/biztalk/, 2003.

[Mül02] M. Müller: Zahlungssysteme im E-Commerce, In: M. Schögel, T. Tomaczak und C. Belz (Hrsg.), Roadm@p to E-Business, Seite 848–865. Thexis, St. Gallen, 2002.

[Mye94] J. Myers: POP3 AUTHentication command. RFC 1734, Internet Engineering Task Force, Dezember 1994.

[Net03] Netegrity Inc: Siteminder: Overview, November 2003. http://www.netegrity.com/products/products.cfm?page=SMoverview.

[Nie94] J. Nielsen: Usability Engineering. Academic Press, Boston, 1994.

[OAS03] Organization for the Advancement of Structured Information Standards: OASIS - Who we are - Mission, 2003. http://www.oasis-open.org/who/.

[OP02] A. Osterwalder und Y. Pigneur: An e-Business Model Ontology for Modeling e-Business, In: 15th Bled Electronic Commerce Conference - e-Reality: Constructing the e-Economy, Bled, Slovenia, Juni 2002.

[OPR02] M. Ortega, M. A. Perez und T. Rojas: A Systemic Quality Model for Evaluating Software Products, In: 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, Florida, USA. International Institute of Informatics and Systemics, Juli 2002.

[pA02] paybox.net AG: Mobile Payment Delivery Made Simple, 2002. http://www.paybox.net.

[Per96] C. Perkins: IP Mobility Support. RFC 2002, Internet Engineering Task Force, Oktober 1996.

[Per00] C. Perkins: Mobile IP Joins Forces with AAA. IEEE Personal Communications, 7(4):59–61, August 2000.

[PR83] J. Postel und J. Reynolds: Telnet Protocol Specification. RFC 854, Internet Engineering Task Force, Mai 1983.

[PR85] J. Postel und J. Reynolds: File Transfer Protocol. RFC 959, Internet Engineering Task Force, Oktober 1985.

[RHKS01] C. Rensing, Hasan, M. Karsten und B. Stiller: A Survey on AAA Mechanisms, Protocols, and Architectures and a Policy-based Approach beyond: Ax. Technical Report 111, Eidgenössische Technische Hochschule - Institut für Technische Informatik und Kommunikationsnetze, Mai 2001. ftp://ftp.kom.e-technik.tu-darmstadt.de/pub/papers/RHKS01-1-paper.pdf.

[RHKS02] C. Rensing, Hasan, M. Karsten und B. Stiller: AAA: A Survey and a Policy-Based Architecture and Framework. IEEE Network, 16(6):22–27, November 2002.

[Rig00] C. Rigney: RADIUS Accounting. RFC 2866, Internet Engineering Task Force, Juni 2000.

[RLS99] P. Reichl, S. Leinen und B. Stiller: A Pracitcal review of Pricing and Cost Recovery for Internet Services, In: IEW'99: 2nd Internet Economics Workshop, Berlin, Berlin, Mai 1999.

[Roe02] U. Roedig: Firewall-Architekturen für Multimedia-Applikationen. Dissertation, Technische Universität Darmstadt, 2002.

[RSC+02] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley und E. Schooler: SIP: Session Initiation Protocol. RFC 3261, Internet Engineering Task Force, Juni 2002.

[RWRS00] C. Rigney, S. Willens, A. Rubens und W. Simpson: Remote Authentication Dial In User Service (RADIUS). RFC 2865, Internet Engineering Task Force, Juni 2000.

[SBJT02] M. Schögel, B. Birkhofer, M. Jazbec und T. Tomaczak: Roadm@p to E-Business - Eine Methode für den erfolgreichen Umgang mit Technologien in der marktorientierten Unternehmensführung, In: M. Schögel, T. Tomaczak und C. Belz (Hrsg.), Roadm@p to E-Business, Seite 16–67. Thexis, St. Gallen, 2002.

[Sch98] K. Schmeh: Safer Net: Kryptographie im Internet und Intranet. dpunkt-Verlag, Heidelberg, 1998.

[Sch02] K. Schögel: Bezugsrahmen der Geschäftsmodellierung, In: M. Schögel, T. Tomaczak und C. Belz (Hrsg.), Roadm@p to E-Business, Seite 374–399. Thexis, St. Gallen, 2002.

[SFPW98] B. Stiller, G. Fankhauser, B. Plattner und N. Weiler: Charging and Accounting for Integrated Internet Services - State off the Art, Problems, and Trends, In: INET'98: The Internet Summit, Genf, Schweiz. Internet Society, Juli 1998.

[Sim94] W. Simpson: The Point-to-Point Protocol (PPP). RFC 1661, Internet Engineering Task Force, Juli 1994.

[Sim96] W. Simpson: PPP Challenge Handshake Authentication Protocol (CHAP). RFC 1994, Internet Engineering Task Force, August 1996.

[Slo94] M. Sloman: Policy Driven Management For Distributed Systems. Plenum Press Journal of Network and Systems Management, 2(4):333–336, Dezember 1994.

[SRL98] H. Schulzrinne, A. Rao und R. Lanphier: Real Time Streaming Protocol (RTSP). RFC 2326, Internet Engineering Task Force, April 1998.

[SS02] B. Skiera und M. Spann: Preisdifferenzierung im Internet, In: M. Schögel, T. Tomaczak und Ch. Belz (Hrsg.), Roadmap to E-Business, Seite 270–284. Thexis, St. Gallen, 2002.

[SSGS00] D. Scheuermann, S. Schwiderski-Grosche und B. Struif: Usability of Biometrics in Relation to Electronic Signatures; EU-Studie; GMD Report Nr. 118. Technical Report 118, GMD, Juli 2000.

[STB02] M. Schögel, T. Tomaczak und C. Belz: Roadm@p to E-Business - Wie Unternehmen das Internet für erfolgreiche Geschäfte nutzen - Vorwort , In: M. Schögel, T. Tomaczak und C. Belz (Hrsg.), Roadm@p to E-Business, Seite 10–13. Thexis, St. Gallen, 2002.

[Ste00] R. Steinmetz: Multimedia-Technologie: Grundlagen, Komponenten und Systeme. Springer Verlag, 2000. 3. Auflage (erstmalig mit CD).

[SX01] G. Stone und G. Xie: Network Policy Languages: A Survey and a New Approach. IEEE Network, 15(1):10–21, Januar 2001.

[Tan02] A. S. Tanenbaum: Computer Networks. Prentice Hall PTR, 2002. 4. Auflage.

[TSdL03] A. Taal, G. Sliepen und C. de Laat: A grammar for Policies in a Generic AAA Environment. Internet Draft draft-irtf-aaaarch-generic-policy-03.txt, Internet Engineering Task Force, Februar 2003. Work in progress.

[VCF+00] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat und M. Holdrege: AAA Authorization Framework. RFC 2904, Internet Engineering Task Force, August 2000.

[VdL01] J. Vollbrecht und C. de Laat: Internet Research Task Force - Authentication Authorisation Accounting ARCHitecture Research Group (AAAARCH). URL http://www.irtf.org/charters/aaaarch.html, Mai 2001.

[VS00] A. Vaha-Sipila: URLs for Telephone Calls. RFC 2806, Internet Engineering Task Force, April 2000.

[VVK00] U. Varshney, R.J. Vetter und R. Kalakota: Mobile commerce: a new frontier. IEEE Computer, 33(10):32–38, Januar 2000.

[WL92] T. Woo und S. Lam: Authentication for Distributed Systems. Computer, 25(1):39–52, Januar 1992.

[XML03] XML.ORG: About XML.org, 2003. http://www.xml.org/xml/aboutxml.shtml.

[YPG00] R. Yavatkar, D. E. Pendarakis und R. Guerin: A Framework for Policy-based Admission Control. RFC 2753, Internet Engineering Task Force, Januar 2000.

[ZZC02] T. Zseby, S. Zander und C. Carle: Policy-Based Accounting. RFC 3334, Internet Engineering Task Force, Oktober 2002.

Abkürzungen

AAA	Authentication, Authorization and Accounting
AFS	Advanced File System
API	Application Programming Interface
ASM	Application-Specific Module
AVP	Attribute Value Pair
A^x	Authentication, Authorization, Metering, Accounting, Charging, Auditing, Billing and Payment
CDR	Call Detail Records
CHAP	Challenge Handshake Authentication Protocol
CMS	Cryptographic Message Syntax
COPS	Common Open Policy Service
cXML	Commerce XML
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DSL	Digital Subscriber Line
DTD	Document Type Definition
EAP	Extensible Authentication Protocol
eBML	E-Business Modelling Language
ebXML	Electronic Business XML
FTP	File Transfer Protocol
GSM	Global System for Mobile Communication
HLR	Home Location Register
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Number Authority

ID	Identity
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMSI	International Mobile Subscriber Identity
IP	Internet Prococol
IPDR	Internet Protocol Data Records
IPSec	IP Security
IRTF	Internet Research Task Force
ISDN	Integrated Services Digital Network
ISO	International Standardisation Organisation
ITU	International Telecommunication Union
LAN	Local Area Network
LDAP	Light-weight Directory Access Protocol
MAC	Medium Access Control
NAI	Network Access Identifier
NAS	Network Access Server
OASIS	Organization for the Advancement of Structured Information Standards
OSI	Open Systems Interconnection
PAE	Port Access Entity
PAP	Password Authentication Protocol
PDA	Personal Digital Assistent
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PKI	Public Key Infrastructure
PIN	Persönliche Identifikations Nummer
POP	Post Office Protocol
PPP	Point-to-Point Protocol
PR	Policy Repository
PSTN	Public Switched Telephone Netowrk
QoS	Quality-of-Service
RADIUS	Remote Authentication Dial In User Service
RBE	Rule based Engine
rlogin	Remote Login

RPC	Remote Procedure Call
RSVP	Resource ReSerVation Protocol
RTFM	Realtime Traffic Flow Management
RTSP	Real Time Streaming Protokoll
SCTP	Stream Control Transmission Protocol
SE	Service Equipment
SESAME	Secure European Systems for Applications in a Multivendor Environment
SET	Secure Electronic Transaction
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
SPKI	Simple Public Key Infrastucture
SPI	Security Parameter Index
SSH	Secure Shell
SSL	Secure Socket Layer
SSO	Single-Sign-On
TAN	Transaktions Nummer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UHO	User Home Organization
URI	Uniform Ressource Identifier
URL	Uniform Ressource Locator
VLR	Visitor Location Register
VoIP	Voice over IP
WAN	Wide Area Network
WWW	World Wide Web
XML	eXtensible Markup Language

Anhang A: Eigene Veröffentlichungen

Bücher

Stephan Fischer, Christoph Rensing und Utz Roedig. *Open Internet Security - Von den Grundlagen zu den Anwendungen*. Springer Verlag, Heidelberg, Januar 2000. ISBN 3-540-66814-4.

Christoph Rensing, Susanne Offenbartl und Jan Hansen. *Entwicklung und Einsatz elektronischer Medien als Lehr- und Lernmittel an hessischen Hochschulen*. Hessisches Ministerium für Wirtschaft, Verkehr und Landesentwicklung, Wiesbaden, März 2001. ISBN 3-933732-27-1

Zeitschriften

Christoph Rensing, Hasan, Martin Karsten und Burkhard Stiller. AAA: A Survey and a Policy-Based Architecture and Framework. *IEEE Network*, 16(6):22-27, November 2002.

Patente

Utz Roedig, Ralf Ackermann, Christoph Rensing, Dieter Rohrdrommel, Juergen Schlesinger, and Ralf Steinmetz. Distributed Firewall for Multimedia Applications. Patent Registration EP00113530, Juni 2000.

Konferenzbeiträge

Christoph Rensing, Ralf Ackermann, Utz Roedig, Lars Wolf und Ralf Steinmetz. Sicherheitsunterstützung für Internet Telefonie. In P. Horster (Hrsg.) *Sicherheitsinfrastrukturen - Grundlagen, Realisierungen, Rechtliche Aspekte, Anwendungen*, DuD-Fachbeiträge, Seite 285-296, Vieweg, Wiesbaden, März 1999.

Christoph Rensing, Utz Roedig, Ralf Ackermann, Lars Wolf und Ralf Steinmetz. VDMFA, eine verteilte dynamische Firewallarchitektur für Multimedia-Dienste. In R. Steinmetz (Hrsg.) *Kommunikation in verteilten Systemen (KiVS)*, Darmstadt, Informatik aktuell, Seite 144-157, Springer, Heidelberg, März 1999.

Ralf Ackermann, Christoph Rensing, Stephan Noll-Hussong, Lars Wolf und Ralf Steinmetz. SSS4it - Secure Session Setup für Internet Telefonie. In *Systemsicherheit 2000*, Bremen, Seite 140-150, Vieweg, Wiesbaden, März 2000.

Christoph Rensing, Utz Roedig, Ralf Ackermann und Ralf Steinmetz. A Survey of Requirements and Standardization Efforts for IP-Telephony-Security. In *Workshop "Sicherheit in Netzen und Medienströmen"*, Berlin, Informatik aktuell, Seite 50-60, Springer Heidelberg, September 2000.

Utz Roedig, Ralf Ackermann, Christoph Rensing und Ralf Steinmetz. A Distributed Firewall for Multimedia Applications. In Workshop *"Sicherheit in Netzen und Medienströmen"*, Berlin, Informatik aktuell, Seite 3-16, Springer Heidelberg, September 2000.

Christoph Rensing, Martin Karsten und Ralf Steinmetz. Darstellung der Sicherheitsattribute von Kommunikationsbeziehungen mittels UML. In *Enterprise Security* 2002, Seite 132-143, itVerlag, Hoehenkirchen, März 2002.

Sonstiges

Christoph Rensing. Hacker Praktikum. In *Projektveranstaltungen im Studium an der TUD*, Seite 177-184, TUD Schriftenreihe Wissenschaft und Technik 82, November 2001, ISBN 3-88607-130-8

Christoph Rensing, Hasan, Martin Karsten und Burkhard Stiller. A Survey on AAA Mechanisms, Protocols, and Architectures and a Policy-based Approach beyond: Ax. Technical Report 111, Eidgenössische Technische Hochschule - Institut für Technische Informatik und Kommunikationsnetze, Mai 2001, ftp://ftp.kom.e-technik.tu-darmstadt.de/pub/papers/RHKS01-1-paper.pdf.

Anhang B: Darstellung der Anwendungsfälle

Für jeden einzelnen Anwendungsfall werden in diesem Anhang das Internet-Dienstmodell, der Protokollablauf für die Realisierung der Zugriffskontrolle mit existierenden Systemen, wie in Kapitel 3 beschrieben, die Zugriffskontroll-Policy, wie in Kapitel 4.4.1 definiert, und der Protokollablauf für die Realisierung der Zugriffskontrolle mit der A^x-Architektur vorgestellt. Tabelle 15 zeigt einen Überblick über die in den Anwendungsfällen genutzten und kontrollierten Dienste. In den Anwendungsfällen 4 bis 6 wird auf die Darstellung der Verbindungs- und Internetzugangsdienste verzichtet, da die Anwendungsfälle davon unabhängig sind.

AF	Dienst	Dienstklasse	Zugangs-technologie	Domäne	Authenti fizierung	Ver-fahren	Dyna-misch	Privat
1.1	Kabel-Verbindung zu LAN in Kanzlei	Verbindung	Kabel LAN	Heimat				
1.2	Intranet-Zugang in Kanzlei	Internet-Zugang	Kabel LAN	Heimat				
1.3	Rechnungswesen Deluxe	Anwen-dung, Inhalt			Person Besitz	Ch-Re	statisch	ja
2.1	Wave-LAN Verbin-dung zu LAN in Außenstelle	Verbindung	Funk LAN	fremde				
2.2	Intranet-Zugang in Außenstelle	Internet-Zugang	Funk LAN	fremde	Person Wissen	Ch-Re	statisch	nein
3.1	GSM-Verbindung zu Einwahlknoten von Belgacom	Verbindung	Wählverbin-dung Funk	fremde	Person Besitz/ Wissen	Ch-Re	statisch	nein
3.2	Internet-Zugang von Belgacom	Internet-Zugang	Wählverbin-dung Funk	fremde	nur Iden-tifizie-rung			
3.3	Web-Mailer von WWW-OK	Anwendung			Person Wissen	direkt	statisch	ja
4.1	Web-Fotoalbum von WWW-OK	Anwendung			Person Wissen	direkt	statisch	ja
4.2	Web-Service und Speicherplatz von WWW-OK	Anwendung			Person Wissen	direkt	statisch	ja
5.1	Recherche in Inter-net-Datenbank von Recht & Urteil	Anwen-dung, Inhalt						

Tabelle 15: Merkmale der Anwendungsfälle

AF	Dienst	Dienstklasse	Zugangs-technologie	Domäne	Authenti fizierung	Ver-fahren	Dyna-misch	Privat
5.2	Download der Gesetzestexte von Recht & Urteil	Inhalt				Berech -tigung		
6.1	Videokonferenz von NRW-ON	Anwendung			Person Wissen	direkt	ja	nein
6.2	Backbone QoS Datentransport von Backbone Anbieter	QoS-Trans-port			Person Wissen	direkt	ja	nein

Tabelle 15: Merkmale der Anwendungsfälle

B.1 1. Anwendungsfall

Internet-Dienstmodell.

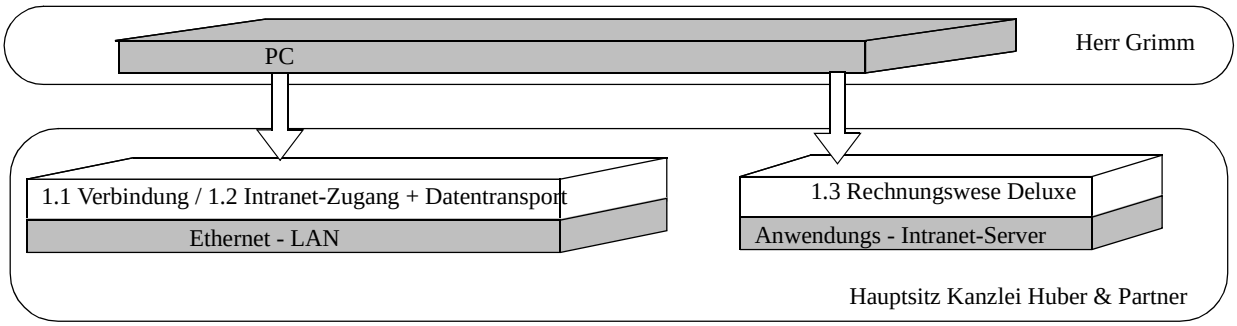

Abbildung 67: Internet-Dienstmodell im Anwendungsfall 1

Direkte Zugriffskontrolle durch die Anwendung “Rechnungswesen Deluxe”.

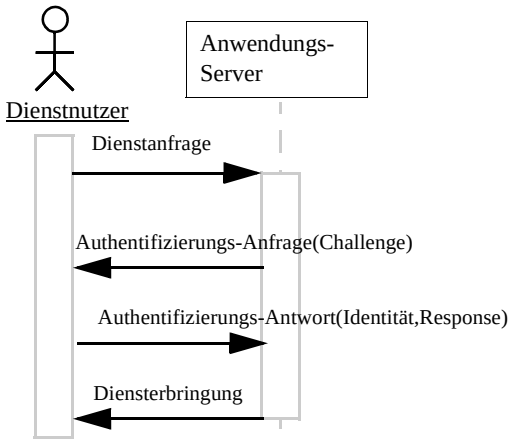

Abbildung 68: Protokollablauf im Anwendungsfall 1.3

Zugriffskontroll-Policy für Huber&Partner.

```
<?xml version="1.0" encoding="UTF-8"?>
<ZugriffskontrollPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSche-
maLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontrolldien-
ste.xsd">
   <ZugriffskontrollPolicyID>Rechnungswesen-Deluxe-bei-HuberundPartner</ZugriffskontrollPolicyID>
   <Dienstanbieter>k.huber_und_partner.de</Dienstanbieter>
   <Dienst-Heimatdienstanbieter-Policy>
      <Dienst>
         <Dienst-ID>22</Dienst-ID>
         <Dienstbezeichnung>Rechnungswesen Deluxe - Huber und  Partner</Dienstbezeichnung>
         <Lokation>file://deluxe/mandanten/*</Lokation>
      </Dienst>
      <Heimatdienstanbieter-Policy>
         <Zugriffskontrolldienste>
            <AuthentifizierungsPolicy>
               <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
               <OrtderDurchfuehrung>
                  <lokal/>
               </OrtderDurchfuehrung>
            </AuthentifizierungsPolicy>
            <AutorisierungsPolicy>
               <PruefungNutzerberechtigungen>
                  <OrtderDurchfuehrung>
                     <lokal/>
                  </OrtderDurchfuehrung>
                  <Authentifizierungs.ID>Benutzerkennung</Authentifizierungs.ID>
               </PruefungNutzerberechtigungen>
            </AutorisierungsPolicy>
         </Zugriffskontrolldienste>
      </Heimatdienstanbieter-Policy>
   </Dienst-Heimatdienstanbieter-Policy>
</ZugriffskontrollPolicy>
```

Nutzung der A^x-Architektur.

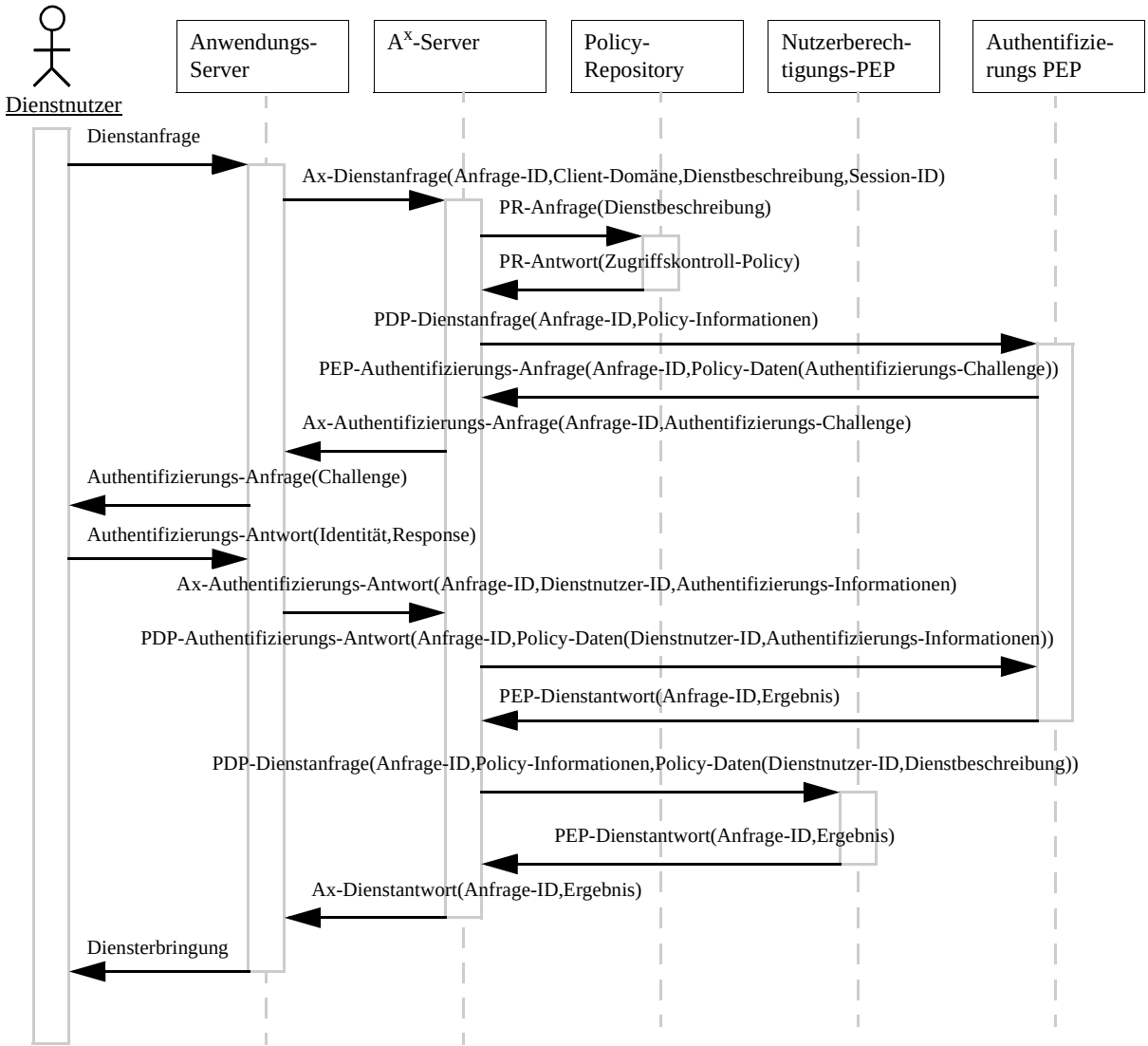

Abbildung 69: Protokollablauf im Anwendungsfall 1.3 - A^x-Architektur

B.2 2. Anwendungsfall

Internet-Dienstmodell.

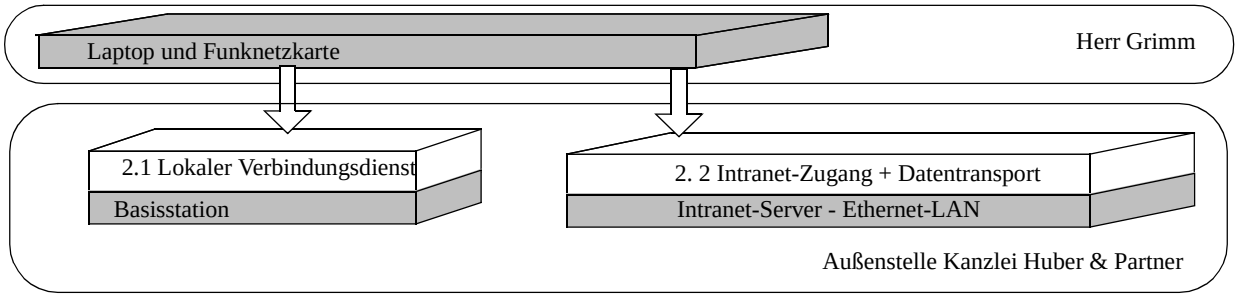

Abbildung 70: Internet-Dienstmodell im Anwendungsfall 2

Zugriffskontrolle mittels 802.1X und RADIUS.

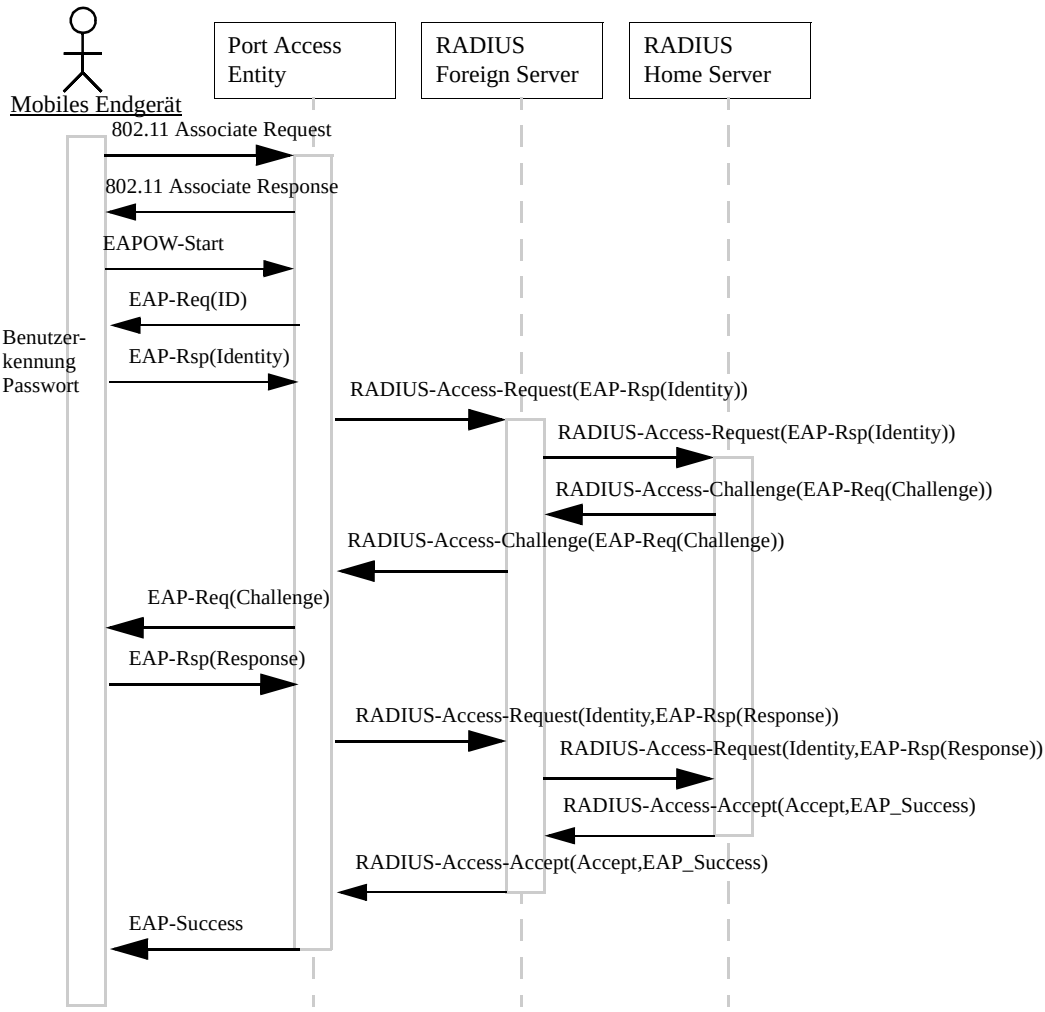

Abbildung 71: Protokollablauf im Anwendungsfall 2.2

Zugriffskontroll-Policy für Intranet-Zugang und Datentransport.

```
<?xml version="1.0" encoding="UTF-8"?>
<ZugriffskontrollPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontrolldienste.xsd">
   <Dienstanbieter>do.huber_und_partner.de</Dienstanbieter>
   <Dienst-Heimatdienstanbieter-Policy>
   <Dienst>
      <Dienst-ID>52</Dienst-ID>
      <Dienstbezeichnung>Internet-Zugang</Dienstbezeichnung>
   </Dienst>
   <Heimatdienstanbieter-Policy>
   <Heimatdienstanbieter>k.huber_und_partner.de</Heimatdienstanbieter>
   <Heimatdienstanbieter>ms.huber_und_partner.de</Heimatdienstanbieter>
   <Zugriffskontrolldienste>
      <AuthentifizierungsPolicy>
         <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
         <OrtderDurchfuehrung>
            <fremderHeimatdienstanbieter/>
         </OrtderDurchfuehrung>
      </AuthentifizierungsPolicy>
   </Zugriffskontrolldienste>
   </Heimatdienstanbieter-Policy>
   <Heimatdienstanbieter-Policy>
   <Heimatdienstanbieter>do.huber_und_partner.de</Heimatdienstanbieter>
   <Zugriffskontrolldienste>
      <AuthentifizierungsPolicy>
         <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
         <OrtderDurchfuehrung>
            <lokal/>
         </OrtderDurchfuehrung>
      </AuthentifizierungsPolicy>
   </Zugriffskontrolldienste>
   </Heimatdienstanbieter-Policy>
   </Dienst-Heimatdienstanbieter-Policy>
</ZugriffskontrollPolicy>
```

Nutzung der A^{x}-Architektur.

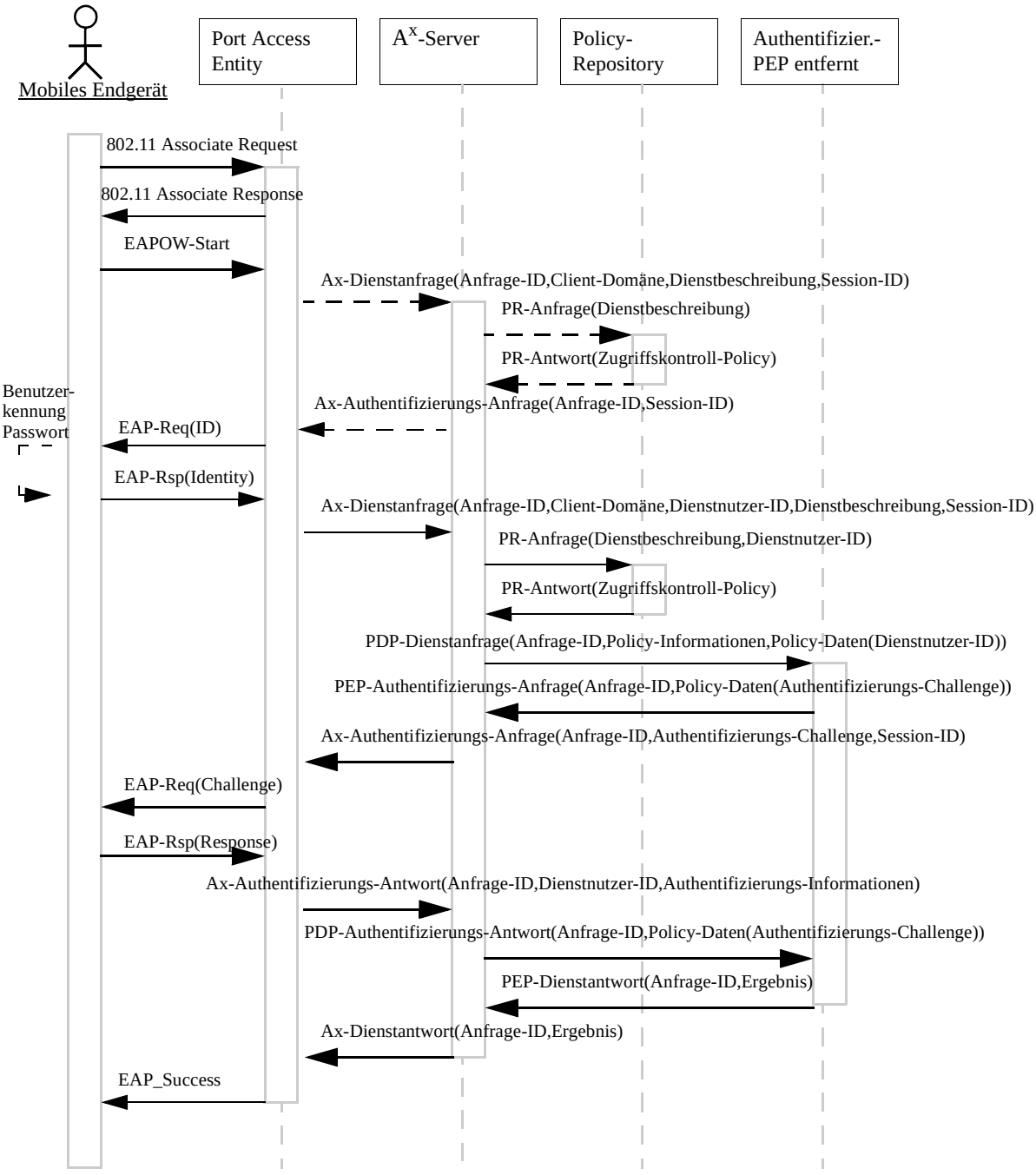

Abbildung 72: Protokollablauf im Anwendungsfall 2.2 - A^{x}-Architektur

B.3 3. Anwendungsfall

Internet-Dientsmodell.

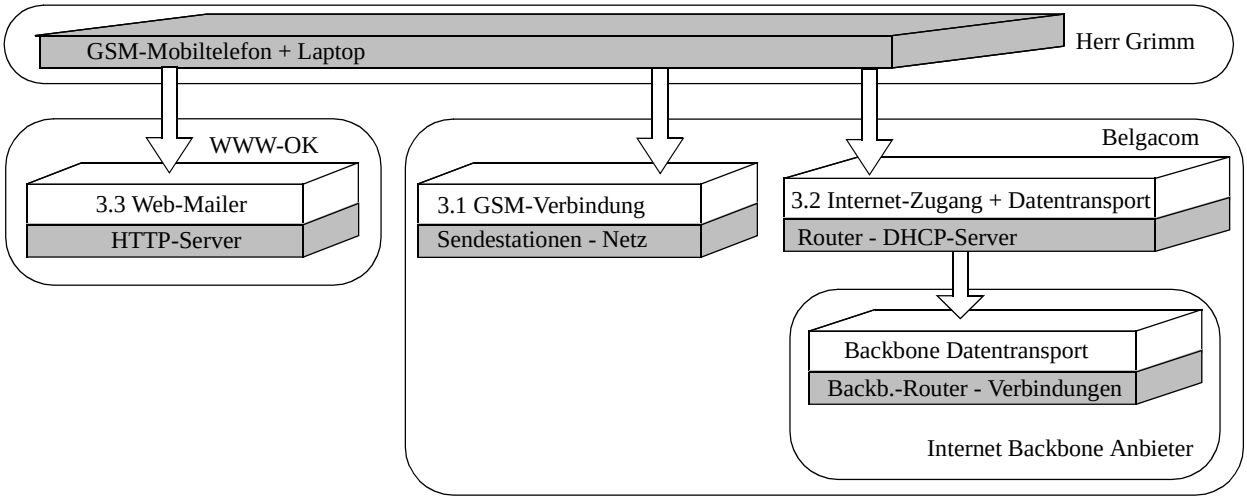

Abbildung 73: Internet-Dienstmodell im Anwendungsfall 3

Call-by-Call und GSM-Zugriffskontrolle durch Belgacom.

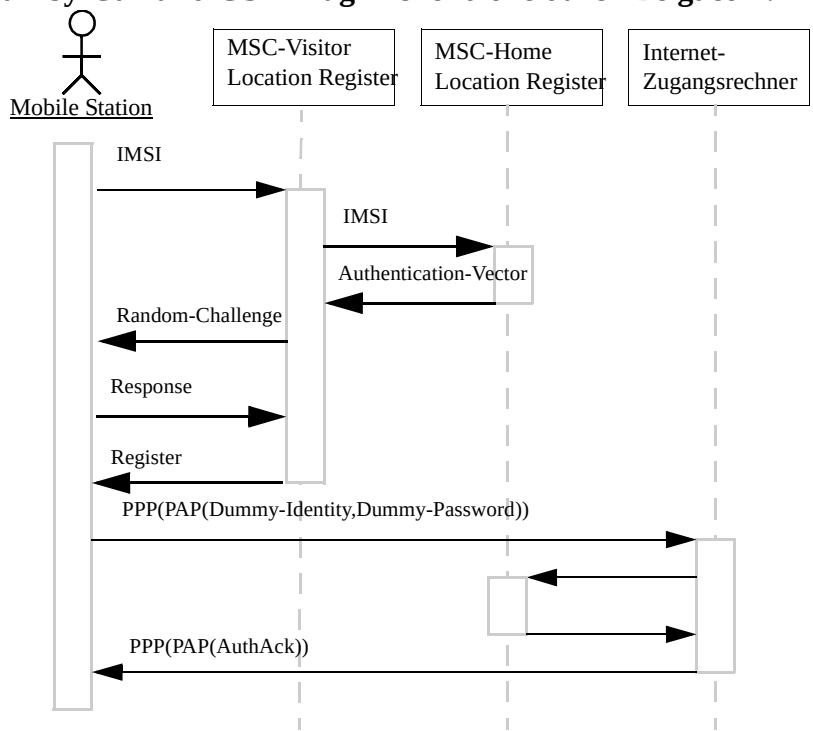

Abbildung 74: Protokollablauf im Anwendungsfall 3.1/3.2

Direkte Zugriffskontrolle durch die Anwendung von WWW-OK.

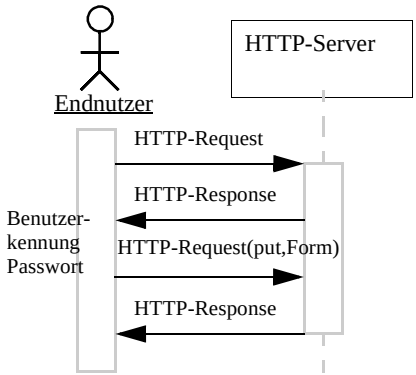

Abbildung 75: Protokollablauf im Anwendungsfall 3.3

Zugriffskontroll-Policy von Belgacom.

```
<?xml version="1.0" encoding="UTF-8"?>
<ZugriffskontrollPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSche-
maLocation=”http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontrolldien-
ste.xsd”>
   <ZugriffskontrollPolicyID>Belgacom</ZugriffskontrollPolicyID>
   <Dienstanbieter>Belgacom.be</Dienstanbieter>
   <Dienst-Heimatdienstanbieter-Policy>
      <Dienst>
         <Dienst-ID>5555</Dienst-ID>
         <Dienstbezeichnung>GSM</Dienstbezeichnung>
      </Dienst>
      <Heimatdienstanbieter-Policy>
         <Heimatdienstanbieter>belgacom.be</Heimatdienstanbieter>
         <Zugriffskontrolldienste>
            <AuthentifizierungsPolicy>
               <AuthentifizierungsID>Teilnehmeranschlussnummer</AuthentifizierungsID>
               <OrtderDurchfuehrung>
                  <lokal/>
               </OrtderDurchfuehrung>
            </AuthentifizierungsPolicy>
         </Zugriffskontrolldienste>
      </Heimatdienstanbieter-Policy>
      <Heimatdienstanbieter-Policy>
         <Heimatdienstanbieter>t-mobile.de</Heimatdienstanbieter>
         <Heimatdienstanbieter>orange.fr</Heimatdienstanbieter>
         <Zugriffskontrolldienste>
            <AuthentifizierungsPolicy>
               <AuthentifizierungsID>Teilnehmeranschlussnummer</AuthentifizierungsID>
               <OrtderDurchfuehrung>
                  <fremderHeimatdienstanbieter/>
               </OrtderDurchfuehrung>
            </AuthentifizierungsPolicy>
         </Zugriffskontrolldienste>
      </Heimatdienstanbieter-Policy>
   </Dienst-Heimatdienstanbieter-Policy>
   <Dienst-Heimatdienstanbieter-Policy>
      <Dienst>
```

```
      <Dienst-ID>5241</Dienst-ID>
      <Dienstbezeichnung>Internetzugang und Basistransport</Dienstbezeichnung>
    </Dienst>
    <Heimatdienstanbieter-Policy>
      <Heimatdienstanbieter>belgacom.be</Heimatdienstanbieter>
      <Zugriffskontrolldienste>
        <AuthentifizierungsPolicy>
          <AuthentifizierungsID>Teilnehmeranschlussnummer</AuthentifizierungsID>
          <OrtderDurchfuehrung>
            <lokal/>
          </OrtderDurchfuehrung>
        </AuthentifizierungsPolicy>
      </Zugriffskontrolldienste>
      <Zugriffskontrolldienste>
        <AuthentifizierungsPolicy>
          <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
          <OrtderDurchfuehrung>
            <lokal/>
          </OrtderDurchfuehrung>
        </AuthentifizierungsPolicy>
      </Zugriffskontrolldienste>
    </Heimatdienstanbieter-Policy>
    <Heimatdienstanbieter-Policy>
      <Heimatdienstanbieter>t-mobile.de</Heimatdienstanbieter>
      <Heimatdienstanbieter>orange.fr</Heimatdienstanbieter>
      <Zugriffskontrolldienste>
        <AuthentifizierungsPolicy>
          <AuthentifizierungsID>Teilnehmeranschlussnummer</AuthentifizierungsID>
          <OrtderDurchfuehrung>
            <fremderHeimatdienstanbieter/>
          </OrtderDurchfuehrung>
        </AuthentifizierungsPolicy>
      </Zugriffskontrolldienste>
      <Zugriffskontrolldienste>
        <AuthentifizierungsPolicy>
          <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
          <OrtderDurchfuehrung>
            <fremderHeimatdienstanbieter/>
          </OrtderDurchfuehrung>
        </AuthentifizierungsPolicy>
      </Zugriffskontrolldienste>
    </Heimatdienstanbieter-Policy>
  </Dienst-Heimatdienstanbieter-Policy>
</ZugriffskontrollPolicy>
```

Zugriffskontroll-Policy von WWW-OK.

```
<?xml version="1.0" encoding="UTF-8"?>
<ZugriffskontrollPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSche-
maLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontrolldien-
ste.xsd">
  <ZugriffskontrollPolicyID>wwwok-webmailer</ZugriffskontrollPolicyID>
  <Dienstanbieter>wwwok.de</Dienstanbieter>
  <Dienst-Heimatdienstanbieter-Policy>
    <Dienst>
```

```
        <Dienst-ID>1010</Dienst-ID>
        <Dienstbezeichnung>Web-Mailer</Dienstbezeichnung>
        <Lokation>file://kunden-id/postfaecher/benutzerkennung/*</Lokation>
      </Dienst>
      <Heimatdienstanbieter-Policy>
        <Zugriffskontrolldienste>
          <AuthentifizierungsPolicy>
            <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
            <OrtderDurchfuehrung>
              <lokal/>
            </OrtderDurchfuehrung>
          </AuthentifizierungsPolicy>
          <AutorisierungsPolicy>
            <PruefungNutzerberechtigungen>
              <OrtderDurchfuehrung>
                <lokal/>
              </OrtderDurchfuehrung>
              <Authentifizierungs.ID>Benutzerkennung</Authentifizierungs.ID>
            </PruefungNutzerberechtigungen>
          </AutorisierungsPolicy>
        </Zugriffskontrolldienste>
      </Heimatdienstanbieter-Policy>
    </Dienst-Heimatdienstanbieter-Policy>
</ZugriffskontrollPolicy>
```

Zugriffskontroll-Policy des Internet Backbone Anbieters.

```
<?xml version="1.0" encoding="UTF-8"?>
<ZugriffskontrollPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSche-
maLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontrolldien-
ste.xsd">
    <ZugriffskontrollPolicyID>Backbone-Basisdatentransport</ZugriffskontrollPolicyID>
    <Dienstanbieter>internet-backbone.com</Dienstanbieter>
    <Dienst-Heimatdienstanbieter-Policy>
      <Dienst>
        <Dienst-ID>5236</Dienst-ID>
        <Dienstbezeichnung>Internet-Zugang und Basis-Datentransport</Dienstbezeichnung>
      </Dienst>
      <Heimatdienstanbieter-Policy>
        <Zugriffskontrolldienste/>
      </Heimatdienstanbieter-Policy>
    </Dienst-Heimatdienstanbieter-Policy>
</ZugriffskontrollPolicy>
```

Nutzung der A^x-Architektur durch Belgacom.

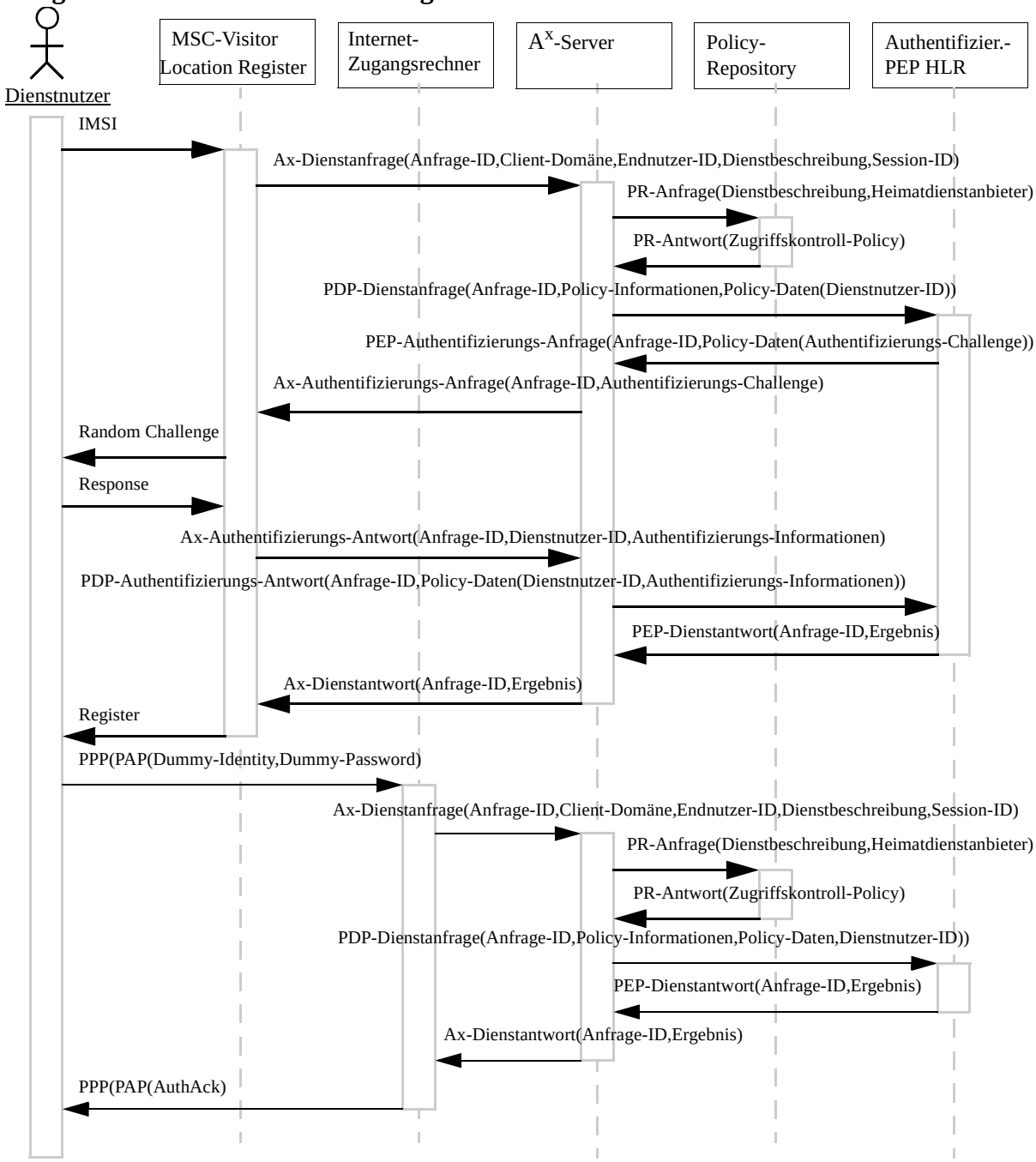

Abbildung 76: Protokollablauf im Anwendungsfall 3.1/3.2 - A^x-Architektur

Nutzung der A^x-Architektur durch WWW-OK.

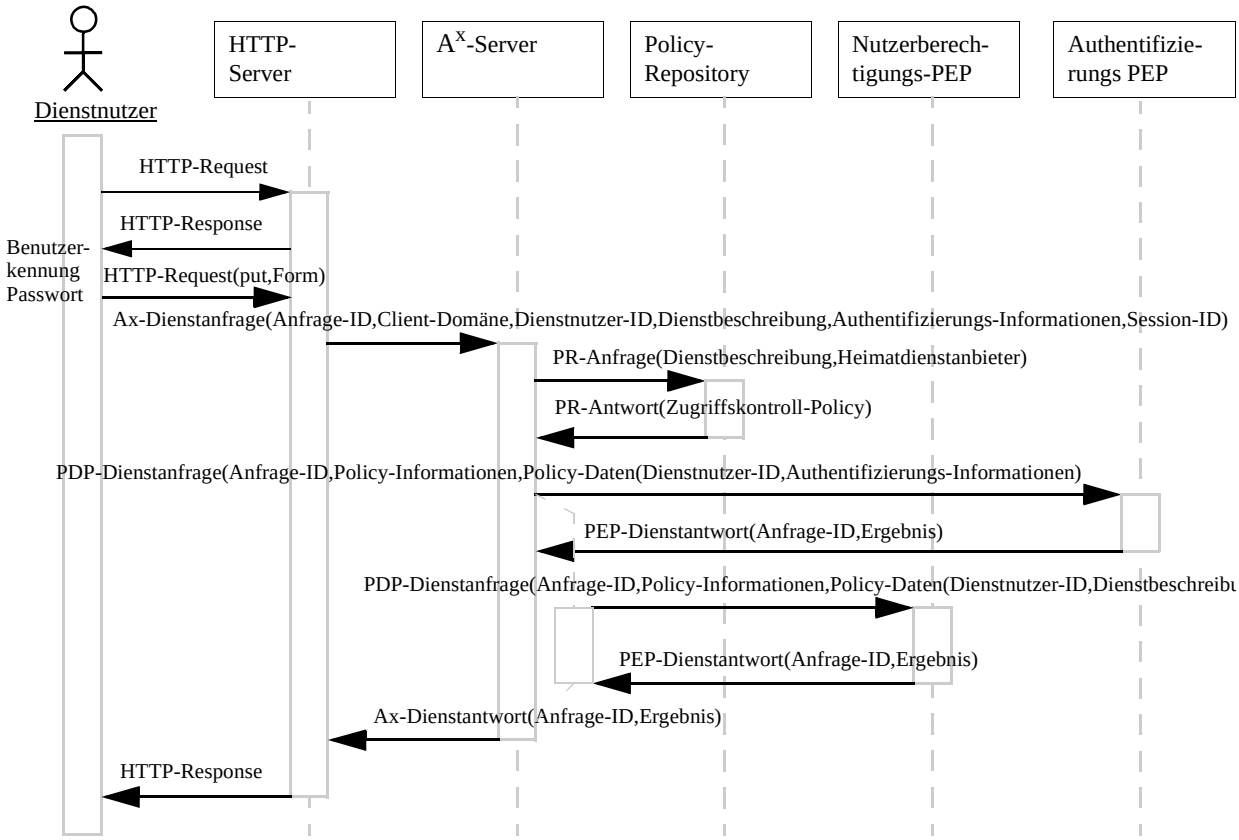

Abbildung 77: Protokollablauf im Anwendungsfall 3.3 - A^x-Architektur

B.4 4. Anwendungsfall

Internet-Dienstmodell.

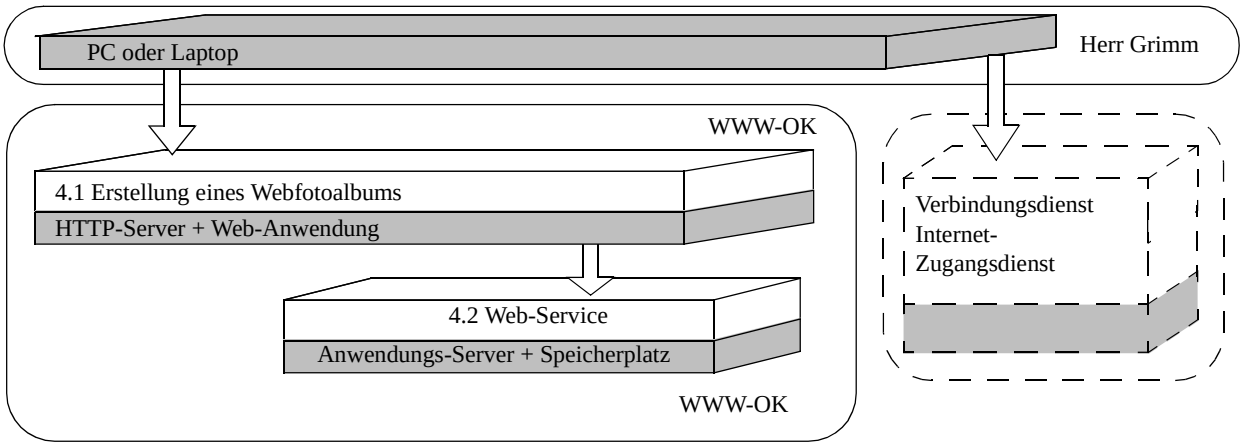

Abbildung 78: Internet-Dienstmodell im Anwendungsfall 4

Direkte Zugriffskontrolle durch die Anwendung.

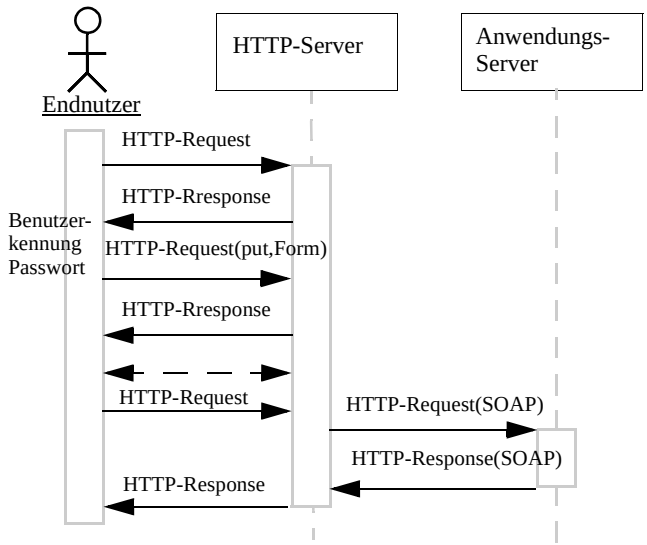

Abbildung 79: Protokollablauf im Anwendungsfall 4.1/4.2

Zugriffskontroll-Policy von WWW-OK.

```
<?xml version="1.0" encoding="UTF-8"?>
<ZugriffskontrollPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontrolldienste.xsd">
  <Dienstanbieter>wwwok.de</Dienstanbieter>
  <Dienst-Heimatdienstanbieter-Policy>
    <Dienst>
      <Dienst-ID>1020</Dienst-ID>
      <Dienstbezeichnung>Web-Fotoalbum</Dienstbezeichnung>
      <Lokation>file://kunden-id/htdocs/fotoalbum/*</Lokation>
    </Dienst>
    <Heimatdienstanbieter-Policy>
      <Zugriffskontrolldienste>
        <AuthentifizierungsPolicy>
          <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
          <OrtderDurchfuehrung>
            <lokal/>
          </OrtderDurchfuehrung>
        </AuthentifizierungsPolicy>
        <AutorisierungsPolicy>
          <PruefungNutzerberechtigungen>
            <OrtderDurchfuehrung>
              <lokal/>
            </OrtderDurchfuehrung>
            <Authentifizierungs.ID>Benutzerkennung</Authentifizierungs.ID>
          </PruefungNutzerberechtigungen>
        </AutorisierungsPolicy>
      </Zugriffskontrolldienste>
    </Heimatdienstanbieter-Policy>
  </Dienst-Heimatdienstanbieter-Policy>
  <Dienst-Heimatdienstanbieter-Policy>
    <Dienst>
      <Dienst-ID>15</Dienst-ID>
      <Dienstbezeichnung>Web-Service-Thumbnails</Dienstbezeichnung>
    </Dienst>
    <Heimatdienstanbieter-Policy>
      <Heimatdienstanbieter>wwwok.de</Heimatdienstanbieter>
      <Zugriffskontrolldienste>
        <AuthentifizierungsPolicy>
          <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
          <OrtderDurchfuehrung>
            <lokal/>
          </OrtderDurchfuehrung>
        </AuthentifizierungsPolicy>
        <AutorisierungsPolicy>
          <PruefungNutzerberechtigungen>
            <OrtderDurchfuehrung>
              <lokal/>
            </OrtderDurchfuehrung>
            <Authentifizierungs.ID>Benutzerkennung</Authentifizierungs.ID>
          </PruefungNutzerberechtigungen>
        </AutorisierungsPolicy>
```

```
        </Zugriffskontrolldienste>
      </Heimatdienstanbieter-Policy>
   </Dienst-Heimatdienstanbieter-Policy>
</ZugriffskontrollPolicy>
```

Nutzung der A^x-Architektur.

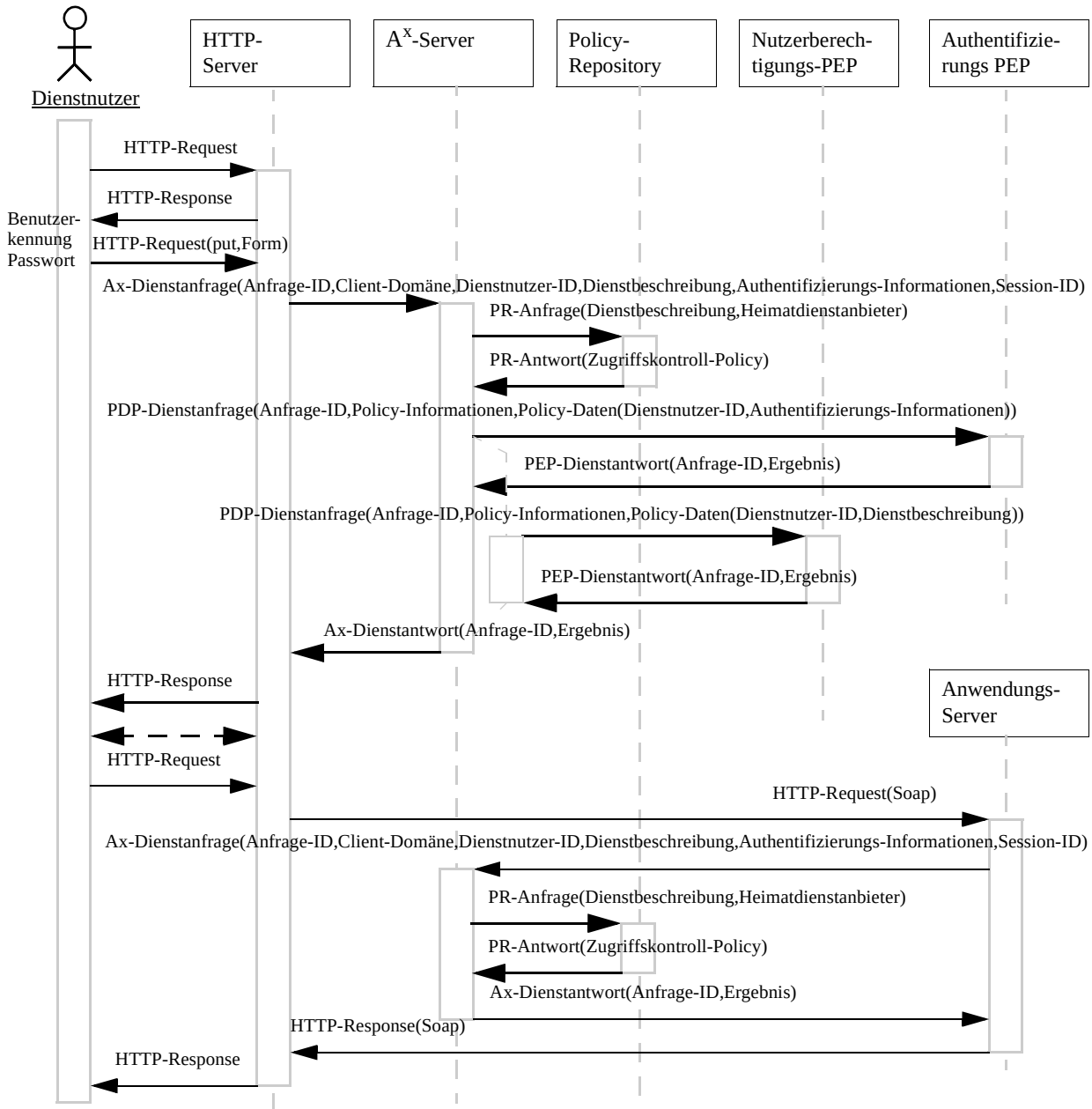

Abbildung 80: Protokollablauf im Anwendungsfall 4.1/4.2 - A^x-Architektur

B.5 5. Anwendungsfall

Internet-Dienstmodell.

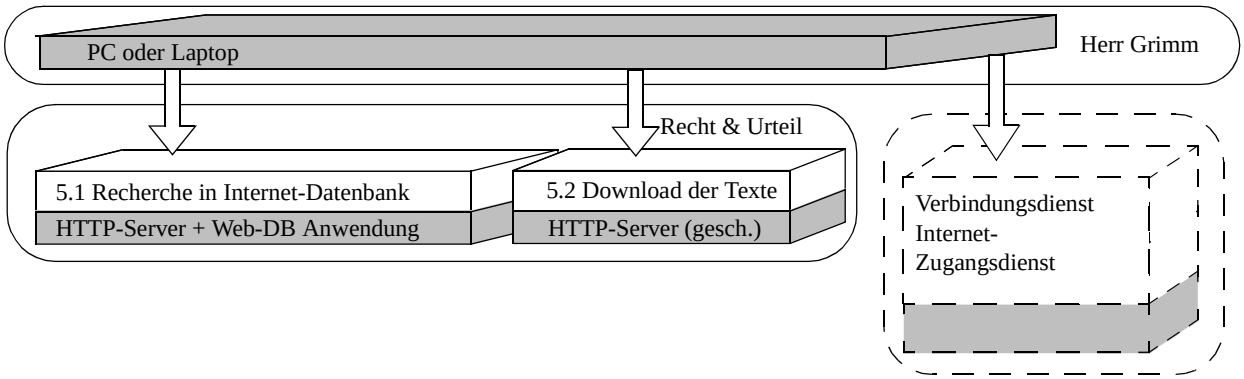

Abbildung 81: Internet-Dienstmodell im Anwendungsfall 5

Direkte Zugriffskontrolle mittels Autorisierung über Kreditkarteninformationen.

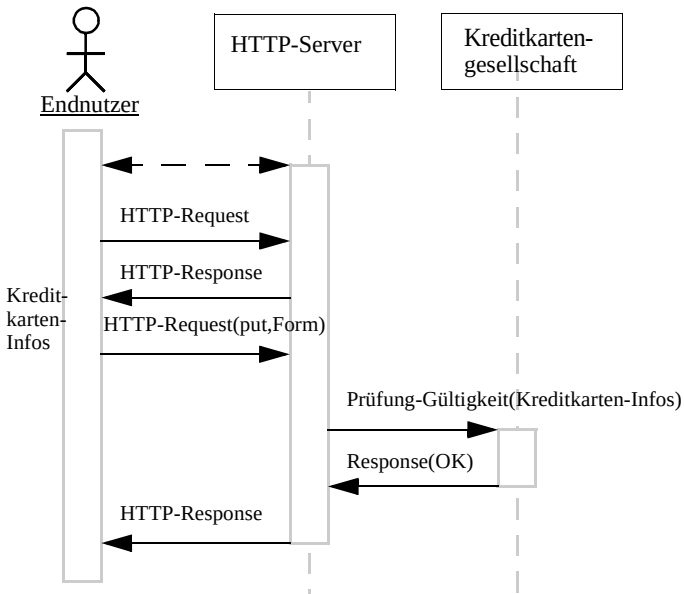

Abbildung 82: Protokollablauf im Anwendungsfall 5.2

Zugriffskontroll-Policy von Recht&Urteil.

```
<?xml version="1.0" encoding="UTF-8"?>
<ZugriffskontrollPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontrolldienste.xsd">
   <ZugriffskontrollPolicyID>rechtundurteil-webdienste</ZugriffskontrollPolicyID>
   <Dienstanbieter>recht-und-urteil.de</Dienstanbieter>
   <Dienst-Heimatdienstanbieter-Policy>
      <Dienst>
         <Dienst-ID>5401</Dienst-ID>
         <Dienstbezeichnung>Web-Datenbank</Dienstbezeichnung>
      </Dienst>
      <Heimatdienstanbieter-Policy>
         <Zugriffskontrolldienste/>
      </Heimatdienstanbieter-Policy>
   </Dienst-Heimatdienstanbieter-Policy>
   <Dienst-Heimatdienstanbieter-Policy>
      <Dienst>
         <Dienst-ID>54</Dienst-ID>
         <Dienstbezeichnung>HTTP-Download Urteile</Dienstbezeichnung>
         <Lokation>file://urteile/kostenpflichtig</Lokation>
      </Dienst>
      <Heimatdienstanbieter-Policy>
         <Zugriffskontrolldienste>
            <AutorisierungsPolicy>
               <PruefungBerechtigungsnachweis>
                  <OrtderDurchfuehrung>
                     <lokal/>
                  </OrtderDurchfuehrung>
                  <TypBerechtigungsnachweis>Kreditkarte</TypBerechtigungsnachweis>
               </PruefungBerechtigungsnachweis>
            </AutorisierungsPolicy>
         </Zugriffskontrolldienste>
         <Zugriffskontrolldienste>
            <AuthentifizierungsPolicy>
               <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
               <OrtderDurchfuehrung>
                  <lokal/>
               </OrtderDurchfuehrung>
            </AuthentifizierungsPolicy>
            <AutorisierungsPolicy>
               <DynamischeAutorisierung>
                  <OrtderDurchfuehrung>
                     <lokal/>
                  </OrtderDurchfuehrung>
                  <TypDynamischeAutorisierung>Kontenstand</TypDynamischeAutorisierung>
               </DynamischeAutorisierung>
            </AutorisierungsPolicy>
         </Zugriffskontrolldienste>
      </Heimatdienstanbieter-Policy>
   </Dienst-Heimatdienstanbieter-Policy>
</ZugriffskontrollPolicy>
```

Nutzung der A^x-Architektur.

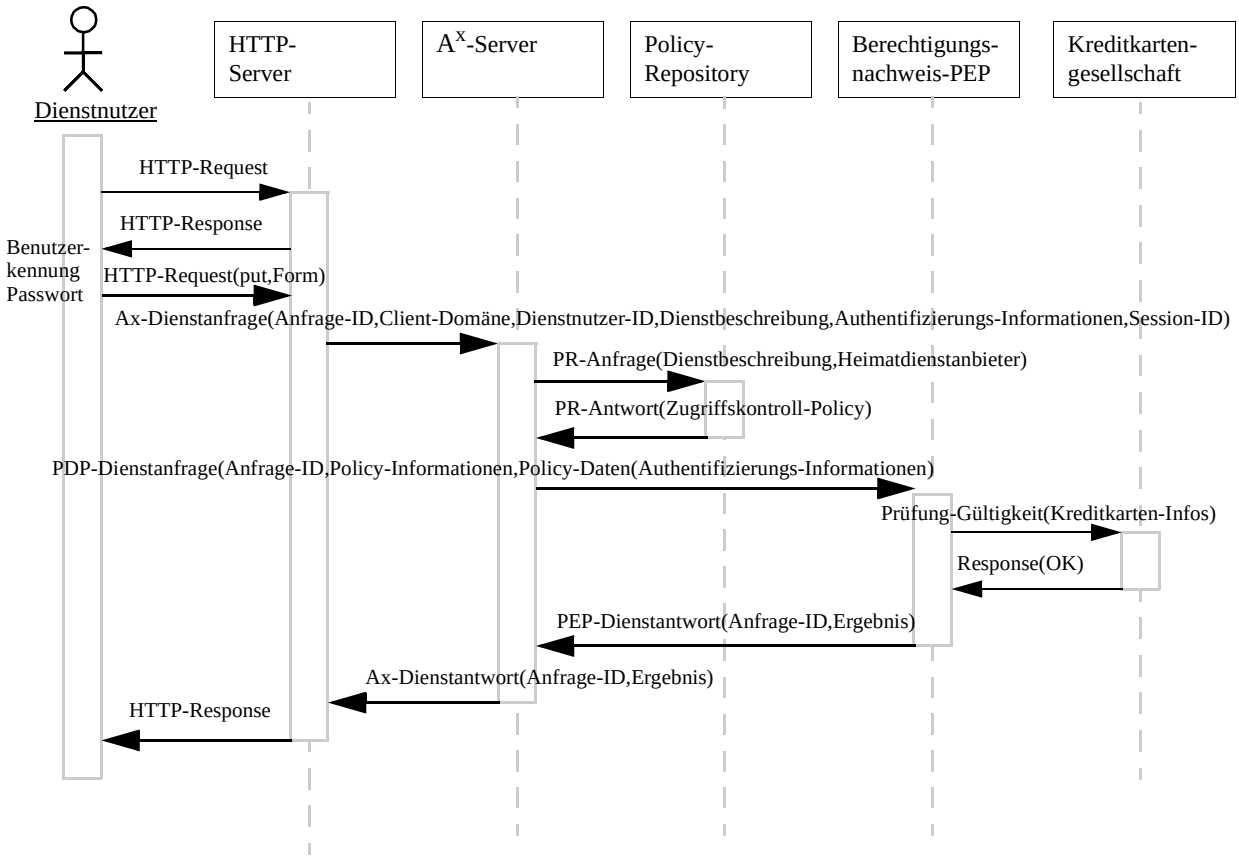

Abbildung 83: Protokollablauf im Anwendungsfall 5.2 - A^x-Architektur

B.6 6. Anwendungsfall

Internet-Dienstmodell.

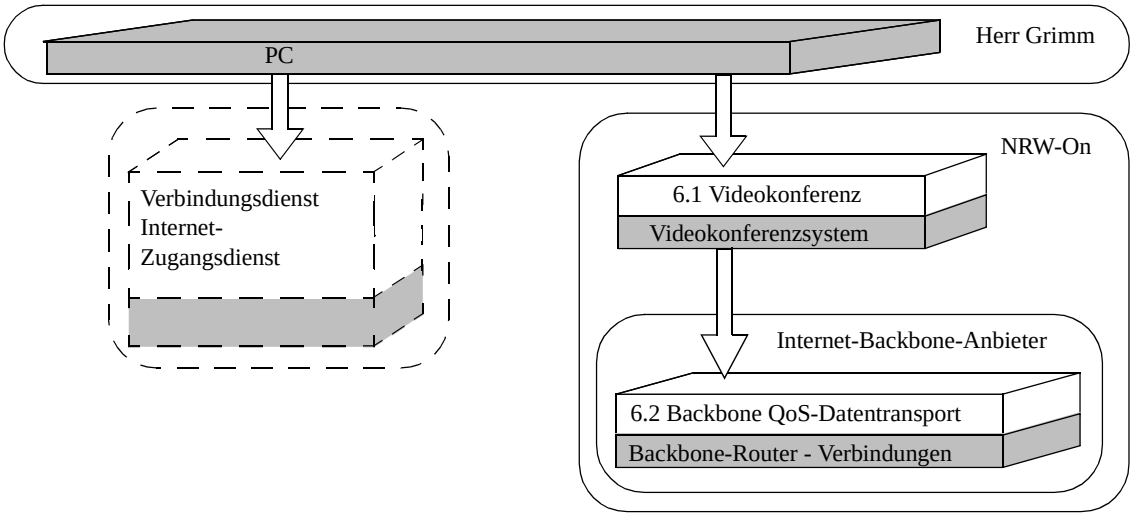

Abbildung 84: Internet-Dienstmodell im Anwendungsfall 6

Direkte Zugriffskontrolle und Zugriffskontrolle durch IntServ Router.

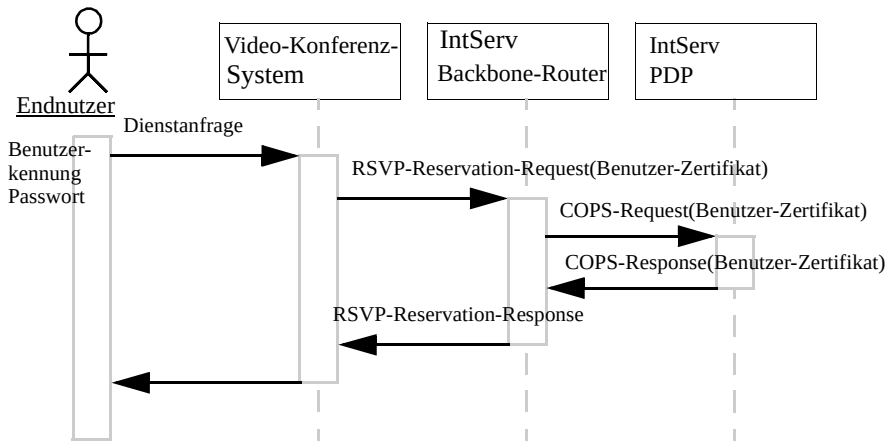

Abbildung 85: Protokollablauf im Anwendungsfall 6.1/6.2

Zugriffskontroll-Policy von NRW-On.

```
<?xml version="1.0" encoding="UTF-8"?>
<ZugriffskontrollPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSche-
maLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontrolldien-
ste.xsd">
   <ZugriffskontrollPolicyID>NRW-On ZugriffskontrollPolicy</ZugriffskontrollPolicyID>
   <Dienstanbieter>nrw-on.de</Dienstanbieter>
   <Dienst-Heimatdienstanbieter-Policy>
      <Dienst>
         <Dienst-ID>65</Dienst-ID>
         <Dienstbezeichnung>Internet-Zugang</Dienstbezeichnung>
      </Dienst>
      <Heimatdienstanbieter-Policy>
         <Zugriffskontrolldienste/>
      </Heimatdienstanbieter-Policy>
   </Dienst-Heimatdienstanbieter-Policy>
   <Dienst-Heimatdienstanbieter-Policy>
      <Dienst>
         <Dienst-ID>555</Dienst-ID>
         <Dienstbezeichnung>Videokonferenz</Dienstbezeichnung>
      </Dienst>
      <Heimatdienstanbieter-Policy>
         <Zugriffskontrolldienste>
            <AuthentifizierungsPolicy>
               <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
               <OrtderDurchfuehrung>
                  <lokal/>
               </OrtderDurchfuehrung>
            </AuthentifizierungsPolicy>
            <AutorisierungsPolicy>
               <DynamischeAutorisierung>
                  <OrtderDurchfuehrung>
                     <lokal/>
                  </OrtderDurchfuehrung>
                  <TypDynamischeAutorisierung>Pruefung-Gruppenzugehoerigkeit</TypDynamischeAu-
torisierung>
               </DynamischeAutorisierung>
            </AutorisierungsPolicy>
         </Zugriffskontrolldienste>
         <Zugriffskontrolldienste>
            <AuthentifizierungsPolicy>
               <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
               <OrtderDurchfuehrung>
                  <lokal/>
               </OrtderDurchfuehrung>
            </AuthentifizierungsPolicy>
            <AutorisierungsPolicy>
               <DynamischeAutorisierung>
                  <OrtderDurchfuehrung>
                     <lokal/>
                  </OrtderDurchfuehrung>
                  <TypDynamischeAutorisierung>Kontenstand</TypDynamischeAutorisierung>
               </DynamischeAutorisierung>
```

```
            </AutorisierungsPolicy>
         </Zugriffskontrolldienste>
      </Heimatdienstanbieter-Policy>
   </Dienst-Heimatdienstanbieter-Policy>
</ZugriffskontrollPolicy>
```

Zugriffskontroll-Policy des Internet-Backbone-Anbieters.

```
<?xml version="1.0" encoding="UTF-8"?>
<ZugriffskontrollPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSche-
maLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/xml/schema-zugriffskontrolldien-
ste.xsd">
   <ZugriffskontrollPolicyID>backbone-qos</ZugriffskontrollPolicyID>
   <Dienstanbieter>internet-backbone.com</Dienstanbieter>
   <Dienst-Heimatdienstanbieter-Policy>
      <Dienst>
         <Dienst-ID>6541</Dienst-ID>
         <Dienstbezeichnung>QoS Reservierung</Dienstbezeichnung>
      </Dienst>
      <Heimatdienstanbieter-Policy>
         <Zugriffskontrolldienste>
            <AuthentifizierungsPolicy>
               <AuthentifizierungsID>Benutzerkennung</AuthentifizierungsID>
               <OrtderDurchfuehrung>
                  <lokal/>
               </OrtderDurchfuehrung>
            </AuthentifizierungsPolicy>
            <AutorisierungsPolicy>
               <PruefungNutzerberechtigungen>
                  <OrtderDurchfuehrung>
                     <lokal/>
                  </OrtderDurchfuehrung>
                  <Authentifizierungs.ID>Benutzerkennung</Authentifizierungs.ID>
               </PruefungNutzerberechtigungen>
            </AutorisierungsPolicy>
         </Zugriffskontrolldienste>
      </Heimatdienstanbieter-Policy>
   </Dienst-Heimatdienstanbieter-Policy>
</ZugriffskontrollPolicy>
```

Nutzung der A^x-Architektur.

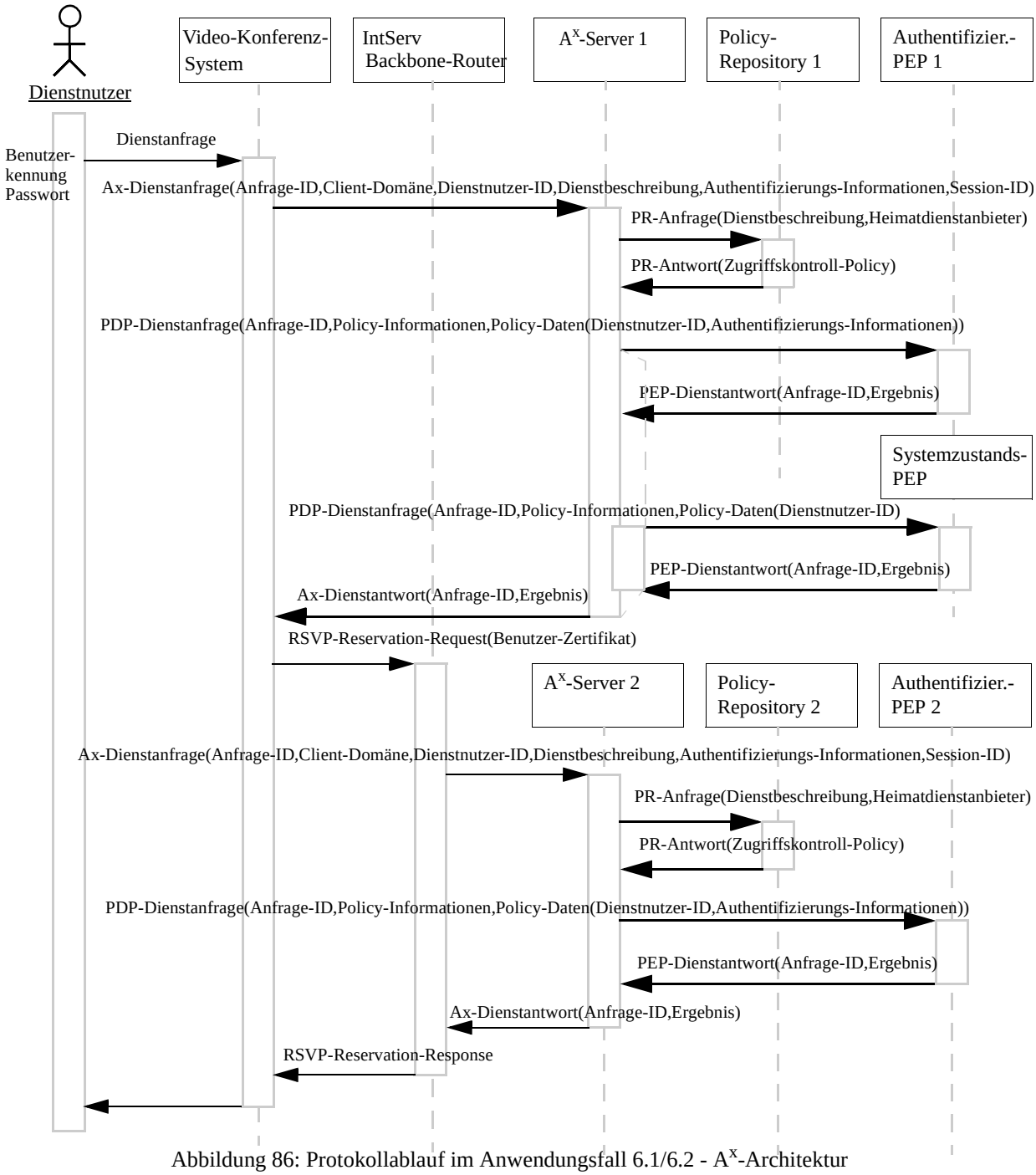

Abbildung 86: Protokollablauf im Anwendungsfall 6.1/6.2 - A^x-Architektur

Anhang C: Policy-Beschreibungssprachen

C.1 Die Policy-Sprache zur Beschreibung von Geschäftsmodellen

Überblick über die Sprachelemente.

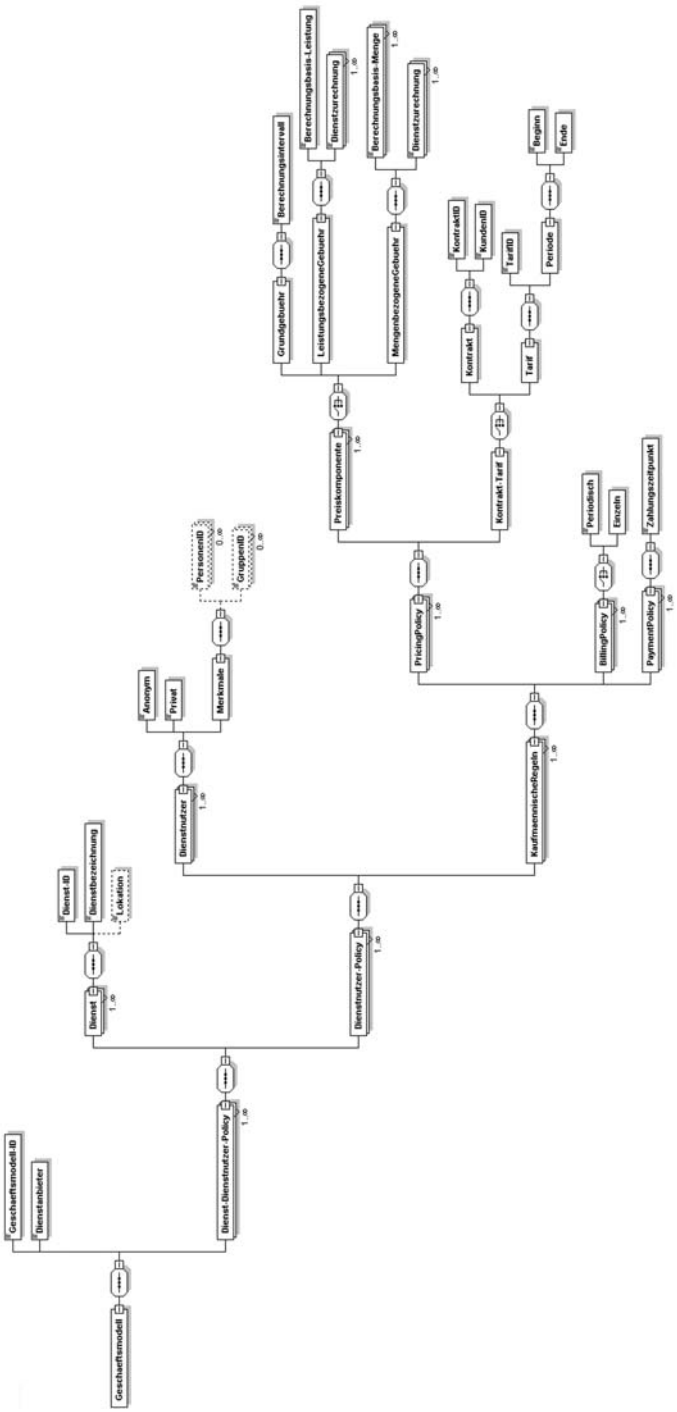

Abbildung 87: Elemente der Policy-Sprache für Geschäftsmodelle

XML-Schema der Policy-Sprache.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attribute-
FormDefault="unqualified">
   <xs:element name="Geschaeftsmodell">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="Geschaeftsmodell-ID" type="xs:string"/>
            <xs:element name="Dienstanbieter" type="xs:anyURI"/>
            <xs:element name="Dienst-Dienstnutzer-Policy" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="Dienst" maxOccurs="unbounded">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="Dienst-ID" type="xs:string"/>
                              <xs:element name="Dienstbezeichnung" type="xs:string"/>
                              <xs:element name="Lokation" type="xs:anyURI" minOccurs="0"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                     <xs:element name="Dienstnutzer-Policy" maxOccurs="unbounded">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="Dienstnutzer" maxOccurs="unbounded">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="Anonym" type="xs:boolean"/>
                                       <xs:element name="Privat" type="xs:boolean"/>
                                       <xs:element name="Merkmale">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="PersonenID" type="xs:string" minOc-
curs="0" maxOccurs="unbounded"/>
                                                <xs:element name="GruppenID" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
                                             </xs:sequence>
                                          </xs:complexType>
                                       </xs:element>
                                    </xs:sequence>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="KaufmaennischeRegeln">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="PricingPolicy" maxOccurs="unbounded">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="Preiskomponente" maxOc-
curs="unbounded">
                                                   <xs:complexType>
                                                      <xs:choice>
                                                         <xs:element name="Grundgebuehr">
```

```
                                                <xs:complexType>
                                                   <xs:sequence>
                                                      <xs:element name="Berechnungsintervall"
type="Berechnungsintervall-Grundgebuehr"/>
                                                   </xs:sequence>
                                                </xs:complexType>
                                             </xs:element>
                                             <xs:element name="LeistungsbezogeneGebuehr">
                                                <xs:complexType>
                                                   <xs:sequence>
                                                      <xs:element name="Berechnungsbasis-Leis-
tung" type="Berechnungsbasis-Leistungsbezug"/>
                                                      <xs:element name="Dienstzurechnung"
type="Dienstnutzer" maxOccurs="unbounded"/>
                                                   </xs:sequence>
                                                </xs:complexType>
                                             </xs:element>
                                             <xs:element name="MengenbezogeneGebuehr">
                                                <xs:complexType>
                                                   <xs:sequence>
                                                      <xs:element name="Berechnungsbasis-Menge"
type="Berechnungsbasis-Mengenbezug" maxOccurs="unbounded"/>
                                                      <xs:element name="Dienstzurechnung"
type="Dienstnutzer" maxOccurs="unbounded"/>
                                                   </xs:sequence>
                                                </xs:complexType>
                                             </xs:element>
                                          </xs:choice>
                                       </xs:complexType>
                                    </xs:element>
                                    <xs:element name="Kontrakt-Tarif">
                                       <xs:complexType>
                                          <xs:choice>
                                             <xs:element name="Kontrakt">
                                                <xs:complexType>
                                                   <xs:sequence>
                                                      <xs:element name="KontraktID" type="xs:string"/
>
                                                      <xs:element name="KundenID" type="xs:string"/
>
                                                   </xs:sequence>
                                                </xs:complexType>
                                             </xs:element>
                                             <xs:element name="Tarif">
                                                <xs:complexType>
                                                   <xs:sequence>
                                                      <xs:element name="TarifID" type="xs:string"/>
                                                      <xs:element name="Periode">
                                                         <xs:complexType>
                                                            <xs:sequence>
                                                            <xs:element name="Beginn"
type="xs:dateTime"/>
                                                            <xs:element name="Ende"
type="xs:dateTime"/>
```

```
                                                                    </xs:sequence>
                                                                </xs:complexType>
                                                            </xs:element>
                                                        </xs:sequence>
                                                    </xs:complexType>
                                                </xs:element>
                                            </xs:choice>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="BillingPolicy" maxOccurs="unbounded">
                            <xs:complexType name="Rechnungsstellung">
                                <xs:choice>
                                    <xs:element name="Periodisch" type="Rechnungsperiode"/>
                                    <xs:element name="Einzeln"/>
                                </xs:choice>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="PaymentPolicy" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="Zahlungszeitpunkt" type="Zahlungszeit-
punkt"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:simpleType name="Berechnungsintervall-Grundgebuehr">
    <xs:restriction base="xs:string">
        <xs:enumeration value="jaehrlich"/>
        <xs:enumeration value="quartalsweise"/>
        <xs:enumeration value="monatlich"/>
        <xs:enumeration value="woechentlich"/>
        <xs:enumeration value="taeglich"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Berechnungsbasis-Leistungsbezug">
    <xs:restriction base="xs:string">
        <xs:enumeration value="je Darstellung"/>
        <xs:enumeration value="je Sitzung"/>
        <xs:enumeration value="je Download"/>
```

```
            <xs:enumeration value="je Transaktion"/>
            <xs:enumeration value="je QoS Reservierung"/>
            <xs:enumeration value="je Klick"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Berechnungsbasis-Mengenbezug">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Dauer"/>
            <xs:enumeration value="Volumen"/>
            <xs:enumeration value="Qualitaet"/>
            <xs:enumeration value="Teilnehmer"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Dienstnutzer">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Sender"/>
            <xs:enumeration value="Empfaenger"/>
            <xs:enumeration value="Anfrager"/>
            <xs:enumeration value="Anbieter"/>
            <xs:enumeration value="Dritter"/>
            <xs:enumeration value="Alle"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Rechnungsperiode">
        <xs:restriction base="xs:string">
            <xs:enumeration value="jaehrlich"/>
            <xs:enumeration value="quartalsweise"/>
            <xs:enumeration value="monatlich"/>
            <xs:enumeration value="wochentlich"/>
            <xs:enumeration value="taeglich"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="Zahlungszeitpunkt">
        <xs:restriction base="xs:string">
            <xs:enumeration value="vor Diensterbringung"/>
            <xs:enumeration value="nach Diensterbringung"/>
            <xs:enumeration value="gleichzeitig"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

C.2 XML Spezifikation des Geschäftsmodells von NRW-On

```
<?xml version="1.0" encoding="UTF-8"?>
<Geschaeftsmodell xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSche-
maLocation="http://www.kom.e-technik.tu-darmstadt.de/~rensing/schema-geschaeftsmodell.xsd">
   <Geschaeftsmodell-ID>Business-Modell-VT-Konf</Geschaeftsmodell-ID>
   <Dienstanbieter>NRW-On</Dienstanbieter>
   <Dienst-Dienstnutzer-Policy>
      <Dienst>
         <Dienst-ID>5555</Dienst-ID>
         <Dienstbezeichnung>Videokonferenz</Dienstbezeichnung>
      </Dienst>
      <Dienst>
         <Dienst-ID>5554</Dienst-ID>
         <Dienstbezeichnung>Telefonkonferenz</Dienstbezeichnung>
      </Dienst>
      <Dienstnutzer-Policy>
         <Dienstnutzer>
            <Anonym>false</Anonym>
            <Privat>false</Privat>
            <Merkmale>
               <GruppenID>Pivatkunde</GruppenID>
            </Merkmale>
         </Dienstnutzer>
         <KaufmaennischeRegeln>
            <PricingPolicy>
               <Preiskomponente>
                  <MengenbezogeneGebuehr>
                     <Berechnungsbasis-Menge>Teilnehmer</Berechnungsbasis-Menge>
                     <Berechnungsbasis-Menge>Dauer</Berechnungsbasis-Menge>
                     <Dienstzurechnung>Anfrager</Dienstzurechnung>
                  </MengenbezogeneGebuehr>
               </Preiskomponente>
               <Kontrakt-Tarif>
                  <Tarif>
                     <TarifID>Privatkunde</TarifID>
                     <Periode>
                        <Beginn>2003-01-01T00:00:00</Beginn>
                        <Ende>2003-06-30T24:00:00</Ende>
                     </Periode>
                  </Tarif>
               </Kontrakt-Tarif>
            </PricingPolicy>
            <BillingPolicy>
               <Einzeln/>
            </BillingPolicy>
            <PaymentPolicy>
               <Zahlungszeitpunkt>gleichzeitig</Zahlungszeitpunkt>
            </PaymentPolicy>
         </KaufmaennischeRegeln>
      </Dienstnutzer-Policy>
      <Dienstnutzer-Policy>
         <Dienstnutzer>
            <Anonym>false</Anonym>
```

```
            <Privat>false</Privat>
            <Merkmale>
               <GruppenID>Geschaeftskunde</GruppenID>
            </Merkmale>
         </Dienstnutzer>
         <KaufmaennischeRegeln>
            <PricingPolicy>
               <Preiskomponente>
                  <Grundgebuehr>
                     <Berechnungsintervall>monatlich</Berechnungsintervall>
                  </Grundgebuehr>
               </Preiskomponente>
               <Preiskomponente>
                  <MengenbezogeneGebuehr>
                     <Berechnungsbasis-Menge>Teilnehmer</Berechnungsbasis-Menge>
                     <Berechnungsbasis-Menge>Dauer</Berechnungsbasis-Menge>
                     <Dienstzurechnung>Anfrager</Dienstzurechnung>
                  </MengenbezogeneGebuehr>
               </Preiskomponente>
               <Kontrakt-Tarif>
                  <Tarif>
                     <TarifID>Geschaeftskunde</TarifID>
                     <Periode>
                        <Beginn>2003-01-01T00:00:00</Beginn>
                        <Ende>2003-06-30T24:00:00</Ende>
                     </Periode>
                  </Tarif>
               </Kontrakt-Tarif>
            </PricingPolicy>
            <BillingPolicy>
               <Periodisch>monatlich</Periodisch>
            </BillingPolicy>
            <PaymentPolicy>
               <Zahlungszeitpunkt>nach Diensterbringung</Zahlungszeitpunkt>
            </PaymentPolicy>
         </KaufmaennischeRegeln>
      </Dienstnutzer-Policy>
   </Dienst-Dienstnutzer-Policy>
</Geschaeftsmodell>
```

C.3 Die Policy-Sprache zur Beschreibung der kaufmännischen Unterstützungsdienste

Überblick über die Sprachelemente.

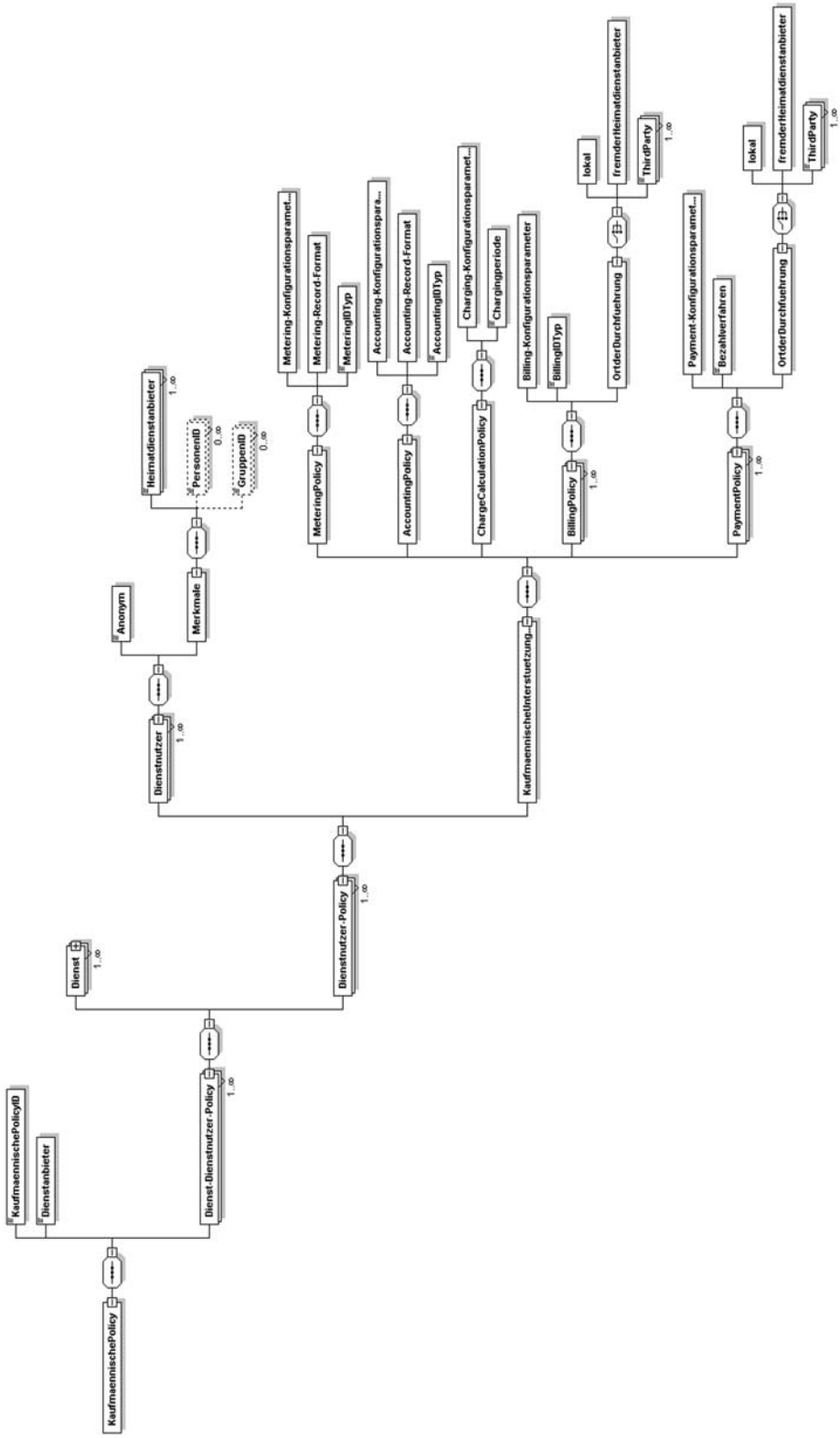

Abbildung 88: Elemente der Policy-Sprache für kaufmännische Unterstützungsdienste

XML-Schema der Policy-Sprache.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attribute-
FormDefault="unqualified">
   <xs:element name="KaufmaennischePolicy">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="KaufmaennischePolicyID" type="xs:string"/>
            <xs:element name="Dienstanbieter" type="xs:anyURI"/>
            <xs:element name="Dienst-Dienstnutzer-Policy" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="Dienst" maxOccurs="unbounded">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="Dienst-ID" type="xs:string"/>
                              <xs:element name="Dienstbezeichnung" type="xs:string"/>
                              <xs:element name="Lokation" type="xs:anyURI" minOccurs="0"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                     <xs:element name="Dienstnutzer-Policy" maxOccurs="unbounded">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="Dienstnutzer" maxOccurs="unbounded">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="Anonym" type="xs:boolean"/>
                                       <xs:element name="Merkmale">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="Heimatdienstanbieter" type="xs:string"
maxOccurs="unbounded"/>
                                                <xs:element name="PersonenID" type="xs:string" minOc-
curs="0" maxOccurs="unbounded"/>
                                                <xs:element name="GruppenID" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
                                             </xs:sequence>
                                          </xs:complexType>
                                       </xs:element>
                                    </xs:sequence>
                                 </xs:complexType>
                              </xs:element>
                              <xs:element name="KaufmaennischeUnterstuetzungsdienste">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="MeteringPolicy">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="Metering-Konfigurationsparameter"/>
                                                <xs:element name="Metering-Record-Format"/>
                                                <xs:element name="MeteringIDTyp" type="MeteringIDTyp"/>
```

```
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="AccountingPolicy">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Accounting-Konfigurationsparameter"/>
                            <xs:element name="Accounting-Record-Format"/>
                            <xs:element name="AccountingIDTyp" type="AccountingIDTyp"/
>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="ChargeCalculationPolicy">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Charging-Konfigurationsparameter"/>
                            <xs:element name="Chargingperiode" type="Chargingperiode"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="BillingPolicy" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Billing-Konfigurationsparameter"/>
                            <xs:element name="BillingIDTyp" type="BillingIDTyp"/>
                            <xs:element name="OrtderDurchfuehrung">
                                <xs:complexType>
                                    <xs:choice>
                                        <xs:element name="lokal"/>
                                        <xs:element name="fremderHeimatdienstanbieter"/>
                                        <xs:element name="ThirdParty" type="xs:anyURI"
maxOccurs="unbounded"/>
                                    </xs:choice>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="PaymentPolicy" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Payment-Konfigurationsparameter"/>
                            <xs:element name="Bezahlverfahren" type="Bezahlverfahren"/>
                            <xs:element name="OrtderDurchfuehrung">
                                <xs:complexType>
                                    <xs:choice>
                                        <xs:element name="lokal"/>
                                        <xs:element name="fremderHeimatdienstanbieter"/>
                                        <xs:element name="ThirdParty" type="xs:anyURI"
maxOccurs="unbounded"/>
                                    </xs:choice>
                                </xs:complexType>
                            </xs:element>
```

```
                                        </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="Chargingperiode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="einzeln"/>
    <xs:enumeration value="monatlich"/>
    <xs:enumeration value="wochentlich"/>
    <xs:enumeration value="taeglich"/>
    <xs:enumeration value="stuendlich"/>
    <xs:enumeration value="minuetlich"/>
    <xs:enumeration value="sekuendlich"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="MeteringIDTyp">
  <xs:restriction base="xs:string">
    <xs:enumeration value="IP-Adresse"/>
    <xs:enumeration value="Benutzerkennung"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="AccountingIDTyp">
  <xs:restriction base="xs:string">
    <xs:enumeration value="IP-Adresse"/>
    <xs:enumeration value="Teilnehmeranschlussnummer"/>
    <xs:enumeration value="Benutzerkennung"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="BillingIDTyp">
  <xs:restriction base="xs:string">
    <xs:enumeration value="KundenID"/>
    <xs:enumeration value="TemporaereID"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="Bezahlverfahren">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Rechnungsbasiert"/>
    <xs:enumeration value="GuthabenkontoSoftware"/>
    <xs:enumeration value="GuthabenkontoHardware"/>
    <xs:enumeration value="Kreditkartenverfahren"/>
    <xs:enumeration value="elektronischKontenbasiert"/>
    <xs:enumeration value="elektronischGeldartig"/>
    <xs:enumeration value="elektronischDialer"/>
```

```
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

C.4 Die Policy-Sprache zur Beschreibung von Zugriffskontrolldiensten

Überblick über die Sprachelemente.

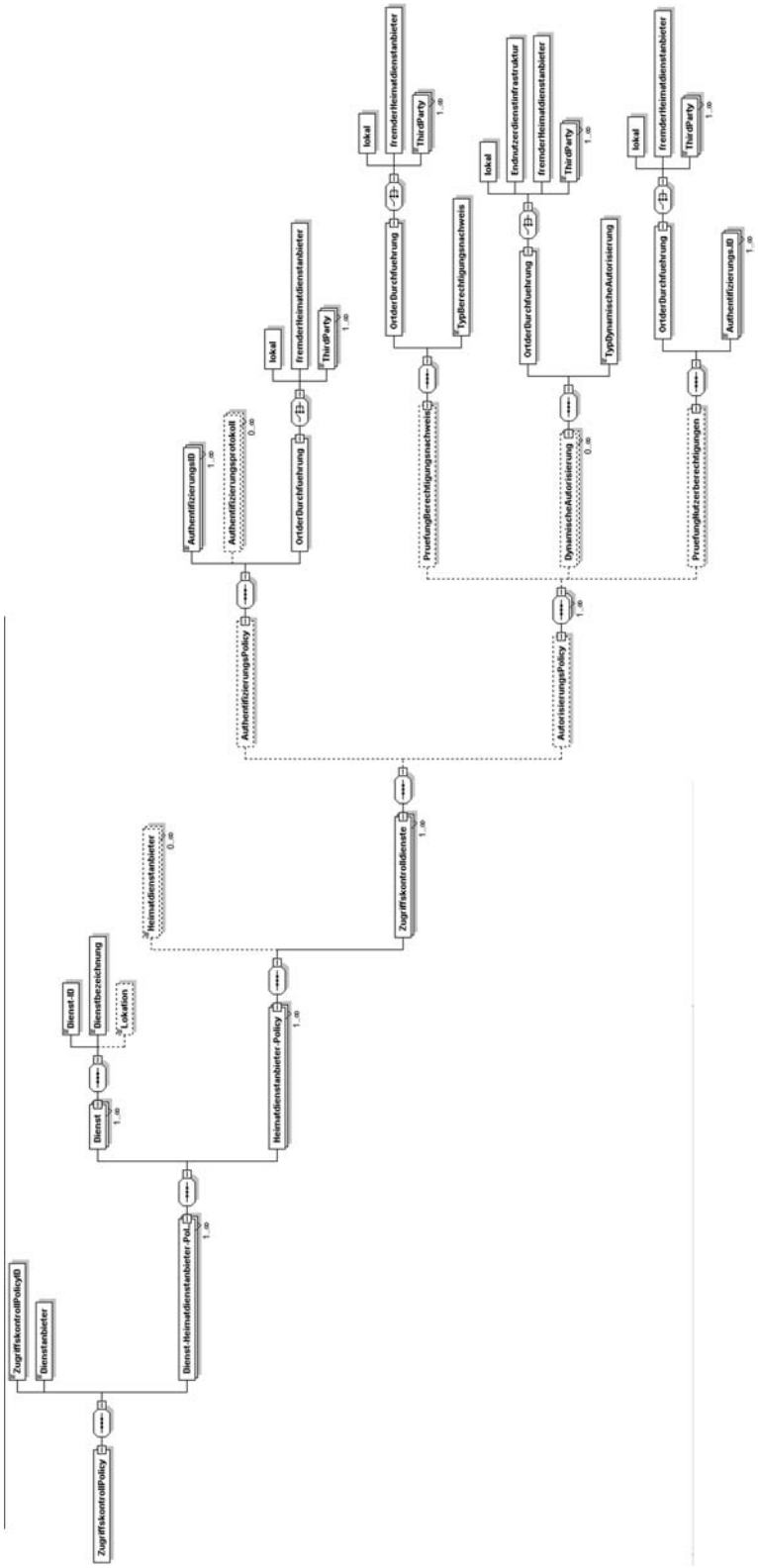

Abbildung 89: Elemente der Policy-Sprache für Zugriffskontrolldienste

XML-Schema der Policy-Sprache.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attribute-
FormDefault="unqualified">
   <xs:element name="ZugriffskontrollPolicy">
      <xs:complexType>
         <xs:sequence>
            <xs:element name="ZugriffskontrollPolicyID" type="xs:string"/>
            <xs:element name="Dienstanbieter" type="xs:anyURI"/>
            <xs:element name="Dienst-Heimatdienstanbieter-Policy" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name="Dienst" maxOccurs="unbounded">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="Dienst-ID" type="xs:string"/>
                              <xs:element name="Dienstbezeichnung" type="xs:string"/>
                              <xs:element name="Lokation" type="xs:anyURI" minOccurs="0"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                     <xs:element name="Heimatdienstanbieter-Policy" maxOccurs="unbounded">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element name="Heimatdienstanbieter" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
                              <xs:element name="Zugriffskontrolldienste" maxOccurs="unbounded">
                                 <xs:complexType>
                                    <xs:sequence>
                                       <xs:element name="AuthentifizierungsPolicy" minOccurs="0">
                                          <xs:complexType>
                                             <xs:sequence>
                                                <xs:element name="AuthentifizierungsID" type="Authentifi-
zierungsID" maxOccurs="unbounded"/>
                                                <xs:element name="Authentifizierungsprotokoll" minOccurs="0"
maxOccurs="unbounded"/>
                                                <xs:element name="OrtderDurchfuehrung">
                                                   <xs:complexType>
                                                      <xs:choice>
                                                         <xs:element name="lokal"/>
                                                         <xs:element name="fremderHeimatdienstanbieter"/>
                                                         <xs:element name="ThirdParty" type="xs:anyURI"
maxOccurs="unbounded"/>
                                                      </xs:choice>
                                                   </xs:complexType>
                                                </xs:element>
                                             </xs:sequence>
                                          </xs:complexType>
                                       </xs:element>
                                       <xs:element name="AutorisierungsPolicy" minOccurs="0">
                                          <xs:complexType>
                                             <xs:sequence maxOccurs="unbounded">
                                                <xs:element name="PruefungBerechtigungsnachweis" minOc-
```

```
curs="0">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="OrtderDurchfuehrung">
                                                    <xs:complexType>
                                                        <xs:choice>
                                                            <xs:element name="lokal"/>
                                                            <xs:element name="fremderHeimatdienstanbi-
eter"/>
                                                            <xs:element name="ThirdParty"
type="xs:anyURI" maxOccurs="unbounded"/>
                                                        </xs:choice>
                                                    </xs:complexType>
                                                </xs:element>
                                                <xs:element name="TypBerechtigungsnachweis"
type="Berechtigungsnachweis"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:element name="DynamischeAutorisierung" minOccurs="0"
maxOccurs="unbounded">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="OrtderDurchfuehrung">
                                                    <xs:complexType>
                                                        <xs:choice>
                                                            <xs:element name="lokal"/>
                                                            <xs:element name="Endnutzerdienstinfrastruk-
tur"/>
                                                            <xs:element name="fremderHeimatdienstanbi-
eter"/>
                                                            <xs:element name="ThirdParty"
type="xs:anyURI" maxOccurs="unbounded"/>
                                                        </xs:choice>
                                                    </xs:complexType>
                                                </xs:element>
                                                <xs:element name="TypDynamischeAutorisierung"
type="DynamischeKomponente"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:element name="PruefungNutzerberechtigungen" minOc-
curs="0">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="OrtderDurchfuehrung">
                                                    <xs:complexType>
                                                        <xs:choice>
                                                            <xs:element name="lokal"/>
                                                            <xs:element name="fremderHeimatdienstanbi-
eter"/>
                                                            <xs:element name="ThirdParty"
type="xs:anyURI" maxOccurs="unbounded"/>
                                                        </xs:choice>
```

```
                                                    </xs:complexType>
                                                </xs:element>
                                                <xs:element name="Authentifizierungs.ID"
type="AuthentifizierungsID" maxOccurs="unbounded"/>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="Berechtigungsnachweis">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Kreditkarte"/>
      <xs:enumeration value="GuthabenkontoHardware"/>
      <xs:enumeration value="MicropaymentGeldartig"/>
      <xs:enumeration value="Ticket"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="DynamischeKomponente">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Kontenstand"/>
      <xs:enumeration value="Kreditkartengueltigkeit"/>
      <xs:enumeration value="Systemzustand"/>
      <xs:enumeration value="Pruefung-Gruppenzugehoerigkeit"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="AuthentifizierungsID">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Teilnehmeranschlussnummer"/>
      <xs:enumeration value="Benutzerkennung"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

C.5 XML-Schema für eine Session-Beschreibung

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attribute-
FormDefault="unqualified">
  <xs:element name="Sessionbeschreibung">
    <xs:annotation>
      <xs:documentation>Comment describing your root element</xs:documentation>
```

```
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="SessionID" type="xs:string"/>
          <xs:element name="Session-Beginn" type="xs:dateTime"/>
          <xs:element name="Session-Ende" type="xs:dateTime" minOccurs="0"/>
          <xs:element name="Authentifizierungs-Status" type="xs:string" minOccurs="0"/>
          <xs:element name="Authentifizierungs-Protokoll" type="xs:string" minOccurs="0"/>
          <xs:element name="Authentifizierungs-Ort" type="xs:string" minOccurs="0"/>
          <xs:element name="UserID" type="xs:anyURI" minOccurs="0"/>
          <xs:element name="GeraeteID-Typ" type="GeraeteID" minOccurs="0"/>
          <xs:element name="GeraeteID" type="xs:string" minOccurs="0"/>
          <xs:element name="Session-Kind" type="xs:string" minOccurs="0"/>
          <xs:element name="Session-Eltern" type="xs:string" minOccurs="0"/>
          <xs:element name="Dienstspezifika" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Dienstanbieter" type="xs:string"/>
                <xs:element name="DienstID" type="xs:integer"/>
                <xs:element name="Dienst-Bezeichnung" type="xs:string"/>
                <xs:element name="GruppenID" type="xs:string" minOccurs="0"/>
                <xs:element name="MeteringIDTyp" type="MeteringIDTyp" minOccurs="0"/>
                <xs:element name="MeteringID" type="xs:string" minOccurs="0"/>
                <xs:element name="AccountingIDTyp" type="AccountingIDTyp" minOccurs="0"/>
                <xs:element name="AccountingID" type="xs:string" minOccurs="0"/>
                <xs:element name="Billing-Ort" type="xs:string" minOccurs="0"/>
                <xs:element name="Payment-Ort" type="xs:string" minOccurs="0"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:simpleType name="MeteringIDTyp">
      <xs:restriction base="xs:string">
        <xs:enumeration value="IP-Adresse"/>
        <xs:enumeration value="Benutzerkennung"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="AccountingIDTyp">
      <xs:restriction base="xs:string">
        <xs:enumeration value="IP-Adresse"/>
        <xs:enumeration value="Teilnehmeranschlussnummer"/>
        <xs:enumeration value="Benutzerkennung"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="GeraeteID">
      <xs:restriction base="xs:string">
        <xs:enumeration value="IP-Adresse"/>
        <xs:enumeration value="Teilnehmeranschlussnummer"/>
      </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Index

T

V

Z

Lebenslauf

Name:	Christoph Rensing
Geburtstag:	04.07.1967

Schule und Studium

1973 - 1977	Grundschule Düsseldorf
1977 - 1986	Görres-Gymnasium Düsseldorf Abschluß: Abitur 06/1986
1987 - 1993	Studium Wirtschaftsinformatik, Universität Mannheim Abschluß: Diplom-Wirtschaftsinformatiker, 3/1993

Berufliche Tätigkeit

04/1993 - 02/1998	Geschäftstführender Gesellschafter der Klink & Rensing Systemberatungsgesellschaft mbH
seit 03/1998	Wissenschaftlicher Mitarbeiter im Fachbereich Elektrotechnik und Informationstechnik der Technischen Universität Darmstadt
seit 01/1999	Geschäftsführer des Hessischen Telemedia Technologie Kompetenz-Centers / httc e.V.