Das Papier wurde vorgestellt auf dem 5. Workshop *Sicherheit in vernetzten Systemen* des DFN-CERT und DFN-PCA, Hamburg, 4.-5. März 1998

# Smart Card gesicherte Web-Umgebung und rollenbasierte Zugriffskontrolle im administrativen Bereich[1]

**Teil 1 des Beitrages: *"Abgesicherte Internet-Umgebungen mit Hilfe rollenbasierter Zugriffsmechanismen für WWW- und Email-Dienste"***

**Jörg Bartholdt <panther@prz.tu-berlin.de>**
**Klaus Nagel <klausn@prz.tu-berlin.de>**

**Technische Universität Berlin**

**Februar 1998**

## 1.0 Einleitung

In einer öffentlichen Verwaltung wie der TU Berlin werden viele administrative Vorgänge noch per Hand und Formular abgewickelt. Auch wenn in den Abteilungen bereits EDV-gestützte Systeme eingesetzt werden, geschieht der Informationsaustausch mit anderen Abteilungen noch über Hauspost, Fax oder Telefon. Bei schützenswerten Informationen (insbesondere personenbezogenen Daten) ist eine schnelle Auskunft selbst per Telefon nicht möglich.

Die Vernetzung der bestehenden Systeme mit Hilfe des Internet als preiswertes Transportmedium kann hier Abhilfe schaffen, stößt aber auf datenschutzrechtliche Bedenken. Unbefugte können Daten mitlesen, abfragen, gefälschte Daten einspielen, etc.

Eine Lösung ist der Aufbau eines virtuellen privaten Netzwerks, zu denen nur Befugte Zutritt erhalten können. Innerhalb des privaten Netzwerks ist dann eine differenzierte Zugriffskontrolle notwendig, damit jeder Angestellte nur an die für ihn vorgesehenen Daten herankommen kann.

Unser Ansatz ist die Implementierung eines TRUSTCENTER, mit dessen Hilfe ein virtuelles privates Netz aufgebaut werden kann, das den Sicherheitsanforderungen von Datenschützern, Personalrat, Mitarbeitern, etc. entspricht. Der Kommunikationsablauf in der Verwaltung soll damit beschleunigt, die Verfügbarkeit von Informationen erhöht werden, ohne Unbefugten Zugriff zu diesen Informationen zu ermöglichen.

Kern des TRUSTCENTER ist der TRUSTCENTER-Rechner. Er hat die Aufgabe, die Internet-Infrastruktur zu nutzen, um den Mitarbeitern Sicherheitsdienste anzubieten.

Das TRUSTCENTER ist zugleich eine Zertifizierungsstelle für öffentliche Schlüssel der Mitarbeiter und versorgt den TRUSTCENTER-Rechner mit diesen Informationen.

1. Die Arbeit entstand teilweise im Rahmen des ESPRIT-Projektes E2S (End-to-End Security over the Internet). Weitere Informationen: http://www.e2s.com

## 2.0 Zugriffskontrolle

Die Zugriffskontrolle ist eine Funktion, die mit Hilfe von Zugriffskontrollinformationen, Regelwerken und Parametern des Zugriffs (kontextabhängig) diesen erlauben oder verbieten kann. Auf die Entscheidung können viele Faktoren Einfluß haben:

- Initiator und Ziel des Zugriffs
- Gruppen, zu denen Initiator und Ziel gehören
- Sicherheitsstufen (*levels*) von Initiator und Ziel
- Rollen von Initiator und Ziel
- die Art des Zugriffs
- Kontexte (z.B. Ort und Zeit)
- oder Kombinationen davon.

Die Summe der Regeln wird als Sicherheitspolitik (*Security Policy*) bezeichnet. So werden die Bedingungen festgelegt, unter denen ein Initiator auf ein Ziel zugreifen darf.

Man unterscheidet Sicherheitspolitiken nach verschiedenen Merkmalen:

**Wer** vergibt die Rechte?

- *mandatory policy* – Ein System (z.B. das Betriebssystem) vergibt und erzwingt generelle Regeln für die Zugriffskontrolle.
- *discretionary policy* - Je Ziel wird eine Zugriffkontrolle vom Eigentümer konfiguiert (Beispiel ist das Dateisystem unter UNIX).

**Wie** werden die Rechte vergeben?

- *Identitätsbasiert* - Je nach Initiatoren und Ziel.
- *Kontextbasiert* - Nach äußeren Umständen (z.B. zu welcher Tageszeit, von welchem Rechner).

**Wo** werden die Rechte zugeordnet/gespeichert?

- Beim Ziel - Access-Control-Matrizen (z.B. beim UNIX-Dateisystem).
- Beim Initiator - Vollmachten (*Credentials* oder *Tickets*), die bei einem Zugriff mitgegeben werden.
- Gemischt - Sicherheitsstufen, Sicherheitskennzeichen (*Security levels*, *Security labels*).

Die Implementierung von Zugriffskontrollen kann sich verschiedener Mittel bedienen. In herkömmlichen Zugriffskontrollmodellen werden die Zugriffskontrollinformationen fast ausschließlich in Zugriffskontrollisten gehalten, da die Integritätssicherung der Zugriffskontrollinformationen entfällt.

Für eine Zugriffskontrolle ist entscheidend, daß der Initiator mit hinreichender Sicherheit identifiziert werden kann. Ein Rechnersystem ist im allgemeinen nur in der Lage,

einen Benutzer zu authentifizieren, nicht aber zu identifizieren. Diese beiden Begriff haben eine unterschiedliche Bedeutung:

Üblicherweise erhält man Zugang zu einem Rechnersystem mit Hilfe einer UserID und einem Paßwort. Es wird davon ausgegangen, daß das Wissen des Paßwortes die Person eindeutig identifiziert, so daß diese Person für alle Aktionen, Dateien, Prozesses, etc. unter dieser UserID verantwortlich ist. Ein Paßwort kann aber einer anderen Person mitgeteilt werden, oder sie erhält es auf unrechtmäßige Weise. Dann kann sie sich unter dieser UserID einloggen, ohne daß das System dieses Eindringen erkennen kann.


Abbildung 1 – Authentifikation und Identifikation

## 2.1 Rollenbasierte Zugriffskontrolle

Problematisch wird die Zuordnung eines Zugriffsrechts für ein Paar (Initiator und Ziel, siehe), wenn es viele Initiatoren und viele Ziele gibt. Bei *n* Initiatoren und *m* Zielen gibt es *n*m* Zugriffsrechtsbeschreibungen. Nehmen wir an, ein Angestellter hat Zugriff auf fünf Ziele (Datenbanken, Informationsdienste, etc.). Sollte dieser die Organisation verlassen, müssen alle fünf Administratoren der Ziele alle Rechte dieses Angestellten löschen und für einen Neueingestellen dann wieder einrichten. Das ist aufwendig und fehlerträchtig.

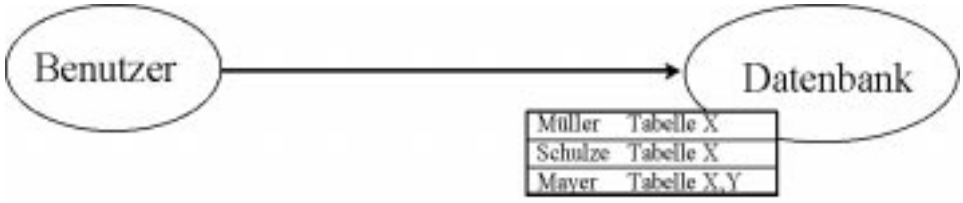



Abbildung 2 – Zugriffskontrolle einer Datenbank

Für große Systeme bietet sich statt eines personen-bezogenen ein rollen-basiertes System an, da nicht das Individuum entscheidend für die Zugriffskontrolle ist, sondern die Rolle, die es innerhalb der Organisation besetzt.

Rollen sind eine erweiterte *Sichtweise* des Themas Zugriffskontrolle; sie sind nicht zu verwechseln mit z.B. Zugriffskontrollisten, die ein *Mittel* zur Implementierung von Zugriffskontrollen sind.

Rollen dienen der Zusammenfassung von inhaltlich zusammengehörenden Rechten wie in Abbildung 3 gezeigt, z.B. das Lese- und Schreibrecht auf bestimmte Daten in der Datenbank. Rollen können hierarchisch strukturiert sein, d.h. man kann den Rollen “Forschungsleiter Projekt A” mit dem Lese- und Schreibrecht auf Forschungsdaten des Projekts A und “Budgetleiter Projekt A” mit dem Lese- und Schreibrecht auf die Finanzdaten von Projekt A eine übergeordnete Rolle “Projektleiter A” zuordnen. Jemand, der diese Rolle zugeordnet bekommt, besitzt damit auch die untergeordneten Rollen.

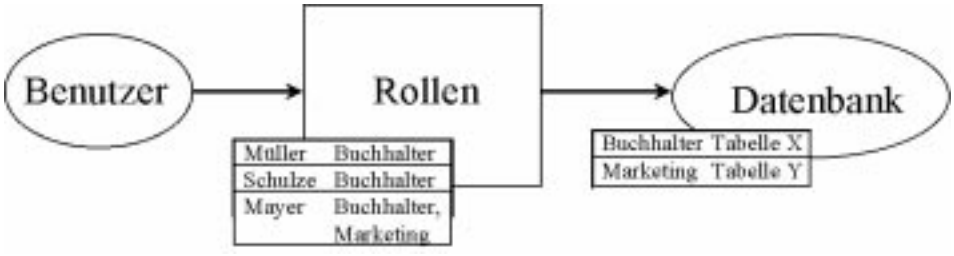



Abbildung 3 – Rollenbasierte Zugriffskontrolle einer Datenbank

Rollen können als einander ausschließend markiert werden. Es gibt Rollen, die semantisch unvereinbar sind. Das wären in einer Bank zum Beispiel die Rolle des “Kassierers” und des “Kassenprüfers”. Ein Kassierer kann nicht gleichzeitig seine Entlastung erklären!

Rollen-basierte Zugriffskontrolle ist ein stark diskutiertes Forschungsthema, bei dem es zu einigen Problemen noch keine allgemein anerkannten, klaren Lösungen gibt.

Eine strenge Hierarchisierung (baumartige Struktur) ergibt ein einfaches System, wirft aber einige Probleme auf. Eine Möglichkeit bei einer rollenorientierten Sichtweise ist es, zwei Rollen als einander ausschließend zu deklarieren. Eine Person darf nicht gleichzeitig beide Rollen innehaben. Ein weiteres Problem ist die Administration der Rollen selbst. Rollen sollen die Zugriffsmöglichkeiten ihrer Inhaber auf die notwendigen, erlaubten Informationsbestände und Dienste beschränken. In der Fachliteratur wird oft von einer statischen Rollenverteilung ausgegangen, da ausgespart bleibt wie die Administration der Rollenstruktur zugriffskontrolliert wird. Wird ein Rollenadministrator, der vollen Zugriff auf die Zuordnung von Personen zu Rollen und der Hierarchisierung der Rollen untereinander hat, mit der Verwaltung beauftragt, ist dieser in der Lage, jeden Dienst selbst in Anspruch zu nehmen. Da zur Betreuung und Pflege des Systems zusätzliche Arbeitskräfte benötigt werden, muß untersucht werden, wie ein Mißbrauch verhindert werden kann.

Abgesehen von dem oben gesagten wird *eine* Person allein in großen Organisationen nicht in der Lage sein, alle Rollen mit ihren Rechten und Pflichten zu kennen. Eine zentrale Administration des Rollensystems ist in diesem Fall sinnlos, da diese eine Person wieder nur auf Weisungen von anderer Seite angewiesen ist.

Das obige Bankbeispiel besagt, daß die Rolle eines “Kassierers” mit der des “Kassenprüfers” unvereinbar ist. Innerhalb einer Filiale ist das sicherlich korrekt. Wohl aber wäre es möglich, daß der “Kassierer” der Filiale A, “Kassenprüfer” der Filiale B wird. Eine Rolle muß also auch in ihrem Kontext ausgewertet werden und kann zu unterschiedlichen Ergebnissen bzgl. der Widersprüchlichkeit von Rollen und der Erteilung von Rechten führen. Ein möglicher Ansatz wird von Luigi Giuri und Pietro Iglio in ihrem Artikel “Role Templates for Content-Based Access Control” verfolgt, die die Bildung von Rollenvorlagen (*role templates*) vorschlagen. Eine Rollenvorlage mit dem Kontext in Form eines oder mehrerer Parameter wird zu einer Instanz der Rolle. Die Vorlage “Kassierer” wird zusammen mit dem Parameter “Filiale A” zur Rolle “Kassierer(Filiale A)”. Diese unterscheidet sich hinsichtlich der Rechte von “Kassierer(Filiale

B)” und vor allem hinsichtlich der Widersprüchlichkeit gegenüber den Rollen “Kassenprüfer(Filiale A)” und “Kassenprüfer(Filiale B)”.

Diese Problematik wird durch Roshan Thomas von einer anderen Seite angegangen. Sein Vorschlag “TEAM based Access Control” versucht, erst eine Strukturierung des Kontexts (in sog. Teams) vorzunehmen und innerhalb des Kontexts dann Rollen zu verwenden. Der Vorschlag über Rollenvorlagen baut erst die Rollenvorlagenhierarchie auf und instanziiert sie mit den Kontexten. Diese Idee verwenden auch wir in unserem Modellierungsansatz.

## 3.0 Anwendungsszenarien

Bei der Entwicklung des Zugriffskontrollsystems standen die Anforderungen aus Verwaltungen insbesondere der TU-Verwaltung im Vordergrund. Aber auch Anwendungsszenarien durch das E2S-Projekt wurden herausgearbeitet. Im folgenden werden nur zwei kleine Beispiele angegeben.

Beispiel 1: Projektkontendatenbank

An der Technischen Universität Berlin werden von den Instituten und Einrichtungen wie der Prozeßrechnerverbundzentrale (PRZ) zahlreiche Drittmittelprojekte durchgeführt, die zentral und hauptverantwortlich von dem Forschungsreferat der Zentralen Universitätsverwaltung (ZUV) verwaltet werden. So erhalten die Projektleiter in der Regel nur einmal monatlich die Kontoauszüge ihrer Projekte, anhand derer sie die institutseigene Kontoführung überprüfen können. Durch Einführung unseres Systems hätten die Projektleiter zu jeder Zeit Zugriff auf den aktuellen Kontostand ihrer Projekte.

Beispiel 2: Kunde-Verkäufer-Verhältnis

Das E2S-Projekt behandelt den Bereich Business-to-Business Electronic-Commerce. Grundszenario ist, daß ein Kunde über das Internet einen Katalog durchblättern und ggf. für ihn spezielle Preislisten einsehen kann, dazu ist eine Kunden-Authentifizierung zwingend notwendig. Der Kataloganbieter nimmt die Bestellungen entgegen und reicht sie an die Lieferanten weiter. Dabei kann der Kataloganbieter auf eigene Rechnung oder im Auftrag des Lieferanten einen Katalog im Internet offerieren. Das Unterschreiben einer Bestellung durch einen Mitarbeiter oder eine Mitarbeiterin einer Firma verpflichtet aber nicht die Person zum Kauf, sondern die Firma. Er/sie handelt in der Rolle des Einkäufers für dieses Unternehmen. Sollte er/sie die Firma verlassen, bleibt die Vertragsbindung zwischen Firma und Lieferant bestehen. Ggf. betreut dann ein anderer Mitarbeiter dieses Geschäft. Private Schlüssel können dann nicht nur für Personen sinnvoll sein, sondern auch für Rollen innerhalb von Organisationen, die für die Organisation nach außen auftreten und bestimmte Kompetenzen haben. Mit Hilfe des Zugriffskontrollsystems läßt sich die Bestellung dann durch einen der Rolle zugeordneten Schlüssel unterschreiben. Die rechtlichen Konsequenzen dieser Unterschrift trägt die Firma. Das Unternehmens-TRUSTCENTER überprüft, ob die Bestellung die Kompetenzen des Mitarbeiters nicht überschreitet und signiert mit dem Rollenschlüssel die Bestellung. Der Einkäufer hat so nie Kenntnis vom geheimen Schlüssel. Der TRUSTCENTER-Dienst kann von dem Kataloganbieter auch als vertrauenswürdige Dienstleistung gegenüber den Kunden erbracht werden.

# 4.0 Modellierung

Um ein Zugriffskontrollsystem implementieren zu können, bedarf es zunächst eines Modells einer Verwaltungsorganisation. Mit Hilfe des Modells soll die Organisation hinsichtlich relevanter Daten für die Zugriffskontrolle beschrieben werden. Das Modell soll Mittel für diese Beschreibung liefern, die eine leichte Übertragung der Informationen in ein Computersystem ermöglichen.

Bei diesem Modell handelt es sich um einen objektorientierten Ansatz. Der schon beschriebene Nachteil, daß ein rollenbasiertes Zugriffskontrollsystem mit steigender Anzahl der Rollen unübersichtlicher wird, legt eine weitere Strukturierung der Rollen nahe. Zunächst werden Einheiten einer Organisation identifiziert. Als Einheit kommen einzelne Stellen, Personen, Arbeitsgruppen, Projekte, Datenbanken, Fachbereiche, Abteilungen und so weiter in Frage. Über die Vergabe von Rollen macht man sich nur innerhalb dieser Einheiten Gedanken.

Als Beispiel greifen wir uns einige Arbeitsgruppen und Datenbanken, sowie Personen, die in einer Firma arbeiten, und einige Objekte aus der Stellenhierarchie heraus.

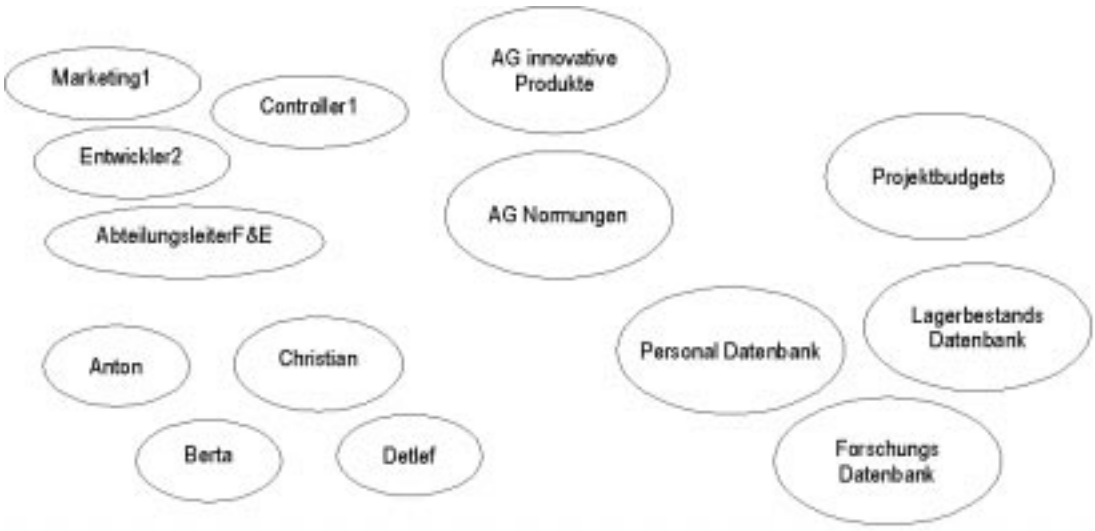



Abbildung 4 – Beispielobjekte

Personenobjekte sind von allen anderen Objekten zu unterscheiden, da sie als Repräsentanten von Personen natürlich keine Sammlung von Mitgliedern führen können.

## 4.1 Rollen, Mitgliedschaften und Rechte

Innerhalb der Objekte etablieren wir Rollen, die die Zuständigkeiten repräsentieren. In einer Arbeitsgemeinschaft (AG) gibt es einen Leiter, diverse Entwickler und Marketingmitarbeiter. Der Zugriff auf die Forschungsdaten und Projektbugets ist nach Projekten gegliedert.

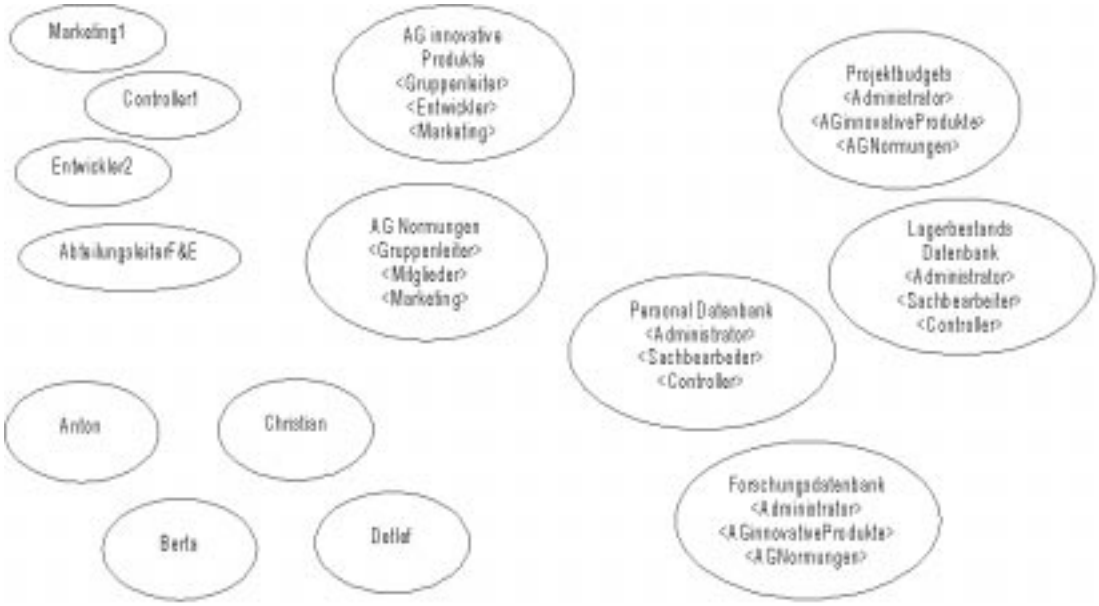



Abbildung 5 - Beispielrollen

Jede Rolle kann Mitglieder aufnehmen und damit Verantwortlichkeit und Rechte an die Mitglieder weitergeben. Beispielsweise hat Anton die Stelle Entwickler1 inne, die der Entwicklung innovativer Produkte zugeordnet ist. Berta hat die Stelle Abteilungsleiter Forschung und Entwicklung und darf auf die Forschungsdatenbank und Projektbudgets zugreifen. Rollen können nicht nur Mitglieder aufnehmen, sondern ihrerseits auch Mitglied in einer anderen Rolle sein, so daß sich indirekte Mitgliedschaften ergeben.

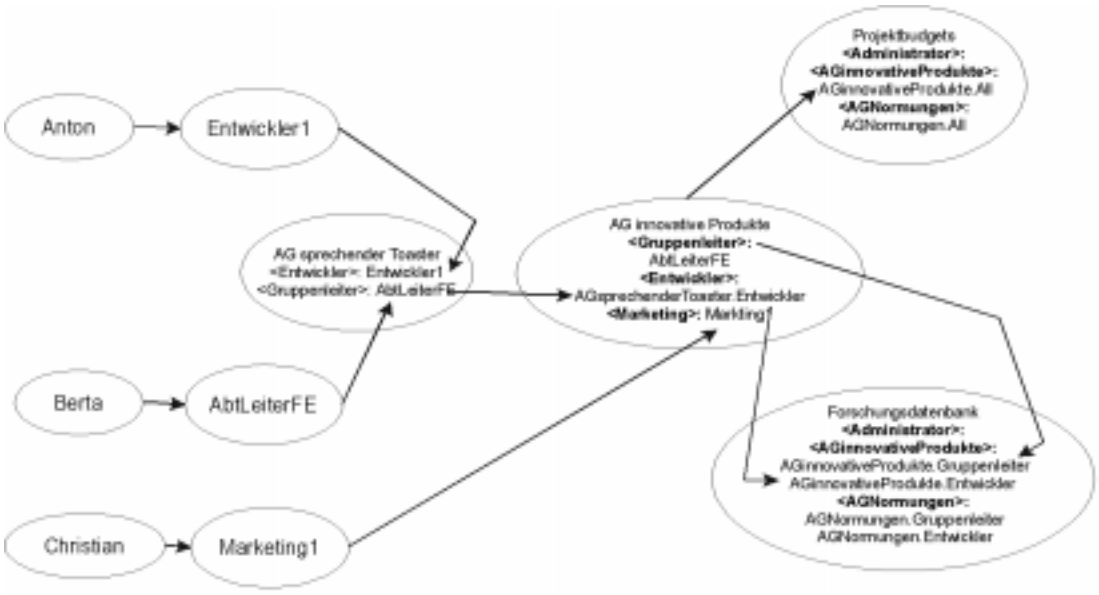



Abbildung 6 - Mitgliedschaften

Wichtig ist, daß Änderungen in einem Objekt ohne Auswirkungen auf alle anderen Objekte stattfinden. Jemand, der volles Administrationsrecht für ein Objekt hat (z.B. Zuordnen von Mitgliedern in Rollen, Zuordnen von Rechten zu Rollen im Objekt “Forschungsdatenbank”), kann dies nur für dieses Objekt tun. Damit läßt sich die Verant-

wortung an die zuständigen Mitarbeiter weiterleiten und es gibt keinen Rollenadministrator, der über “Superrechte” verfügt.

Die Arbeitsgemeinschaft “innovative Produkte” sieht z.B. folgendermaßen aus:

| Rolle | Mitglieder | Rechte | ist Mitglied bei |
|---|---|---|---|
| Erzeuger | Firmenchef | [cross.gif] | |
| Gruppenleiter | AbtLeiterFE | changeObject | Forschungsdaten.AGinnovativeProdukte |
| Entwickler | Entwickler1, Entwickler2, Entwickler3 | | Forschungsdaten.AGinnovativeProdukte |
| Marketing | Marketing1 | | |
| Alle | [cross.gif] | listObject | ProjektBudget.AGinnovativeProdukte |
| Jeder | [cross.gif] | - | [cross.gif] |

Der Gruppenleiter ist der Abteilungsleiter Forschung und Entwicklung (diese Stelle hat Berta inne). Er darf das Objekt “innovativeProdukte” verwalten (Recht “changeObject”) und ist Mitglied in der Rolle AGinnovativeProdukte der Forschungsdatenbank. Entwickler (z.B. Anton, der ja die Stelle Entwickler1 innehat) dürfen ebenfalls auf die Forschungsdatenbank zugreifen, hat aber kein Administrationsrecht für das Objekt. Der Marketingmitarbeiter Marketing1 (z.B. Christian) hat zunächst scheinbar gar keine Rechte; allerdings gibt es innerhalb von Objekten für Verwaltungszwecke drei feste Rollen:

1. <Erzeuger>: Der oder die Eigentümer des Objekts dürfen alle Rechte, die dieses Objekt zur Verfügung stellt, benutzen. Allen anderen Rollen muß ein Recht erst zugewiesen werden, bevor die Mitglieder es ausüben können. Wird ein neues Objekt erzeugt, so wird der Erzeuger automatisch auf die Mitgliederliste des <Erzeuger> gesetzt.
2. <Alle>: Diese Rolle führt keine eigene Mitgliederliste, sondern die Menge aller Mitglieder aller anderen Rollen dieses Objekts bilden die Mitglieder. Dieser Rolle können Rechte oder Mitgliedschaften zugewiesen werden und dienen der Vereinfachung, damit nicht z.B. eine Mitgliedschaft für alle nicht jeder Rolle einzeln zugeordnet werden muß. Im Beispiel wird davon Gebrauch gemacht, in dem die Rolle <Alle> Mitglied in der Budgetdatenbank ist.
3. <Jeder>: Objekte können Rechte ohne Einschränkung vergeben. Sie werden dazu der Rolle <Jeder> zugeordnet, die ebenfalls keine Mitgliederliste führt, sondern in dieser Rolle ist per Definition jedes authentifizierte Mitglied des Zugriffskontrollsystems.

## 4.2 Methoden

Das Zugriffskontrollsystem arbeitet objektorientiert. Die Administration des System wird durch das System selbst zugriffskontrolliert. Dazu stellen die Objekte Methoden zum Hinzufügen und Löschen von Mitgliedern, Rollen, Rechten zu Rollen, etc. zur Verfügung. Ein Recht erlaubt im allgemeinen den Zugriff auf mehrere Methoden (siehe Abbildung 7). So gibt es z.B. ein Recht listObject, das den Aufruf aller Methoden erlaubt, die keine Veränderungen an diesem Objekt vornehmen; es können aber alle

Informationen aus dem Objekt ausgelesen werden, z.B. welche Rollen es gibt, welche Mitglieder und Rechte die Rollen haben, etc.

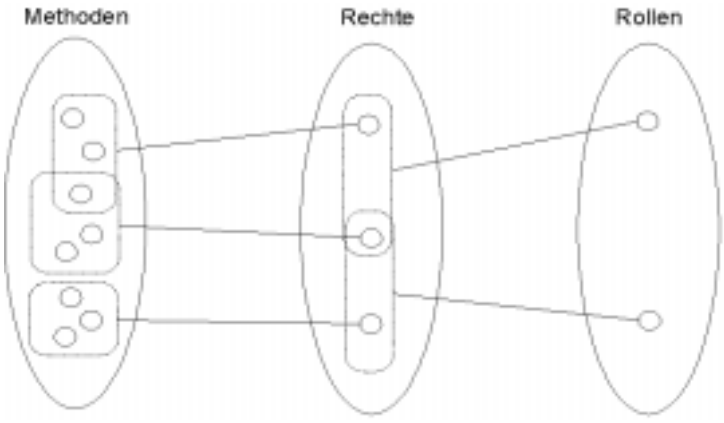



Abbildung 7 – Rollen – Rechte - Methoden

Eine andere Gruppe von Methoden wird beim Verbinden mit entfernten Diensten (z.B. einer Datenbank) benutzt. Der Datenbank wird vom TRUSTCENTER die Rolle mitgeteilt, unter der sich ein Benutzer verbinden möchte. Auf Seiten der Datenbank werden nur die Rollennamen eingerichtet und bekommen ein bestimmtes Rechteprofil. Das ausgehandelte Ticket wird dem Browser des Benutzers übermittelt, und dieser kann fortan für eine begrenzte Zeit im Rahmen der Gültigkeit des Ticktes Anfragen an die Datenbank stellen.

## 4.3 Attribute

Objekte besitzen Attribute. Attribute haben einen Namen, einen Typ, einen Wert und eine Liste der Lese- bzw. Schreibberechtigten Rollen. Sie dienen dazu, Informationen über das Objekt für verschiedene Zwecke zu speichern:

- Darstellungsinformationen zu diesem Objekt (für das HTML-Fontend GIF-Bilder, beschreibender Text, etc.).
- Bei Personenobjekten der Name, Vorname, Personalnummer, Telefon, etc., die zur Auswertung von Zugriffskontrollregeln benutzt werden.
- Email-Adresse und Public-Key-Identifier für das sichere Email-System, das parallel zum Zugriffskontrollsystem auf dem TRUSTCENTER läuft (siehe Workshop-Beitrag von Thomas Hildmann [Hild98]).

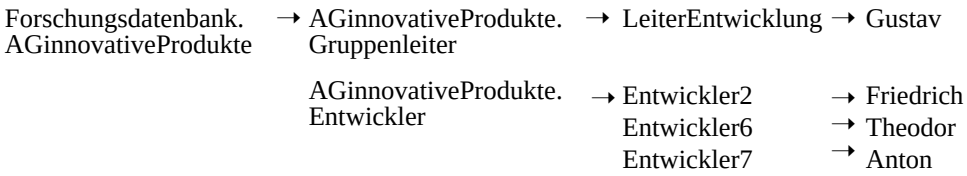
Hier ein Auszug aus der Attributliste, wie sie in dieser Implementierung verwendet wird:

| Typ | Name | Beschreibung |
|---|---|---|
| Description | Name | Name des Objekts, wie er einem Benutzer angezeigt wird. Er darf damit etwas länger sein und auch Leerzeichen enthalten. |
| Mail | apra-mail | Internet mail adresse des Objekts |
| Mail | x400-name | X400 Namensbezeichnung des Objekts. Diese wird zur Auffindung des öffentlichen Schlüssels für sichere Email verwendet |

| | | |
|---|---|---|
| Personal | Name | Persönliche Daten zu der Person. Die Angaben werden beim Erstellen der Person (z.B. bei ihrer Einstellung) eingetragen. |
| | ... | |

## 4.4 Nachrichtenmechanismus

Die Objekte und ihrer Rollen können als Adressaten von Emails verwendet werden. Wird eine Nachricht an eine Rolle eines Objekts geschickt, wird sie an alle Mitglieder vervielfältigt. Sollten diese Mitglieder keine Personenobjekte sein, so vervielfältigt das Mitglied die Nachricht an alle seine Mitglieder. Eine Nachricht an die Rolle AGinnovativeProdukte der Forschungsdatenbank geht folgenden Weg:




Der Nachrichtenmechanismus wird durch das Sichere Email System (siehe Workshop-Beitrag von Thomas Hildmann [Hild98]) erbracht.

Das System bietet zusätzlich die Möglichkeit, Nachrichten nicht an die Mitglieder weiterzuleiten, sondern in einem eigenen Briefkasten zu lagern. Sollte der entsprechende Sachbearbeiter krank sein, kann ein Stellvertreter diese Post bearbeiten. Läge sie bereits für den Sachbearbeiter verschlüsselt in dessen Briefkasten, käme der Stellvertreter nicht mehr an diese Informationen heran.

## 4.5 Makros

Der einfachste Ansatz des Zugriffkontrollmechanismus ist es, einer Person unter einer bestimmten Rolle Zugang zu einem entfernten Dienst zu verschaffen. Dabei werden vom Objekt eine oder mehrere Methoden zur Verfügung gestellt, die eine Verbindung zu einem Dienst aufbauen. Dabei wird nicht der Benutzer selbst bei dem Dienst angemeldet, sondern “jemand in der Rolle des ...”.

Das Ausüben eines Rechts kann nicht nur von dem Innehaben einer Rolle abhängen, sondern auch von weiteren Parametern. Makros sind *parametrisierte Methoden* (*Method-Templates*), bei denen neben der Mitgliedschaft in der geforderten Rolle noch Kontextvariablen ausgewertet werden können, um die Zugriffskontrollentscheidung treffen zu können. Dazu zählen z.B. die aktuelle Uhrzeit oder das aufrufenden Individuum. Bei Abfragen von Gehaltsabrechnungen darf jeder Mitarbeiter bzw. jede Mitarbeiterin nur seine bzw. ihre Abrechnung einsehen. Würde man die Einsichtnahme ausschließlich über Rollen realisieren, wäre für jeden Mitarbeiter die Einrichtung eine anderen Rolle notwendig. Genau dieses soll aber vermeiden werden und kann durch ein Makro realisiert werden. Daß die Auswertung der Zugriffskontrollinformationen auf das Individuum zurückgreifen kann, durchbricht eine strenge Rollen-Orientierung. Für die Modellierung solcher Informationszugriffe ist es aber notwendig.

Auch in den Fällen, in denen ein Verbinden mit dem entfernten Dienst nicht notwendig ist, kann ein Makro definiert werden. Es sendet Abfragen im Namen des Zugriffskon-

trollsystems an den entfernten Dienst und stellt die Antwort dem Benutzer zur Verfügung. Es ist für diese Rolle kein Benutzerbereich bei dem Dienst notwendig; dem Benutzer wird keine direkte Verbindung verschafft. Makros können in diesem Fall sogar als *applikationssspezifische Filter* verwendet werden. Sie können Anfrage und Antwort inhaltlich filtern (z.B. werden nur Datensätze geliefert, bei denen ein bestimmter Zahlungsbetrag kleiner einem Höchstbetrag ist), bevor es sie weiterleitet. So ist eine noch feinere Unterscheidung als eine Ja/Nein-Entscheidung für einen Methodenaufruf möglich.

# 5.0 Architektur

Das Zugriffskontrollsystem ist WWW-basiert. Die Zugriffe auf entfernte Dienste sowie die Administration des Systems selbst wird mit Hilfe eines Browser vorgenommen. Die Benutzer bekommen damit eine einheitliche Schnittstelle zu allen Diensten, die über die sichere Web-Umgebung angeboten werden.

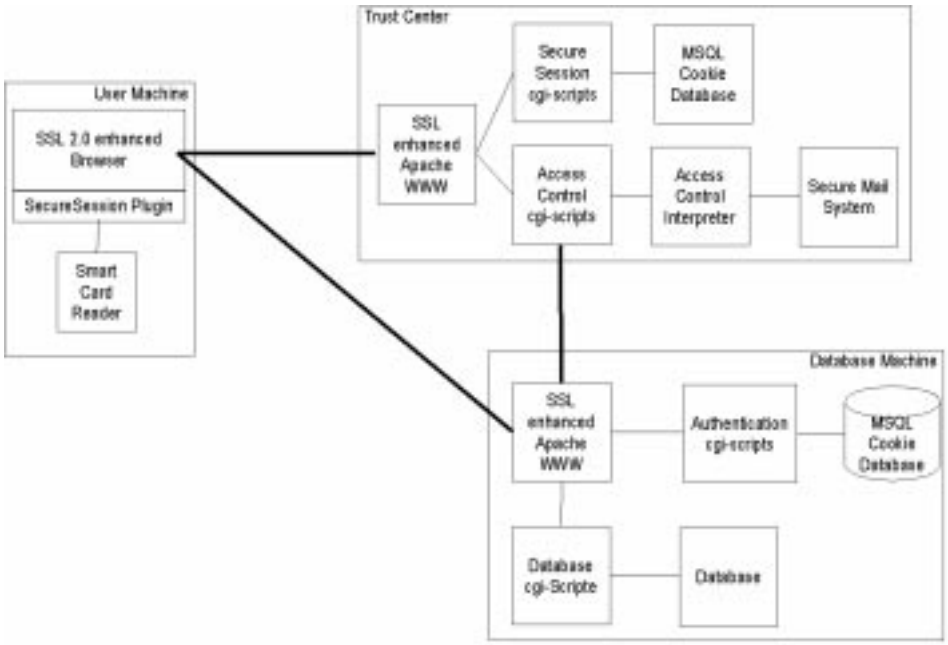



Abbildung 8 – Architektur des Zugriffskontrollsystems über das Internet

## 5.1 Technische Mittel

Wir haben versucht auf frei erhältliche Softwarepakete bzw. im Rahmen des E2S-Projekts entwickelte Komponenten zurückzugreifen. Das führte uns auf folgende Liste der verwendeten externen Produkte.

Auf der Benutzerseite:

- Standard SSL-fähiger Web-Browser (Netscape Navigator)
- GCR400/500 Smart Card Reader/Writer von GemPLUS (Frankreich)
- GPK2000 Smart Cards von GemPLUS (Frankreich)
- Secude von Secude GmbH, http://www.Secude.com/

Auf der Serverseite:

- Apache 1.2.0 WWW-Server
- SSLeay 0.8.1 SSL Paket ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL/SSLeay-0.8.1.tar.gz
- Apache 1.2.0 + SSLeay 0.8.1 Patch
- PHP/FI CGI-scripting language, http://www.vex.net/php
- MiniSQL 2.0 Datenbanksystem, http://www.Hughes.com.au/
- Secude 5.0.

## 5.2 Ablauf

Zunächst wird eine Secure Session aufgebaut. Sie besteht aus vertraulicher, gegenseitig authentifizierter Kommunikation. Für die Sicherung der Vertraulichkeit und für die Authentifizierung der Server gegenüber den Benutzer-Clients verwenden wir SSL. Einzig die Benutzerauthentifizierung erfolgt durch eine Smartcard. Der Einsatz von Smartcards bietet einen sicheren Schutz vor dem Ausspionieren des privaten Schlüssels. Wenn private Schlüssel auf Festplatten gespeichert und nur durch ein Passwort gesichert sind, können sie über entsprechende Software (z.B. ActiveX) “gestohlen” werden und die Person genießt keine Mobilität und kann nicht an einem anderen Computer arbeiten. Wird der private Schlüssel auf einer Smartcard gespeichert, kann er die Smartcard nicht verlassen. Ein eigener Prozessor führt die Verschlüsselung mit dem privaten Schlüssel durch. Die Smartcard wird noch durch eine PIN geschützt, so daß auch bei Verlust kein Schaden entstehen kann. Im Gegensatz zu einer Private-Key-Datei, deren Passwort man auf dem eigenen Rechner beliebig lange versuchen kann zu “knacken”, verweigert die Smartcard nach dreimaliger Fehleingabe den Dienst.

Für die kryptographischen Routinen und zur Ansteuerung des SmartCard-Lesers wird SecuDE verwendet. Für den Browser gibt es ein Plug-In, das aus dem Browser heraus die SecuDE Funktionen aufruft.

Nach erfolgreicher Authentifizierung wird ein Cookie generiert, der es dem Benutzer gestattet, das TRUSTCENTER zu benutzen, ohne bei jeder Anfrage erneut seine PIN eingeben zu müssen (*single sign-on*). Die Gültigkeit eines solchen Cookies ist zeitlich begrenzt. Der Cookie wird nur über die SSL-gesicherte Verbindung übertragen, so daß er als gemeinsames temporäres Geheimnis zwischen Benutzer und TrustCenter angesehen werden kann. Dieses gemeinsame Geheimnis reicht dem TrustCenter für die Authentifizierung einer Anforderung aus. Sollte das Ticket ablaufen, wird der Benutzer wieder auf die Eingangsseite zur Authentifizierung gegenüber dem TrustCenter verwiesen.

Der Benutzer kann jedes Objekt des Organisationsmodells ansehen und Aktionen ausführen (Verbinden zu entfernten Diensten, Administration des Objekts, Versenden von Emails). Beim Navigieren durch das Modell sendet der Browser den Cookie mit, der von den Zugriffskontrollscripten geprüft wird und erhält dann eine entsprechende HTML-Seite. Die Scripte benutzen den Access Control Model Interpreter (ACMI), um die notwendigen Informationen aus dem Modell zu erhalten.

Der Ablauf beim Verbindung mit einem entfernten Dienst ist in Form eines Timeline-Diagramms in Abbildung 9 dargestellt. Der “Klick” des Benutzers ruft eine URL auf,

durch den das Verbindungsscript ausgeführt wird. Es verbindet sich über ssh mit der gewünschten Datenbank. Durch das ssh-Protokoll ist eine gegenseitige starke Authentifizierung gewährleistet und fordert einen Cookie für jemanden in der Rolle des XYZ an. Dieser Cookie wird an den Benutzer weitergereicht. Mit dieser Antwort schickt das Script ein “Location:”-Parameter mit, der den Browser dazu veranlaßt, nicht die Seite darzustellen, sondern sogleich die unter Location angegebene URL abzufragen. Diese zeigt dann auf die gewünschte Datenbank. Für diesen Cookie gilt Vergleichbares wie für den TRUSTCENTER-Cookie. Er ist zeitlich begrenzt und wird auf seiten der Datenbank gespeichert. Bei Anfragen an die Datenbank wird erst der Cookie überprüft und dann die Anfrage ausgeführt.

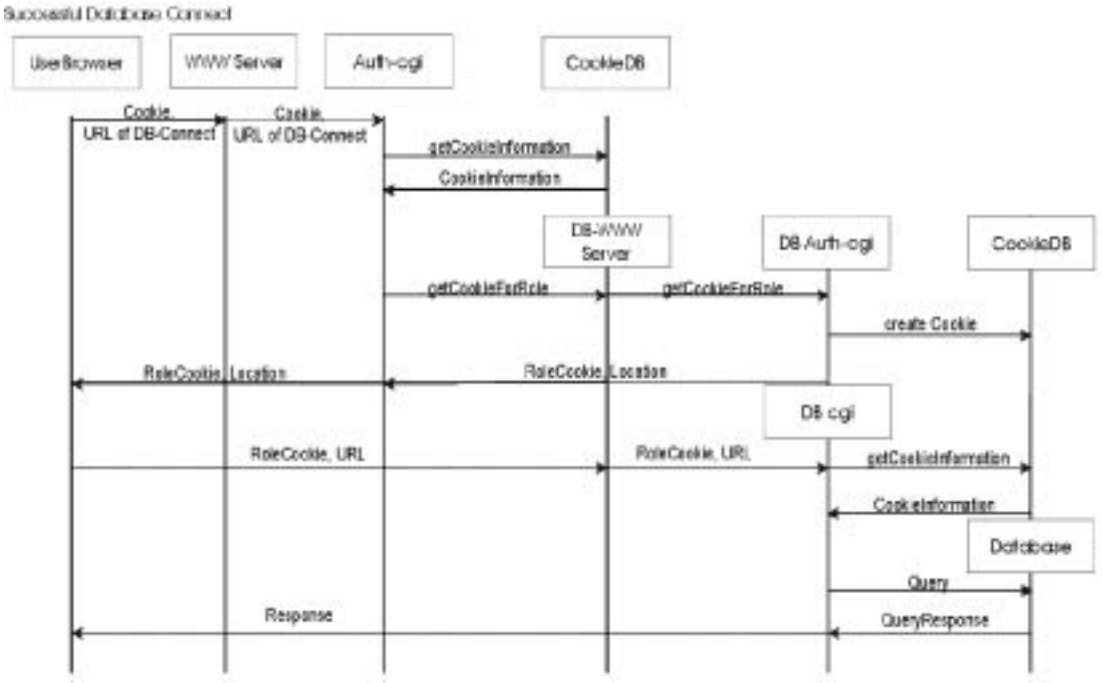



Abbildung 9 – Timeline Diagramm beim Verbinden zu einem entfernten Dienst

## 5.3 Secure Sessions

Die Realisierung eines Single-Sign-On Systems ist unabhängig vom TrustCenter und kann auch bei anderen Web-basierten Diensten verwendet werden. Daher wurde ein unabhängiges Secure-Session-Konzept auf Basis von Smartcards zur Benutzerauthentifizierung entwickelt.

Neben dem SSL-fähigen Browser wird ein Secure-Session Plug-In und eine Smartcard plus Smartcard-Leser verwendet. Es gibt auch eine vollständig Software-basierte Lösung, bei der der private Schlüssel auf der Festplatte, durch ein Passwort geschützt, gespeichert wird, so daß keine zusätzliche Hardware notwendig ist. Damit entstehen wieder die weiter oben aufgeführten Probleme, so daß sie nur zu Testzwecken empfohlen wird. Die Smartcard nimmt den privaten Schüssel (die Secude-PSE, *Personal Security Environment*) auf. Er verläßt die Smartcard nie, da ein Chip auf der Karte die digitale Signatur berechnet. Das Plug-in reagiert auf MIME-Types der Form application/x-auth1. Eingebettet in diese Antwort ist

```
<BAKO>
    <FORM ACTION="https://TRUSTCENTER/cgi-bin/auth.cgi">
```

```
    <INPUT TYPE="HIDDEN" NAME="STATUS" VALUE="SignedAuthentication">
    <INPUT TYPE="HIDDEN" NAME="RANDOM" VALUE="12344">
    <INPUT TYPE="SUBMIT" NAME="SEND" VALUE="Authenticate">
    </FORM>
</BAKO>
```

SecuDE arbeitet mit X.400-Namen. Unterschrieben wird die Zufallszahl zusammen mit dem eignen Namen (C=DE,O=TU-Berlin,CN=JoergBartholdt,SN=54543) und wird an den Server zurückgesandt. Bei erfolgreicher Authentifizierung sendet das TRUSTCENTER die persönliche Eingangsseite für den Benutzer.

## 5.4 Cookie Datenbank

Für die Speicherung der Cookies verwenden wir MiniSQL, eine für Testzwecke frei verfügbare Datenbank. Das Authentifizierungsscript trägt hier den Cookie ein, den es dem Benutzer zuordnet. Die Tabelle hat folgenden Aufbau:

| Feld | Beschreibung |
|---|---|
| cookieid | eine Zufallszahl (*Number once used; abgekürzt: NONCE*) |
| ipaddr | IP Adresse, von der die Authentifizierung vorgenommen wurde. Eine Sitzung kann *nicht* unterbrochen und an anderer Stelle wieder aufgenommen werden (*Session mobilty* wird nicht unterstützt). |
| expire | Gültigkeitsdauer des Cookies. Sie wird in Sekunden seit dem 1.1.1970 00:00 Uhr angegeben (*UNIX-Time-Format*). |
| userid | Benutzerkennung des authentifizierten Benutzers. Wird diese Tabelle von entfernten Diensten benutzt, wird sie die Rollenkennung beinhalten, für die dieser Dienst den Cookie an das TRUSTCENTER herausgegeben hat, welches ihn dann an den Benutzer weiterleitet. |

Neben der Notwendigkeit der Existenz des Cookies darf die Gültigkeitsdauer nicht überschritten sein. Wird ein Cookie einem Client übermittelt, kann u.a. ein Gültigkeitszeitraum angegeben werden. Normalerweise verwerfen Browser einen Cookie, wenn er ungültig wird und schicken ihn nicht mehr mit. Unter Sicherheitsaspekten darf man sich natürlich auf dieses Verhalten nicht verlassen.

## 5.5 BNF

Für eine Implementierung wurde eine Beschreibung des Modells in erweiterter Bacus-Naur-Form verwendet. Hier nur ein kurzer Auszug:

```
<Modell>    :=  <Object>+
<Object>    :=  Object (<CRLF>
                    name : <ID><CRLF>
                    description : <Reference><CRLF>
                    <Role>*
                    <Cap>*
                    <Attr>*
                    <Method>*
                ) <CRLF>
```

```
<Role>       :=  <RoleClass> (<CRLF>
                     name : <ID><CRLF>
                     description : <Reference><CRLF>
                     <Member>*
                     <CapRef>*
                     [<Mailbox>]
                 ) <CRLF>
<Cap>        :=  Cap (<CRLF>
                  name : <ID><CRLF>
                  description : <Reference><CRLF>
                 ) <CRLF>
<Attr>       :=  Attr (<CRLF>
                  name: <ID><CRLF>
                  type: <ID><CRLF>
                  value: <Value><CRLF>
                  readallow: <ObjectRole>*<CRLF>
                  writeallow: <ObjectRole>*<CRLF>
                 ) <CRLF>
<ObjectRole>:= <ID>[.<ID>]
<RoleClass> := Role | CreatorRole | PublicRole | AllRole | OwnerRole
<ID>         :=  [a-zA-Z][a-zA-Z0-9_]*
```

# 6.0 Zusammenfassung

Das TRUSTCENTER bietet Infrastrukturdienste an, mit dessen Hilfe ein virtuelles privates Netzwerk auf Basis des Internet aufgebaut wird. Das rollenbasierte Zugriffskontrollsystem modelliert die Organisation mit ihren Stellen, Personen, Abteilungen, Gruppen, etc. So lassen sich viele heterogene Dienste sinnvoll und mit geringem Aufwand administrieren. Die Rechte ergeben sich aus den Zuständigkeiten der Mitarbeiter innerhalb der Organisation, so z.B. der Bestellbefugnis der Mitarbeitet an der TUB. Die Bereitstellung der Dienste über das Web sichert die Verfügbarkeit auf allen Plattformen, und die Sicherung der Kommunikation durch starke Kryptographie und die Authentifizierung mit Smartcards baut ein virtuelles privates Netz trotz Nutzung des Internet auf. Die Mitarbeiter haben keine weitere Arbeit durch die Sicherheitsmechanismen, da alle Arbeiten wie z.B. das Key-Management durch das TrustCenter übernommen werden.

## Quellen


[Hild98] Thomas Hildmann, Klaus Nagel, Sichere und erweiterte Email Umgebung für den Einsatz in Verwaltungen, Proposal 5. Workshop DFN-Cert, 1998

[Rosh97] Team based Access Control, Roshan Thomas, Second ACM Workshop on Role-Based Access Control, 1997

[Guir97] Role Templates for Content based Access Control, Luigi Guiri & Pietro Iglio, Second ACM Workshop on Role-Based Access Control, 1997