Ein Benutzerkonzept für kollaborative Applikationen am Beispiel von ShaRef

Studierender: Nick Nabholz
Projektbetreuer: Dr.Klaus Wolfertz
Hochschule für Wirtschaft, Verwaltung und Technik Zürich
Abteilung für Informatik, 14. September 2005

Zusammenfassung

Inhalt dieser Arbeit ist ein Benutzerkonzept für kollaborative Applikationen. Dabei sind Zugriffsschutz und Zusammenarbeit die widersprüchlichen Anforderungen. Das Benutzerkonzept schützt die Ressourcen vor dem Zugriff unberechtigter Benutzer und implementiert daher eine Form von Access Control. Es unterstützt auch die Zusammenarbeit der Benutzer, die am besten nach dem Wiki Prinzip funktioniert. Dabei bleibt das Benutzerkonzept skalierbar.

Der Prototyp für die Beispiel Applikation ShaRef zeigt, dass das Konzept funktionsfähig ist. Mit einer Server seitigen Steuerung für Zugriffskontrolle und einem Eclipse Plugin für die Benutzeroberfläche werden die Anforderungen von ShaRef erfüllt.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Zielsetzung und Vorgehen

Moderne Computer Systeme erlauben verschiedenen Benutzern den Zugriff auf die gleichen Ressourcen. Dies ist eine Voraussetzung für die Zusammenarbeit mit Hilfe von Computer Systemen.[1] Die kollaborative Arbeitsweise führt in vielen Fällen zu besseren Resultaten. Kollaborative Applikationen nutzen daher diese Fähigkeit der Computer Systeme und ermöglichen die Zusammenarbeit der verschiedenen Benutzer. Nicht jeder Benutzer möchte seine Ressourcen mit allen anderen Benutzern teilen. Viele kollaborative Applikation benötigen daher einen Mechanismus, der den Zugriff der Benutzer auf die Ressourcen reguliert. Dieser Mechanismus heisst Access Control oder Zugriffsschutz.

Access Control erlaubt oder unterbindet den Zugriff eines Benutzers auf eine Ressource. Damit eine Implementierung von Access Control diese Aufgabe erfüllen kann, werden Informationen zur den Berechtigungen der einzelnen Benutzer für die verschiedenen Ressourcen benötigt. Die Verwaltung dieser Informationen kann bei grösseren Systemen mit vielen Benutzern und vielen Ressourcen einen beträchtlichen Aufwand verursachen. Damit das System skalierbar bleibt, muss diese Aufgabe verteilt werden und kann nicht an die System Administratoren delegiert werden.

1.1 Zielsetzung

Ziel dieser Arbeit ist ein Benutzerkonzept für kollaborative Applikationen. Dabei sind Zugriffsschutz und Zusammenarbeit die widersprüchlichen Anforderungen. Das Benutzerkonzept soll Ressourcen vor dem Zugriff unberechtigter Benutzer schützen und muss daher eine Form von Access Control implementieren. Es soll aber auch die Zusammenarbeit der Benutzer unterstützen, die am besten nach dem Wiki Prinzip [2] funktioniert. Dabei soll das Benutzerkonzept skalierbar bleiben.

Mit einem Prototyp für die Beispiel Applikation ShaRef wird das Konzept auf seine Funktionsfähigkeit geprüft. Es interessiert auch die Frage, ob das beschriebene Konzept Applikations übergreifend implementiert werden kann. Der Prototyp soll hier zumindest Anhaltspunkte liefern.

---

[1] A.Dix spricht in seinem Beitrag von den Artefakten der Arbeit. Diese sind entweder computerisiert oder nicht computerisiert. Sie sind Gegenstand der Kommunikation, und werden zum Kommunikationsmittel. Dix, Alan:Computer Supported [Cooperative] Work, 1994, S.10f.

[2] Das Wiki Prinzip verzichtet auf jede Form von Access Control und ist im Abschnitt 2.2 beschrieben.

1.2 Vorgehen

Jedes Benutzerkonzept mit Access Control muss die Benutzer und ihre Berechtigungen für die Ressourcen verwalten. Die Art und Weise wie diese Aufgabe gelöst wird ist entscheidend für die Skalierbarkeit des Systems. Die zentralisierte Lösung führt bei wachsendem System zur Ueberlastung. Damit die Skalierbarkeit des Systems gewährleistet werden kann, muss die Aufgabe mit einem verteilten Ansatz gelöst werden.

Im Kapitel 2 werden als erstes die bekanntesten Konzepte für Access Control vorgestellt. Anschliessend wird das Wiki Prinzip präsentiert, ein in der Praxis bewährtes, kollaboratives Benutzerkonzept. Es soll als Basis für das gesuchte Benutzerkonzept dienen. Das Wiki Prinzip kennt keinen Zugriffsschutz. Es bleibt also zu klären, wie eine Form von Access Control integriert werden kann. Dazu untersucht der Autor das Konzept der selbstregulierten Arbeitsgruppe. Dieses Konzept erlaubt die Modularisierung der Verwaltung von Benutzern und Berechtigungen. Diese Eigenschaft der selbstregulierten Arbeitsgruppe, kann für die Lösung dieses Problems ausgenutzt werden.

In Kapitel 3 entwickelt der Autor das Benutzerkonzept. Ausgehend vom kollaborativen Wiki-Prinzip wird das Benutzerkonzept entwickelt. Mit Hilfe der selbstregulierten Arbeitsgruppe wird das Wiki-Prinzip mit Zugriffsschutz ergänzt. Das System bleibt dabei skalierbar. Anschliessend wird untersucht, wie die Skalierbarkeit und die Kollaborativität des Systems noch verbessert werden können.

Der Autor hat einen Prototyp des Konzepts für die Beispiel Applikation ShaRef implementiert. Der Prototyp soll die Machbarkeit und Funktionsfähigkeit des in Kapitel 3 gefundenen Benutzerkonzepts aufzeigen. Der Prototyp liegt auf einem zentralen Rechner und wird von verschiedenen Applikationen gemeinsam genutzt.

Im Kapitel 5 wird die Implementierung des Prototyps beschrieben. Die Server Applikation ist als Java RMI Applikation implementiert. Es handelt sich um eine verteilte Applikation mit klassischer drei Schichten Architektur. Die Steuerung besteht aus zwei Komponenten, Session Manager und Group Manager. Der Session Manager verwaltet die Benutzer und die Arbeitssitzungen. Der Group Manager verwaltet die Arbeitsgruppen und die Berechtigungen der Benutzer.

Prototyp und Objektmodell werden im Kapitel 6 an den Anforderungen der Beispiel Applikation gemessen. Es wird aufgezeigt, wie die Anforderungen der Beispiel Applikation ShaRef mit dem beschriebenen Konzept erfüllt werden können. Im Kapitel 4 des Dokuments *ShaRef Use Cases* der ShaRef Entwickler Gruppe sind detaillierte Anwendungsfälle für das Benutzerkonzept formuliert. Der Prototyp wird auf diese Anwendungsfälle hin überprüft.

Im Kapitel 7 zieht der Autor eine positive Bilanz. Mit dem gewählte Ansatz ist ein kollaboratives, skalierbares Benutzerkonzept mit Zugriffsschutz möglich. Der Prototyp ist funktionsfähig. Der grosse Test steht aber noch aus. Die Beispiel Applikation ShaRef wird den Prototyp zur Steuerung der Berechtigungen nutzen. Erst in der Anwendung wird

sich zeigen, ob das Konzept die Erwartungen bezüglich Kollaborativität und Skalierbarkeit wirklich erfüllt.

Offene Fragen und ein Ausblick schliessen die Arbeit ab. Die Anforderungen für eine vollständige Implementierung des Konzepts werden beschrieben. Auf die noch nicht untersuchten Applikations seitigen Probleme und Möglichkeiten wird hingewiesen.

2 Grundlagen

Das gesuchte Benutzerkonzept muss einen Zugriffsschutz implementieren. Dazu werden im folgenden Abschnitt die bekanntesten Konzepte für Access Control untersucht. Weiter wird das Verhältnis von Access Control und Authentisierung geklärt.

Das gesuchte Benutzerkonzept soll die Zusammenarbeit der Benutzer unterstützen. Ein erfolgreiches Konzept für Zusammenarbeit ist das Wiki-Prinzip. Diese Form von Computer basierter Zusammenarbeit verzichtet auf Access Control. Applikationen mit einem Wiki Benutzerkonzept haben sich in der Praxis als besonders kollaborativ erwiesen. Das gesuchte Benutzerkonzept soll auf dem Wiki-Prinzip aufbauen.

Zu klären ist, wie der Zugriffsschutz in ein Wiki basiertes Benutzerkonzept integriert werden kann. Das Konzept der selbstregulierten Arbeitsgruppe bietet die Möglichkeit zur Modularisierung des Benutzerkonzepts. Damit lässt sich, wie später gezeigt wird, der Zugriffsschutz in ein Wiki basiertes Benutzerkonzept integrieren.

2.1 Access Control

Objekte, auf die der Zugriff der Benutzer reguliert werden soll, heissen Ressourcen. Access Control regelt den Zugriff der Benutzer auf die Ressourcen in Computer Systemen. Die Aufgabe der Berechtigungen ist es, die Benutzer mit den Ressourcen zu verknüpfen. Eine Berechtigung erlaubt es einem Benutzer bestimmte Operationen auf einer Ressource auszuführen. Die beiden bekanntesten Konzepte für Access Control werden hier vorgestellt.

2.1.1 Role-based Access Control(RBAC)

Role-based Access Control verknüpft die Benutzer mit den Berechtigungen über Rollen. Eine Rolle kann Berechtigungen für verschiedenste Ressourcen beinhalten. Wird einem Benutzer eine Rolle zugewiesen, hat er alle Berechtigungen der Rolle zur Verfügung.

Mit diesem Konzept lassen sich die Funktionen einer Organisation modellieren. Jedem Mitarbeiter wird die seiner Funktion oder Stellung entsprechende Rolle zugewiesen. Mit der Rolle bekommt er die zur Ausübung seiner Funktion benötigten Berechtigungen. Die Rollen bleiben, die Benutzer können wechseln. Wechselt der Benutzer die Funktion in der Organisation, zum Beispiel im Zuge einer Beförderung, wechselt er auch die Rolle. Das Konfigurieren und Zuweisen der Rollen ist die Aufgabe eines System Administrators. Die zentrale Verwaltung von Berechtigungen durch Administratoren ist eine Voraussetzung für RBAC.

'Role Based Access Control (RBAC) bases access control on the function a user has in a organization. A role can be thought as a set of permissions within the context of a

organization. A user can not pass access permissions to other users at his discretion.' [3]

Mehrere Benutzer können über die gleichen Rollen und damit über Berechtigungen für die gleichen Ressourcen verfügen. Rollen lassen sich auch hierarchisieren. Hierarchisch höhere Rollen beinhalten jeweils alle Berechtigungen der hierarchisch untergeordneten Rollen. Hierarchisierung der Rollen kann helfen die Anzahl der Rollen pro Benutzer zu reduzieren.

RBAC macht direkt keine Aussage zur Verteilung der Entscheidungskompetenz. In der Praxis führt dieses Konzept zu einer Zentralisierung. System Administratoren verwalten die Rollen der Benutzer. Die normalen Benutzer sind an der Verteilung von Berechtigungen nicht beteiligt. RBAC ist daher nur bedingt skalierbar.

2.1.2 Discretionary Access Control (DAC)

'Discretionary Access Control (DAC) permits granting and revolving of access privileges to the discretion of the individual user. The user may grant privileges for the objects under his control to other users without intercession of a system administrator.' [3]

DAC ist ein verteilte Lösung. Die Verwaltung der Berechtigungen ist Sache der Benutzer. Der Ersteller einer Ressource ist per Definition der Besitzer der Ressource. Er verfügt über alle Berechtigungen. Er kann nach Belieben Berechtigungen für seine Ressourcen an andere Benutzer erteilen. Er kann auch einem anderen Benutzer eine administrative Berechtigung für seine Ressource erteilen. Der Berechtigte kann dann ebenfalls Berechtigung für die Ressource an andere Benutzer zuweisen.

Mit DAC wird die Entscheidungskompetenz verteilt. DAC ist daher skalierbar. Aber DAC ist nur bedingt geeignet für Zusammenarbeit. Die Ressourcen sind im Besitz der einzelnen Benutzer. Es liegt an jedem einzelnen Benutzer, Berechtigungen für seine Ressourcen an die anderen Benutzer zu vergeben. Die Schaffung von grösseren Räumen mit gemeinsamen Berechtigungen bedarf der Zusammenarbeit der Benutzer. Dabei wird jeder der beteiligten Benutzer mit administrativen Aufgaben konfrontiert.

2.1.3 Authentisierung

Access Control steuert den Zugriff der Benutzer auf die Ressourcen. Die Erlaubnis zum Zugriff auf Ressourcen basiert auf der Identität des Benutzers. David Ferraiolo beschreibt in [Role 2003] das Verhältnis von Access Control und Authentisierung. Jede Form von Zugriffsschutz basiert auf Authentisierung. Authentisieren bedeutet die Indentität eines Benutzers überprüfen. Die meist verbreitete Form von Authentisierung ist ein Passwort.

[3] Ferraiolo, David et al.:[Role 1992] Role-Based Access Control, 1992, S.3

Der Benutzer eines Systems gibt dem System seine Identität über einen Benutzernamen bekannt. Mit dem Passwort kann das System überprüfen, ob es sich wirklich um den Benutzer handelt. Das Konzept geht natürlich davon aus, dass nur der Benutzer selber das korrekte Passwort kennt. Die verschiedenen Möglichkeiten zur Authentisierung und ihre Vor- und Nachteile sind nicht Thema dieser Arbeit.

2.2 Das Wiki Prinzip

Das gesuchte Benutzerkonzept soll die Zusammenarbeit der beteiligten Benutzer unterstützen. Das Wiki Prinzip ist ein in der Praxis bewährtes, kollaboratives Benutzerkonzept. Es soll als Basis für das gesuchte Benutzerkonzept dienen. Das Wiki Prinzip definiert sich über die Abwesenheit von Zugriffskontrolle. Alle Benutzer haben die vollen Berechtigungen. [4]

Alle Benutzer sind gleichgestellt. Jeder Benutzer hat die vollen Berechtigungen. Jeder Benutzer hat das Recht neue Ressourcen zu kreieren und bestehende Ressourcen zu modifizieren. Dabei spielt es keine Rolle, welcher Benutzer die Ressource erstellt hat.

Das Konzept ist skalierbar und unterstützt die Zusammenarbeit. Obwohl, oder gerade weil ein klassisches Wiki keinen Zugriffsschutz implementiert, funktioniert diese Form der Zusammenarbeit. Ein Beispiel mit sehr vielen Benutzern ist Wikipedia, eine kollaborative Enzyklopädie. Nach Angabe von [Wikipedia] umfasst dieses kollaborative Projekt Artikel in mehr als 100 Sprachen. Seit Mai 2001 wurden allein in deutscher Sprache 273'508 Artikel verfasst.

Diese Form der Zusammenarbeit funktioniert oft, aber nicht immer. Interessant ist in diesem Zusammenhang auch die folgende Zeitungsnotiz. *'...Das hat die renommierte Zeitung* Los Angeles Times *dazu animiert, einen Leitartikel von Web-Usern als Wiki schreiben zu lassen. Allerdings ging der Schuss nach hinten los: Anstatt eines klugen Kommentars stellten immer mehr User primitive Statements online...'* [5]

2.3 Selbstregulierte Arbeitsgruppen

Sabine Pietruschka beschreibt das Konzept der selbstregulierte Arbeitsgruppe. *'Die Uebertragung von Entscheidungs- und Verhaltenskontrolle stellt ein Definitionsmerkmal der selbstregulierten Arbeitsgruppe dar. Die Autonomie führt zu einer Erweiterung der Handlungsspielräume und den damit verbundenen positiven Effekten. Das Gestaltungsprinzip*

[4] Vergleiche Leuf, Bo et al.:The[Wiki Way], 2003, What's a 'Wiki'? S.13f.

[5] Aber der der Schuss kann auch nach hinten losgehen, 20 Minuten, Dienstag 28.Juni 2005, S.25

der Gruppenarbeit ist, sowohl den individuellen als auch den kollektiven Handlungsspielraum zu erweitern.' [6]

Die selbstregulierte Arbeitsgruppe verfügt per Definition über einen hohen Grad an Autonomie. Selbstregulation, Selbstbestimmung Selbstverwaltung bezeichnen verschiedene Formen mit zunehmender Autonomie. Die selbstregulierte Arbeitsgruppe verfügt über Entscheidungsbefugnisse für Aufgabenausführung, gruppeninterne Führung, Gruppenmitgliedschaft und interne Aufgabenverteilung. Die selbstregulierte Arbeitsgruppe ist als fester Bestandteil der Arbeitsorganisation gedacht. Ein Gruppensprecher vertritt die Gruppe nach aussen. Er wird von der Gruppe selbst bestimmt.

Das Konzept der selbstregulierten Arbeitsgruppe verbindet zwei, für diese Arbeit interessante, Eigenschaften. Mit dem Konzept der selbstregulierten Arbeitsgruppe kann Entscheidungskompetenz an die einzelnen Arbeitsgruppen verteilt werden. Diese Verteilung der Aufgaben macht das System skalierbar. Mit dem Konzept der selbstregulierten Arbeitsgruppe lässt sich zudem ein Benutzerkonzept modularisieren. Modularisieren bedeutet in diesem Fall, die Kombination verschiedener Benutzerkonzepte. Die Autonomie der Arbeitsgruppe bedeutet die mindestens partielle Entkopplung von Innen und Aussen. So kann zum Beispiel ausserhalb und innerhalb der Arbeitsgruppe jeweils ein anderes Benutzerkonzept angewendet werden.

[6] Pietruschka, Sabine: [Arbeitsgruppen], 2003, Autonomie von Arbeitsgruppen, S.11f.

3 Ein Benutzerkonzept für kollaborative Applikationen

Eine Applikation mit Zugriffsschutz ist ein System. Das System soll für Zusammenarbeit und Skalierbarkeit optimiert werden. Wie im Kapitel 2 gezeigt, lassen sich im System verschiedene, für Access Control relevante, Objekte identifizieren. Es gibt Benutzer und Ressourcen. Und es gibt Beziehungen zwischen diesen Objekten, die sich in Form von Berechtigungen manifestieren.

In diesem Kapitel wird ausgehend vom kollaborativen Wiki-Prinzip ein Benutzerkonzept mit Access Control entwickelt. Mit dem Konzept der selbstregulierten Arbeitsgruppe wird das Wiki-Prinzip mit Zugriffsschutz ergänzt. Dieser Ansatz hat auch den Vorteil, dass die Skalierbarkeit durch die selbstregulierte Arbeitsgruppe bereits gewährleistet ist.[7]

3.1 Lösungsansatz

Das kollaborative Benutzerkonzept basiert auf dem Wiki-Prinzip. Mit Hilfe der selbstregulierten Arbeitsgruppe wird die fehlende Zugriffskontrolle hinzugefügt. Die Arbeitsgruppe fasst eine Anzahl von Benutzern und Ressourcen zu einer Einheit zusammen, wobei sich diese Einheit auf die Verwaltung der Berechtigungen bezieht. Die Mechanismen zur Verteilung der Berechtigungen innerhalb der Arbeitsgruppe sind vom äusseren Berechtigungsraum entkoppelt.

Innerhalb der Arbeitsgruppe kommt das Wiki-Prinzip zum Einsatz. Alle Benutzer sind gleichgestellt. Jeder Benutzer hat die vollen Berechtigungen auf allen Ressourcen der Arbeitsgruppe. Jeder Benutzer hat das Recht neue Ressourcen zu kreieren und bestehende Ressourcen zu modifizieren. Dabei spielt es keine Rolle, welcher Benutzer die Ressource ursprünglich erstellt hat. Alle von den Mitgliedern der Arbeitsgruppe erstellten Ressourcen sind im Besitz der Arbeitsgruppe.

Obwohl innerhalb der Arbeitsgruppe das Wiki-Prinzip zur Anwendung kommt verfügt die Arbeitsgruppe aus System Sicht über einen Zugriffsschutz. Nur die Mitglieder der Arbeitsgruppe können auf deren Ressourcen zugreifen. Benutzer, die nicht Mitglied der Arbeitsgruppe sind, haben keinen Zugriff. Es stellt sich die Frage, wer die Mitglieder der Arbeitsgruppe bestimmt. Das Konzept der selbstregulierten Arbeitsgruppe wie es in Abschnitt 2.3 beschrieben ist, gibt dazu einen Hinweis. Die Gruppenmitgliedschaft liegt im Entscheidungsbereich der Arbeitsgruppe. Es ist Sache der Arbeitsgruppe die Verteilung der Berechtigungen an ihre Mitglieder zu regulieren. Es sind also die Mitglieder

[7] Die Skalierbarkeit eines Systems aus selbstregulierten Arbeitsgruppen beruht auf der damit verbundene Verteilung der Aufgaben.

Abbildung 1: Selbstregulierte Arbeitsgruppe und WIKI

der Arbeitsgruppe selbst, die neue Mitglieder in eine Gruppe aufnehmen können, oder bestehende Mitglieder aus einer Gruppe entfernen können. Diese Lösung steht auch im Einklang mit der Forderung nach Skalierbarkeit. Die Delegation der Verwaltung der Mitgliedschaft an einen System Administrator ist aus diese Blickwinkel nicht möglich.

3.2 Ein kollaboratives Benutzerkonzept mit Access Control

Der Grundbaustein des Benutzerkonzepts ist die Arbeitsgruppe, wie in Abbildung 1 skizziert und im Abschnitt 3.2 beschrieben. Die praktizierte absolute Gleichbehandlung aller Mitglieder der Gruppe scheint dem Autor nicht Praxis bezogen. Die Arbeitsgruppe, wie im vorhergehenden Abschnitt skizziert, wird modifiziert.

3.2.1 Arbeitsgruppe

Der Lösungsvorschlag modifiziert den im Abschnitt 3.2 präsentierte Arbeitsgruppe. Die praktizierte absolute Gleichbehandlung aller Mitglieder der Gruppe scheint dem Autor nicht Praxis bezogen. Nicht alle Mitglieder wollen und können sich mit den administrativen Belangen der Gruppe beschäftigen. Es macht wohl auch keinen Sinn alle Mitglieder für diese Aufgabe auszubilden. [8]

Aus diesen Gründen führt der Autor für die Zusammenarbeit innerhalb der Arbeitsgruppe Rollen ein. Das Rollen Konzept eignet sich für die Organisation von Zusammenarbeit. Die Berechtigungen für die Ressourcen der Arbeitsgruppe können in Form von Rollen an die Mitglieder der Arbeitsgruppe verteilt werden. Verschiedene Berechtigungen werden jeweils zu einer Rolle zusammengefasst. Die Rollen reflektieren die verschiedenen Funktionen der Mitglieder einer Arbeitsgruppe. Jedem Mitglied einer Arbeitsgruppe wird eine

[8] Auch in der Beschreibung der selbstregulierten Arbeitsgruppe durch Sabine Pietruschka im Abschnitt 2.3 sind nicht alle Mitglieder der Gruppe gleichgestellt. So verfügt die Gruppe zum Beispiel über einen Gruppensprecher, der die Arbeitsgruppe nach aussen vertritt.

Rolle zugeteilt. Mitglieder mit gleichen Rollen haben die gleichen Berechtigungen für die selben Ressourcen. Die vom Wiki-Prinzip implizierte Kollaborativität geht nicht verloren.

Der im Abschnitt 3.2 beschriebene Lösungsansatz kann als Arbeitsgruppe mit einer einzigen Rolle interpretiert werden. Es ist die obligatorischen Administratoren Rolle. Jede Arbeitsgruppe benötigt mindestens einen Administrator. Die Benutzer in der Administratoren Rolle organisieren die Arbeitsgruppe. Ein Administrator kann einem beliebigen Benutzer eine Rolle für die Arbeitsgruppe zuweisen und auch wieder entziehen. Er kann auch weitere Administratoren hinzufügen. Dies ist eine Schwäche des Konzepts. Der Administrator kann auch sich selbst entfernen. Eine Arbeitsgruppe kann aber ohne Administrator nicht mehr geändert werden. Eine Implementierung des Konzepts muss hier Vorkehrungen treffen.

Neu kommt jetzt die Mitglieder Rolle hinzu. Benutzer in der Mitglieder Rolle haben keine administrativen Berechtigungen. Die Mitglieder Rolle beinhaltet ausschliesslich Berechtigungen für die Ressourcen der Arbeitsgruppe. Das Konzept erlaubt es auch verschiedene Mitglieder Rollen zu definieren mit unterschiedlichen Berechtigungen für die Ressourcen der Arbeitsgruppe.

3.2.2 System

Das System besteht aus einer Reihe von solcher Arbeitsgruppen. Zum Aufbau eines funktionsfähigen Systems werden verschiedene Arbeitsgruppen Typen benötigt. Es gibt drei Haupttypen.

- Ressourcen basierte Arbeitsgruppen sind in der Applikation mit Zugriffsschutz reflektiert. Die Ressourcen einer solchen Arbeitsgruppe spiegeln die Ressourcen einer Applikation mit Zugriffsschutz. Die Mitglieder einer solchen Arbeitsgruppe bearbeiten gemeinsam die Ressourcen der Arbeitsgruppe.
- Die System Arbeitsgruppe kontrolliert das System und ist der Ursprung aller Berechtigungen. Es gibt nur eine System Arbeitsgruppe und die existiert von Anfang an. Am Anfang gibt es genau einen Benutzer mit der Rolle des System Administrators. Dieser Benutzer kann weitere System Administratoren ernennen. Neben der System Administratoren Rolle gibt es weitere Rollen. Diese Rollen beinhalten die Berechtigungen zum Erstellen von neuen Arbeitsgruppen. Ein System Administrator kann diese System Rollen an beliebige Benutzer vergeben. Diese wiederum können Arbeitsgruppen erstellen und Rollen für diese Arbeitsgruppen an weitere Benutzer vergeben.
- Administrative Arbeitsgruppen verwalten nicht direkt die Ressourcen einer Applikation und müssen nicht in einer Applikation reflektiert werden. Diese Arbeitsgruppen können zur Strukturierung der Verwaltung von Berechtigungen genutzt werden. Dieser Arbeitsgruppen Typ wird erst für die im Abschnitt 3.3 beschriebenen Erweiterungen benötigt.

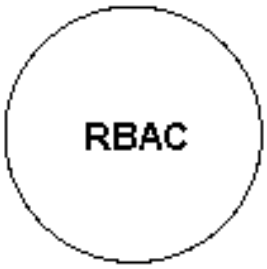

Abbildung 2: Selbstregulierte Arbeitsgruppe und RBAC

Mit diesen Arbeitsgruppen Typen lässt sich ein funktionsfähiges System aufbauen. Dabei bestehen keine direkte Beziehungen zwischen den einzelnen Arbeitsgruppen. Der Fluss der Berechtigungen geht von der System Arbeitsgruppe über die Benutzer zu den Mitgliedern der anderen Arbeitsgruppen.

3.3 Modell erweitern

In diesem Abschnitt wird das Modell nochmals erweitert. Der Ansatz ist einfach. An die Stelle eines Benutzers im System soll auch eine weitere Arbeitsgruppe treten können. Zwischen den Arbeitsgruppen entstehen Beziehungen und Abhängigkeiten. Es findet ein Austausch von Berechtigungen statt. Die Berechtigungen können direkt von Arbeitsgruppe zu Arbeitsgruppe fliessen.

Wie in Abbildung 2 skizziert kommt im Innern der Arbeitsgruppe das für die Zusammenarbeit günstige RBAC zur Anwendung. Mit der Erweiterung der Spielregeln treten die Arbeitsgruppen zueinander in Beziehung. Diese Beziehung zwischen einzelnen Arbeitsgruppen kann als eine Art Arbeitsgruppen DAC interpretiert werden. Die Berechtigungen für die nehmende Arbeitsgruppen werden nach dem Belieben der gebenden Arbeitsgruppe erteilt.[9]

An die Stelle eines Benutzers im System kann eine beliebige Arbeitsgruppe treten. Im folgenden wird untersucht, wie mit diesem Mechanismus Skalierbarkeit und Kollaborativität des Systems gesteigert werden können.

[9]Mit dem gleichen Recht könnte diese Beziehung aus Sicht der gebenden Arbeitsgruppe auch als RBAC interpretiert werden. Die nehmende Arbeitsgruppe erhält eine Rolle für die gebende Arbeitsgruppe.

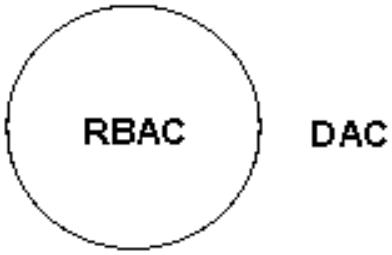

Abbildung 3: Selbstregulierte Arbeitsgruppe, RBAC und DAC

3.3.1 Skalierbarkeit verbessern

Den Benutzer durch eine Arbeitsgruppe ersetzen bedeutet auch, einer Arbeitsgruppe eine Rolle für eine weitere Arbeitsgruppe zu erteilen. Die Berechtigungen gehen von einer Arbeitsgruppe zur anderen Arbeitsgruppe über. Dabei kann zwischen der gebenden und der nehmenden Arbeitsgruppe unterschieden werden. Die gebende Arbeitsgruppe gibt der nehmenden Arbeitsgruppe eine Rolle für ihre Ressourcen.

Die nehmende Arbeitsgruppe wird die neuen Berechtigungen an ihre Mitglieder verteilen. Da es sich hier um eine administrative Arbeit handelt, werden die Administratoren der nehmenden Arbeitsgruppe die Verteilung übernehmen. Sie können die bestehenden Rollen der nehmenden Arbeitsgruppe um Berechtigungen für die Ressourcen der gebenden Arbeitsgruppe erweitern.[10]

Mit diesem Mechanismus kann die Skalierbarkeit des Systems verbessert werden. Skalierbarkeit heisst die Entscheidungskompetenzen bei wachsenden Systemen verteilen. So kann zum Beispiel die Ueberlastung eines ersten Administrators abgebaut werden, indem dieser eine öfters verlangte Rolle nicht mehr selber verteilt. Statt dessen weist er die Rolle einer administrativen Arbeitsgruppe zu. Die Administratoren dieser Gruppe können dann selbständig die mit der Rolle verbundenen Berechtigungen an weitere Benutzer verteilen, indem sie die Benutzer zu Mitgliedern der Gruppe machen. Der erste Administrator kann so den Entscheid über die Erteilung einer Rolle an die Administratoren der Gruppe delegieren.

Eine Arbeitsgruppe kann im Regelfall eine oder mehrere Ressourcen Typen aus einer Applikation verwalten. Mit der beschriebenen Erweiterung kann diese Grenze abgebaut werden. Gebende und nehmende Arbeitsgruppe müssen sich nicht auf die gleiche Applikation beziehen. Das Konzept erlaubt es eine Arbeitsgruppe für den Applikations übergreifenden Zugriffsschutz zu erweitern. Die Rollen der nehmenden Arbeitsgruppe beinhalten dann

[10] Hierarchisierte Rollen sind für die Verteilung in der nehmenden Arbeitsgruppe besonders interessant. So besteht die Möglichkeit zur Differenzierung. Einige Rollen könnten mit hierarchisch untergeordneten Rollen weniger stark erweitert werden.

auch Berechtigungen für die Ressourcen der gebenden Arbeitsgruppe, deren Ressourcen aus einer anderen Applikation kommen können.

3.3.2 Kollaborativität erhöhen

Mit der Erweiterung aus dem vorigen Abschnitt lassen sich also die Rollen einer Arbeitsgruppe durch Berechtigungen für Ressourcen anderer Arbeitsgruppen erweitern. Damit kann der Raum mit gemeinsamen Berechtigungen für die Mitglieder einer Arbeitsgruppe vergrössert werden. Die Kollaborativität des Systems kann also erhöht werden.

Die Schaffung solcher Arbeitsgruppen übergreifender Räume kann aber nur durch die Zusammenarbeit der verschiedenen Administratoren der verschiedenen Arbeitsgruppen erreicht werden. Die verteilten Entscheidungskompetenzen sind für die beschriebene Rollenbildung über mehrere Arbeitsgruppen hinderlich. Der Administrator einer Arbeitsgruppe hat keine Berechtigungen zur Erweiterung der Rollen seiner Arbeitsgruppe mit den Berechtigungen einer anderen Arbeitsgruppen. [11]

Die Lösung könnte wie folgt aussehen. Der System Administrator vergibt die Berechtigungen zum Erstellen von Arbeitsgruppen an eine weitere Arbeitsgruppe. Dies könnte eine rein Administrative Arbeitsgruppe sein. Es genügt neben der Administrator Rolle eine Mitglieder Rolle zu definieren. Alle Mitglieder dieser Arbeitsgruppe haben die Berechtigung zum Erstellen von neuen Arbeitsgruppen. Nicht der Ersteller, sondern die administrative Arbeitsgruppe des Benutzers wird jetzt Administrator von neuen Arbeitsgruppe. Damit kann der Benutzer die Arbeitsgruppe administrieren, ist aber nicht der einzige Benutzer mit administrativen Berechtigungen. Alle Mitglieder der administrativen Arbeitsgruppe werden zu Administratoren der neuen Arbeitsgruppe. Damit entsteht die Möglichkeit mehrere Arbeitsgruppen zentral zu verwalten und entsprechend ausgedehnte Wiki Räume zu schaffen.

Für dieses Konzept ist es wichtig, ob ein Benutzer als Mitglied einer Arbeitsgruppe handelt oder als direkt vom System ermächtigter Benutzer. Und was passiert bei doppelter Berechtigung? Das System benötigt Information zum Kontext einer Handlung. [12]

---

[11] Dies ist auch der selbe Grund wieso sich DAC nur bedingt für die Zusammenarbeit eignet. In DAC Systemen liegen Eigentum und Berechtigungen der einzelnen Ressourcen beim jeweiligen Ersteller der Ressource. Bereits die Schaffung grösserer Räume mit gemeinsamen Berechtigungen über mehrere Ressourcen ist ein organisatorisches Problem. Siehe dazu auch Sandhu, R./Munawer Q.: How To Do [Discretonary] Access Control Using Roles, ACM Press 1998, S.51.f

[12] siehe dazu auch Edwards, W. Keith: Policies and Roles in [Collaborative] Applications, ACM Press 1996, Abschnitt *The Importance of Context*

3.4 Zusammenfassung

Der Vorschlag für das kollaborative Benutzerkonzept basiert auf dem Wiki-Prinzip. Mit Hilfe der selbstregulierten Arbeitsgruppe wird die fehlende Zugriffskontrolle hinzugefügt. Die Arbeitsgruppe fasst eine Anzahl von Benutzern und Ressourcen zu einer Einheit zusammen. Diese Einheit bezieht sich auf die Berechtigungen. Innerhalb der Arbeitsgruppe kommt das Wiki-Prinzip zur Anwendung. Das Wiki-Prinzip wird lokal auf die Mitglieder der Arbeitsgruppe beschränkt.

Aus praktischen Gründen wird die Arbeitsgruppe modifiziert. Innerhalb der Arbeitsgruppe wird RBAC eingeführt. Die obligatorische Administratoren Rolle übernimmt die Verwaltung von Mitgliedern und deren Rollen. Daneben gibt es Mitglieder Rollen ohne administrative Berechtigungen.

Um ein funktionierendes System aufzubauen sind verschiedene Arbeitsgruppen Typen nötig. Es gibt genau eine System Arbeitsgruppe. Die System Arbeitsgruppe kennt neben der System Administratoren Rolle weitere Rollen mit den Berechtigungen zum Erstellen von anderen Arbeitsgruppen. Benutzer in diesen Rollen können Arbeitsgruppen erstellen und Benutzer zur Mitarbeit berechtigen.

An die Stelle eines Benutzer im System kann neu auch eine Arbeitsgruppe treten. [13] Damit werden direkte Beziehungen zwischen den Arbeitsgruppen möglich. Es wird gezeigt, wie mit diesem Mechanismus Skalierbarkeit und Kollaborativität des Systems verbessert werden können.

[13] Bei diesem Vorgang gibt es eine (Berechtigungs) gebende und eine nehmende Arbeitsgruppe. Die Beziehung zwischen einzelnen Arbeitsgruppen können als eine Art Arbeitsgruppen DAC interpretiert werden. Die Berechtigungen für die nehmende Arbeitsgruppen werden nach dem Belieben der gebenden Arbeitsgruppe erteilt.

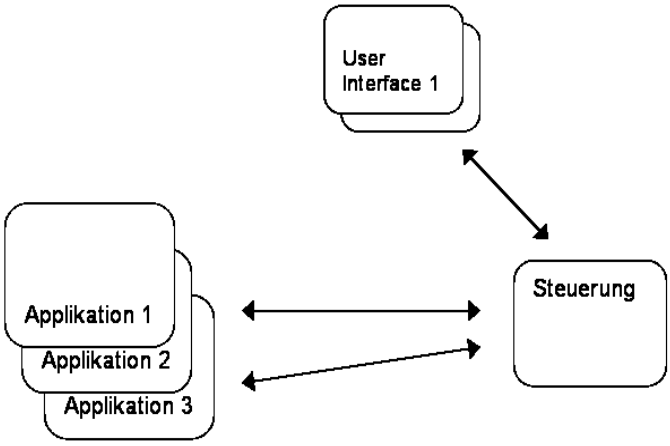

Abbildung 4: System

4 Prototyp

Der Autor hat für die Beispiel Applikation ShaRef einen Prototyp implementiert. Damit soll das Konzept auf Praxis Tauglichkeit getestet werden. Im Kapitel 6 werden Konzept und Prototyp mit Blick auf die Anforderungen von ShaRef überprüft.

Es handelt sich um ein verteiltes System mit vielen Benutzern. Der Prototyp differenziert daher zwischen Applikation und Steuerung. Die Applikationen sind verteilt. Die Steuerung befindet sich auf einem zentralen Rechner. Die Benutzer Oberfläche für die Steuerung ist ebenfalls eine verteilte Applikation. Das System wie in Abbildung 4 skizziert hat eine typische Client-Server Architektur. Es besteht eine Aufgabenteilung zwischen Steuerung und den Applikationen.

Zugriffsschutz benötigt Informationen über Ressourcen und Berechtigungen der Benutzer. Der Entscheid, ob ein Benutzer eine Operation ausführen darf, oder nicht, basiert auf diesen Informationen. Zuallerletzt muss der unberechtigten Benutzer daran gehindert werden eine Operation auszuführen. In einem System gemäss Abbildung 4 stellt sich die Frage der Aufgabenverteilung. Aus praktischen Gründen ist es vorteilhaft möglichst viel Funktionalität Server seitig zu implementieren, diese muss dann nur einmal implementiert werden.

4.1 Applikations seitige Funktionalität

Die Applikations Entwickler müssen den Zugriffsschutz implementieren. Im konkreten Fall müssen die Entwickler entscheiden was für Arbeitsgruppen Typen sie den Benutzern der Applikation für die Verwaltung der Ressourcen anbieten wollen. Sie müssen für jede Arbeitsgruppe die zugehörigen Ressourcen Typen festlegen und pro Arbeitsgruppen Typ eine passende Menge von Rollen festlegen

Die Umsetzung der Zugriffskontrolle in der Applikation ist ebenfalls Sache der Applikations Entwickler. Zuerst müssen für alle möglichen Operationen auf den Ressourcen die benötigten Rollen festgelegt werden. Danach müssen die Operationen für den Zugriffsschutz erweitert werden. Bevor eine bestimmte Operation auf einer Ressource ausgeführt wird, wird die Applikation einen Aufruf der Steuerung ausführen. Sie wird die sich über die Rolle des aktuellen Benutzers für die Arbeitsgruppe informieren. Auf Grund des Resultats muss die Applikation dann die Operation ausführen oder eben nicht ausführen.

Die Applikations Entwickler müssen die Arbeitsgruppe definieren. Die Arbeitsgruppe muss aber auch in der Steuerung abgebildet werden. Um Access Control zu nutzen muss eine Applikation die Arbeitsgruppen bei der Steuerung registrieren. Weiter muss die Applikation jede Ressource einer entsprechenden Arbeitsgruppe zuordnen.

Die Arbeit der Applikations Entwickler kann mit einem Server seitigen Applikations Layer für die Steuerung erleichtert werden. In diesem Layer existieren die verschiedenen Typen von Arbeitsgruppen mit den zugehörigen Rollen. Der Prototyp implementiert eine solche Schicht für die Arbeitsgruppen von ShaRef im Code. Ein Applikations Layer auf dem Server könnte konfigurierbar gehalten werden. Bei den benötigten Informationen geht es um den Arbeitsgruppen Typ und um einige Angaben zu den Rollen der Arbeitsgruppe.

4.2 Server seitige Funktionalität

Der Zugriffsschutz entscheidet, ob ein bestimmter Benutzer eine bestimmte Operation auf einer bestimmten Ressource ausführen darf oder eben nicht ausführen darf. Die Steuerung verwaltet die Berechtigungen der Benutzer für die Ressourcen und kann so die Anfragen der Applikationen nach der Rolle eines Benutzers für eine Arbeitsgruppe beantworten.

Die Steuerung implementiert nicht alle im Benutzerkonzept beschriebene Funktionalität. Sie soll aber aufzeigen, dass die im Konzept formulierten Gedanken funktionsfähig sind. Bei der Auswahl der zu implementierenden Funktionalität wurden die Anforderungen von ShaRef berücksichtigt.

Eine Einschränkung betrifft die Menge der Rollen pro Arbeitsgruppe. Der Prototyp kann pro Arbeitsgruppen Typ nur einen hierarchischen Satz von Rollen verarbeiten. Auch mit dieser Einschränkung können die Anforderungen des ShaRef Projekts vom Prototyp erfüllt werden. Diese Einschränkung vereinfacht die Server seitige und auch die Applikations seitige Implementierung der Zugriffskontrolle.

Der Prototyp implementiert die System Arbeitsgruppe. Die System Arbeitsgruppe hat einen hierarchischen Satz von Rollen. Es gibt neben der obligatorischen Administratoren Rolle eine Rolle mit der Berechtigung zum Erstellen von Bibliographien und Gruppen und eine weitere Rolle mit der Berechtigung zum Erstellen von Bibliographien. Die Berechtigung zum Erstellen eines Benutzer ist nicht kontrolliert. An der Mitarbeit interessierte Personen können sich ohne Eingriff eines System Administrators selber registrieren. Der

registrierte Benutzer verfügt als solcher über keine für das System relevanten Berechtigungen, dieses Vorgehen birgt also keine Risiken. Der registrierte Benutzer ist sichtbar für andere Benutzer. Diese können dem neuen Benutzer Berechtigungen in Form von Rollen für Arbeitsgruppen zuweisen. Damit wird der Benutzer erst zur Mitarbeit befähigt.

Die Steuerung kann ohne Änderungen an den inneren Strukturen beliebige Arbeitsgruppen Typen verarbeiten. Zur Unterstützung der Applikations Entwickler von ShaRef implementiert der Prototyp einen Server seitigen Applikations Layer für ShaRef. Darin sind die Bezeichnungen der Arbeitsgruppen Typen von ShaRef und die zugehörigen Rollen definiert. Der Applikations Layer für Sharef auf dem Server ist fürr den Prototypen fest im Code implementiert.

- Die Bibliographie ist eine Ressourcen basierte Arbeitsgruppe für ShaRef. Sie verwaltet eine Sammlung von Literatur Referenzen - das sind die Ressourcen. Weiter implementiert der Prototyp ein hierarchische Satz von Rollen für die Bibliographie. Da gibt es eine Leser Rolle, eine Benutzer Rolle mit Schreibrechten und die obligatorische Administratoren Rolle.
- Der Gruppe ist eine administrative Arbeitsgruppe für ShaRef. Die Arbeitsgruppe definiert neben der Administratoren Rolle nur die Mitglieder Rolle ohne spezielle Berechtigungen.

Der Prototyp implementiert einige der in den Abschnitten und beschriebenen Erweiterung des Modells für Zugriffsschutz. Wie im Konzept beschrieben erlaubt es auch der Prototyp an die Stelle eines Benutzers im System eine beliebige Arbeitsgruppe zu setzen. Das bedeutet konkret, dass in einer Arbeitsgruppe eine weitere Arbeitsgruppe eine Rolle übernehmen kann.

Nicht implementiert ist die Funktionalität zur differenzierten Verteilung der erhaltenen Berechtigungen auf die einzelnen Rollen der nehmenden Arbeitsgruppe. Erteilt der Administrator einer Arbeitsgruppe einer weiteren Arbeitsgruppe eine Rolle für die erste Arbeitsgruppe, so gehen die entsprechenden Berechtigungen an alle Mitglieder der zweiten Arbeitsgruppe. Wie später gezeigt wird, können mit dieser Vereinfachung die Anforderungen von ShaRef trotzdem erfüllt werden.

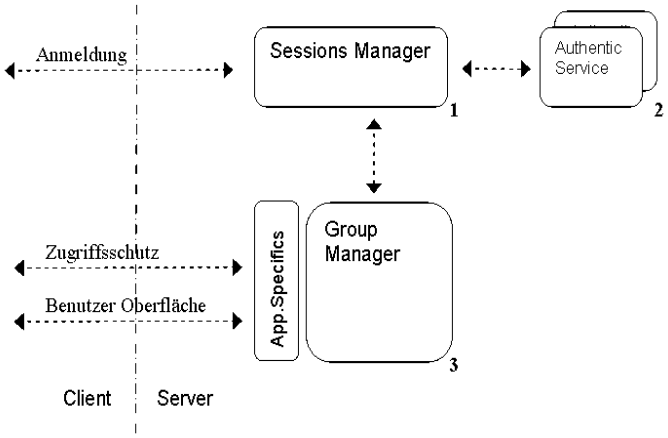

Abbildung 5: Server Komponenten

5 Implementierung

Das System wird als Java RMI Applikation[14] implementiert. Es ist eine verteilte Applikation in klassischer drei Schichten Architektur. Auf dem Server befindet sich das Datenbank Management System und die Applikations Logik. Die dritte Schicht sind die Applikationen mit Zugriffsschutz.

Arbeitsgruppen und Benutzer werden durch Server seitige Objekte repräsentiert. Die gesamte Information über Benutzer und Berechtigungen sind in den Server Objekten im Arbeitsspeicher. Anpassungen an den Berechtigungen und Auswertung von Benutzer Anfragen erfolgen direkt auf den Server Objekten.

Informationen über Benutzer und Berechtigungen werden mit dem Datenbank Management System (DBMS) persistent gemacht. Jede Änderung an einem Objekt der Applikations Logik wird in einer Transaktion auch in die Datenbank geschrieben. Damit könne bei einem Server Ausfall oder bei einem Server Neustart alle Benutzer- und Arbeitsgruppen Objekte rekonstruiert werden.

Die Steuerung besteht aus mehreren Komponenten. Bei der Komponenten Bildung wurde auf eine klare Aufgabenteilung und schmale Schnittstellen geachtet.

Der Group Manager hat zwei Schnittstellen. Eine Schnittstelle bedient die Applikationen mit Zugriffsschutz. Sie beantwortet die Fragen nach der Rolle eines Benutzer für eine bestimmte Arbeitsgruppe. Die zweite Schnittstelle wird für die Programmierung von Benutzeroberflächen zur Verwaltung von Arbeitsgruppen genutzt.

Der Session Manager verwaltet die Arbeitssitzungen der Benutzer. Er stellt die Schnitt-

[14] siehe dazu William Grosso: Java [RMI]

stelle für die Anmeldung der Benutzer zur Verfügung. Für die Authentisierung der Benutzer nutzt der Session Manager einen Authentisierungs Dienst. Der Prototyp implementiert eine eigenen Authentisierungs Dienst, kann aber auch externe Dienste nutzen. Der Austausch zwischen Session Manager und Group Manager beschränkt sich im auf wenige Informationen zum Benutzer und zur Arbeitssitzung.

Der Autor hat eine Benutzeroberfläche für die Verwaltung der Arbeitsgruppen entwickelt. Das Eclipse Plugin kann in den ShaRef Rich Client[15] integriert werden. Es wird im Kapitel 5.3 detailliert beschrieben.

5.1 Group Manager

Der Group Manager ist die Server Komponente für die Verwaltung der Benutzer und ihrer Berechtigungen für die Arbeitsgruppen. Die erste Aufgabe des Group Managers ist es, die Fragen nach den Berechtigungen eines Benutzers zu beantworten. Damit er diese Aufgabe wahrnehmen kann muss jede Applikation die neuen Arbeitsgruppen registrieren.

Der Benutzer der eine Arbeitsgruppe registriert, erhält die Administratoren Rolle für die neue Arbeitsgruppe. Er kann jetzt weiteren Benutzern eine Rolle für diese Arbeitsgruppe zuweisen. Mit diesen Informationen kann der Group Manager die Frage nach der Rolle eines Benutzers für eine bestimmte Arbeitsgruppe beantworten.

Der Group Manager verwaltet Arbeitsgruppen. Neben der System Arbeitsgruppe und den administrativen Arbeitsgruppen, spiegelt er auch die Ressourcen basierten Arbeitsgruppen der Applikationen. Jede Arbeitsgruppe existiert als Objekt in der Applikations Logik der Steuerung.

Es sind nicht nur die Ressourcen der Applikationen, die einen Zugriffsschutz benötigen. Auch die Arbeitsgruppen Objekte im Group Manager müssen geschützt werden. Es geht um die administrativen Berechtigungen in der Arbeitsgruppe. Das sind die Berechtigung neue Mitglieder in die Arbeitsgruppe aufzunehmen oder bestehende Mitglieder aus der Arbeitsgruppe zu entfernen. Auch das Ändern der Rolle eines Mitglieds der Arbeitsgruppe ist eine administrative Berechtigungen.

In einer Arbeitsgruppe verfügen nur die Administratoren die über solche Berechtigungen. Der Group Manager muss bei allen Operationen, die zu Änderungen an den administrativen Berechtigungen führen, überprüfen ob der ausführende Benutzer ein Administrator der betroffenen Arbeitsgruppe ist. Einige Methoden des Group Managers generieren also einen weiteren Aufruf des Group Managers.

Bei der Implementierung des Prototyps wurde versucht die Steuerung mit möglichst

[15] Der ShaRef Rich Client wird auf Basis von Eclipse RCP implementiert

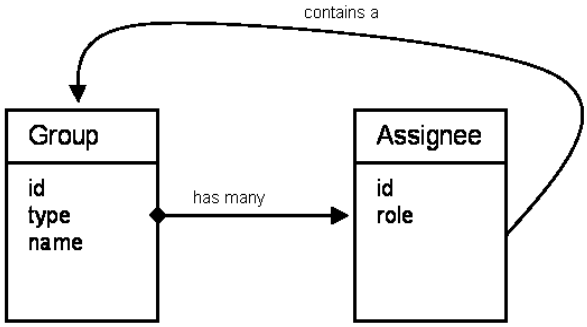

Abbildung 6: Klassen Diagramm

wenigen Klassen zu realisieren. Die Umsetzung des Konzepts der selbstorganisierten Arbeitsgruppe ist in Abbildung 6 dargestellt. [16]

Neben den im Arbeitsgruppe gibt es noch die Benutzer. Auch der Benutzer ist als Objekt im Group Manager gespiegelt. Das Objekt benötigt ebenfalls einen Zugriffsschutz. Um die Anzahl der Objekt Typen möglichst klein zu halten wird auch der Benutzer als Arbeitsgruppe modelliert - wenn auch mit beschränkter Funktionalität.

Die Interna der Benutzer Arbeitsgruppe sind minimal. Die einzige Resource dieser Arbeitsgruppe ist der Benutzer selber. Die Administratoren Rolle wird vom Benutzer selbst belegt. Es gibt keine weiteren Rollen. Auch können keine Administratoren hinzugefügt oder entfernt werden. Die Hauptfunktion eines Benutzers ist es Berechtigungen auf anderen Arbeitsgruppen auszuüben. Der Benutzer erhält Berechtigungen in Form von Rollen für andere Arbeitsgruppen. Die Fähigkeit Berechtigungen auf andere Arbeitsgruppen auszuüben, ist für die Benutzer Arbeitsgruppe natürlich und leicht verständlich, wirft aber für andere Arbeitsgruppen Typen Fragen auf. Die Verallgemeinerung diese Vorgangs für weitere Arbeitsgruppen Typen ist bereits im Abschnitt 3.3 beschrieben.

Eine Arbeitsgruppe hat gemäss Objektmodell zwei Hauptfunktionen.

- Verwalten der Arbeitsgruppen eigenen Rollen
  Dies ist die Aufgabe des Administrators. Die Arbeitsgruppe hat die Funktionalität für das Zuweisen der Arbeitsgruppe eigenen Rollen an andere Arbeitsgruppen und das Verteilen der Berechtigungen auf die eigenen Rollen.

- Ermitteln der Berechtigungen eines Benutzers
  Diese Aufgabe kann von der Arbeitsgruppe automatisch erledigt werden. Sie kann die Rollen eines Benutzers für die Arbeitsgruppe selbst und die aus dem Erweiterungs Mechanismus resultierenden Rollen für andere Arbeitsgruppe ermitteln.

[16] Der Prototyp ist mit der Programmiersprache Java implementiert. Objekttypen können mit dieser Programmiersprache direkt durch Klassen abgebildet werden.

Die Umsetzung des Modells für Zugriffsschutz benötigt zwei Klassen und ist im Klassendiagramm in Abbildung 6 dargestellt. Arbeitsgruppen und Benutzer werden von der Group Klasse modelliert. Die Beziehungen zwischen den Arbeitsgruppen und den Benutzern können mit der Assignee Klasse abgebildet werden.

Aus dem Diagramm wird ersichtlich, dass es im wesentlichen um eine Zuordnung von Group Klasse zu Group Klasse geht. Mit der Zuordnung fliessen die Berechtigungen. Die zugeordnete Group Klasse wird in eine Assignee Klasse eingebettet. Hier finden Methoden und Attribute zur Regulierung des Uebergangs der Berechtigungen Platz.

Group Klasse und Assignee Klasse implementieren einen Zugriffsschutz. Alle Methoden von Group Klasse und Assignee Klasse, die irgendwelche Daten verändern, sind abgesichert. Konkret verlangen diese Methoden beim Aufruf den Session Key als Argument. Vor Ausführung der Operation überprüfen diese Methoden, die Berechtigung des aufrufenden Benutzers mit Hilfe des Schlüssel und den gleichen Methoden des Group Managers, die auch von den Applikationen mit Zugriffsschutz verwendet werden.

Die Erweiterung zur Verteilung von Berechtigungen anderer Arbeitsgruppen auf die Rollen der Arbeitsgruppe ist implementiert. Dabei wird nicht zwischen den verschiedenen Rollen der Arbeitsgruppe differenziert. Alle Mitglieder einer Arbeitsgruppe profitieren ohne Unterschied von solchen Berechtigungen. Eine entsprechender Zusatz könnte aber in der Assignee Klasse Platz finden.

5.2 Session Manager

Der Session Manager ist die Server Komponente für die Verwaltung der Benutzer und der Arbeitssitzungen. Er stellt die Schnittstelle für die Anmeldung der Benutzer zur Verfügung. Bei der Anmeldung eines Benutzers muss der Session Manager den Benutzer authentisieren. Für die Authentisierung der Benutzer nutzt der Session Manager einen Authentisierungs Dienst. Der Prototyp implementiert eine eigenen Authentisierungs Dienst, kann aber auch externe Dienste nutzen.

Das System besteht aus den Server Komponenten für Zugriffsschutz und den Applikationen mit Zugriffsschutz. Das verteilte System benötigt an verschiedenen Stellen die Information zur Identität des Benutzers. Mit dem Session Konzept soll diese Information im ganzen System verfügbar gemacht werden.

Dies wäre auch möglich, durch wiederholte Aufrufe des Authentisierungs Dienstes. Dabei muss aber jedes Mal das Passwort über das Netz. Dieser Vorgang sollte über eine sichere Verbindung gehen.

Das Session Konzept ermöglicht die einmalige Authentisierung des Benutzers am Anfang der Arbeitssitzung. Der Benutzer erhält dann einen Schlüssel, den Session Key. Im System wird der authentifizierte Benutzer durch den den Session Key repräsentiert. Der Schlüssel

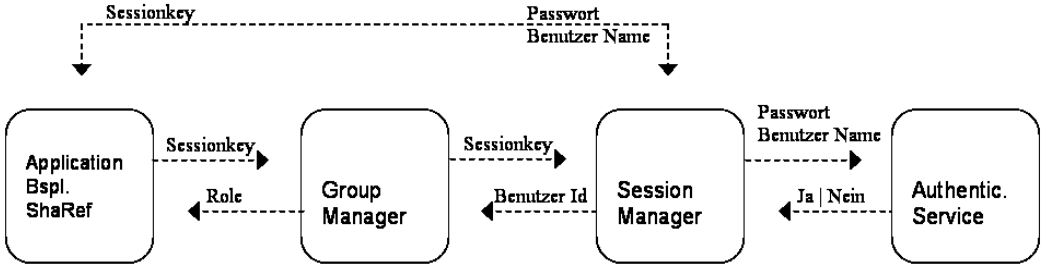

Abbildung 7: Session Manager

hat eine begrenzter Gültigkeitsdauer. Ein gestohlener Schlüssel kann nur für kurze Zeit eingesetzt werden.

Das System zur Authentisierung der Benutzer ist in Abbildung 7 dargestellt. Der Benutzer einer Applikation muss sich beim Start der Applikation authentisieren. Ueber eine Anmelde-Dialog sendet der Benutzer den Benutzernamen und das Passwort an den Session Manager. Der Session Manager nutzt den Authentisierungs Dienst zur Ueberprüfung des Passworts. Ist das Passwort korrekt generiert der Session Manager einen Schlüssel. Die Applikation erhält den Schlüssel zurück.

Die Applikation kann mit dem Schlüssel zum Beispiel die Rollen eines Benutzers für eine bestimmte Arbeitsgruppe ermitteln. Dazu senden Sie den Schlüssel an den Group Manager. Der Group Manager wiederum leitet diesen an den Session Manager weiter, der den Schlüssel auf Gültigkeit überprüft. Ist der Schlüssel gültig erhält der Group Manager die Identifikation des Benutzers zurück. Damit kann er die Rolle des Benutzers für die besagte Arbeitsgruppe ermitteln. Zu allerletzt erhält die Applikation die Rolle des Benutzers und entscheidet auf dieser Basis, ob der Benutzer die Operation ausführen darf, oder nicht. [17]

5.3 Eclipse Plugin

Der Rich Client für ShaRef ist auf der Eclipse Plattform implementiert. Der Autor hat für den Eclipse Rich Client von ShaRef ein ergänzendes Plugin für das Management der Berechtigungen implementiert.

Die Eclipse Plattform basiert auf einer Plugin Architektur. Das heisst konkret, dass der Eclipse Kern eigentlich keine andere Funktionalität besitzt, als Plugins [18] auszuführen.

[17] Die Kommunikation zwischen den Komponenten ist für den Prototyp mit Java RMI realisiert. Die Information werden nicht verschlüsselt. Um das System abzusichern sollte zumindest der Anmelde Vorgang über eine gesicherte Verbindung ausgeführt werden.

[18] Plugins sind Komponenten, die ohne weiteres zu einer bestehenden Applikation hinzugefügt wer-

Die Eclipse Plattform kann als Laufzeitumgebung für eigene Programme genutzt werden. Die Entwickler einer Applikation können von einer Anzahl Dienste und vorbereiteter Komponenten zum Aufbau der Benutzeroberfläche profitieren.

Der Rich Client für ShaRef basiert auf der Eclipse Plattform und lässt sich ebenfalls mit Plugins erweitern. Es ist daher nahe liegend den Rich Client für ShaRef mit einem Plugin für das Management von Berechtigungen zu erweitern. Das Plugin für ShaRef besteht im wesentlichen aus zwei Benutzerdialogen, die Funktionalität auf dem Server aufrufen. Der Group Explorer dient zur Bearbeitung von Arbeitsgruppen. Mit dem Group Finder kann der Benutzer eine Arbeitsgruppen selektieren.

5.3.1 Group Finder

Der Group Finder dient der Selektion von Arbeitsgruppen oder Benutzern. Für jeden Arbeitsgruppen Typ gibt es eine separate Instanz des Group Finders. Der Group Finder hat drei Anzeigefelder. Das mittlere Anzeigefeld zeigt alle vorhandenen Arbeitsgruppen eines bestimmten Typs, respektive alle Benutzer des Systems. Die Arbeitsgruppen oder Benutzer sind alphabetisch geordnet. Im unteren Anzeigefeld wird ein beschreibender Text zur selektierten Arbeitsgruppe oder zum selektierten Benutzer angezeigt. Das obere Anzeigefeld ist ein Textfeld. Der Benutzer kann hier einen Filter angeben. Der Filter ist ein beliebige Zeichenfolge. Im mittleren Textfeld werden jeweils nur solche Arbeitsgruppen oder Benutzer angezeigt, die die entsprechende Zeichenfolge im Namen enthalten.

5.3.2 Group Explorer

Der Group Explorer dient zur Bearbeitung von Arbeitsgruppen. Der Benutzer kann über die Befehle in der Symbolleiste einen entsprechenden Group Finder aufrufen eine beliebige Arbeitsgruppe oder eine Benutzer selektieren. Die Arbeitsgruppe oder der Benutzer wird dann zur Ansicht und Bearbeitung in das Anzeigefeld des Group Explorers eingefügt. Die Auswahl an Arbeitsgruppen im Group Explorer kann vom Benutzer selbst bestimmt werden und ist persistent. Das heisst, dass der Group Explorer wenn er geschlossen und wieder geöffnet wird die gleichen Arbeitsgruppen anzeigt. In der Symbolleiste existiert auch ein Befehl zum Entfernen von Arbeitsgruppen.

Eine Arbeitsgruppe wird im Group Explorer mit allen Mitgliedern angezeigt. Der Arbeitsgruppen Typ wird mit einem entsprechenden Symbol kommuniziert. Mit den gleichen Symbolen werden auch der Typ der Mitglieder der Arbeitsgruppe angezeigt. Zusätzlich

---

den können. Die bestehende Applikation integriert das Plugin und stellt dem Benutzer die zusätzliche Funktionalität zur Verfügung.

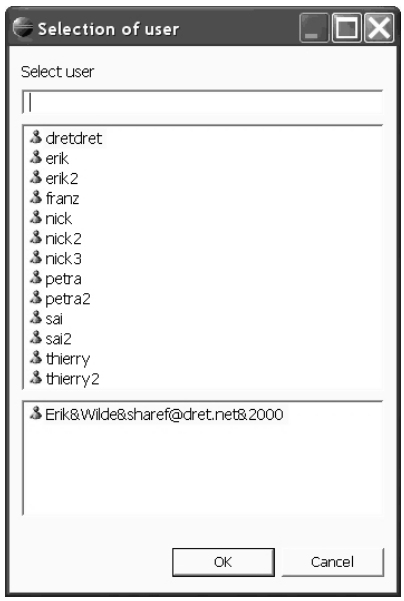

Abbildung 8: Group Finder

zum Namen wird auch die Rolle des Mitglieds für die Arbeitsgruppe angezeigt. Die Reihenfolge der Mitglieder folgt der Hierarchisch der Rollen. Zuoberst sind also immer die Administratoren.

Das Kontext Menu bieten die nötige Funktionalität zum Hinzufügen von neuen Mitgliedern zu einer Arbeitsgruppe oder Entfernen von bestehenden Mitgliedern aus einer Arbeitsgruppe. Auch können die Rollen der Mitglieder innerhalb der Arbeitsgruppe geändert werden.

5.4 Zusammenfassung

Der Prototyp implementiert das im Kapitel 3 formulierte Konzept. Es sind nicht alle im beschriebenen Möglichkeiten implementiert. Die Auswahl erfolgte mit Blick auf die Anforderungen der Beispiel Applikation ShaRef.

Der Prototyp implementiert mehrere Arbeitsgruppen Typen. Mit der Bibliographie wird eine Ressourcen basierte Arbeitsgruppe für ShaRef implementiert. Dazu kommen ein Beispiel für eine Administrative Arbeitsgruppe und die System Arbeitsgruppe.

Einige der im Abschnitt 3.3 beschriebenen Erweiterungen des Modells sind implementiert. Jede Arbeitsgruppe kann eine beliebige Rolle in einer anderen Arbeitsgruppe ausfüllen. Dabei fliessen die damit verbundenen Berechtigungen an die Mitglieder der Arbeitsgrup-

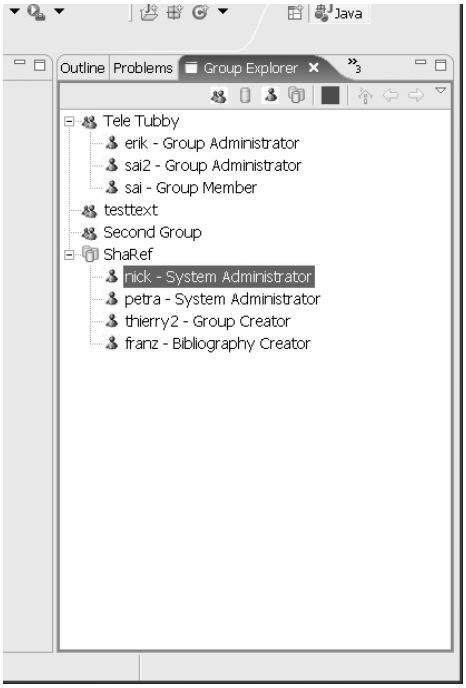

Abbildung 9: Group Explorer

pe. Nicht implementiert sind die im Abschnitt beschriebenen Mechanismen zur gezielten Erweiterung von einzelner Rollen mit Rollen für andere Arbeitsgruppen.

6 Beurteilung am Beispiel von ShaRef

Die Ueberprüfung von Konzept und Prototyp orientiert sich an den Anforderungen der Beispiel Applikation ShaRef. ShaRef ist eine Lösung für das kollaborative Verwalten von Literatur Referenzen. Das ShaRef-Projekt ist Teil der ETH World Initiative. Dieses Programm zielt auf die Entwicklung und Einführung von Technologien für Kommunikation und Zusammenarbeit. [19]

Im Abschnitt 6.1 wird aufgezeigt, wie die Anforderungen von ShaRef bezüglich Zugriffsschutz mit dem Prototypen abgedeckt werden können. Das zentrale Objekt von ShaRef ist die Bibliographie, eine Sammlung von Literatur Referenzen. Die Bibliographie wird im Prototyp als Arbeitsgruppe modelliert.

Die Anforderungen an die Authentisierung werden im Abschnitt 6.2 behandelt. Der Auftraggeber ETH World möchte ShaRef an den Authentisierungsdienst NETHZ anbinden. Mitarbeiter mit einem solchen Konto solchen sich mit NETHZ für ShaRef authentisieren können. ShaRef soll aber auch von externen Mitarbeitern oder Projektteilnehmern ohne NETHZ Konto genutzt werden können.

Es existieren sehr detailliert formulierten Anforderungen an das Benutzerkonzept. Sie sind in Form von Anwendungsfällen im UseCase Dokument 'Use Cases for Group Setting and Bibliography Administration' der ShaRef Projektgruppe festgehalten. Im Abschnitt 6.3 werden diese Anwendungsfälle ausgewertet.

6.1 Bibliographie

Das für ShaRef zentrale Objekt mit Zugriffsschutz heisst Bibliographie. Eine Bibliographie ist eine Sammlung von Literatur Referenzen. Die Literatur Referenzen in einer Bibliographie sind bezüglich Zugriffsschutz nicht differenziert.

Die Anforderungen an den Zugriffsschutz einer Bibliographie lassen sich wie folgt zusammenfassen. Eine ShaRef Bibliographie ist entweder privates Objekt einzelner Benutzer, kann aber auch von einer Gruppe von Benutzern gemeinsam genutzt werden. Privat- oder Gruppen- Bibliographie sollen einem erweiterten Benutzerkreis im Lesemodus zugänglich gemacht werden können. Weiter kann der Besitzer einer Bibliographie diese im Web publizieren. Diese Anforderungen können mit dem Prototyp wie folgt erfüllt werden:

Die Bibliographie wird als Arbeitsgruppe modelliert. Zur Abbildung der Berechtigungen in der geforderten Differenzierung, werden drei Rollen benötigt. Es gibt die im Konzept obligatorische Administrator-Rolle, eine Benutzer-Rolle und eine Leser-Rolle. Diese Rollen sind hierarchisch organisiert. Der Benutzer verfügt über alle Berechtigungen eines

[19] siehe auch Erik Wilde: A Tool for Bibliography Management and Sharing: The [ShaRef] Project

Lesers. Der Administrator verfügt über alle Berechtigungen eines Benutzers.Die einzelnen Rollen verfügen über folgende Berechtigungen.

- *Administrator Rolle*
  Der Administrator einer Bibliographie kann gemäss Objektmodell einem beliebigen Benutzer eine Rolle zuteilen und auch wieder entziehen. Weiter verfügt er über die Berechtigung die Bibliographie zu publizieren.

- *Benutzer Rolle*
  Der Benutzer einer Bibliographie kann neue Literatur Referenzen in die Bibliographie einfügen. Er kann auch bestehende Referenzen in der Bibliographie anpassen oder aus der Bibliographie entfernen.

- *Leser Rolle*
  Der Leser einer Bibliographie hat Lesezugriff auf alle in einer Bibliographie vorhandenen Referenzen. Er kann eine Kopie einer solchen Referenz erstellen und in eine eigene Bibliographie als eigenständiges Objekt einfügen.

6.2 Authentisierung

Die Anforderungen an die Authentisierung lassen sich wie folgt zusammenfassen. Der Auftraggeber ETH World möchte ShaRef an den Authentisierungsdienst NETHZ anbinden. Denn die meisten Mitarbeiter der ETH Zürich verfügen bereits über ein solches Konto. Diese Mitarbeiter sollen sich mit NETHZ für ShaRef authentisieren, d.h. mit Benutzername und Passwort von NETHZ auch für den Zugang zu ShaRef nutzen können. ShaRef soll aber auch von externen Mitarbeitern oder Projektteilnehmern ohne NETHZ Konto genutzt werden können. [20]

Der Prototyp implementiert neben der Anbindung an NETHZ einen eigenen Authentisierungsdienst. Neue Benutzer von ShaRef müssen sich registrieren. Sie wählen einen eindeutigen ShaRef Benutzernamen. Verfügt der Benutzer bereits über ein NETHZ Konto kann er den den NETHZ Benutzernamen verwenden. Um den ShaRef Authentisierungsdienst zu nutzen muss der Benutzer ein Passwort wählen. Verfügt er über ein NETHZ Konto kann er stattdessen 'Authentisierung über NETHZ' wählen. Ein Konflikte entsteht, wenn ein NETHZ Benutzername bereits von einem anderen Benutzer als ShaRef Benutzername verwendet wird. Der Prototyp bietet dafür eine Lösung mit einem Alias. Der Benutzer kann dann das NETHZ Passwort verwenden, hat aber einen anderen ShaRef Benutzernamen. [21]

[20] Details zum NETHZ Dienst auf der Webseite [NETHZ] der ETH Zürich.

[21] Für diese Lösung reduziert sich die Anbindung an NETHZ auf die Überprüfung von Passwörtern. Zur Verhinderung des beschriebenen Konflikts wird die Anbindung wesentlich aufwändiger.

Die geforderte Integration externer Mitarbeiter oder Projektteilnehmer kann ohne Eingriffe von System Administratoren erfolgen. Die Mitarbeit eines Externen an einer Projekt-Bibliographie erfordert zwei Schritte.

1. Der Externe Mitarbeiter registriert sich für ShaRef, wie in 6.2 beschrieben. Er ist jetzt für die Administratoren der Projekt Bibliographie sichtbar.
2. Ein Administrator der Projektbibliographie erteilt dem externen Mitarbeiter die Benutzer Rolle. Damit ist dieser Berechtigt eigene Referenzen in der Projekt- Bibliographie abzulegen und bestehende Referenzen zu bearbeiten.

Für die Verteilung der System Benutzer Rolle sind die System Administratoren verantwortlich. Sie müssen den Benutzern diese Rolle zuweisen. Für die Benutzer mit NETHZ Konto könnten diese Rolle bei der Registrierung neuer Benutzer automatisch zugewiesen werden.

6.3 ShaRef Use Case Dokument

Die ShaRef Projekt Gruppe hat Dokument mit den wichtigsten Use Case für ShaRef erarbeitet. Das Kapitel 4 'Use Cases for Group Setting and Bibliography Administration' enthält die für das Benutzerkonzept relevanten Anteile. Das Dokument befindet sich auf der beiliegenden CD.

Die nachfolgende zwei Tabellen enthalten die relevanten Use Cases mit der Angabe, ob der Use Case vom Prototyp erfüllt werden kann (x = Erfüllt, o = nicht erfüllt). Der Autor hat zu diesen Use Cases jeweils einen JUNIT Test Code geschrieben. Der Code befindet sich ebenfalls auf der CD.

Nr	Titel	Beschreibung	Erfüllt
4.1	Create Groups	Ein Benutzer mit der entsprechenden System Berechtigung erstellt eine neue Gruppe. Der Benutzer ist der Administrator dieser Gruppe.	x
4.2	Add User or Group to a Group	Der Administrator fügt einer Gruppe einen beliebigen Benutzer oder eine beliebige Gruppe hinzu.	x
4.3	Appoint a Group Administrator	Der Administrator einer Gruppe kann einem anderen Mitglied er Gruppe die Administratoren Rolle zuweisen.	x
4.4	Group Administrator steps back	Der Administrator einer Gruppe kann von seiner Rolle als Administrator zurücktreten. Das System überprüft, ob die Gruppe noch andere Administratoren besitzt. Falls ja, wird erhält der Administrator eine normale Mitglieder Rolle.	x
4.5	Remove User or Group from a Group	Der Administrator schliesst ein beliebiges Mitglied der Gruppe, sei es ein Benutzer oder sei es eine Gruppe, aus der Gruppe aus.	x

Tabelle 1: Auswertung gemäss Use Case Dokument ShaRef (Teil 1)

Nr	Titel	Beschreibung	Erfüllt
4.6	Delete a Group	Der Administrator löscht eine Gruppe. Das System überprüft, ob die zu löschende Gruppe nicht letzter Administrator einer anderen Gruppe ist. Ist dies der Fall verweigert das System die Operation.	x
4.7	Create a Bibliography	Ein Benutzer erstellt eine neue Bibliographie. Der Benutzer ist Administrator der neuen Bibliographie.	x
4.8	Assign Read Permission	Der Administrator einer Bibliographie weist einem beliebigen Benutzer die Leser Rolle zu.	x
4.9	Publish Bibliography	Der Administrator publiziert eine Bibliographie. Bemerkung: Dieser Vorgang hat keinen direkten Zusammenhang mit der Zugriffskontrolle.	O
4.10	Transfer Bibliography to a new Owner	Der Administrator einer Bibliographie überträgt die Bibliographie an einen anderen Benutzer oder an eine Gruppe. Bemerkung: dieser Use Case ist nicht direkt implementiert. Durch eine Kombination von Use Case 4.3 mit Use Case 4.4 kann das gewünschte Resultat erzielt werden.	x
4.11	Delete a Bibliography	Der Administrator einer Bibliographie löscht die Bibliographie. Das System überprüft, ob die zu löschende Bibliographie nicht letzter Administrator einer anderen Gruppe ist. Ist dies der Fall verweigert das System die Operation.	x

Tabelle 2: Auswertung gemäss Use Case Dokument ShaRef (Teil 2)

7 Schlussfolgerungen

Jedes Benutzerkonzept mit Access Control muss die Benutzer und ihre Berechtigungen für die Ressourcen verwalten. Die Art und Weise wie diese Aufgabe gelöst wird ist entscheidend für die Skalierbarkeit des Systems. Damit die Skalierbarkeit des Systems gewährleistet werden kann, muss die Aufgabe mit einem verteilten Ansatz gelöst werden. Das Benutzerkonzept ist auch der der Schlüssel zur kollaborativen Applikation. Das Wiki-Prinzip spielt dabei eine zentrale Rolle.

Der gewählte Ansatz verhilft dem Wiki-Prinzip mittels der selbstregulierten Arbeitsgruppe zum geforderten Zugriffsschutz. Das Resultat ist ein kollaboratives, skalierbares Benutzerkonzept mit Zugriffsschutz. Der Prototyp ist funktionsfähig. Der grosse Test steht aber noch aus. Die Beispiel Applikation ShaRef wird den Prototyp zur Steuerung der Berechtigungen nutzen. Der Prototyp für ShaRef erfüllt die in Form von Anwendungsfällen festgelegten Anforderungen der ShaRef Projektgruppe. Der grosse Praxis Test steht aber noch aus. Die Beispiel Applikation ShaRef wird den Prototyp zur Steuerung der Berechtigungen nutzen. In der Anwendung wird sich zeigen, ob das Konzept die Erwartungen bezüglich Kollaborativität und Skalierbarkeit wirklich erfüllt.

8 Ausblick, Offene Fragen

Der Prototyp für ShaRef zeigt die Funktionsfähigkeit des Konzepts. Einige der im Kapitel sect:entwurf beschriebenen Erweiterungen sind aber nicht implementiert. So kann zwar eine beliebige Arbeitsgruppe den Platz eines Benutzers im System übernehmen. Die gezielte Verteilung der Berechtigungen anderer Arbeitsgruppen auf die Rollen einer zweiten Arbeitsgruppe ist mit dem Prototyp nicht möglich. Datenmodell und Applikations Logik müssen dazu erweitert werden. Es ist auch nicht geklärt, wie diese zusätzliche Funktionalität in der Benutzeroberfläche angeboten werden kann.

Die vorliegende Arbeit nutzt das Konzept der selbstregulierten Arbeitsgruppe für die Verwaltung von Benutzern, Berechtigungen und Ressourcen. Das System besteht aus den Applikationen mit Zugriffsschutz und einer Server seitigen Komponente. Der funktionierender Zugriffsschutz ist ein Resultat des Zusammenspiels von Server Komponente und Applikation. Der Prototyp implementiert das Konzept als Server Komponente. Die Applikations seitigen Aufgaben werden nur am Rande behandelt. Die Möglichkeiten die sich zum Beispiel für einen Applikations übergreifenden Zugriffsschutz bieten sind nicht untersucht.

Literatur

[Arbeitsgruppen] Pietruschka Sabine: Führung selbstregulierter Arbeitsgruppen, München und Mering:Rainer Hampp Verlag, 2003

[Cooperative] Dix, Alan: Computer Supported Cooperative Work: A Framework, in: Rosenberg, Duska und Hutchison, Chris(Eds.), Design Issues in CSCW, S.9-26, London:Springer-Verlag London Limited 1994

[Collaborative] Edwards, W.Keith: Policies and Roles in Colaborative Applications, in: Proceedings of ACM Conference on Computer-Supported Cooperative Work, Boston MA:ACM Press 1996

[Discretonary] Sandhu, R./Munawer Q.: How To Do Discretonary Access Control Using Roles, in: Proceedings of the 3rd ACM Workshop on RBAC, 1998, S.47-54, Fairfax, VA:ACM Press 1998

[NETHZ] ETH Zürich: nethz Authentisierung und Autorisierung, 31.3.2005, http://www.id.ethz.ch/services/list/nethz_db (14.08.2005 9:45 MESZ)

[RMI] Grosso, William: Java RMI, Sebastopol , CA 95472: O'Reilly & Associates, Inc., 2002

[Role 1992] Ferraiolo, David/Kuhn, D.Richard: Role-Based Access Control, in: Proceedings of 15th National Security Conference, 1992, S.554-563

[Role 2003] Ferraiolo, David/Kuhn, D.Richard/Chandramouli, Ramaswamy: Role-Based Access Control, Boston/London:Artech House, 2003

[ShaRef] Erik Wilde: A Tool for Bibliography Management and Sharing: The ShaRef Project, D-Lib Magazine, Vol. 10, No. 9, September, 2004

[Wiki Way] Leuf, Bo/Cunningham, Ward: The Wiki Way, 2nd Printing, Boston und viele weitere:Addison Wesley, 2004

[Wikipedia] Deutschsprachige Wike Community: Hautpseite Wikipedia, Version vom 10. August 2005, http://de.wikipedia.org/wiki/Hauptseite (13.08.2005 9:00 MESZ)