Ein Framework für föderiertes Sicherheitsmanagement

Habilitationsschrift im Fach Informatik
an der Fakultät für Mathematik und Informatik der
Ludwig-Maximilians-Universität München

von

Helmut Reiser

2008

Ein Framework für föderiertes Sicherheitsmanagement

Habilitationsschrift im Fach Informatik
an der Fakultät für Mathematik und Informatik der
Ludwig-Maximilians-Universität München

von

Helmut Reiser

Tag der Einreichung: 27. Mai 2008

Fachmentorat:

Professor Dr. Heinz-Gerd Hegering, Ludwig-Maximilians-Universität München
Professor Dr. Martin Wirsing, Ludwig-Maximilians-Universität München
Professor Dr. Claudia Linnhoff-Popien, Ludwig-Maximilians-Universität München
Professor Dr. Burkhard Stiller, Universität Zürich, Schweiz, Eidgenösische Technische Hochschule Zürich, Schweiz

Danksagung

Diese Arbeit wurde während meiner Tätigkeit am Lehrstuhl für Kommunikationssysteme und Systemprogrammierung begonnen und am Leibniz-Rechenzentrum (LRZ) der Bayerischen Akademie der Wissenschaften abgeschlossen.

In beiden Positionen war und ist Prof. Dr. Heinz-Gerd Hegering mein Mentor, Förderer und Chef. Ihm gebührt mein ganz besonderer und herzlicher Dank, denn er war es, der mich viele Jahre gefördert und auch gefordert hat. Die Arbeit in seinem Team war sehr anregend und seine wissenschaftliche Betreuung hervorragend. Er hat mir immer wieder neue anspruchsvolle und herausfordernde Aufgaben und die entsprechende Verantwortung übertragen, so dass ich mich persönlich weiterentwickeln konnte. Dabei hat er mir immer sehr viel Freiheit gelassen und mich ermutigt eigene Wege zu gehen und neue Projekte zu planen und zu realisieren.

Dank gilt auch meinen Fachmentoren Prof. Dr. Martin Wirsing und Prof. Dr. Claudia Linnhoff-Popien für ihre Unterstützung und ihren persönlichen Einsatz der zum Gelingen dieser Arbeit erheblich beigetragen hat. Bei Prof. Dr. Burkhard Stiller, Zürich möchte ich mich dafür bedanken, dass er sich sofort bereit erklärt meine Arbeit als externer Gutachter im Mentorat zu bewerten.

Der stellvertretende Leiter des LRZ, Dr. Victor Apostolescu und der Abteilungsleiter der Abteilung Kommunikationsnetze (KOM), Alfred Läpple haben mich immer unterstützt und ermuntert diese Arbeit zu vollenden. Die Mitarbeiter der Abteilung KOM und insbesondere die Mitglieder der Gruppe Netzplanung waren mir immer kompetente Diskussionspartner, haben mich unterstützt und mir auch das eine oder andere Mal den Rücken freighalten. Dafür bin ich ihnen sehr dankbar.

Danken möchte ich auch allen früheren und aktiven Kolleginnen und Kollegen des Munich Network Managment Teams (MNM-Team). Die Kompetenz, die Diskussionen und auch die Streitkultur des Teams machen meine wissenschaftliche Heimat aus.

Mein ganz besonderer tiefempfundener Dank gilt meiner Familie. Mathilde für die vorbehaltlose Unterstützung und den langen Atem auf dem gemeinsamen Weg. Meiner Tochter Mathilde Susanna bin ich dankbar, dass sie mich gelehrt und immer wieder sanft daran erinnert hat, dass es neben der (wissenschaftlichen) Arbeit noch andere, sehr freudvolle und sinnstiftende Dinge im Leben gibt :-)

Für Mathilde^2 *Susanna*

Kurzfassung

Föderationen, d.h. Kooperationsformen zwischen rechtlich unabhängigen und selbständigen Unternehmen werden gebildet, um ein gemeinsames Ziel zu verfolgen, das durch die Bildung der Föderation gefördert oder nur mit einer Föderation überhaupt erreichbar wird. Die beteiligten Organisationen bilden mit ihren Individuen und ihren Ressourcen — unter Beibehaltung ihrer lokalen Autonomie — eine virtuelle Organisation. Durch Föderationen und virtuelle Organisationen entsteht der dringende Bedarf nach organisationsübergreifendem Sicherheitsmanagement sowie für interorganisationale Prozesse.

In dieser Arbeit wird deshalb ein Rahmenwerk für föderiertes Sicherheitsmanagement entwickelt. Dazu werden Grids, die als Phänotyp für Föderationen angesehen werden können, als technische Basis betrachtet. Es wird eine in dieser Weise neuartige, umfassende Sicherheits- und Anforderungsanalyse durchgeführt, auf deren Basis eine Szenario-unabhängige Hierarchie von Sicherheitsdienstklassen definiert und Abhängigkeitsgrade zwischen den Sicherheitsdiensten eingeführt werden. Desweiteren wird ein neues allgemeingültiges Klassifikations- und Bewertungsschema für interorganisationale Sicherheitsmechanismen in Form eines Kriterienkataloges entwickelt. Dieser Kriterienkatalog stellt das strukturbildende Element für eine umfassende und Szenario-unabhängige Analyse und Bewertung von Sicherheitsmechanismen dar. Es werden auf dieser Basis Mechanismen für alle Sicherheitsdienste in der Hierarchie der Sicherheitsdienstklassen untersucht und bewertet. Als ein wichtiges Ergebnis dieser Vorgehensweise wird eine Defizit- und Schwachstellenanalyse der bestehenden Sicherheitsmechanismen verschiedener Middleware-Technologien erarbeitet. Für zwei bisher sicherheitstechnisch nur unbefriedigend abgedeckte Dienstklassen werden dann konkrete Verbesserungen vorgeschlagen. Ein rekursiver Trust Algorithmus dient der dynamischen Berechnung von Trust Levels zwischen den Partnern und der Übertragung von Vertrauenswerten. Mit Implanted Chain Certificates (IChC) wird ein Verfahren zum effizienten verteilten Gruppenmanagement und zur dezentralen Autorisierung auf Basis von Zertifikatsketten angegeben. Die Arbeit schließt mit einem Vorgehensmodell, das die Anwendung des Rahmenwerkes erläutert. Dabei werden die in dieser Arbeit entwickelten Szenario-unabhängigen Konzepte so aufbereitet, dass sie von Sicherheitsverantwortlichen oder Sicherheitsadministratoren in einer Analyse- und Synthesephase in ihrem konkreten Szenario angewendet werden können.

Abstracts

Cooperations between legally independent and autonomous organizations, called federations, are formed with the objective of a common target. The federation promotes the achievement of this objective or it is even a necessary prerequisite for achieving the cooperation's goal. The participating organizations of a federation with their individuals and their resources establish a so called virtual organization retaining their autonomy and protection of their interests. Due to federations and virtual organizations there are strong needs for a security management crossing organizational boundaries and inter-organizational processes.

A framework for federated security management is therefore developed within this paper. Grids, which can be regarded as a phenotype of federations, build the technical basis of the research presented in this work. A comprehensive security and requirements analysis is the basis for the development of a scenario independent hierarchy of security service classes which contains a definition of dependency levels between the security services. Furthermore a universal classification and rating schema in terms of a systematically developed criteria catalog for inter-organizational security mechanisms has been evolved. The criteria catalog is used in a structuring way for a comprehensive and scenario independent analysis, review and rating of security mechanisms. Security services are implemented by security mechanisms and therefore mechanisms for all security services of the hierarchy have been analyzed. One conclusion of this work is a breakdown of deficiencies and weaknesses of existing security mechanisms of different middleware technologies. Improvement for two service classes has been proposed. For dynamic calculation and transmission of trust levels between partners a new recursive trust algorithm is proposed. With Implanted Chain Certificates (IChC) also a method for efficient and distributed group membership management and decentralized authorization based on certificate chains has been evolved. At the end of the paper a process model of the presented framework points out how to apply the framework. Chief security officers or security administrators might advantageously use the scenario independent concepts within an analysis phase and a synthesis phase when designing security policies and a security architecture for their concrete scenarios.

INHALT

Kapitel 1

Einführung

Inhaltsverzeichnis

In der hoch spezialisierten und arbeitsteiligen Wirtschaft sind Unternehmen und Organisationen häufig gezwungen Dienstleistungen oder Teile ihrer Produktion an andere Unternehmen auszulagern (**Outsourcing**). Auch bei der Entwicklung neuer Produkte oder der Bereitstellung von Diensten zeigt sich häufig, dass rechtlich unabhängige Unternehmen, für beschränkte Zeit, Kooperationen eingehen, um gemeinsame Projekte zu verwirklichen. Beispielsweise hatte der Automobilhersteller BMW Group im Jahr 2004 rund 60.000 Unternehmen, die bei BMW als Kooperationspartner registriert waren und in verschiedensten Projekten mit BMW zusammenarbeiteten [Metz 04].

Grid Computing: gemeinsame Nutzung von Ressourcen

In den späten 90er Jahren ist das Konzept des **Grid Computings** entstanden. Die Grundidee der Grid-Bewegung ist es IT-Ressourcen einschließlich IT-Dienste in großem Umfang gemeinsam zu nutzen. Das heißt, Grid Computing beschäftigt sich mit der koordinierten, flexiblen und sicheren gemeinsamen Nutzung von Rechenleistung, Anwendungen, Daten, Speicherplatz oder Netzwerk–Ressourcen durch Organisationen, die sowohl geographisch weit verteilt als auch hoch dynamisch sind [globus, GGFa, grid.org, FKT 01].

Föderation: Kooperation unabhängiger u. selbständiger Unternehmen

Aus beiden Szenarien lassen sich einige charakteristische Gemeinsamkeiten ableiten. Es kooperieren rechtlich unabhängige und selbstständige Unternehmen, um ein bestimmtes Ziel zu erreichen; sie bilden eine so genannte **Föderation**. Im Normalfall bringen alle Beteiligten einen bestimmten Anteil ihrer Ressourcen ein und nutzen die dadurch entstehende föderierte Infrastruktur. Die Beteiligten bilden ein virtuelles Unternehmen unter Beibehaltung der lokalen Autonomie sowohl in technischer, organisatorischer als auch in rechtlicher Hinsicht; im Grid-Kontext spricht man in diesem Zusammenhang von einer Virtuellen Organisation (vgl. Abb. 1.1). Insbesondere gibt es i.a. keine ausgezeichnete Organisation, die die Befugnis hat die Verwendung bestimmter Technologien und Standards festzulegen oder organisatorische Vorgaben zu machen. Die beteiligten Unternehmen verwenden ihre vorhandene Technologie und sind im Normalfall auch nicht bereit ihre Policies grundlegend zu ändern, um an einer Föderation teilzunehmen.

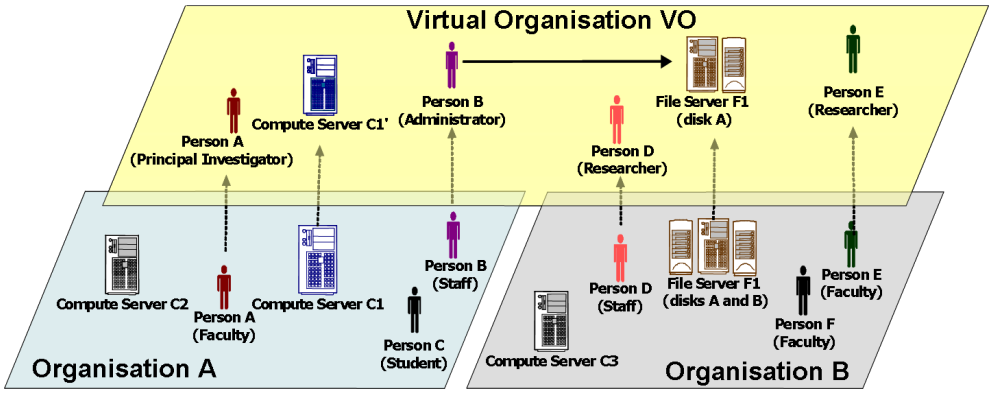

Abbildung 1.1: Virtuelle Organisation, nach [FoCh 05]

Dies führt zu technisch und administrativ sehr heterogenen Strukturen, die auch noch hoch dynamisch sein können. Es entsteht eine IT–Landschaft eines virtuellen Unternehmens, welche zwangsweise die Grenzen der einzelnen Organisation überschreitet. Eine solche interorganisationale IT–Infrastruktur, bestehend aus gleichberechtigten und autonomen Organisationseinheiten, stellt eine große Herausforderung für das effiziente und effektive Management dar.

1.1 Fragestellung

Ziel: einheitlich hohes und durchgehendes Sicherheitsniveau

Bei der Bildung interorganisationaler Infrastrukturen spielen die Sicherheit und die Sicherheitseigenschaften der Infrastruktur und die dabei erforderlichen organisationsübergreifenden Interaktionen eine zentrale Rolle. Organisationen, die Ressourcen einbringen und an einer Föderation teilnehmen sollen, werden dies nur tun, wenn sie sicher sein können, dass ihre Ressourcen und Daten ausreichend geschützt werden können und die Interaktionen zwischen den an der Föderation beteiligten Organisationen authentisch und vertrauenswürdig sind. Im Hinblick auf die entstehende Gesamtinfrastruktur muss ein einheitlich hohes und durchgehendes Sicherheitsniveau erreicht werden.

geringe zusätzliche Kosten

Die IT–Infrastruktur einer interorganisationalen Kooperation wird für das gemeinsame Projekt i.d.R. nicht neu geplant und beschafft, sondern es werden die Ressourcen der „gewachsenen“ Infrastrukturen der beteiligten Organisationen in die Föderation eingebracht und dann gemeinsam genutzt. Der Betrieb und das Management der Infrastruktur soll möglichst keine oder nur geringe zusätzlichen Aufwände und Kosten bei der einzelnen beteiligten Organisation verursachen.

konfliktäre Ziele

In der Realisierung und beim Betrieb erweisen sich diese Ziele in der Regel als konkurrierend. Aus Gründen der Kosteneffizienz und wegen der Beibehaltung

der lokalen Autonomie wird keine durchgehende Sicherheits– und Managementarchitektur für die Föderation entwickelt, sondern es werden bestehende heterogene Konzepte und Architekturen übernommen. Die (organisations–) lokalen Systeme müssen für die Föderation an den Organisationsgrenzen gekoppelt werden. Dies führt häufig zu einer redundanten Datenhaltung und es werden umfangreiche Abbildungsprozesse an den Organisationsgrenzen benötigt. Daneben ist die Einhaltung eines einheitlich hohen globalen Sicherheitsniveaus ausgesprochen schwierig, wenn die Infrastruktur durch die Koppelung heterogener lokale Sicherheitsmechanismen entsteht.

Koppelung lokaler Sicherheitsarchitekturen

Auch bei der Abwehr von Gefahren sowie der Erkennung und Verhinderung von Angriffen versucht jede Organisation die eigenen Ressourcen mit bestehenden eigenen Konzepten und Mechanismen zu schützen. Durch die Autonomie der beteiligten Organisationen liegt der Fokus der Gefahrenabwehr somit bisher üblicherweise auf lokalen Ressourcen. Eine globale Sichtweise fehlt; keine der Organisationen fühlt sich verantwortlich für die Gefahrenabwehr des interorganisationalen Gesamtsystems. Auch hierdurch entstehen wieder Redundanz und zusätzliche Kosten ohne die Vorteile und möglichen Synergieeffekte einer föderativen Gefahrenabwehr zu nutzen.

Für viele Sicherheitsanforderungen existiert eine große Anzahl an Sicherheitsdiensten und –mechanismen, welche die Sicherheitsanforderungen realisieren. Häufig wurden diese Mechanismen jedoch unter der Annahme einer intraorganisationalen Verwendung realisiert und sind deshalb nicht ohne weiteres auf interorganisationale Szenarios zu übertragen.

viele konkurrierende Sicherheitsmechanismen

Beim Aufbau einer interorganisationalen Kooperations– und Sicherheitsarchitektur stellen sich im wesentlichen drei Fragen:

1. Welche Sicherheitsanforderungen sind für das konkrete Szenario überhaupt relevant und müssen umgesetzt werden?
2. Welche Mechanismen sind grundsätzlich in der Lage diese Anforderungen umzusetzen? Da es in vielen Fällen mehrere verschiedene Mechanismen gibt, wird eine Entscheidungsunterstützung und eine Auswahlmetrik benötigt.
3. Für jeden Sicherheitsmechanismus muss entschieden werden, ob dieser lokal, zentral oder föderiert umzusetzen ist. In ähnlicher Weise stellt sich die Frage, wer die Policies für die entsprechenden Verfahren festlegen darf und muss. Werden Policy-Entscheidungen lokal, zentral oder föderiert getroffen?

Das **Ziel dieser Arbeit** ist es, ein Gesamtkonzept für ein föderiertes Sicherheitsmanagement zu entwerfen. Ein solches Gesamtkonzept ist bislang noch nicht vorgeschlagen worden. Die Fragestellung ist sowohl wissenschaftlich herausfordernd, weil es gilt ein Rahmenwerk zu entwickeln, in das etliche bisher isoliert entstandene Lösungen von Einzelproblemen integriert werden müssen, als auch von hoher Praxisrelevanz, da es gilt, den Sicherheitsverantwortlichen (Chief Security Officer, CSO) und den Sicherheitsadministra-

Leitfaden für Bewertung und Auswahl von Sicherheitsmechanismen

tor bei der Beantwortung vieler relevanter Fragen zu unterstützen und ihnen einen Leitfaden in die Hand zu geben, mit dem sie in die Lage versetzt werden Sicherheitsmechanismen zu bewerten und abhängig vom Szenario und von ihren konkreten Anforderungen auszuwählen. Dazu sind folgende Teilfragestellungen bzw. Teilprobleme zu behandeln:

- Systematische Untersuchung von Sicherheitsanforderungen
- Entwicklung einer Taxonomie und eines Kriterienkataloges zur Bewertung und eines Leitfadens zu Auswahl lokaler, zentraler oder föderierter Sicherheitsmechanismen zur Realisierung der Sicherheitsanforderungen
- Konzeptentwicklung für die effiziente und effektive Koppelung lokaler Sicherheits- und Managementarchitekturen
- Aufbau einer Sicherheitsföderation mit:
  - Realisierung eines einheitlich hohen Sicherheitsniveaus
  - Realisierung einer Ende–zu–Ende Sicherheit auf heterogenen Sicherheitsinfrastrukturen
  - Minimierung von Redundanz
- Verwendung des Kriterienkataloges zur Durchführung einer Defizit- und Schwachstellenanalyse
- Entwicklung eines Vorgehensmodells zur Ableitung einer Sicherheitsarchitektur

Eine allgemeingültige konkrete Lösung, die für alle möglichen Szenarios gilt, kann es allerdings nicht geben. Daher kann diese Arbeit nur ein allgemeines Framework, ein Vorgehensmodell und Entscheidungshilfen vorgeben, die für jedes konkrete Szenario instantiiert werden müssen.

Systematische Untersuchungen von Managementkonzepten im Umfeld von Föderationen sind neu. Sie wurden, wie bereits erwähnt, bisher nur für isolierte Fragestellungen durchgeführt. Eine integrale Betrachtung ist jedoch sowohl im Umfeld von Outsourcing, Provider–Ketten als auch Grid Computing unverzichtbar und von wachsender Bedeutung.

1.2 Vorgehensmodell

Das Vorgehensmodell und der Aufbau dieser Arbeit wird in Abbildung 1.2 dargestellt. Dabei sind die Bereiche, in denen Konzepte, Verfahren, wissenschaftliche Beiträge und Ergebnisse im Rahmen dieser Arbeit entwickelt wurden, schattiert dargestellt.

Kapitel 2: Anforderungsanalyse

In Kapitel 2 werden repräsentative Szenarien für interorganisationale Ko-

operationen und Outsourcing sowie das Grid Computing vorgestellt. Basierend auf diesen Szenarien werden die typischen Charakteristika interorganisationaler Kooperationen abgeleitet. Diese Szenarien, die Prinzipien der OSI-Sicherheitsarchitektur und Arbeiten zur Sicherheit in Grids werden verwendet, um eine Ableitung der Sicherheitsanforderungen in Föderationen durchzuführen. Zur Strukturierung und Priorisierung der Sicherheitsanforderungen werden in Abschnitt 2.3 Abhängigkeiten zwischen den Sicherheitsmechanismen untersucht und eine Diensthierarchie abgeleitet. Das Kapitel 2 schließt mit einer Zusammenfassung der Auswirkungen dieser Ergebnisse auf die Aufgabenstellung.

In Kapitel 2 wird auch gezeigt, dass unter Berücksichtigung der technischen Umsetzung von Föderationen Grids und Grid-Middleware-Technologien als Phänotyp für Föderationen verwendet werden können und die weiteren Untersuchungen in dieser Arbeit daher auf Grids und Grid-Technologien, ohne Beschränkung der Allgemeinheit, fokussiert werden können.

Kapitel 3: Kriterienkatalog

Ein Ziel dieser Arbeit ist die Bewertung und Klassifikation bestehender Sicherheitsmechanismen. Um dies realisieren zu können, wird in Kapitel 3 ein anwendungsfallspezifischer, aber szenario-unabhängiger Kriterienkatalog erstellt. Dieser Katalog besteht aus einer Methodik zur Erstellung und Anwendung (vgl. Abschnitt 3.1) und den eigentlichen Kriterien (vgl. Abschnitt 3.2).

Kapitel 4: Bewertung v. Sicherheitsmechanismen

In Abschnitt 4.1 werden die Grundlagen von Grids eingeführt. Einerseits umfasst dies die für die Fragestellungen dieser Arbeit relevante Standardisierung in diesem Umfeld. Andererseits wird mit den wichtigsten Grid Middleware-Technologien die technische Umsetzung des Grid Computing vorgestellt. Mit diesen Vorarbeiten ist das Fundament für eine Bewertung existierender Sicherheitsmechanismen gelegt. Die in Kapitel 2 neu entwickelte Diensthierarchie und die abgeleiteten Dienstabhängigkeiten werden als strukturbildendes Element für den Rest des Kapitels 4 verwendet, um hier die Sicherheitsmechanismen zu gliedern und themenfokussiert zu analysieren. Mit dem in Kapitel 3 vorgestellten, unter integralen Gesichtspunkten entwickelten Kriterienkatalog werden die Sicherheitsmechanismen im Umfeld der drei wichtigsten Middlewares Globus, UNICORE und LCG/gLite untersucht und bewertet. Am Ende von Kapitel 4 erfolgt eine Zusammenfassung der Bewertung und es wird eine Defizit- und Schwachstellenanalyse durchgeführt. Dabei zeigt sich, dass viele der in Kapitel 2 als notwendig klassifizierten Sicherheitsanforderungen nur unzureichend oder überhaupt nicht durch geeignete Sicherheitsmechanismen umgesetzt werden.

Kapitel 5: Zusätzliche Mechanismen

Ausgehend von dieser Analyse werden in Abschnitt 5 neue Sicherheitsmechanismen für das Trust Level Management und ein verteiltes Gruppenmanagement vorgestellt.

Kapitel 6: Anwendung des Rahmenwerks

Kapitel 6 beschreibt, wie ein Sicherheitsverantwortlicher, entweder lokal für seine Infrastruktur oder global für eine ganzes Grid, das Rahmenwerk anwenden kann. Das Vorgehensmodell umfasst zwei Phasen: die Analyse- und die Synthesephase.

Das Kapitel 7 fasst die Arbeit zusammen.

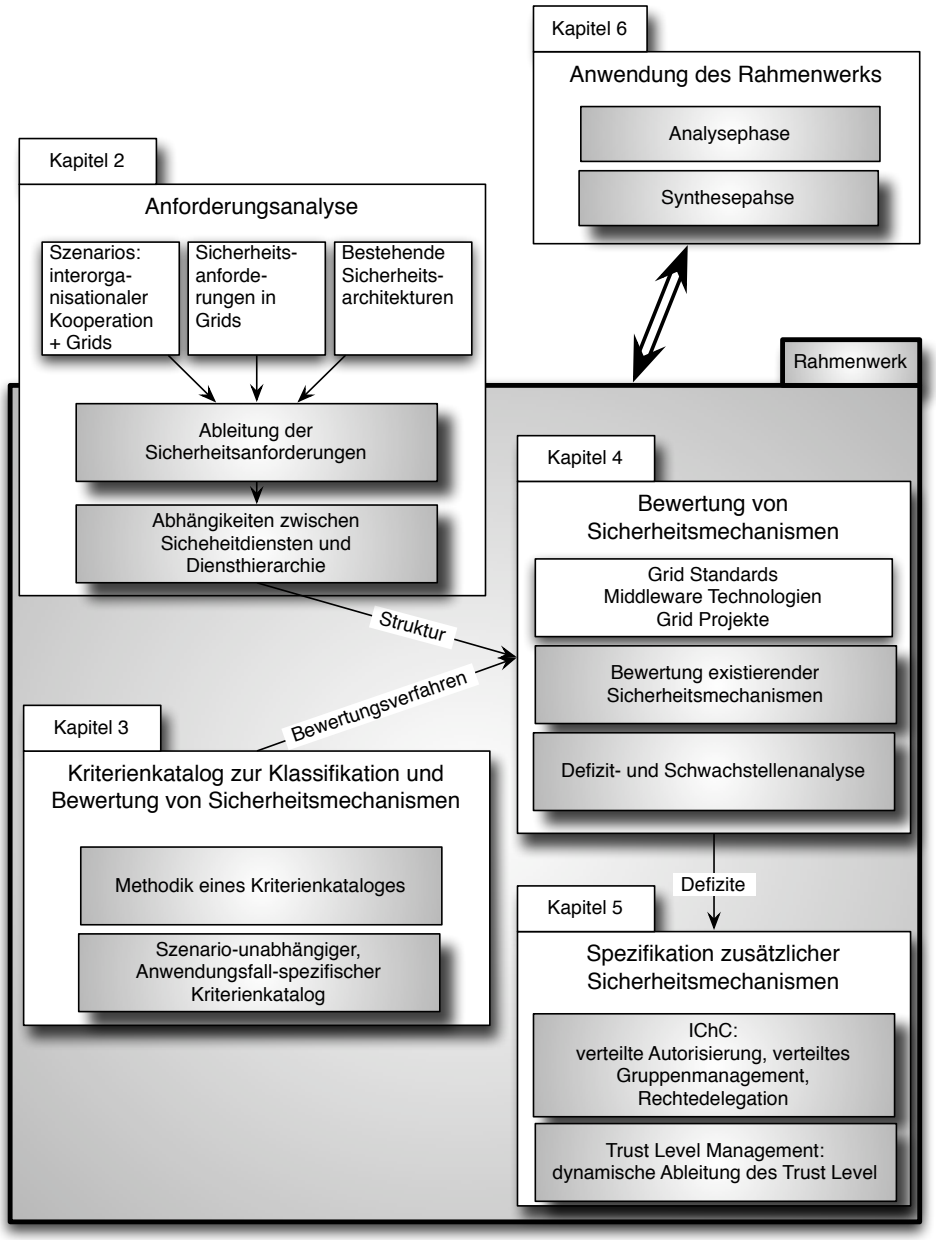

Abbildung 1.2: Vorgehensmodell und Aufbau der Arbeit

Kapitel 2

Problembeschreibung und Anforderungsanalyse

Inhaltsverzeichnis

Das Ziel dieses Kapitels ist die systematische Ableitung von Sicherheitsanforderungen, deren Abhängigkeiten untereinander sowie einer Diensthierarchie für Sicherheitsdienste. Dazu werden charakteristische Anwendungsszenarios für föderative Kooperationen vorgestellt und die repräsentativen gemeinsamen Merkmale abgeleitet.

Ausgehend von den Anforderungen, die sich aus der OSI-Sicherheitsarchitektur, der einschlägigen Literatur und Grid-spezifischer Sicherheitsanforderungen ergeben, wird eine Strukturierung und Priorisierung der Anforderungen erarbeitet.

Die Sicherheitsanforderungen werden durch Sicherheitsdienste umgesetzt. In Abschnitt 2.3 wird ein Abhängigkeitsbegriff zwischen Sicherheitsdiensten

eingeführt. Dieser wird verwendet, um eine Gruppierung in Dienstklassen vorzunehmen und eine Diensthierarchie abzuleiten. Alle Analysen erfolgen Szenario- und Technologie-unabhängig, d.h. sie gelten ganz allgemein und können für alle föderativen Kooperationen verwendet werden.

Die Diensthierarchie kann als strukturbildendes Element und Hilfsmittel für eine Szenario-spezifische Sicherheitsanalyse dienen und wird von einem Sicherheitsverantwortlichen bei der Analyse seines konkreten Anwendungsszenarios genau so eingesetzt (vgl. auch Kapitel 6).

2.1 Anwendungsszenarien für interorganisationale Kooperationen

Im folgenden Abschnitt 2.1.1 wird ein repräsentatives Beispiel einer interorganisationalen Kooperation in der Industrie vorgestellt. Abschnitt 2.1.2 führt in das Grid Computing ein, das vorwiegend in der Wissenschaft genutzt wird, um interorganisational und international zu kooperieren. Abschnitt 2.1.3 fasst die aus den vorangegangenen Abschnitten gewonnenen Charakteristika interorganisationaler Kooperation zusammen.

2.1.1 Interorganisationale Kooperation und Outsourcing in der Industrie

Die Entwicklung und Produktion von Gütern und Dienstleistungen kann heutzutage häufig nur noch in Arbeitsteilung und in Kooperation verschiedener Unternehmen erfolgen. Beispielsweise bilden bereits in der Entwicklungsphase von Fahrzeugkomponenten die Zulieferer, externe Planungsbüros und Mitarbeiter interner Planungs– und Entwicklungsabteilungen des Automobilherstellers Projektgruppen, um gemeinsam die entsprechende Komponente zu entwerfen und zu entwickeln. Betrachtet man die Nutzung von IT–Systemen in solchen Fällen, wird schnell ersichtlich, dass Mechanismen eines Sicherheitskonzeptes, das auf einer in sich abgeschlossenen Heimat– und Sicherheitsdomäne basiert, nicht ohne weiteres umsetzbar sind.

heterogene Sicherheitsdomänen und -policies

Die Projektpartner stammen aus verschiedenen Organisationen und damit auch aus unterschiedlichen Sicherheitsdomänen, in denen unterschiedliche Sicherheits-Policies gelten und unterschiedliche Sicherheitsmechanismen zum Einsatz kommen. Ein erfolgreicher Abschluss des Entwicklungsprojektes ist aber nur möglich, wenn die Partner in der Lage sind, auch IT–gestützt zusammenzuarbeiten und ihre IT-Ressourcen auch gemeinsam zu nutzen. Trotzdem will natürlich kein Unternehmen auf die lokale Autonomie sowie die eigene Entscheidungsfreiheit verzichten. Auch sollen lokale Sicherheits–Policies nicht unterlaufen werden können. In der Praxis werden

deshalb häufig in jedem der beteiligten Unternehmen spezielle (Sicherheits–) Domänen für das Projekt gebildet, diesen werden dezidierte Ressourcen nur für das Projekt zugewiesen und den Partnern aus den anderen Unternehmen eigene lokale Kennungen eingerichtet. In der praktischen Umsetzung entsteht durch dieses Vorgehen erheblicher Mehraufwand, es kommt zu Redundanz, Inkonsistenzen und u.U. zu einer Verminderung des (globalen) Sicherheitsniveaus. Dies lässt sich beispielhaft am Identitätsmanagement in solchen interorganisationalen Firmenkooperationen zeigen. Das **Identitätsmanagement** befasst sich mit der zweifelsfreien Identifikation einer Entität (zum Beispiel eines Mitarbeiters des Unternehmens) und deren Repräsentation im IT–System (z.B. durch eine Kennung).

Problem: Redundanz, Inkonsistenzen, zus. Aufwand

Bei der BMW Group wurde bspw. für jeden Projektmitarbeiter aus einem anderen Unternehmen eine so genannte Extern–Kennung von BMW angelegt und auch gepflegt. Die anderen Unternehmen machen entsprechendes in ihrer Heimatdomäne für die Mitarbeiter von BMW und die anderen externen Projektpartner. Dies führt zu einer erheblichen Redundanz, da bei n Projektpartnern die Informationen über einen Mitarbeiter n–fach erfasst, verarbeitet, gespeichert und gepflegt werden. Bei der BMW Group bestanden im Jahr 2004 Beziehungen zu rund 60.000 Unternehmen, die den Status Kooperationspartner besaßen [Metz 04]. Unter der Annahme, dass in jedem Unternehmen m Mitarbeiter für ein Projekt abgestellt werden und es p Projekte gibt, müssen bei n Unternehmen in Summe $O(n \cdot m \cdot p)$ Kennungen erfasst, verarbeitet, gespeichert und gepflegt werden. Nachdem die Daten mehrfach und an verschiedenen Stellen erfasst werden, kommt es zwangsweise zu Inkonsistenzen in den Datenbeständen mit allen negativen Folgen für das Sicherheitsniveau. Für ein Unternehmen ist es häufig schwierig nachzuvollziehen, bei welchem Partner–Unternehmen ein Mitarbeiter überhaupt Extern–Kennungen besitzt. Folglich ist es auch nicht immer gewährleistet, dass das Partnerunternehmen darüber informiert wird, falls ein Mitarbeiter das Unternehmen verlässt. In diesem Fall hat diese Person bei Partnerunternehmen weiterhin Zugang zu Ressourcen des Projektes.

Für den einzelnen Mitarbeiter bedeutet dies, dass er für jedes beteiligte Unternehmen eine eigene Zugangskennung erhält und sich ein eigenes Passwort dafür merken sollte. Da dies durchaus einen nicht unerheblichen Aufwand für den Mitarbeiter bedeutet, wird der häufig auf allen Systemen dasselbe Passwort verwenden. Der Mitarbeiter muss sich auf den Systemen jedes Unternehmens gesondert anmelden. Selbst wenn es innerhalb eines Unternehmens eine Single–Sign–On Lösung geben sollte, lässt sich diese nicht auf die Föderation übertragen.

organisationslokale Sichtweise bestimmend

Auch im Hinblick auf die Rechtevergabe (Autorisierung) und deren Umsetzung durch die Zugriffskontrolle ergeben sich neue Probleme. Die technische Zugriffskontrolle erfolgt bei den konkreten Ressourcen und damit organisationslokal. Die Unternehmen wollen die alleinige und volle Entscheidungshoheit über ihre Ressourcen behalten. Dementsprechend müssen für jeden Nutzer, der diese Ressourcen nutzt, organisationslokal Rechte vergeben werden.

Die Heimatdomäne wird über diese Rechte häufig weder informiert noch hat sie auf diese Rechtevergabe Einflussmöglichkeiten. Dies führt dazu, dass im Heimatunternehmen die Menge der Rechte, die ein Mitarbeiter besitzt, nicht mehr in vollem Umfang bekannt ist. Abhängig von den Rechten, die ein Projektmitarbeiter bei einem Partnerunternehmen besitzt, können dem Heimatunternehmen Folgelasten und Kosten entstehen. Das Heimatunternehmen hat aber keine direkten Möglichkeiten auf die Autorisierung beim Partner Einfluss zu nehmen oder selbst die Rechte des eigenen Mitarbeiters beim Partner zu beschränken.

Verringerung des Sicherheitsniveaus

Global betrachtet führen alle diese Probleme zu einer Verringerung des Sicherheitsniveaus.

2.1.2 Grids

Das Grid Computing hat sich ursprünglich aus Bestrebungen heraus entwickelt, die Ressourcen nationaler Supercomputing-Zentren zu verbinden und einfach nutzen zu können. Die „Rechenpower“ sollte so einfach zugreifbar sein wie der elektrische Strom aus dem Stromnetz (engl. Power Grid). Auf diese Aspekte bezieht sich auch die frühe Definition des Grid Computing von Ian Foster und Carl Kesslemann [FoKe 99]:

Definition Grid

„*A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.*“

In dieser Definition steht der Aspekt einer technischen Infrastruktur im Vordergrund. Diese sollte eine zuverlässige, konsistente und kostengünstige Nutzung von Höchstleistungsrechnern quasi von „überall“ ermöglichen. Obwohl in dieser Definition nicht explizit genannt, ist die Überwindung der Domänengrenzen zwischen technisch und rechtlich autonomen Organisationen bereits ein wichtiges Thema. In den frühen Tagen des Grid waren jedoch die technischen Fragen einer Infrastruktur von entscheidender Bedeutung. Im Laufe der Zeit hat sich dieser sehr technische Fokus der Koppelung von Höchstleistungsrechenzentren doch deutlich verallgemeinert. Heute steht nicht mehr nur die Nutzung von Supercomputern im Vordergrund, sondern es wird ganz allgemein von Ressourcen gesprochen. Unter diesen Ressourcenbegriff fallen Rechner, Software, Daten, Speicher- und Plattenplatz, aber auch andere Ressourcen wie wissenschaftliche Großgeräte (z. B. Beschleunigerring am CERN, astronomische Teleskope, Satelliten) u. ä. Die Nutzung dieser Ressourcen ist weiter gefasst. Es soll nicht nur der Austausch von Daten und Jobs, sondern der direkte Zugriff auf die Ressource ermöglicht werden. Bei der Nutzung von Grids rückt das kollektive und kollaborative Lösen von Problemen und das „Teilen“ von Ressourcen in den Vordergrund. Eine gemeinsame Nutzung verteilter Ressourcen, die von verschiedenen autonomen Organisationen zur Verfügung gestellt werden, erfordert natürlich große Kontroll- und Einflussmöglichkeiten für den Ressourcen-Anbieter und Absprachen zwischen Nutzern und Anbietern von Ressourcen. Die entscheiden-

verallgemeinerter Ressourcenbegriff

kollektives u. kollaboratives Problemlösen

de Neuerung in der Definition von Grid ist jedoch die Virtualisierung und der Begriff der „Virtuellen Organisation (VO)“. Die spätere Arbeit von Foster, Kesselmann und Tuecke zur Definition des Begriffes Grid trägt die Virtuelle Organisation bereits im Titel [FKT 01].

Zusammenfassend lässt sich ein Grid charakterisieren als eine zielorientierte, kontrollierte sowie koordinierte Ressourcen-Teilung und Benutzung in dynamischen, skalierbaren und verteilten virtuellen Organisationen.

Virtualisierung und Transparenz im Grid

Eine Grid-Infrastruktur ermöglicht einem Nutzer den Zugriff auf Ressourcen, die i.d.R. auch außerhalb der lokalen administrativen Domäne liegen, d. h. die Kollaboration erfolgt Domänen-übergreifend. Unter einer **Domäne** wird eine technisch (und häufig auch rechtlich) autonome Organisationseinheit verstanden, die einer einheitlichen operationalen Verantwortung unterliegt und in der es ein einheitliches Betriebskonzept mit entsprechenden Policies gibt. Die Beteiligten, die Domänen-übergreifend ein gemeinsames Ziel verfolgen oder gemeinsame Interessen haben, teilen sich ihre Ressourcen. In [FKT 01] wird dieses „Teilen“ näher spezifiziert.

Domäne = autonome Organisations-einheit

„*The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a virtual organization.*“

Außerdem wird hier der für das Grid zentrale Begriff der Virtuellen Organisation (VO) eingeführt. Eine **Virtuelle Organisation** wird bestimmt durch eine Menge von Individuen, Ressourcen und/oder Institutionen, die sich durch Regeln (Policies) definiert, die das Teilen der Ressourcen festlegen. Eine virtuelle Organisation umfasst also verschiedene reale Organisationen und damit verschiedene autonome Domänen.

Definition Virtuelle Organisation

Aber nicht nur im Hinblick auf die organisatorische Gliederung, sondern auch bei den Ressourcen selbst, wird das Prinzip der Virtualisierung angewendet. Ein Grid-Benutzer soll und muss im Idealfall überhaupt nicht mehr wissen, wo sich eine Ressource (z. B. der Speicherplatz für Daten und Replikate dieser Daten) befindet. Er gibt seinen Auftrag an das Grid und das Grid sorgt dafür, dass dieser Auftrag ausgeführt wird. Für den Nutzer ist die Diensterbringung durch das Grid transparent (im Sinne von nicht sichtbar). Diese Transparenz kann Ort, Zeit und sogar die verwendete Technologie umfassen. Der Benutzer muss sich nicht mehr darum kümmern, in welcher Domäne (Ort), zu welcher Zeit und mit welcher konkreten technischen Realisierung der Ressource sein Auftrag ausgeführt wird. Zusammenfassend lässt sich feststellen, dass ein Grid durch die folgenden Punkte charakterisiert ist:

transparente Dienstnutzung

- Rechtlich und technisch autonome Organisationen bilden die Grid-Infrastruktur
- (i. Allg.) keine zentrale Kontrolle der Ressourcen
- Verschattung der Domänengrenzen bei gleichzeitig großer Kontroll- und Einflussmöglichkeit für Ressourcen–Anbieter
- Gemeinsame Nutzung von Ressourcen (Resource Sharing)
- Kollektive und kollaborative Problemlösung
- Virtualisierung und Transparenz
- Bildung skalierbarer virtueller Organisation (VO)

Arten von Grids

Grids lassen sich in Abhängigkeit ihres Verwendungszweckes klassifizieren. Diese Vorgehensweise wurde bspw. in der Britischen „e-Science Gap Analysis“ [FoWa 03b, FoWa 03a] verfolgt. In dieser Studie wurden im April 2003 in Großbritannien Wissenschaftler zum Zustand und Anwendungsspektrum von Grid-Technologien befragt. Aus dieser Befragung entstand eine Defizit-Analyse sowie eine Klassifikation von Grids nach ihrem Verwendungszweck.

Eine gängigere und allgemein breiter akzeptierte Klassifikation von Grids richtet sich nach der Art der vernetzten Objekte. Hier werden die beiden Klassen Ressource-Grids und Service Grids unterschieden. Bei den **Ressource-Grids** steht die gemeinsame Nutzung von Ressourcen aller Art im Vordergrund. Das Grid virtualisiert und vereinheitlicht dabei den Zugang zu heterogenen Ressourcen. Die Klasse der Ressource Grids wird weiter unterteilt in Computing Grids, Data-Grids sowie Access Grids. Beim **Computing-Grid** steht die Verfügbarkeit von Rechenleistung im Vordergrund. Das **Daten-Grid** (manchmal auch als File-Grid bezeichnet) soll erhebliche Mengen von Daten zuverlässig und ausfallsicher speichern und einfach wieder zugänglich machen. Das Daten-Grid stellt auch Methoden zur effizienten Suche oder ggf. zur lokalen Verdichtung oder (Vor-) Filterung der Daten zur Verfügung. Insbesondere in wissenschaftlichen Großexperimenten lässt sich nicht immer zwischen Computing- und Daten-Grid trennen. In diesen Fällen kommt eine Mischform aus Computing- und Daten-Grid zum Einsatz. Hier fallen sehr große Datenmengen an, die auch innerhalb des Grid weiterverarbeitet oder analysiert werden. Beim **Access-Grid** handelt es sich um eine Infrastruktur, die spezifische Werkzeuge zur Kollaboration beinhaltet, wie z. B. Video-Konferenz-Systeme mit Whiteboard und gemeinsam nutzbaren Applikationen.

Ressource-Grid
Computing-Grid
Daten-Grid
Access-Grid

Den **Service-Grids** liegt eine Service Oriented Architecture (SOA), im Deutschen auch als dienstorientierte Architektur bezeichnet, zugrunde. Ein Service in diesem Kontext ist als eine Funktionalität definiert, die über eine standardisierte Schnittstelle in Anspruch genommen werden kann. Die technische

Service-Grid

Abbildung auf eine konkrete Ressource wird dabei für den Nutzer verschattet. Die Dienste eines Service Grid werden unterliegenden Ressourcen eines Ressource Grids zugeordnet.

2.1.3 Charakteristika interorganisationaler Kooperation

Aus den beiden sehr unterschiedlichen Szenarios, dem Grid Computing sowie Kooperation und Outsourcing, lassen sich gemeinsame Merkmale ableiten, welche die interorganisationale Kooperation charakterisieren.

Diese ist gekennzeichnet durch eine Entwicklung weg von privat im eigenen Unternehmen genutzten Diensten hin zu gemeinsam genutzten, geteilten und sogar gemeinsam erbrachten Diensten. Dabei ist eine möglichst hohe Interoperabilität entscheidend für den Erfolg einer Kooperation. Diese Interoperabilität muss über Organisationsgrenzen und heterogene Ressourcen hinweg möglich sein. Besonders wichtig ist dabei die Berücksichtigung verschiedenster Policies. Es bedarf der Möglichkeit Policies über Organisationsgrenzen hinweg auszutauschen (**Cross Organisational Policy Exchange**) und diese auf heterogenen Umgebungen in operationale Policies zu transformieren.

Policy Austausch über Domänengrenzen

In interorganisationalen Umgebungen gibt es verschiedene Policies aus unterschiedlichen Verantwortungsbereichen, z.B. Policies der Heimat- (Local Policy) und der Gast-Domäne (Target-Policy), Nutzer-Policies, Policies für Ressourcen, von Anbietern usw. Die gesamte Policy–Menge innerhalb der interorganisationalen Kooperation (VO-Policy) beinhaltet zwangsweise Konflikte und Widersprüche, die über geeignete Mechanismen aufgelöst werden müssen.

Das Wesen der Diensterbringung hat sich weg von statisch konfigurierten Diensten hin zur dynamischen Diensterbringung gewandelt. Auch bei den Ressourcen zeichnet sich eine zunehmende Virtualisierung ab. Für die Nutzer in einer Kooperation ist die technische Ausprägung einer Ressource und deren Lokalisation von untergeordneter Bedeutung und im Extremfall völlig unerheblich. Wichtig ist die aufgabenzentrierte Problemlösung mit Hilfe von Diensten und geeigneten Ressourcen. Für den Nutzer ist es wichtig, dass er Ressourcen in ausreichendem Umfang und in adäquater Ausprägung zur Verfügung hat; man spricht hier von Co-Allokation multipler Ressourcen. Wo, von wem und auf welchen realen Ressourcen diese virtualisierten Ressourcen erbracht werden, sollte für den Nutzer transparent sein.

dynamische Diensterbringung

Virtualisierung

Aus Sicht des Ressourcen-Betreibers ist das Ressourcen–Management jedoch von eminenter Wichtigkeit. Ein Ressourcen-Betreiber möchte in der Lage sein, den Zugriff auf seine Ressourcen zu vergeben und ggf. auch zu beschränken. Er möchte im Regelfall die volle Kontrolle über seine realen Ressourcen behalten. Ein Grundsatz, der sich in vielen Fällen durch die unabhängige rechtliche Stellung der beteiligten Partner ergibt, ist die Forderung nach vollkommener lokaler Autonomie, d.h. die totale Entscheidungsbefug-

lokale Autonomie

nis und –freiheit über eigene Ressourcen. Diese lokale Autonomie wird nur durch eine explizite Delegation von Rechten an „fremde“ Kooperationspartner aufgeweicht.

VO-Management

Auch organisationstheoretisch spiegelt sich in der Bildung skalierbarer virtueller Organisationen die interorganisationale Kooperation wider. Dabei ist es wichtig, das der VO zugrunde liegende gemeinsame Regelwerk geeignet festzulegen und die Umsetzung technisch zu unterstützen. Die Etablierung, der Betrieb, die Administration und die Auflösung einer VO muss durch geeignete Managementprozesse und –Verfahren unterstützt werden (eine Architektur für das VO-Management wurde im Rahmen von [Schi 07] entwickelt).

Trust Management

Die Basis für jede Art der Kooperation sind Vertrauensbeziehungen zwischen den beteiligten Partnern. Ist die Anzahl der Partner sehr klein, sind die Partner alle untereinander bekannt und ist die Art der Kooperation sehr eng, können diese Vertrauensbeziehungen implizit gepflegt werden. Je größer Art und Anzahl der Partner sind, umso schwieriger und aus sicherheitstechnischen Erwägungen gefährlicher wird dieser Ansatz. Grundlage jeder VO–Bildung muss daher ein formalisiertes und explizites Trust Management sein.

Jede der an einer VO beteiligten Domänen betreibt eigene Sicherheitseinrichtungen wie z.B. Firewalls, Intrusion-Detection- und Monitoring-Systeme, etc., die an der Domänengrenze einen sicheren internen von einem unsicheren externen Bereich separieren. Diese Unterscheidung gilt grundsätzlich auch für alle Dienstanforderungen und jede Art der Kommunikation der Kooperationspartner. Um eine effektive und effiziente Kooperation möglich zu machen, sollen die Sicherheitseinrichtungen, bei einer bestimmungsgemäßen Verwendung der gemeinsamen Infrastruktur, so wenig wie möglich in Erscheinung treten und die Mitglieder der VO so wenig wie möglich behindern. Hier manifestiert sich ein Konflikt zwischen einem angemessenen Sicherheitsniveau und der Nutzbarkeit der gemeinsamen Infrastruktur.

In der folgenden Liste, werden die Charakteristika interorganisationaler Kooperation nochmal kurz zusammengefasst:

Charakteristika interorganisationaler Kooperation

- Entwicklung von privaten Diensten hin zu gemeinsam genutzten und geteilten Diensten
- Interoperabilität über Organisationsgrenzen, operationale Policies und heterogene Ressourcen hinweg
- umfassendes Ressource-Management für Ressourcen-Anbieter
- Austausch von Policies über Organisationsgrenzen hinweg
- Policy Mapping (User-, Site-local-, Target-site und VO–Policies)
- statische versus dynamischer Diensterbringung
- Übergang von realen Ressourcen hin zu virtualisierten Ressourcen
- Bildung skalierbarer virtueller Organisationen (VOs)

- Vollkommene lokale Autonomie (Entscheidungsbefugnis über eigene Ressourcen)
- Delegation von Rechten
- Trust Level Management
- Einfache Funktion über Schutzsysteme (z.B. Firewalls) hinweg; ohne Verringerung des Sicherheitsniveaus

2.1.4 Grids als Phänotyp für Föderationen

Die Bildung und Aufrechterhaltung von Föderationen ist bei der Etablierung und dem Betrieb von Grid-Infrastrukturen systemimmanent. Der fundamentale Zweck und die Motivation zum Aufbau einer Grid-Infrastruktur ist die interorganisationale Kooperation; die gemeinsame, effiziente und zielgerichtete Nutzung von verteilten Ressourcen. Damit stellen Grids eine phänotypische Realisierung von Föderationen dar. Eine föderative Zusammenarbeit ist naturgemäß die Basis für ein Grid.

Grid-Middleware als technische Plattform

Zur technischen Umsetzung von Grids und damit von Föderationen kann auf Grid Middleware-Technologien zurückgegriffen werden. Eine Grid Middleware ist die technische Plattform zur Realisierung einer Föderation (vgl. Abschnitt 4.1). Aus diesen Gründen werden im folgenden Grids und entsprechende Middleware-Technologien als charakteristische Vertreter für Föderationen betrachtet, obwohl gegenwärtig Grids außerhalb des Forschungsbereichs noch keine breite Verbreitung gefunden haben.

Obwohl in der Wirtschaft und in den meisten Industrieunternehmen noch keine Grid-Techniken eingesetzt werden, lassen sich Fragen und Problemstellungen der interorganisationalen Kooperation (wie z.B. das BMW-Szenario aus Abschnitt 2.1.1) auf vergleichbare Fragestellungen in Grids abbilden.

2.2 Anforderungsanalyse zur Ableitung von Sicherheitsanforderungen

Vorgehensweise bei der Risikoanalyse

Die Sicherheitsanforderungen, die ein konkretes System erfüllen soll, werden im Rahmen des Sicherheits-Engineering mit Hilfe einer Bedrohungs- und Risikoanalyse abgeleitet [Bund 04, Bund 05b, Bund 05c, Bund 05a, Bund 05d, Gree 06, Jaqu 07]. Zuerst werden in einer **Bestandsaufnahme** alle Schutzobjekte wie z.B. Rechensysteme, Dienste, Datenbestände usw. ermittelt. In der **Bedrohungsanalyse** wird für jedes Objekt aus der Bestandsaufnahme untersucht und dokumentiert, welchen potentiellen Bedrohungen und Gefahren das Objekt ausgesetzt ist. Daran anschließend werden die Eintrittswahrscheinlichkeiten, genauer die **Erwartungswerte** für den Eintritt eines

Schadens und die entsprechenden **Schadenshöhen** ermittelt, um das quantitative Risiko berechnen und die **Risiken priorisieren** zu können. Die schwierigsten Schritte dabei sind die Ermittlung der Eintrittswahrscheinlichkeiten und der Schadenshöhen. Aus den priorisierten Risiken werden dann Sicherheitsanforderungen abgeleitet.

Diese Arbeit entwickelt ein Rahmenwerk für föderiertes Sicherheitsmanagement, das einen generischen Leitfaden — unabhängig vom konkreten Szenario — liefern soll. Aus diesem Grund ist es erforderlich allgemeinere Sicherheitsanforderungen zu bestimmen, die dann nochmal für das jeweilige konkrete Anwendungsszenario im Rahmen einer Risikoanalyse verfeinert werden sollten.

Als Grundlage und als weitere Quelle zur Ermittlung der Sicherheitsanforderungen werden einerseits die OSI–Sicherheitsarchitektur, andererseits einschlägige Vorarbeiten aus der Grid-Literatur verwendet.

2.2.1 OSI–Sicherheitsarchitektur

Die OSI–Sicherheitsarchitektur schützt die Kommunikation zwischen heterogenen Rechensystemen, nicht jedoch die Rechensysteme selbst. Obwohl die OSI–Sicherheitsarchitektur spezifiziert wurde, um Kommunikationssysteme und nicht verteilte Systeme zu sichern, sind die angegebenen Konzepte so allgemeingültig, dass sie auch auf das dieser Arbeit zugrunde liegende Szenario angewendet werden können.

Die OSI–Sicherheitsarchitektur [X.800, X.810, ISO 10181-1] ist sowohl als X– als auch als ISO–Standard veröffentlicht. Alle X.81y Standards werden bei der ISO unter ISO 10181-x geführt. Die OSI–Sicherheitsarchitektur erweitert das ISO–OSI Referenzmodell (OSI–RM) [ISO 7498] und beschreibt Sicherheitsdienste und –mechanismen. Dienste, die in der Lage sind, eine Sicherheits-Policy [Gree 06] durchzusetzen bzw. zu erfüllen, werden als **Sicherheitsdienste** bezeichnet. Ein Sicherheitsdienst wird durch **Sicherheitsmechanismen** realisiert. Aus einfachen Basisdiensten lassen sich dabei auch komplexere Sicherheitsdienste kombinieren. Ein Sicherheitsdienst der OSI–Sicherheitsarchitektur ist immer an eine bestimmte Schicht des OSI–RM gebunden.

Sicherheitsdienste werden durch Sicherheitsmechanismen realisiert

Folgende Sicherheitsdienste werden gefordert:

Authentisierung

- Die **Authentisierung (Authentication)** liefert die Gewissheit über die Identität einer Entität. Die Authentisierung ist nur im Kontext einer Relation zwischen einem so genannten **Principal**, d.h. der Entität, die authentisiert werden soll, und einem **Verifier**, der Entität, die die Authentisierung durchführt, sinnvoll [X.811].

Zugriffskontrolle

- Das primäre Ziel der **Zugriffskontrolle (Access Control)** ist die Verhinderung von nicht autorisierten Operationen auf Rechen– oder Kommunikationssystemen [X.812].

Verbindlichkeit

- Der Sicherheitsdienst, der **Verbindlichkeit (Non Repudiation)** gewährleistet, umfasst die Erzeugung, die Verifikation und die Speicherung von Belegen sowie die spätere Wiederherstellung und wiederholte Verifikation dieser Belege mit dem Ziel, Kontroversen über das Auftreten von Ereignissen oder Aktionen aufzulösen [X.812]. Ein stattgefundenes Ereignis oder eine durchgeführte Aktion kann später nicht geleugnet werden. Beispiele für solche Aktionen sind das Verschicken von Nachrichten, der Aufruf einer entfernten oder lokalen Operation, u.ä.

Vertraulichkeit

- **Vertraulichkeit (Confidentiality)** bezeichnet die Eigenschaft von Information, nicht autorisierten Personen, Entitäten oder Prozessen nicht zugänglich zu sein und von diesen auch nicht enthüllt werden zu können [X.814].

Integrität

- Der Dienst, der die **Integrität (Integrity)** von Daten und Attributen sichert, kann unautorisierte Veränderung, Löschung, Einfügung, Erzeugung und das Wiedereinspielen erkennen oder verhindern [X.815].

Sicherheitsaudit und –alarme

- Daneben wird auch ein Dienst für **Sicherheitsaudits (Security Audit)** und **–alarme (Security Alarms)** spezifiziert. Ein Sicherheitsaudit ist eine unabhängige Revision und Prüfung von Systemdatensätzen und Aktivitäten. Der Sicherheitsauditdienst unterstützt einen **Auditverantwortlichen (Audit Authority)** durch die Möglichkeit, Ereignisse und Aktionen zu spezifizieren, auszuwählen und zu verwalten, die in einem **Sicherheitsaudit–Pfad (Security Audit Trail)** aufgezeichnet werden.
  Ein Sicherheitsalarm ist eine Warnung an eine Person oder einen Prozess, die anzeigt, dass eine Situation eingetreten ist, die eine rechtzeitige Aktion erfordert. Der Zweck eines solchen Alarms umfasst u.a. die Meldung von Sicherheitsverletzungen, das Überschreiten von Schwellwerten oder andere sicherheitsrelevante Ereignisse [X.816].

Sicherheitsdienste bilden Hierarchie

Im Rahmen der Arbeiten zu [Reis 01] wurde gezeigt, dass die Sicherheitsdienste nach OSI eine Hierarchie (vgl. Abbildung 2.1) bilden.

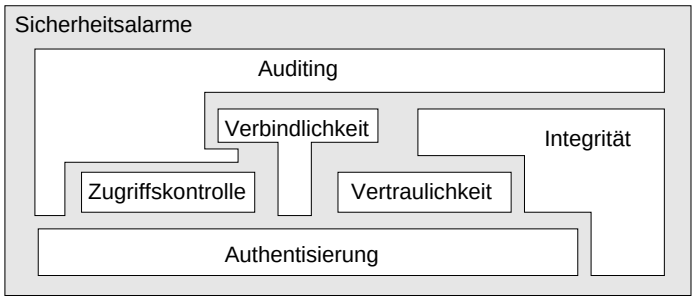

Abbildung 2.1: Hierarchie der OSI–Sicherheitsdienste

Die Authentisierung dient als Basisdienst für die Zugriffskontrolle, die Vertraulichkeit sowie die Verbindlichkeit. Ohne eine sichere und zweifelsfreie

Authentisierung von Kommunikationspartnern kann keine Zugriffskontrolle erfolgen. Rechte werden immer an Entitäten vergeben, die ihre Identität über den Authentisierungsdienst belegen müssen. Analog muss auch für eine vertrauliche Kommunikation die Identität des Kommunikationspartners zweifelsfrei feststehen.
Verbindlichkeit lässt sich auf unterschiedliche Arten realisieren. Auf jeden Fall muss aber die Identität des Akteurs bestimmt werden. Diese kann dann mit der durchgeführten Aktion in eine (manipulations–) sichere „Log–Datei“ geschrieben werden. In diesem Fall verwendet der Verbindlichkeitsdienst den Authentisierungs– und den Auditingdienst als Basisdienste. Falls die Verbindlichkeit der Aktion durch eine digitale Signatur des Akteurs sichergestellt wird, dienen der Vertraulichkeits– und der Authentisierungsdienst als Basis. Entsprechend gibt es auch verschiedene Möglichkeiten, einen Integritätsdienst zu realisieren, der entweder völlig ohne Basisdienste auskommt oder sich auf den Authentisierungsdienst oder den Vertraulichkeits– bzw. den Verbindlichkeitsdienst abstützt.

Der Dienst, der Sicherheitsalarme erzeugt, ist eine Querschnittsfunktionalität, die bei Sicherheitsverletzungen in allen anderen Diensten zur Verfügung stehen muss. Gleiches gilt für den Auditing–Dienst, der kritische und sicherheitsrelevante Aktionen in einem Log speichert. Der Auditing–Dienst kann aber selbst wieder Basisdienst sein, z.B. für den Verbindlichkeitsdienst.

2.2.2 Sicherheitsanforderungen in Grids

Sicherheitsanforderungen aus Grids lassen sich aus verschiedenen Quellen ableiten. Im folgenden werden diese Quellen kurz dargestellt, um dann die dort angeführten Sicherheitsanforderungen zusammenzufassen.

e-Science Gap Analysis

In Großbritannien wurden Anfang 2003 rund 80 Wissenschaftler zum Stand und den Defiziten von Grid-Infrastrukturen im Hinblick auf deren Anwendbarkeit im e-Science Programm befragt. Daraus resultierend entstand die *e-Science Gap Analysis* [FoWa 03b, FoWa 03a, AzMa 02, BAK+ 02]. Sowohl diese Studie als auch einschlägige Vorarbeiten [Chiv 03] beschäftigen sich auch mit Unzulänglichkeiten der Sicherheitsinfrastruktur.

EGEE

Auch im DataGRID Projekt [DataGRID] und dem Nachfolgeprojekt EGEE (vgl. Abschnitt 4.1.11) [EGEEa, EGEEc, EGEE-Tec] wurden Sicherheitsanforderungen für Grid-Infrastrukturen untersucht [Data 02, Data 01] und die Anforderungen der Grid-Nutzer ermittelt [EGEE 04c].

GGF OGF

Im Rahmen des Global Grid Forums (GGF), bzw. der Nachfolgeorganisation Open Grid Forum (OGF) (vgl. Abschnitt 4.1.1), beschäftigen sich verschiedene Working Groups mit der Spezifikation einer Open Grid Service Architecture (OGSA) und der Open Grid Service Infrastructure (OGSI). Innerhalb von OGSA befasst sich die Security Working Group (OGSA-SEC-WG) mit Sicherheitsanforderungen [MCLS 03, MSWS 03] und möglichen Sicherheitsarchitekturen im Kontext von OGSA [SWT+ 02, IBMi 02]. Ausgehend von typischen Grid Use Cases werden in [HuTh 01] Sicherheits–

Implikationen abgeleitet. In [NJD+ 02] werden Herausforderungen an eine OGSA-Sicherheitsarchitektur identifiziert und Sicherheitsanforderungen abgeleitet

Die Autoren gehen von drei Herausforderungen (Challenges) aus, die absolut kritisch für die Sicherheit von Grids sind (vgl. Abbildung 2.2).

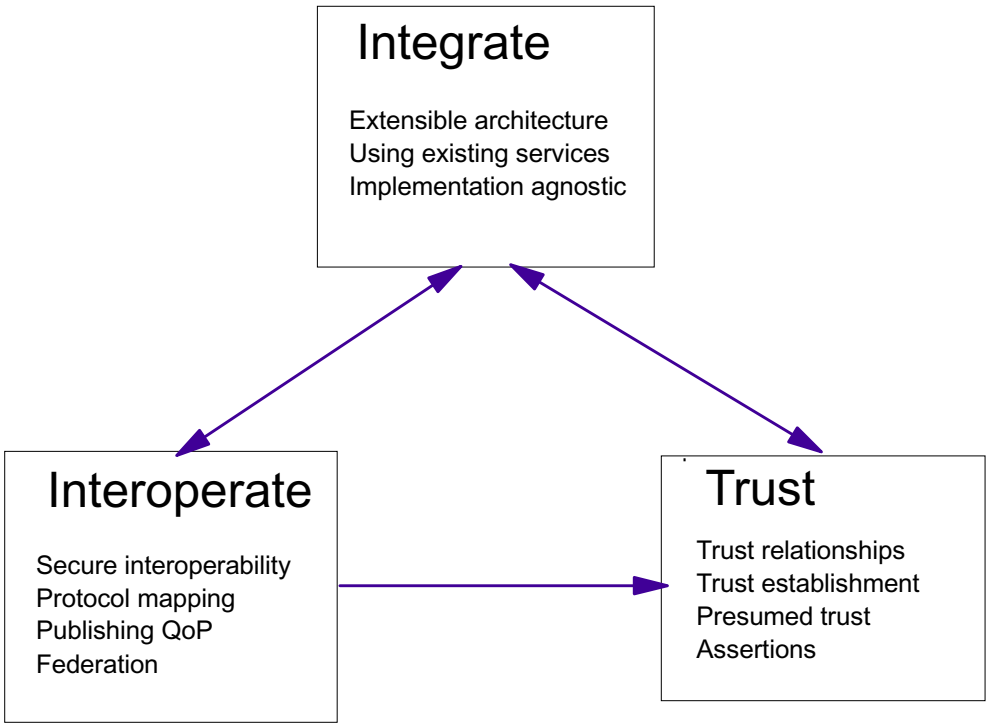

Abbildung 2.2: Herausforderungen für die Sicherheit in Grid-Umgebungen [NJD+ 02]

1. **Integration Challenge:** Da bestehende Sicherheitsinfrastrukturen nicht einfach ausgetauscht oder ersetzt werden können, muss die Sicherheitsarchitektur in der Lage sein bestehende und neue Sicherheitsmechanismen integrieren können. — Integration Challenge

2. **Interoperability Challenge:** Um eine interorganisationale Architektur realisieren zu können, müssen die Sicherheitsdienste auf Ebene der Protokolle, der Policies und des Identitätsmanagements interoperabel sein. Jede beteiligte Domäne muss in der Lage sein eigene Policies zu spezifizieren; diese müssen ausgetauscht und von den Beteiligten interpretierbar sein. Interoperabilität im Identity Management meint, dass Mechanismen benötigt werden, um einen Benutzer einer Domäne in einer Anderen identifizieren und authentisieren zu können. In der Arbeit von [NJD+ 02] wird bereits erkannt, dass es zwar schön wäre, wenn eine Identität über alle Domänen hinweg einheitlich definiert werden könnte, dies aber absolut unrealistisch ist. — Interoperability Challenge

3. **Trust Relationship Challenge:** Der Aufbau und das Validieren von Ver- — Trust Relationship Challenge

trauensbeziehungen zwischen am Grid beteiligten Domänen ist für sehr viele Dienste unumgänglich. Die Tatsache, dass das Grid die dynamische Erzeugung und das Management von Benutzer–kontrollierten transienten Diensten ermöglicht, erschwert das Trust Management zusätzlich. Jeder Benutzer kann solche transienten Dienste erzeugen und dazu eigenen Quell–Code zur Ausführung bringen.

Zur Erzeugung transienter Dienste ist es erforderlich, diese abhängig von der Identität des Nutzers zu autorisieren. Sowohl der ausführende Nutzer als auch die Gast–Domäne spezifizieren Sicherheitspolicies, die in Einklang zu bringen sind (**Policy Mapping**). Der Benutzer, der beispielsweise spezifiziert, wer seinen Dienst nutzen darf, muss sich darauf verlassen können, dass diese Policy von der Gast–Domäne auch durchgesetzt wird.

Policy Mapping

Die transienten Dienste müssen in der Gast–Domäne stellvertretend für den Benutzer Aktionen initiieren können und selber weitere transiente Dienste in anderen Domänen starten können. Das Recht, stellvertretend für den Benutzer zu agieren, sowie entsprechende Zugriffsrechte auf Ressourcen der Gast–Domänen und die entsprechenden Nutzungs–Policies müssen vom Benutzer delegierbar sein (**Delegation**).

Delegation

Alle Sicherheitsdienste und -mechanismen, die im Grid-Umfeld eingesetzt werden, müssen sich diesen Herausforderungen stellen. Dementsprechend sind die in Abbildung 2.2 angegebenen Challenges nützliche und sinnvolle Kriterien zur Bewertung und werden deshalb im Kriterienkatalog in Abschnitt 3.2 aufgenommen werden.

Sicherheitsanforderungen in Grids

Aus der OSI-Sicherheitsarchitektur als Basis, der in diesem Teilabschnitt angegebenen Literatur zu Sicherheitsanforderungen in Grids sowie aus [CDS 08, Bund 07, Bund 05d, Bund 05a, Bund 05b, Bund 04, Ecke 06] wurde eine umfassende Liste von Sicherheitsanforderungen abgeleitet, die eine Grid-Sicherheitsarchitektur durch Dienste und Mechanismen umsetzen muss:

Authentisierung

- **Authentisierung** (Authentication) im Sinne der OSI–Sicherheitsarchitektur (vgl. S. 16)

Identifikation

- **Identifikation**: Die Überprüfung einer Entität, d.h. die Feststellung der Identität (z.B. durch Überprüfen des Personalausweises), und die Vergabe eines eindeutigen Identifikators (z.B. eines Benutzer–Namens) wird als Identifikation bezeichnet.

Single Sign On

- **Single Sign On (SSO)**: Ein Benutzer, der sich erfolgreich authentisiert hat, soll diese Authentisierung für eine beschränkte Zeit auch für weitere OGSA-Ressourcen nutzen zu können ohne sich wiederholt anmelden zu müssen.

Autorisierung

- **Autorisierung** (Authorization): Hierunter versteht [NJD+ 02] die Möglichkeit durch Spezifikation von Autorisierungs–Policies den Zugriff auf OGSA-Ressourcen zu beschränken. Der Vorgang der Zugriffs-

kontrolle (nach OSI vgl. S. 16) wird nicht weiter betrachtet. Demgegenüber fordert [Chiv 03] bereits eine verteilte Autorisierung.

- **Zugriffskontrolle** (Access Control): im Sinne der OSI-Sicherheitsarchitektur (vgl. S. 16) — Zugriffskontrolle
- **Delegation** von Rechten und Policies: Bei den Rechten wird hier eine eingeschränkte Delegation von der **Impersonation** , d.h. der Übernahme aller Rechte eines Subjekts unterschieden. Subjekte (denen Rechte zugeordnet sind) können sowohl Nutzer als auch Ressourcen sein, die im Auftrag eines Nutzers handeln. — Delegation
- **Lifetime Management**: Rechte (**Credentials**), die ein Benutzer erhält, sollen nur beschränkte Gültigkeit besitzen (**Credential Lifespan**). Da ein vom Benutzer initiierter Job im Grid aber länger laufen kann als diese Gültigkeitsdauer umfasst, muss es Mechanismen zur Verlängerung dieser Rechte geben (**Credential Renewal**). — Lifetime Management
- **Gruppenmanagement** (**Group Membership Management**): Für Kooperationsaufgaben, zur vereinfachten Autorisierung usw. ist es erforderlich dynamisch Gruppen bilden und auch wieder auflösen zu können. Dabei soll die Bildung von Gruppen nicht nur auf Organisations– oder VO–Ebene definiert werden, sondern selbst individuelle Nutzer sollen in die Lage versetzt werden Gruppen zu einem bestimmten Zweck zu etablieren [HuTh 01]. — Gruppenmanagement
- **Vertraulichkeit** (Confidentiality): Während [NJD+ 02] lediglich die Vertraulichkeit des Transportsystems und damit eineVertraulichkeit der Kommunikation fordert, sieht [HuTh 01] auch den Bedarf nach vertraulicher Speicherung von Daten und damit auch das Problem der **Schlüsselhinterlegung** (**Key Escrow**). Daten, die von einem Benutzer in verschlüsselter Form gespeichert wurden, müssen unter bestimmten Voraussetzungen auch dann noch zugänglich sein, wenn der Benutzer den Schlüssel verloren oder das Unternehmen verlassen hat. — Vertraulichkeit
- **Integrität** (nach OSI, vgl. S. 17) der Nachrichten (Message Integrity). Daneben ist die Integrität der gespeicherten Daten von zentraler Bedeutung. Im medizinischen Bereich muss die Integrität der Daten für 30 Jahre gewährleistet sein, um bspw. auf potentielle Schadensersatzansprüche (§ 199 BGB; Schadenersatzansprüche aus Verletzung des Lebens, Körpers und der Gesundheit) reagieren zu können. — Integrität
- **Verbindlichkeit** (Non Repudiation): im Sinne der OSI Sicherheitsarchitektur (vgl. S. 2.2.1). — Verbindlichkeit
- **Datenschutz** (**Privacy**): Entitäten (Nutzer oder Organisationen) müssen Datenschutz–Policies festlegen können. Diese sind geeignet und nachvollziehbar umzusetzen. Für bestimmte Bereiche (z.B. medizinische Forschung und klinische Versorgung oder auch bei bestimmten Telekommunikationsdiensten) gelten sehr enge gesetzliche Rahmen, die eine Nachvollziehbarkeit der Datenspeicherung als auch die Nachvollziehbarkeit — Datenschutz

der Datenübertragung verlangen (**Auditability** und **Trackability**, vgl. auch Abschnitt 4.5.1). Eng mit diesen Anforderung verbunden ist die Forderungen nach

Anonymisierung

- **A**nonymisierung (Anonymisation): In diesem Fall soll ein Nutzer Dienste anonym, z.B. mittels eines Pseudonyms, nutzen können und weder während der Dienstnutzung noch im Nachhinein identifizierbar sein. Es gibt Fälle, in denen eine anonyme Nutzung von Grid-Diensten rechtmäßig, wünschenswert oder sogar notwendig sein kann [SWT+ 02]. Dies ist bspw. in vielen medizinischen Studien eine absolut zwingende Anforderung (vgl. z.B. MediGrid [Comm 05]).

Policy Austausch

- **Austausch von Policies**: Zur Aushandlung eines Sicherheitskontextes zwischen Dienstnutzer und Dienstprovider müssen dynamisch Sicherheitspolicies ausgetauscht werden.

Logging

- **Sicheres Logging**: Alle Dienste, insbesondere die Sicherheitsdienste müssen in die Lage versetzt werden Nutz- und Zugriffsdaten oder sonstige Events mit einem Zeitstempel zu versehen. Der Begriff „sicher“ umfasst hier verlässlich und richtig, d.h. die Speicherung darf nicht unterbrechbar oder im nachhinein änderbar sein. Sicheres Logging kann als Basis für Verbindlichkeit (vgl. S. 17), Auditing und Notariatsdienste (Notarization) genutzt werden.

Auditing

- **Auditing** umfasst die externe Prüfung (und ggf. Zertifizierung) von Sicherheitseigenschaften und die Einhaltung eines vorgeschriebenen Sicherheitsniveaus.

Sicherheitsalarme

- **Sicherheitsalarme** oder allgemeine Sicherheitsevents, sind Nachrichten, die vom Sicherheitsdienste bei kritischen Ereignissen oder bei Zustandsänderungen erzeugt werden und an eine Informationssenke (z.B. ein Managementsystem) verteilt werden.

Sandboxing und Virtualisierung

- **Sandboxing** und **Virtualisierung**: Unter dem Begriff Sandboxing werden alle Verfahren zusammengefasst, die Ressourcen, welche ein Benutzer einer fremden Organisation benutzt, von der lokalen Infrastruktur, auf der die Ressource betrieben wird, isolieren. Damit sollen Schäden, die durch Angriffe, missbräuchliche oder fehlerhafte Nutzung an der lokalen Infrastruktur entstehen könnten, minimiert werden. Zur Erreichung dieses Ziels werden auch Virtualisierungsverfahren genutzt.

Mapping

- **Mapping**: Nachdem interorganisationale Szenarien unterschiedliche Sicherheitsdomänen umfassen, bedarf es Mechanismen, um eine Übersetzung zwischen diesen Domänen zu gewährleisten. Dies umfasst die Übersetzung von unterschiedlichen Namensschemata, das Mapping von Policies und Rechten.

Assurance

- **Assurance**: Diese Anforderung meint die Zusicherung der Einhaltung eines bestimmten Sicherheitsniveaus und die Information über umgesetzte Sicherheitsmechanismen. Diese Information kann für die Entscheidung verwendet werden, in welcher Domäne ein Dienst realisiert wird.

- **Trust Level Management**: Zwischen den beteiligten Partner bestehen implizite oder explizite Vertrauensbeziehungen. Das Trust Level Management befasst sich mit der Formalisierung von Vertrauen und der Ermittlung sowie Bewertung von Vertrauens-Level und deren sicherem und zweifelsfreiem Austausch. Trust Level Management

- **Manageability**: Alle Sicherheitsdienste und -mechanismen, die Sicherheitsanforderungen realisieren, müssen auch verwaltet, administriert und ins lokale Sicherheitsmanagement integriert werden. Manageability

- **Firewall traversal**: Ein Haupthinderniss für dynamisches interorganisationales Grid Computing sind heutige Firewalls. Eine Sicherheitsarchitektur muss dieser Tatsache Rechnung tragen, d.h. es müssen Konzepte entwickelt werden, die Firewall für Grid-Dienste durchlässiger zu machen ohne die lokale Firewall Policy und das lokale Sicherheitsniveau zu gefährden. Firewall traversal

2.3 Sicherheitsdienste in Föderationen: Dienstabhängigkeiten und Diensthierarchie

Im folgenden Abschnitt wird eine Diensthierarchie, vergleichbar mit der bei den OSI-Sicherheitsdiensten, abgeleitet und es werden Abhängigkeiten zwischen den Diensten ermittelt. Für die Zwecke dieser Arbeit werden eine starke sowie eine schwache Abhängigkeit definiert.

Eine **starke Abhängigkeit** zwischen den Diensten A und B liegt dann vor, wenn der Dienst A ohne den Dienst B nicht erbracht werden kann. Der Dienst B stellt also einen **Basisdienst** für A dar. starke Abhängigkeit

Ein Beispiel für einen solchen Basisdienst ist die Integritätssicherung. In Föderationen sind Mechanismen erforderlich, um Veränderungen und Manipulationen an Authentisierungsdaten, Rechten, vertraulichen Daten oder Sicherheitsalarmen erkennen zu können. Ist es nicht möglich bspw. Manipulationen an Rechten zu erkennen oder zu verhindern, ist eine sichere Autorisierung nicht möglich. Ein weiteres Beispiel stellt ein Firewall-Dienst dar. Die Einhaltung und richtige Umsetzung der Firewall-Policies kann nur überprüft werden, wenn ein aussagekräftiges Logging zur Verfügung steht. Damit besteht eine starke Abhängigkeit zwischen dem Firewall-Dienst und dem Logging.

Ein Dienst A, der in einer **schwachen Abhängigkeiten** von B steht, kann in der Regel auch ohne B erbracht werden. Es besteht allerdings insofern eine Abhängigkeit, dass der Dienst B, unter bestimmten Bedingungen, die Diensterbringung von A erleichtern kann. schwache Abhängigkeit

Ein Beispiel einer solchen schwachen Abhängigkeit stellen Datenschutz und Anonymisierung dar. Datenschutz kann in vielen Fällen ohne Anonymisierung realisiert werden, andererseits ist Anonymisierung ein nützliches Hilfsmittel um Datenschutz zu realisieren. Nur in bestimmten Fällen, z.B. bei der gesetzlichen Vorgabe zur Anonymisierung von Patientendaten bei wissenschaftlichen Studien, besteht eine starke Abhängigkeit zwischen Datenschutz und Anonymisierung. Ein weiteres Beispiel sind Sicherheitsalarme. Für die meisten Sicherheitsdienste kann es hilfreich sein ein zentrales Eventmanagement zu besitzen und dort dezentral erzeugte Sicherheitsalarme zu empfangen und zu verarbeiten. Auf diese Weise ist es möglich den Sicherheitsstatus in einer Föderation zu überwachen. Die Sicherheitsdienste erzeugen oft lokale Alarme und funktionieren auch ohne zentrale Event-Konsole. Deshalb ist die Abhängigkeit zwischen den Diensten und den Sicherheitsalarmen als schwach einzustufen.

paarweise Abhängigkeitsanalyse von Sicherheitsdiensten

Als Basis für die Ableitung einer Hierarchie innerhalb der Sicherheitsdienste wurden die im vorangegangenen Abschnitt ermittelten Sicherheitsanforderungen verwendet und systematisch Abhängigkeiten analysiert. Dabei wurden Paare von Sicherheitsdiensten miteinander in Beziehung gesetzt und es wurde geprüft, ob und welche Art von Abhängigkeit zwischen ihnen besteht. Diese Analyse kann unabhängig von der konkreten Realisierung des Sicherheitsdienstes mit einem bestimmten Sicherheitsmechanismus durchgeführt werden und ist damit auch unabhängig von der technischen Realisierung eines bestimmten Dienstes.

Tabelle 2.1 fasst die Abhängigkeiten zwischen den in Abschnitt 2.2 abgeleiteten Sicherheitsdiensten zusammen. Ein Kreuz ( ) steht dabei für eine starke, ein Haken ( ) für eine schwache Abhängigkeit.

Gruppierung in Dienstklassen

Die Tabelle 2.1 wird dann als Basis für eine Gruppierung der Sicherheitsdienste in Dienstklassen verwendet. Eine Strukturierung nach der Häufigkeit von starken und schwachen Abhängigkeiten ergibt zwei Cluster. Bei den starken Abhängigkeiten finden sich Policy Management, Trust Level Management, Authentisierung, Autorisierung und Assurance. Der Cluster mit schwachen Abhängigkeiten beinhaltet: Auditing, Sicherheitsalarme und Logging.

Der letztgenannte Cluster umfasst nur Dienste des Sicherheitsmanagements und kann dementsprechend als Dienstklasse „Sicherheitsmanagement Basisdienste“ übernommen werden. Im ersten Cluster sind Dienste zusammengefasst, von denen viele andere Dienste abhängen, d.h. dieser Cluster könnte die Dienstklasse „Basisdienste“ bilden. Allerdings fallen unter funktionalen Gesichtspunkten die Authentisierung sowie die Autorisierung etwas aus dem Rahmen. Daher wird dieser Cluster in zwei Dienstklassen unterteilt. Die Dienste Authentisierung, Autorisierung, Identifikation, Gruppenmanagement, Single Sign On und Zugriffskontrolle werden in der Dienstklasse „AAI/VO-Dienste“ zusammengefasst.

Unter funktionalen Gesichtspunkten finden sich die beiden weiteren Dienstklassen der „Datensicherheitsdienste“ und der „Privacy Dienste“. Der Firewall-Dienst gehört zur Dienstklasse der Gefahrenabwehr. Tabelle 2.2

Abhängigkeit (Dienst erfordert Sub-Dienst)	Anonymisierung	Assurance	Auditing	Authentisierung	Autorisierung	Datenschutz	Firewall-Traversal	Gruppen-Management	Identifikation	Integrität	Logging	Policy Management	Sandboxing	Sicherheitsalarme	Single Sign On	Trust Level Management	Verbindlichkeit	Vertraulichkeit	Zugriffskontrolle
Anonymisierung		✗	✓	✗	✗	✗						✗				✗			
Assurance												✗							
Auditing		✗										✗							
Authentisierung		✗	✓							✗	✓	✗			✓		✗		✓
Autorisierung		✗	✓	✗		✓				✗	✓	✗					✗		
Datenschutz	✓	✗	✓		✗							✗					✗		✓
Firewall-Traversal		✗	✓	✗	✗						✗	✗					✗		✓
Gruppen-Management		✗	✓	✗	✗					✓	✓	✗					✗		✓
Identifikation		✗	✓			✓				✓		✗					✗		
Integrität		✗	✓									✗		✗			✗		✓
Logging		✗	✓									✗	✓	✓	✓	✓	✗		✓
Policy Management		✗	✓																
Sandboxing		✗	✓									✗							✓
Sicherheitsalarme		✗	✓							✗	✓	✗							✓
Single Sign On		✗	✓		✗					✗	✓	✗					✗		
Trust Level Management		✗	✓								✓	✗							
Verbindlichkeit		✗	✓	✗	✗						✗	✗							
Vertraulichkeit		✗	✓	✗						✗		✗							
Zugriffskontrolle		✗	✓	✗	✗	✓					✓	✗		✓				✓	

Tabelle 2.1: Sicherheitsdienste: starke ( ) und schwache ( ) Abhängigkeit

fasst diese Gruppierung nach den Dienstklassen unter Berücksichtigung der Dienstabhängigkeiten zusammen.

Hierarchie der Sicherheits-dienstklassen

Die ermittelten Dienstklassen können jetzt untereinander nochmals als gesamte Klasse auf Abhängigkeitenbeziehungen untereinander untersucht und abhängig davon gruppiert werden. Das Ergebnis dieser Analyse ist die in Abbildung 2.3 dargestellte Hierarchie der Sicherheitsdienstklassen. In der Abbildung weiter oben stehende Dienstklassen benötigen zur Realisierung ausgewählte Dienste aus den unterliegenden Dienstklassen. Die seitlich angeordneten Dienstklassen sind für das gesamte Rahmenwerk wichtig, haben aber keien Abhängigkeitsbeziehungen zu den anderen.

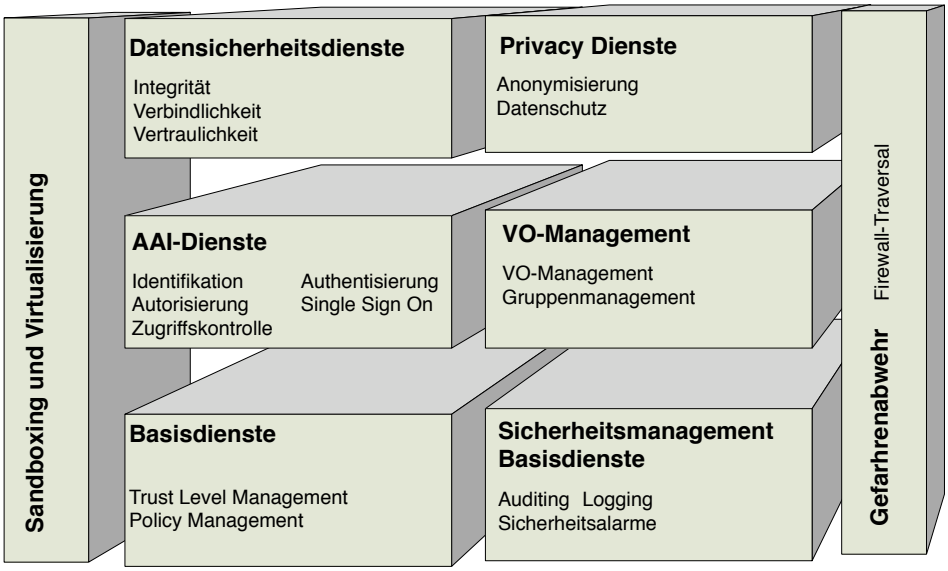

Abbildung 2.3: Hierarchie der Sicherheitsdienstklassen

2.4 Auswirkungen auf die Aufgabenstellung

Unterstützung von Sicherheitsverantwortlichen

Das Hauptziel dieser Arbeit ist die Unterstützung von Sicherheitsverantwortlichen und Sicherheitsadministratoren, die im Rahmen der interorganisationalen Kooperation an Föderationen teilnehmen oder sogar Sicherheitsverantwortung für die gesamte Föderation besitzen. Die Unterstützung wird dabei insbesondere die Phasen Anforderungsanalyse und Design sowie Instantiierung einer Föderation unterstützen. Das Rahmenwerk kann aber auch genutzt werden, um bereits bestehende Föderationen in der Betriebsphase sicherheitstechnisch zu analysieren und zu bewerten, um damit ggf. aktuell genutzte Sicherheitsdienste zu optimieren oder durch bessere zu ersetzen.

Dienstklassen	Basisdienste			AAI-Dienste						Datensicherheits-dienste			Privacy-Dienste	Sandboxing	Gefahren-abwehr	Sicherheits-management Basisdienste		
Abhängigkeit (Dienst erfordert Sub-Dienst)	Assurance	Policy Management	Trust Level Management	Authentisierung	Autorisierung	Identifikation	Gruppen-Management	Zugriffskontrolle	Single Sign On	Integrität	Verbindlichkeit	Vertraulichkeit	Anonymisierung	Datenschutz	Sandboxing	Firewall-Traversal	Auditing	Logging	Sicherheitsalarme
Anonymisierung	x	x	x	x	x									x			✓		✓
Assurance		x															✓		✓
Auditing	x	x															✓		✓
Authentisierung	x	x	x	x		x				x	x						✓	✓	✓
Autorisierung	x	x	x	x						x	x						✓	✓	✓
Datenschutz	x	x	x		x							x	✓				✓		✓
Firewall-Traversal	x	x	x	x	x			✓									✓	x	✓
Gruppen-Management	x	x	x	x	x					✓							✓	✓	✓
Identifikation	x	x	x	x						✓				✓			✓		
Integrität	x	x															✓		✓
Logging	x	x															✓		
Policy Management	x																✓		
Sandboxing	x	x						x				✓					✓	✓	✓
Sicherheitsalarme	x	x						x		x	✓						✓	✓	
Single Sign On	x	x	x	x						x	x						✓	✓	✓
Trust Level Management	x	x															✓	✓	
Verbindlichkeit	x	x		x	x												✓	x	✓
Vertraulichkeit	x	x		x						x							✓		✓
Zugriffskontrolle	x	x	x	x	x							✓		✓			✓	✓	✓

Tabelle 2.2: Sicherheitsdienste: Gruppierung in Dienstklassen (starke ( ) und schwache ( ) Abhängigkeit)

Im Kapitel 2.2 wurden die Sicherheitsanforderungen an Föderationen abgeleitet. Dazu wurden die OSI-Sicherheitsarchitektur vorgestellt sowie Grid-spezifische Anforderungen ermittelt. Innerhalb der OSI-Sicherheitsarchitektur bilden die Sicherheitsdienste, welche Sicherheitsanforderungen realisieren, eine Hierarchie (vgl. Abschnitt 2.2.1; Abbildung 2.1). Ein vergleichbares Abhängigkeitsmodell wurde als Basis für die weiteren Arbeiten abgeleitet (vgl. Abschnitt 2.3) und um föderationsspezifische Aspekte erweitert.

Dienstklassenhierarchie als strukturbildendes Element

Hilfsmittel für szenariospezifische Sicherheitsanalyse

In Abschnitt 2.3 wurden Dienstabhängigkeiten sowie eine Hierarchie innerhalb der Sicherheitsdienste ermittelt. Dienstabhängigkeiten und Hierarchie sind vollkommen Szenario-unabhängig und stellen umfassend Sicherheitsdienste für Föderationen dar. Die Gliederung der Dienste in Dienstklassen (vgl. Tabelle 2.2) und die in Abbildung 2.3 dargestellte Hierarchie wird als strukturbildendes Element für die Auswahl, die Bewertung und die Analyse der Sicherheitsdienste verwendet werden (vgl. Kapitel 4). Ein Sicherheitsverantworlicher soll diese Informationen zur Analyse seines konkreten Anwendungsszenarios nutzen. Dienstklassen, Dienstabhängigkeiten und Hierarchie stellen ein Hilfsmittel zur szenario–spezifischen Bedrohungs- und Sicherheitsanalyse in Föderationen dar. Auf der einen Seite liefert das Hierarchie–Modell eine Checkliste zur Überprüfung der konkreten Sicherheitsanforderungen, d.h. der Sicherheitsverantwortliche wird in die Lage versetzt für sein Anwendungsszenario relevante Sicherheitsanforderungen abzuleiten. Auf der anderen Seite zeigen die Hierarchie bzw. die Abhängigkeitsbeziehungen zwischen den Sicherheitsdiensten dem Verantwortlichen, welche Basis-Dienste für einen bestimmten Sicherheitsdienst erforderlich sind.

Ableitung einer Szenario-abhängigen Instanz des Rahmenwerks

Der Sicherheitsverantwortliche wird, mit dem in Kapitel 6 präsentierten Vorgehensmodell, in die Lage versetzt aus dem allgemeinen Szenario-unabhängigen Modell eine szenario-abhängige Instanz abzuleiten, welche dann nur noch die Dienste enthalten soll, die für das konkrete Szenario von Interesse sind. In Abbildung 2.4 ist ein (vereinfachtes) Beispiel einer solchen Auswahl angegeben. Das allgemeine Rahmenwerk beinhaltet alle Sicherheitsdienste für eine Föderation. Der Sicherheitsverantwortliche verwendet das in der oberen Hälfte der Abbildung 2.4 angegebene Rahmenwerk als Leitfaden und zur systematischen Analyse seines konkreten Szenarios. Im Rahmen dieser Analyse entscheidet er, welche Sicherheitsdienste überhaupt benötigt werden. Damit schließt die erste Phase (Analyse, vgl. Abschnitt 6.1) der sicherheitstechnischen Anforderungsanalyse in Föderationen ab.

Für die Design- und Realisierungsphase (vgl. auch Abschnitt 6.2) muss der Sicherheitsverantwortliche konkrete Sicherheitsmechanismen auswählen, um den gewünschten Sicherheitsdienst umzusetzen. Für die meisten der vorgestellten Sicherheitsanforderungen existiert in der Praxis häufig eine Vielzahl verschiedener Sicherheitsmechanismen. Eine weitere Aufgabe dieser Arbeit muss es daher sein, den Sicherheitsverantwortlichen Hilfsmittel für die Auswahl konkreter Sicherheitsmechanismen an die Hand zu geben. Zu diesem Zweck wird in Kapitel 3 ein Kriterienkatalog zur Bewertung von Sicherheits-

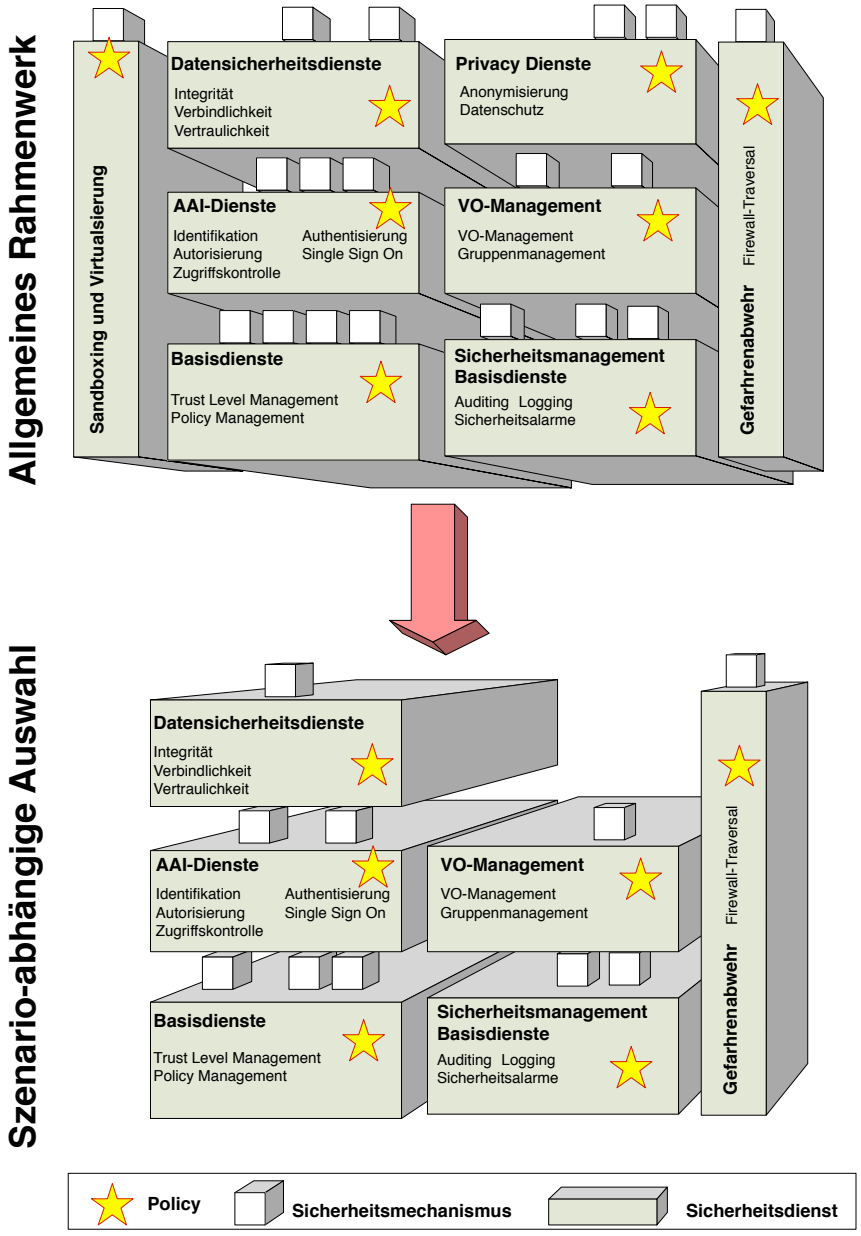

Abbildung 2.4: Exemplarische Ableitung eines szenario-abhängigen Auswahl

mechanismen entwickelt. Die Arbeit wird dann mit Hilfe dieses Kataloges in Kapitel 4 bestehende Mechanismen bewerten und einordnen. Mit dem Kriterienkatalog und dessen exemplarischer Anwendung wird der Sicherheitsverantwortliche in die Lage versetzt, die am besten für sein Szenario geeigneten Sicherheitsmechanismen auszuwählen. Er kann damit zu einer konkreten Instanz einer Sicherheitsarchitektur, die in der unteren Hälfte der Abbildung 2.4 dargestellt wird, kommen.

Die Konfiguration und Nutzung der Mechanismen und Sicherheitsdienste werden mit Hilfe von Policies spezifiziert. Nachdem der Sicherheitsverantwortliche die am besten geeigneten Mechanismen ausgewählt hat, muss er

dann, ggf. in Absprache mit seinen Föderationspartnern, die entsprechenden Policies spezifizieren.

Die Anforderungsanalyse aus Kapitel 2.2, die Dienstabhängigkeiten der Sicherheitsdienste aus Abschnitt 2.3, der Kriterienkatalog aus Abschnitt 3.2 und die Analyse und Bewertung existierende Sicherheitsmechanismen können dann genutzt werden, um eine Defizitanalyse im Hinblick auf Sicherheitsdienste existierender Grid-Middlewares durchzuführen. Die vorliegende Arbeit wird diese Defizite darstellen (vgl. Abschnitt 4.10) und für einige dieser Defizite in Kapitel 5 Lösungsansätze vorschlagen.

Kapitel 3

Klassifikations- und Bewertungsschema für interorganisationale Sicherheitsmechanismen

Inhaltsverzeichnis

In diesem Abschnitt wird ein anwendungsfall-spezifisches, aber szenario-unabhängiges Berwertungs- und Klassifikationsschema für Sicherheitsmechanismen erarbeitet. Den spezifischen Anwendungsfall stellen Föderationen am Beispiel von Grid-Infrastrukturen dar. Das Schema erlaubt es einem Sicherheitsverantwortlichen eine individuelle, vom konkreten Anwendungsszenario abhängige Instanz des Sicherheitsrahmenwerkes zu erstellen und damit festzulegen, welche Kriterien für das abzusichernde Szenario von Interesse und welche Sicherheitsmechanismen für dieses Szenario am besten geeignet sind. Der Sicherheitsverantwortliche muss in einem ersten Schritt die Relevanz jedes Bewertungskriteriums für den konkreten Anwendungsfall festlegen. Das Klassifikationsschema lässt sich ggf. um eigene Kriterien erweitern.

Danach kann eine vom Szenario abhängige Bewertung der Mechanismen erfolgen. Ergebnis dieses Bewertungsschrittes ist eine Ordnung und Priorisierung der Mechanismen. Mit der Analyse ist es auch möglich bestehende Defizite oder Schwachstellen in existierenden Mechanismen und Systemen zu identifizieren.

Im folgenden Abschnitt werden die allgemeine Methodik, die Begriffsdefinitionen sowie Berechnungsverfahren vorgestellt, die für die Erstellung des Kriterienkataloges eingesetzt werden. Abschnitt 3.1 ist unabhängig von einem konkreten Kriterienkatalog spezifiziert und für die Erstellung beliebiger Kriterienkataloge anwendbar. Im Abschnitt 3.2 wird der abstrakte Graph aus der Methodik mit konkreten Kriterien gefüllt. Dieser Abschnitt stellt den für die Bewertung von Sicherheitsmechanismen und -diensten in Grids spezifischen Kriterienkatalog dar.

3.1 Begriffsbestimmung und Methodik zur Erstellung eines Kriterienkataloges

Kriterienkatalog: allgemeines Werkzeug zu Bewertung und Vergleich

Ein Kriterienkatalog stellt ein allgemeines Werkzeug dar, um Szenarien bewerten und vergleichen zu können. Der Vorteil eines solchen Kataloges ist die Vergleichbarkeit verschiedener Szenarios auf unterschiedlichen Abstraktionsstufen. Auf der einen Seite lässt sich die Bewertung eines Szenarios auf eine Punktzahl oder eine Note zurückführen, andererseits kann der Kriterienkatalog aber auch dazu genutzt werden, sehr schnell Bereiche zu identifizieren, welche überproportional gut oder schlecht in das Ergebnis eingehen. Die im folgenden vorgestellten Begriffe leiten sich aus verschiedenen Projekten zur Bewertung von IT-Prozessen, IT-Konzepten und IT-Werkzeugen ab [Sche 99, Giem 00, Bren 02, Enge 07] und basieren auf der Nomenklatur von [BRS 02, Bren 02].

Definition Kriterienkatalog

Ein **Kriterienkatalog** besteht aus einer strukturierten Repräsentation der Kriterien, einem Intervall von Attributwerten (Bewertungen) für die einzelnen Kriterien und einem Gewichtungs- und Berechnungsverfahren, um zu einer Gesamtbewertung zu kommen.

Repräsentation als Graph

Die Struktur des Kataloges wird durch einen gerichteten, zyklenfreien Graphen repräsentiert (vgl. Abbildung 3.1). Die Kriterien werden durch Knoten repräsentiert. Die gerichteten Kanten zwischen den Knoten beschreiben eine (gewichtete) Assoziation zwischen den Kriterien (B ist Teilkriterium von A). Bei den Kriterien lassen sich zwei Arten unterscheiden:

Basiskriterium

- **Basiskriterien** (synonym: **Blattkriterien**) stellen atomare (d.h. keine Teilkriterien umfassende) Kriterien dar, deren Bewertung direkt erfolgt.

Hauptkriterium

- **Hauptkriterien** oder komplexe, zusammengesetzte Kriterien: Ihre Bewertung ist eine Funktion ihrer Beziehungen zu anderen (Haupt- und

Basis-) Kriterien sowie der Eigenschaften dieser Teilkriterien. Die **Wurzel** des Graphen, der die Struktur des Kriterienkataloges darstellt, ist ein ausgezeichnetes Hauptkriterium. Die Bewertung der Wurzel ist eine Funktion aller im Katalog dokumentierten Beziehungen und Attribute.

Einzig die Blattkriterien werden mit Attributbelegungen bewertet, die den Erfüllungsgrad des Kriteriums ausdrücken (Wertung). Im folgenden werden die in Tabelle 3.1 angegebenen Werte verwendet.

Attributbelegung

erreichter Erfüllungsgrad	Symbol	Bewertungszahl b
Kriterium wird voll erfüllt	++	3
Kriterium wird erfüllt	+	2
Kriterium wird teilweise erfüllt	-	1
Kriterium wird nicht erfüllt	--	0

Tabelle 3.1: Abstufungen für Erfüllungsgrad eines Kriteriums [BRS 02]

Die Bewertung eines Hauptkriteriums erfolgt mit Hilfe eines Berechnungsverfahrens über die Werte aller abhängigen Teilkriterien. Die Unterschiede in der Signifikanz einzelner Kriterien werden dabei über die in Tabelle 3.2 zusammengefassten Gewichte ausgedrückt.

Gewichte

erreichter Erfüllungsgrad	Gewichtung g
Kriterium ist außerordentlich wichtig	4
Kriterium ist wichtig	2
Kriterium ist weniger wichtig	1

Tabelle 3.2: Gewichtung für die Signifikanz von Teilkriterien [Bren 02]

Ein sinnvolles und hilfreiches Berechnungsverfahren liefert aussagekräftige und einfach zu interpretierende Werte. Veränderungen an der Struktur des Kataloges, wie z.B. die Hinzunahme oder das Entfernen einzelner Kriterien, sollten in Bezug auf die Vergleichbarkeit der abgeleiteten Werte keine Auswirkungen haben. Aus diesem Grund eignet sich bspw. das arithmetische Mittel nicht zur Berechnung der Bewertung. Das verwendete Verfahren, das diese Voraussetzungen erfüllt, lässt sich durch folgende Formel darstellen:

Berechnungsverfahren

$$b_P = \frac{\sum_{i=1}^{n} b_i g_{iP}}{\sum_{i=1}^{n} g_{iP}} \tag{3.1}$$

Die Bewertungszahl b_P eines Kriteriums P berechnet sich aus der Summe der gewichteten Bewertungszahlen der Teilkriterien, die in P eingehen. Dabei bezeichnet g_{iP} das Gewicht, mit dem Teilkriterium i in P eingeht.

Abbildung 3.1 fasst in einem fiktiven Beispiel die Begriffe und exemplarisch die Berechnungen zusammen. In diesem Beispiel gehen die Kriterien P_{23} und P_{24} mit Gewicht 1, P_{21} mit Gewicht 4 und P_{22} mit Gewicht 2 in die Berechnung von P_2 ein. Im Beispiel ergibt sich nach Anwendung der Formel (3.1) eine Bewertungszahl von 1,5 für P_2.

Betrachtet man den Graphen, so ist der Knoten i ein Kind von P. Falls das Kriterium i selbst wieder aus mehreren Teilkriterien besteht, kann die Formel rekursiv angewendet werden. Die Summe der gewichteten Bewertungszahlen wird mit der Summe der Gewichte normiert. Durch die Normierung bleibt die berechnete Bewertungszahl im Intervall aus Tabelle 3.1 und der Katalog kann sehr flexibel angepasst werden.

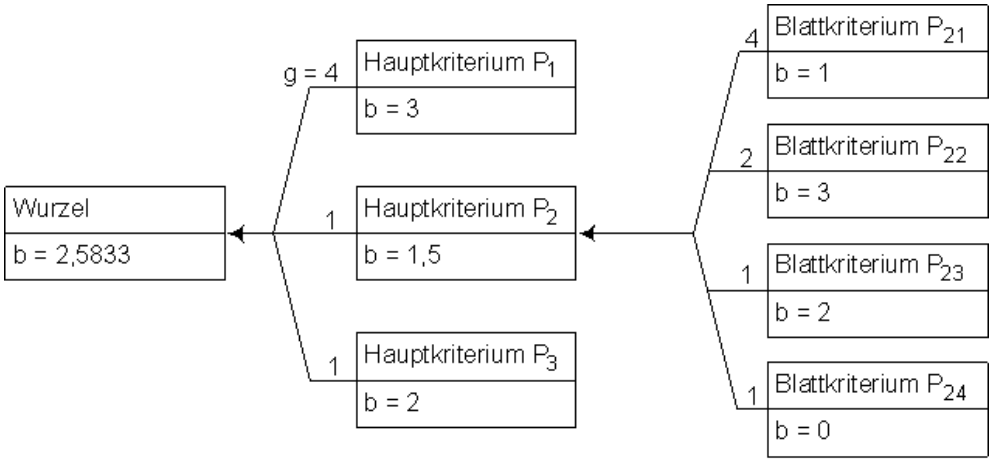

Abbildung 3.1: Kriterienkatalog; allgemeine Struktur

3.2 Kriterien für interorganisationales Sicherheitsmanagement

Ermittlung der Hauptkriterien

Im folgenden wird der im Rahmen diese Arbeit entwickelte Kriterienkatalog zur Bewertung von Sicherheitsmechanismen in Grids vorgestellt. Dieses Konzept ist neu und dient der Ermittlung und Verbesserung des Sicherheitsniveaus. Als Basis für die Ableitung von Kriterien zur Bewertung von Sicherheitsmechanismen wurde [NJD+ 02] mit den dort vorgestellten drei Herausforderungen (Challenges) verwendet (vgl. Abschnitt 2.2.2). In dieser Arbeit werden Integration, Interoperabilität und Trust als absolut kritisch für die Wirksamkeit von Grid-Sicherheitsmechanismen und für den Erfolg der Grid-Technologien selbst gesehen. Dementsprechend wurden diese drei Herausforderungen als Hauptkriterien in den Kriterienkatalog aufgenommen (vgl. Abbildung 3.2) und mit einer hohen Gewichtung versehen. Die in Abschnitt 2.2.2 in der Abbildung 2.2 dargestellten Unterpunkte der Herausforderungen stellen zum Teil sehr technische Aspekte und Beispiele für Herausforderungen dar und sind als solches nicht unmittelbar als Kriterien anwendbar. Aus der Diskussion in [NJD+ 02] und den Erfahrungen aus verschiedenen Grid Projekten lassen sich jedoch allgemeingültigere Basiskriterien zu diesen drei Hauptkriterien ableiten. Diese werden in den nächsten Abschnitten vorgestellt. Außer-

dem lassen sich aus den Challenges weitere Kriterien ableiten, die sich nicht primär in das Dreieck aus Abbildung 2.2 einordnen lassen. Hierunter fallen Delegations- und Mappingkonzepte.

Aus den Erfahrungen der verschiedenen Grid-Projekte [KRSS 06, BgFGR 06c, BgFGR 06a, BgFGR 06b] zeigt sich jedoch, dass [NJD+ 02] Fragen der Betreibbarkeit und des Managements völlig unberücksichtigt lässt. Da diese Aspekte aber auch einen entscheidenden Einfluss auf die Bewertung von Sicherheitslösungen haben, werden die Kriterien um Skalierbarkeit, Administrierbarkeit und Flexibilität erweitert.

Die Sicherheitseigenschaften des Sicherheitsmechanismus selbst ist ein Kriterium, das als so wichtig angesehen wird, dass es im Gegensatz zu [NJD+ 02] auch als Hauptkriterium in den Kriterienkatalog aufgenommen und mit einer hohen Gewichtung versehen wird.

Im folgenden Abschnitt wird der Kriterienkatalog, der sich an diesen Hauptkriterien orientiert, eingeführt und erläutert.

3.2.1 Kriterienkatalog

Der Kriterienkatalog (vgl. Abbildung 3.2) listet alle Kriterien auf. Bei einem Hauptkriterium sind alle Teilkriterien mit ihren Gewichten angegeben. Im Anschluss daran werden die Teilkriterien mit Ihren Teil- bzw. Basiskriterien aufgeführt. Bei den Basiskriterien werden die Bewertungsmaßstäbe entsprechend der Tabelle 3.1 gegliedert.

Festlegung der Gewichte

Die Festlegung der Gewichte kann spezifisch für den konkreten Anwendungsfall erfolgen. Der Sicherheitsverantwortliche kann die Wichtigkeit des einzelnen Kriteriums für seinen spezifischen Fall anpassen und mit diesem ersten Schritt auch den Kriterienkatalog für den konkreten Fall instantiieren.

Im vorgestellten Kriterienkatalog wurden die Gewichte nach den Erfahrungen aus dem D-Grid Projekt (vgl. Abschnitt 4.1.11) vergeben. Auf Ebene 1 des Baumes wird neben den drei wichtigsten Herausforderungen aus [NJD+ 02] auch das Sicherheitsniveau als außerordentlich wichtig klassifiziert und deshalb mit 4 gewichtet. Da in gegenwärtigen Szenarien die Einigung auf gemeinsame Policies und die formale Spezifikation von Policies bereits einen erheblichen Fortschritt darstellt und es in diesen Szenarien derzeitig kaum möglich erscheint, unterschiedliche Policies innerhalb der verschiedenen Domänen überhaupt formal zu spezifizieren, wird das Mapping konfliktärer Policies derzeit als weniger wichtig erachtet und daher mit 1 gewichtet.

KRITERIUM I: Integration

KRITERIUM I: Integration

Die Integration neuer Mechanismen in bestehende Systeme und die Erweiterbarkeit der zu bewertenden Mechanismen werden im Kriterium Integration zusammengefasst. In [NJD+ 02] wird die Herausforderungen weiter un-

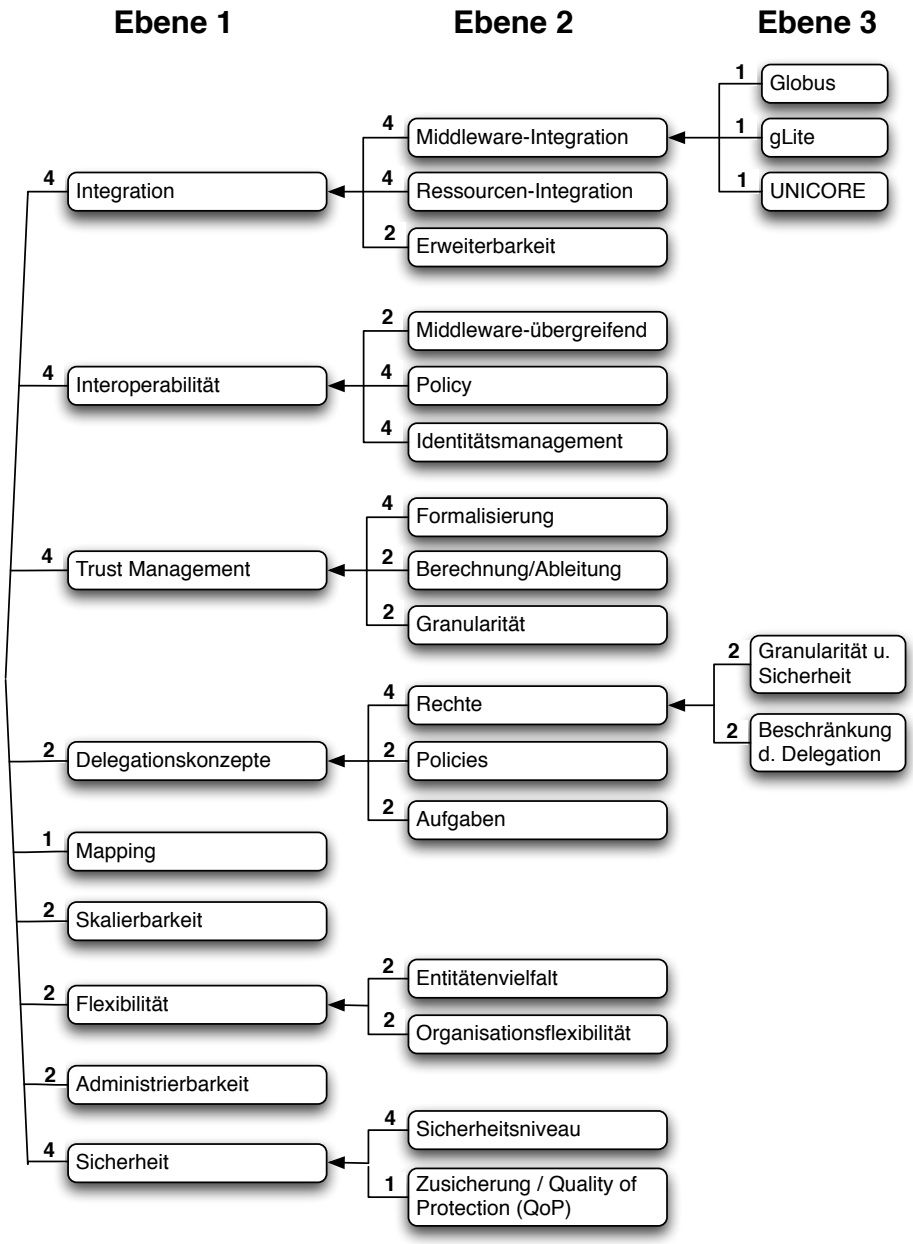

Abbildung 3.2: Kriterienkatalog zur Bewertung von Sicherheitsmechanismen

tergliedert in die Forderung nach erweiterbaren Architekturen. Die Sicherheitsmechanismen sollen sich in bestehende Systeme integrieren lassen und unabhängig von konkreten Implementierungen verwendbar bleiben („Implementation Agnositic“). Aus den Erfahrungen der Grid-Projekte (vgl. Abschnitt 4.1.11) hat sich gezeigt, dass neben der Integration in verwendete Middlewares auch die Integration in die verwendenten Ressourcen entscheidende Bedeutung haben und deshalb mit 4 zu gewichten sind. Die Erweiterbarkeit ist als wichtig zu klassifizieren.

KRITERIUM I: Integration	Gewicht
Middleware-Integration	4
Ressourcen-Integration	4
Erweiterbarkeit	2

KRITERIUM I.1: Middleware-Integration

KRITERIUM I.1: Middleware-Integration

Im Hinblick auf die Middleware-Integration werden die drei wichtigsten Middlewares betrachtet. In der Regel wird eine der Middlewares eine bedeutendere Rolle spielen und deshalb höher gewichtet werden als die anderen. In unserem Fall sind die drei wichtigsten Middlewares als gleich einzustufen und werden daher einheitlich mit 1 gewichtet.

KRITERIUM I.1: Middleware-Integration	Gewicht
Globus	1
gLite	1
UNICORE	1

KRITERIUM I.1.a: Globus
KRITERIUM I.1.b: gLite
KRITERIUM I.1.c: UNICORE

KRITERIEN:Globus, gLite, UNICORE

Für alle drei Blattkriterien gilt folgendes Bewertungsschema:

0 Mechanismus ist als vollständig eigenständige Lösung ohne Möglichkeit der Integration konzipiert; der Mechanismus wird von der Middleware nicht unterstützt.

1 Mechanismus ist als eigenständige Lösung realisiert. Eine eventuelle Integration ist aufwändig

2 Mechanismus ist als Zusatzpaket in die Middleware integrierbar.

3 Mechanismus ist Teil der Middleware

KRITERIUM I.2: Ressourcen-Integration

KRITERIUM I.2: Ressourcen-Integration

0 Der Mechanismus wird von den Ressourcen oder den darauf eingesetzten Betriebssystemen nicht unterstützt. Eine Integration ist nicht möglich.

1 Der Mechanismus wird von den Ressourcen oder den darauf eingesetzten Betriebssystemen teilweise unterstützt. Eine Integration ist aufwändig.

2 Der Mechanismus wird von den Ressourcen oder den darauf eingesetzten Betriebssystemen unterstützt. Eine Integration ist möglich.

3 Der Mechanismus wird von den Ressourcen oder den darauf eingesetzten Betriebssystemen voll unterstützt. Eine nahtlose Integration ist möglich.

KRITERIUM I.3: Erweiterbarkeit

KRITERIUM I.3: Erweiterbarkeit

Dieses Kriterium bewertet, inwieweit der Mechanismus für Erweiterungen offen ist, die denselben Sicherheitsdienst realisieren, und die Möglichkeit den bestehenden Mechanismus um zusätzliche Verfahren zu erweitern:

0 Eine Erweiterung um neue Mechanismen ist nicht möglich.

1 Eine Erweiterung um neue Mechanismen ist sehr aufwändig. Es sind Anpassungen an Middleware-Komponenten und damit eine internationale Abstimmung dieser Änderungen mit den Entwicklern erforderlich.

2 Eine Erweiterung um neue Mechanismen ist aufwändig. Die Middleware lässt sich um eigene Komponenten erweitern, eine Abstimmung ist nur innerhalb des Grids erforderlich.

3 Die Lösung bietet wohldefinierte Schnittstellen, um weitere Mechanismen zu integrieren.

KRITERIUM II: Interoperabilität

KRITERIUM II: Interoperabilität

Die Interoperabilität muss auf verschiedenen Ebenen betrachtet werden; zum einen ist die Ebene der Middleware und der Protokolle, dann die Ebene der Policies und schließlich auch die Frage des Identitätsmanagements. Die Middleware-übergreifende Interoperabilität ist nur in Szenarien außerordentlich wichtig, in denen auch wirklich mehrere Middlewares parallel betrieben werden. Deshalb wird dieses Kriterium mit 2 gewichtet. Die anderen beiden Kriterien sind außerordentlich wichtig.

KRITERIUM II: Interoperabilität	Gewicht
Middleware-übergreifend	2
auf Ebene der Policies	4
beim Identitätsmanagement	4

KRITERIUM II.1: Middleware-übergreifende Interoperabilität

KRITERIUM II.1: Middleware-übergreifende Interoperabilität

Hier wird bewertet, ob der Mechanismus für mehrere Middlewares zur Verfügung steht.

0 Der Mechanismus wird nur von einer Middleware unterstützt.

1 Der Mechanismus wird von zwei Middlewares unterstützt.

2 Der Mechanismus wird von drei Middlewares unterstützt.

3 Der Mechanismus wird von mehr als drei Middlewares unterstützt.

KRITERIUM II.2: Interoperabilität auf Ebene der Policies

KRITERIUM II.2: Interoperabilität auf Ebene der Policies

Auf der Ebene der Policies bedeutet eine sichere Interoperabilität, dass die beteiligten Partner und Institutionen in die Lage versetzt werden ihre Policies zu spezifizieren und auf formalisierte Art und Weise auszutauschen sowie ein Policy Mapping durchzuführen.

0 Kein Policy-Konzept vorgesehen.

1 Policies werden informell spezifiziert; es sind keine Mechanismen zum Policy Mapping vorgesehen.

2 Policies können lokal spezifiziert werden; Policy-Konflikte werden im Einzelfall gelöst.

3 Policy-Spezifikation ist formalisiert und global eindeutig oder es gibt Mechanismen um ein Policy Mapping durchzuführen.

KRITERIUM II.3: Interoperabilität beim Identitätsmanagement

KRITERIUM II.3: Interoperabilität beim Identitätsmanagement

Jede Organisation betreibt ein lokales Identitäts- und Zugriffskontrollsystem (Identity & Access Management System; I&AM). Innerhalb des I&AM werden Namensschemata und Attribute festgelegt. Eine Interoperabilität der Schemata sowie der Verfahren beim ID-Management vermindert Redundanzen, Inkonsistenzen und verringert den Managementaufwand.

0 Es existieren keine Mechanismen zum Identitätsmanagement. Für das Grid müssen parallel zu lokalen ID-Managementsystemen eigene ID-Managementsysteme aufgesetzt werden. Der organisatorische Aufwand für die Abbildung ist hoch. Ein Abstimmung der Namensräume findet nicht statt.

1 Es gibt Mechanismen, um einen Benutzer der Domäne einem Gruppe-Account in der anderen Domäne zuzuordnen. Für das Grid müssen parallel zu lokalen ID-Managementsystemen eigene ID-Managementsysteme aufgesetzt werden. Der organisatorische Aufwand für die Abbildung ist hoch. Ein Abstimmung der Namensräume findet auf Grid-Ebene statt.

2 Es gibt Mechanismen um einen Benutzer der Domäne einem individuellen Account in der anderen Domäne zuzuordnen.Es gibt einheitliche Verfahren um lokale Accounts auf Grid Accounts abzubilden (lokale Übersetzung).

3 Es gibt Mechanismen des föderierten Identitätsmanagements. D.h. die Heimatdomäne ist die einzige autoritative Quelle für Benutzerinformationen. Die entsprechenden Namensschemata sind aufeinander abgestimmt.

KRITERIUM III: Trust Management

KRITERIUM III: Trust Management

Vertrauensbeziehungen zwischen den beteiligten Partner in Grids sind eine fundamentale Basis, um überhaupt Sicherheit in Grids erreichen zu können. In heutigen Systemen und Kooperationen erfolgt das Trust Level Management i.d.R. implizit oder über sehr globale vertragliche Vereinbarungen und dort festgelegte Vertragsstrafen. Diese Techniken sind für überschaubare Kooperationen unter Umständen noch ausreichend. In hoch dynamischen Grid-Umgebungen, in denen zwischen den Partnern keine bilateralen Verträge geschlossen werden können, ist ein Trust Level Management, das sich auf

Vertragsstrafen stützt, weder angemessen noch ausreichend. Außerordentlich wichtig ist hier als grundlegende Basis für ein Trust Level Management eine exakte Formalisierung des **Vertrauenswertes (Trust Levels)**. Dieses Kriterium ist deshalb mit 4 zu gewichten. Aber auch die Wichtigkeit der Berechnung des Vertrauenswertes, dessen Ableitung und Verteilung, darf nicht unterschätzt werden. Ein weiteres wichtiges zu berücksichtigendes Bewertungskriterium ist die Flexibilität und Granularität bei der Spezifikation und Umsetzung.

KRITERIUM III: Trust Managment	Gewicht
Formalisierung	4
Berechnung oder Ableitung von Vertrauenswerten	2
Granularität	2

KRITERIUM III.1: Formalisierung

KRITERIUM III.1: Formalisierung

Bei der Formalisierung des Trust Level Management wird bewertet, inwiefern überhaupt abgestufte Vertrauenswerte angegeben werden können und ob diese für verschiedene Anwendungsfälle oder Dienstklassen unterschiedlich vergeben werden können.

0 Es gibt kein Trust Level Management.

1 Es liegt ein implizites Vertrauensmodell zugrunde. Die Vertrauenswerte sind binär repräsentierbar („Vertrauen“, „kein Vertrauen“). Die Trust Level gelten global für alle Anwendungsfälle und Dienste.

2 Es gibt spezifizierte Vertrauenswerte, die den Partnern zugeordnet werden. Die Trust Level gelten global oder allenfalls für große Klassen von Anwendungsfällen oder Dienste.

3 Es existieren Verfahren, die Partner den spezifizierten Vertrauenswerten zuordnen. Die Trust Level können abhängig von der Anwendungsdomäne oder Dienstspezifisch vergeben werden.

KRITERIUM III.2: Berechnung oder Ableitung von Vertrauenswerten

KRITERIUM III.2: Berechnung oder Ableitung von Vertrauenswerten

Das Trust Level Management ist kein statischer Prozess, der einen Vertrauenswert festlegt, welcher über die gesamte Zeit der Kooperation unverändert bleibt. Im Trust Level Management ist es erforderlich Vertrauen dynamisch neu zu klassifizieren bzw. abhängig vom Verhalten des Partners oder abhängig von der Einschätzung Dritter anzupassen. Reputationssysteme werden auch genutzt, um für neue Partner, abhängig von deren Vertrauen zu bereits bekannten Dritten, initial oder kontinuierlich einen Vertrauenswert festzulegen. Dazu wird die Einschätzung Dritter in Abhängigkeit des Vertrauenswertes des Dritten in die Berechnung einbezogen.

Ein weiterer wichtiger Aspekt bei Reputations- und Bewertungssystemen ist die zeitliche Komponente bzw. die Historie des Verhaltens. Bösartiges oder gutartiges Verhalten des Partners wirkt sich auf dessen Vertrauensbewertung aus. Aspekte, die in diesem Zusammenhang mit in die Berechnung des Vertrauenswertes eingehen, sind die Aktualität der Reputationsinformation (d.h.

die Einschätzung Dritter) und der Verlauf des Verhaltens des zu Bewertenden über die Zeit, d.h. wurde der zu Bewertende nur wenige Male positiv oder negativ auffällig oder befindet er sich in einer kontinuierlichen Verbesserung oder Verschlechterung seines Trust Levels.

0 Es gibt keine Verfahren, um Vertrauen zu berechnen oder abzuleiten. Eine dynamische Anpassung fehlt.

1 Es gibt keine Verfahren, um Vertrauen zu berechnen oder abzuleiten. Die Vertrauenswerte werden periodisch überprüft.

2 Es gibt Reputationsverfahren und mittelbares Vertrauen. Die Vertrauenswerte werden periodisch überprüft.

3 Es gibt Reputationsverfahren und mittelbares Vertrauen. Es gibt eine Historie zu den Partnern, die deren Verhalten bei der Einstufung und Reklassifizierung berücksichtigen. Die Vertrauenswerte werden kontinuierlich ermittelt.

KRITERIUM III.3: Granularität des Trust Level Management

KRITERIUM III.3: Granularität des Trust Level Management

In den gegenwärtig vorherrschenden Kooperationen wird das Trust Level Management nur auf organisatorischer Ebene definiert. Individualisierte Vertrauensverhältnisse der Nutzer oder ihre individuellen Forderungen werden i.d.R. nicht berücksichtigt. Aus dem Bereich des föderierten Identitätsmanagements kommt das Konzept der **Attribute Release Policies (ARPs)** und der **Attribute Acceptance Policies (AAPs)**. Mit Hilfe der ARPs kann jeder Nutzer festlegen, welche Informationen über ihn an welchen Service Provider übermittelt werden dürfen. Mit den AAPs spezifiziert der Provider den Mindestsatz an Informationen, die er benötigt, um seinen Dienst erbringen oder abrechnen zu können. Dieses Grundprinzip lässt sich auch auf das Trust Level Management ausdehnen, wenn Benutzergruppen oder einzelne Nutzer in der Lage versetzt werden, eigene Trust Level abweichend von denen der Heimatdomäne zu spezifizieren. Auf der anderen Seite kann auch der Provider auf individueller oder Gruppenbasis den Trust Level festlegen.

0 Es gibt kein Trust Level Management oder der Vertrauenswert wird nur auf Ebene der Institution festgelegt.

1 Die Ressourcen-Provider sind in der Lage für Gruppen innerhalb einer Organisation unterschiedliche Vertrauenswerte zu vergeben.

2 Die Ressourcen-Provider sind in der Lage individuelle Trust Level zu vergeben.

3 Die Ressourcen-Provider und die Nutzer können eigene Trust Level vergeben.

KRITERIUM IV: Delegationkonzepte

KRITERIUM IV: Delegationkonzepte

Die einfache, effiziente und trotzdem sichere Zusammenarbeit innerhalb ei-

ner Föderation erfordert die Übertragung der zu einer Aufgabenerfüllung notwendigen Rechte. Ohne Delegation von Rechten ist eine interorganisationale Zusammenarbeit kaum möglich. Das Kriterium wird deshalb mit 4 gewichtet. Die Durchführung oder Erledigung von Aufgaben selbst wird durch die Delegation von Tasks oder Jobs realisiert.

Zur Spezifikation der technischen und organisatorischen Rahmenbedingungen werden unabhängige Policies in den beteiligten Organisationen spezifiziert. In einigen Fällen ist es notwendig, diese Policies aus der Quell- in die Ziel-Domäne (und umgekehrt) zu übermitteln, diese auszuwerten und entsprechend zu berücksichtigen.

KRITERIUM IV: Delegation	Gewicht
Rechte	4
Policies	2
Aufgaben	2

KRITERIUM IV.1: Rechtedelegation

KRITERIUM IV.1: Rechtedelegation

Die Rechtedelegation meint die Übertragung eigener Rechte auf einen Stellvertreter (-prozeß), der häufig auch als Proxy bezeichnet wird. Der Stellvertreter handelt dann, ggf. in einer fremden Domäne, im Auftrag des Delegierenden.

Die Delegation von Rechten ist ein zweistufiger Prozess. Zum Einen muss eine möglichst feingranulare Rechtespezifikation und die Beschränkung der delegierten Rechte bewertet werden. Andererseits ist aber auch die Möglichkeit der Einschränkung des Delegationsrechts selbst zu betrachten. Diese beiden Kriterien werden als wichtig erachtet und mit 2 gewichtet.

KRITERIUM IV.1: Rechtedelegation	Gewicht
Granularität und Sicherheit	2
Beschränkung des Delegationsrechts	2

KRITERIUM IV.1.a: Granularität und Sicherheit

KRITERIUM IV.1.a: Granularität und Sicherheit

Für die Zuteilung von Rechten gibt es ein Prinzip in der Sicherheitstheorie, das Art und Menge der zugeteilten Rechte bewertet. Nach diesem **Need to Know Prinzip** erhält jeder Nutzer genau so viele Rechte, wie er für die Erledigung seiner spezifischen und aktuellen Aufgabe benötigt. Optimal für ein Konzept der Rechtedelegation wäre es, dieses Need to Know Prinzip vollständig umsetzen zu können. Das andere Extrem bei der Delegation ist die so genannte **Impersonation**, bei welcher der Stellvertreter alle Rechte des Delegierenden übernimmt.

Die ausschließliche Betrachtung der Delegation von Rechten ist für eine Bewertung nicht ausreichend. Es muss auch der (vorzeitige) Widerruf von Rechten berücksichtigt werden. Im Idealfall sollte der Delegierende jederzeit die Möglichkeit besitzen ein von ihm delegiertes Recht beim Delegierten selbst bzw. über die entsprechenden Sicherheitsmechanismen zurückzunehmen bzw. zu entziehen.

0 Es gibt keine Mechanismen zur Delegation von Rechten.

1 Die Delegation erfolgt als Impersonation, d.h. durch Übertragung aller Rechte.

2 Die Delegation ermöglicht die Vergabe abgestufter Rechte.

3 Die Delegation ermöglicht die Realisierung des Need to Know Prinzips mit Möglichkeiten zum Widerruf.

KRITERIUM IV.1.b: Beschränkung des Delegationsrechtes

KRITERIUM IV.1.b: Beschränkung des Delegationsrechtes

Werden Rechte delegiert, muss auch die Möglichkeit der erweiterten Delegation, d.h. einer Delegation durch den Delegierten betrachtet werden. Für die optimale und breite Ressourcen-Nutzung ist es häufig unabdingbar einen Stellvertreter des Nutzers in die Lage zu versetzen sowohl Aufgaben als auch Rechte des Nutzers weiter an Dritte zu delegieren. Entscheidend hierbei ist jedoch die Möglichkeit dieses Delegationsrecht in Art und Ziel zu beschränken.

0 Eine erweiterte Delegation ist nicht realisierbar oder die Beschränkung des Delegationsrechtes ist nicht möglich.

1 Es gibt grobgranulare Mechanismen, um das Ziel der Delegation zu beschränken, d.h. eine ungehinderte Verbreitung der Rechte zu verhindern.

2 Es gibt Mechanismen, um die Art der Rechtedelegation (welche Rechte dürfen weiter delegiert werden) grobgranular einzuschränken.

3 Es gibt feingranulare Mechanismen, um die Rechte, die weiter delegiert werden dürfen, zu beschränken. Und es gibt feingranulare Mechanismen, um das Ziel der Delegation einzuschränken. Die Spezifikation der Beschränkungen wird automatisiert umgesetzt.

KRITERIUM IV.2: Delegation von Policies

KRITERIUM IV.2: Delegation von Policies

Im interorganisationalen Kontext ist davon auszugehen, dass alle Partner eigene Regeln haben, die z.B. Art und Nutzung von Sicherheitsmechanismen, Diensten und Ressourcen oder die Verwendung von Daten regeln. Im Allgemeinen bedarf es einer Möglichkeit diese Policies, z.B. im Zusammenhang mit der Dienstnutzung oder bei der Konfiguration der Sicherheitsmechanismen, formalisiert auszutauschen. Delegation von Policies wird hierbei in folgendem Sinne verstanden: Der Nutzer, der (Teil-) Aufgaben an eine fremde Domäne übermittelt, hat die Möglichkeit eigene Policies bezüglich Dienstnutzung, Sicherheitsanforderungen und -mechanismen, Datenschutz usw. mit an die Ziel-Domäne zu übertragen.

0 Es sind keine Konzepte zur Delegation von Policies vorhanden.

1 Es gibt informelle Abstimmungen über Policies. Oder: Es gibt Mechanismen zum Austausch, aber keine Festlegungen bezüglich der Spezifikation von Policies.

2 Es gibt Methoden zur formalen Beschreibung von Policies (Policysprachen), aber keine etablierten Mechanismen zum Austausch.

3 Es gibt Methoden zur formalen Beschreibung von Policies (Policysprachen) und institutionalisierte Austauschmechanismen.

KRITERIUM IV.3: Delegation von Aufgaben

KRITERIUM IV.3: Delegation von Aufgaben

Die interorganisationale Kooperation lebt davon, dass Aufgaben delegiert werden können. Das Grundprinzip dabei sollte sein, dass derjenige, der am besten für eine bestimmte Aufgabe geeignet ist, diese auch durchführen sollte. Dies gilt natürlich auch für Aufgaben des Sicherheitsmanagements.

0 Eine Delegation von Aufgaben ist nicht vorgesehen.

1 Die Delegation von vorher fest definierten Aufgaben ist möglich. Die Delegation erfolgt nicht automatisiert.

2 Die Delegation allgemeiner Aufgaben, die nicht vorher fest vorgegeben wurden, ist möglich. Die Delegation erfolgt nicht automatisiert.

3 Die automatisierte Delegation allgemeiner Aufgaben ist möglich.

KRITERIUM V: Mapping

KRITERIUM V: Mapping

In interorganisationalen Kooperationen arbeiten immer autonome Organisationen zusammen, d.h. die Quell-Domäne, die bspw. einen Job in eine Ziel-Domäne delegiert, hat eigene Policies und ein eigenes System, um Rechte zu spezifizieren. Die Ziel-Domäne wiederum besitzt ebenfalls eigene Policies und Systeme zur Rechtespezifikation. Beim Übergang bzw. an der Grenze zwischen diesen beiden Domänen muss eine Abstimmung bzw. eine Abbildung zwischen den verschiedenen Policies bzw. Rechten erfolgen. Für den Fall, dass die Policies oder Rechte konfliktär sind, bedarf es Verfahren diese Konflikte aufzulösen. Diese Abstimmung und Konfliktlösung zwischen den konkurrierenden Rechten und Policies wird als Policy- bzw. Rechte-Mapping bezeichnet.

Der Job wird in der Quell-Domäne, abhängig vom Eigentümer, mit bestimmten Rechten ausgestattet. Die Policies, die für die Ausführung des Jobs, die Rückgabe der Daten, die Sicherheitsanforderungen etc. gelten sollen, werden in der Quell-Domäne spezifiziert. Nach der Übertragung in die Ziel-Domäne muss die Identität des Delegierenden ermittelt und ggf. auf lokale IDs abgebildet werden. Der Job erhält, abhängig von lokalen Regeln der Ziel-Domäne, zusätzliche Rechte oder die Menge seiner Rechte wird beschränkt. Auch in der Ziel-Domäne gelten eigene lokale Policies für den Job, die dessen Ausführung betreffen. Ein Mapping zwischen den verschiedenen Policies ist hier notwendig. Und die Rechte des Jobs müssen mit den lokalen Rechten in der Ziel-Domäne abgestimmt und ggf. auf lokale (Ressourcen-)Rechte abgebildet werden.

0 Es sind keine Mechanismen zur Abstimmung und zur Auflösung von Konflikten vorgesehen oder es gibt keine Möglichkeit in Quell- und Ziel-Domäne Rechte und Policies zu spezifizieren.

1 Die Mapping-Regeln werden individuell in jeder Domäne festgelegt. Die Konfliktauflösung erfolgt in der Ziel-Domäne. Eine Nachvollziehbarkeit für den Nutzer ist nicht gegeben.

2 Es gibt VO-globale Festlegungen für Rechte bzw. Policy-Klassen. Konflikte werden durch ein Ordnungsschema gelöst (z.B. Policies und Reche der Ziel-Domäne sind höherwertig).

3 Es gibt VO-globale Festlegungen für Rechte und Policies und technische Verfahren, die diese in der lokalen Domäne auch abbilden können. Für die Konfliktlösung existieren Protokolle bzw. vordefinierte Verfahren.

KRITERIUM VI: Skalierbarkeit

KRITERIUM VI: Skalierbarkeit

Die Skalierbarkeit eines Sicherheitsmechanismus bzw. des entsprechenden Sicherheitsdienstes muss für Grid-Umgebungen mit betrachtet und bewertet werden. Die Skalierbarkeit setzt die Vergrößerung eines Systems ins Verhältnis zum zusätzlichen Aufwand, der für diese Vergrößerung notwendig ist, bzw. ins Verhältnis zu etwaigen Leistungseinbußen, die durch die Vergrößerung entstehen.

Wenn das Verhältnis aus Vergrößerung und Aufwand größer bzw. gleich eins ist, d.h. bei doppelter Größe entsteht maximal doppelter Aufwand, skaliert das System. Diese Skalierungsfunktion ist allerdings nicht notwendigerweise stetig. Häufig gibt es eine kritische Größe und, falls diese überschritten wird, funktioniert das System überhaupt nicht mehr.

Die Skalierbarkeit, die in diesem Kriterium bewertet wird, umfasst drei Dimensionen:

1. Skalierbarkeit von Hardware und Software
2. organisatorische Skalierbarkeit
3. räumliche Skalierbarkeit

Ein System skaliert räumlich, wenn die geographische Verteilung der Komponenten keinen negativen Einfluss auf die Leistung des Systems hat. In vielen Grids wird von einer weltweiten Verteilung der Nutzer ausgegangen. Sicherheitsmechanismen sollen in einem eng begrenzten Gebiet ebenso funktionieren wie weltweit.

Ein System skaliert in Hardware und Software, wenn zusätzliche Hard- und Software Komponenten zum Grid hinzugenommen werden und es dadurch nicht zu überproportionalem zusätzlichen Aufwand oder zu unverhältnismäßigen Leistungseinbußen kommt. Dies gilt auch für die organisatorische Flexibilität, die die Zunahme der organisatorischen Entitäten wie z.B. die Anzahl der VOs, aber auch die Zunahme der Nutzer betrachtet.

0 Der Mechanismus skaliert in keiner Dimension.

1 Der Mechanismus skaliert in einer Dimension.

2 Der Mechanismus skaliert in zwei Dimensionen.

3 Der Mechanismus skaliert in allen drei Dimensionen.

KRITERIUM VII: Flexibilität

KRITERIUM VII: Flexibilität

Die Flexibilität eines Sicherheitsmechanismus ist in interorganisationalen Kooperationen ein wichtiges Kriterium. Einerseits müssen die Möglichkeiten zur Abbildung der verschiedenen Entitäten im Sicherheitsmechanismus bewertet werden. Andererseits spielt aber auch die organisatorische Flexibilität eine Rolle. Beide Teilkriterien der Flexibilität werden mit 2 gewichtet.

KRITERIUM VII: Flexibilität	Gewicht
Entitätenvielfalt	2
Organisationsflexibilität	2

KRITERIUM VII.1: Entitätenvielfalt

KRITERIUM VII.1: Entitätenvielfalt

Grids bilden komplexe organisatorische Verbünde mit einer Vielzahl von Entitäten. Bei der Umsetzung von Sicherheitsrichtlinien werden abhängig von der Entität Sicherheitsattribute zugeordnet oder die Einhaltung bestimmter Sicherheitsrichtlinien gefordert. Ein Sicherheitsmechanismus kann diese technisch allerdings nur umsetzen, wenn es auch eine technische Abbildung für die jeweilige geforderte Entität gibt. Für die Bewertung wird geprüft, ob folgende Entitäten repräsentiert werden können:

- Nutzer
- Organisation
- Ressource
- VO
- Gruppe
- Rolle
- Code / Implementierung

Die Bewertung wird wie folgt spezifiziert.

0 Der Mechanismus kennt keine Entitäten

1 Der Mechanismus unterstützt eine oder zwei Entitäten.

2 Der Mechanismus unterstützt zwei bis fünf Entitäten.

3 Der Mechanismus unterstützt mehr als fünf Entitäten.

KRITERIUM VII.2: Organisationsflexibilität

KRITERIUM VII.2: Organisationsflexibilität

In Grid-Umgebungen lassen sich Sicherheitsdienste und deren Entscheidungsbefugnis in zentrale und lokale Dienste aufteilen. Im ersten Fall wird der Sicherheitsmechanismus entweder zentral installiert oder aber für alle teilnehmenden Organisationen zentral vorgegeben. Auch die Entscheidung bzw. die Regeln, die bei Entscheidungen zu befolgen sind, werden zentral gefällt oder zentral vorgegeben. Das heißt, die zentrale Instanz hat höchste Priorität und Entscheidungsbefugnis.

Im lokalen Fall werden zwar auch i.d.R. einheitliche Sicherheitsmechanismen vorgegeben. Allerdings liegt hier die Entscheidungsbefugnis (organisations–) lokal. Das heißt, auch wenn es globale Absprachen gibt, entscheidet im Zweifelsfall die lokale Policy.

In den letzten Jahren hat sich aus dem Identitäts– und Access Management (I&AM; Identity and Access Management) heraus der so genannte FIM–Ansatz entwickelt [Homm 07]. Dieses föderierte Identitätsmanagement wird in Abschnitt 4.3.3 genauer erläutert. An dieser Stelle soll nur das Prinzip kurz vorgestellt werden. Grundidee von FIM ist es Identitäts– und Authentisierungsentscheidungen, die eine bestimmte Person betreffen, immer in deren Heimatdomäne entscheiden zu lassen. D.h. die Heimatdomäne ist die einzige autoritative Quelle für Informationen über die Benutzer aus dieser Domäne. Dieser Ansatz bietet viele Vorteile in Bezug auf Skalierbarkeit, Sicherheit und Administrierbarkeit. Auch im Grid-Umfeld beginnen sich FIM–Ansätze langsam im Identitätsmanagement und für die Authentisierung zu etablieren (vgl. z.B. das GridShib Projekt [GridShib]).

Wir glauben allerdings, dass sich das Prinzip der Föderation über das Identitätsmanagement und die Authentisierung hinaus generalisieren und auch für andere Sicherheitsmechanismen anwenden lässt. Hier könnte es sogar sinnvoll sein, das Paradigma der alleinigen autoritativen Quelle aufzuweichen und Entscheidungen gemeinsam zu treffen. Deshalb verwenden wir neben einer zentralen und lokalen auch die föderierte Entscheidung als Klassifikationskriterium.

Unter den Gesichtspunkten der Organisationsflexibilität ist es wichtig, dass ein Sicherheitsmechanismus möglichst an alle Organisationsformen angepasst werden kann. Deshalb werden die Organisationsformen zentral, lokal und föderiert als Dimensionenraum begriffen.

0 Der Mechanismus unterstützt ausschließlich eine Dimension.

1 Der Mechanismus ist für eine Dimension entwickelt, lässt sich aber durch Anpassungen für eine andere Dimension nutzen.

2 Der Mechanismus unterstützt zwei Dimensionen.

3 Der Mechanismus unterstützt alle drei Dimensionen.

KRITERIUM VIII: Administrierbarkeit

KRITERIUM VIII: Administrierbarkeit

Die Administrierbarkeit bewertet den Betriebsaufwand eines Mechanismus. Die Frage, die hier beantwortet werden muss, ist, welcher organisatorische und technische Managementaufwand ist insbesondere in der Betriebsphase, aber auch in den Lebenszyklusphasen Implementierung und Abbau eines Sicherheitsdienstes notwendig.

0 Der Betriebsaufwand ist sehr hoch.

1 Der Betriebsaufwand ist hoch.

2 Der Betriebsaufwand ist gering.

3 Der Betriebsaufwand ist sehr gering.

KRITERIUM IX: Sicherheit

KRITERIUM IX: Sicherheit

Der wichtigste Aspekt zur Bewertung von Sicherheitsmechanismen ist die Sicherheit des Mechanismus selbst. Die Bewertung der Sicherheit eines Verfahrens lässt sich einerseits durch das erreichte oder erreichbare Sicherheitsniveau andererseits durch die Möglichkeiten der Zusicherung der Einhaltung bzw. der Mitteilung des erreichten Sicherheitsniveaus an beteiligte Partner bestimmen (Assurance bzw. Quality of Protection (QoP)).

Während das erste Teilkriterium von außerordentlicher Wichtigkeit ist, hängt die Wichtigkeit der Zusicherung bzw. des QoP vom Anwendungsfall ab. In Anwendungsfällen, in denen eine gesetzliche Verpflichtung zum Nachweis der Einhaltung bestimmter Sicherheitsgrundsätze besteht, würde man dieses Kriterium sicher auch mit 4 bewerten. Im allgemeinen Fall ist die Wichtigkeit geringer und wird deshalb mit 1 bewertet.

KRITERIUM IX: Sicherheit	Gewicht
Sicherheitsniveau	4
Zusicherung / Quality of Protection	1

KRITERIUM IX.1: Sicherheitsniveau

KRITERIUM IX.1: Sicherheitsniveau

Das Sicherheitsniveau besagt, wie stark oder schwach ein Verfahren ist, das dem Sicherheitsmechanismus zugrunde liegt, und wie „gut“ sich damit die Sicherheitsanforderung umsetzen läßt. Besteht ein Sicherheitsmechanismus aus mehreren Prozessschritten, kann das Sicherheitsniveau des Gesamtprozesses immer nur so stark sein wie das schwächste Glied in der Prozesskette. Das Gesamt-Sicherheitsniveau der Prozesskette ist also das Minimum der Niveaus aller seiner Teilprozesse.

Für die Bestimmung des Sicherheitsniveaus gibt es verschiedene Vorgehensmodelle und Normen (z.B. IT-Grundschutz des BSI, ISO 17799, ISO TR 13335) oder formalisierte Kriteriensysteme und entsprechende Zertifizierungsverfahren (z.B. Orange Book, ITSEC, Common Criteria u.ä.), welche

das Sicherheitsniveau eines bestimmten Systems (z.B. eine bestimmte Betriebssystem und Hardwarekombination) oder das Sicherheitsniveau einer Organisation bestimmen [TCSEC, ITSEC, CCIT-1, CCIT-2, CCIT-3]. Für die Bewertung des Sicherheitsniveaus einzelner Mechanismen sind diese Verfahren wegen ihrer Komplexität und des unterschiedlichen Fokus nur bedingt geeignet.

Die Bewertung des Sicherheitsniveaus einzelner Mechanismen erfolgt daher häufig mittelbar über eine Risikomatrix, die Schadenshöhe und Eintrittswahrscheinlichkeit gegenüberstellt (vgl. Abbildung 3.3).

Kategorie	Eintrittswahrscheinlichkeit
niedrig	Der Eintritt des Ereignisses ist sehr unwahrscheinlich, es ist ein sehr hoher Aufwand erforderlich, um einen erfolgreichen Angriff durchzuführen.
mittel	Der Eintritt des Ereignisses ist unwahrscheinlich, es ist ein hoher Aufwand erforderlich, um einen erfolgreichen Angriff durchzuführen.
hoch	Der Eintritt des Ereignisses ist wahrscheinlich, es ist nur ein geringer Aufwand erforderlich, um einen Angriff durchzuführen.

Tabelle 3.3: Einstufung von Eintrittswahrscheinlichkeiten

Durch die Eintrittswahrscheinlichkeit wird eine Aussage über die Wahrscheinlichkeit des Eintretens eines Schadens getroffen. In der Risikomanagement-Praxis wird die Eintrittswahrscheinlichkeit mit einem Wert zwischen 0 und 1 abgeschätzt. Eine solche Abschätzung ist jedoch nur bei Ereignissen möglich, bei denen auf statistische Erfahrungswerte zurückgegriffen werden kann (z. B. Ereignisse aus dem Bereich der Versicherungswirtschaft). Zu Ereignissen aus der modernen Informationstechnik fehlen solche konkreten Erfahrungswerte, so dass sich eine qualitative Abschätzung der Eintrittswahrscheinlichkeit in Kategorien etabliert hat. Ebenso wird bei der Abschätzung des Schadensausmaßes verfahren. Die Einstufung der Eintrittswahrscheinlichkeit und des Schadensausmaßes erfolgt hierbei qualitativ anhand der Kategorien niedrig, mittel, hoch. Die Kategorien werden, wie in Tabelle 3.3 und 3.4 angegeben, konkretisiert.

Aus dieser Klassifikation lässt sich die in Abbildung 3.3 dargestellt Matrix ableiten. Ein Sicherheitsmechanismus wird entsprechend der Kategorien in diese Matrix eingeordnet und mit der dort angegebenen Bewertungszahl klassifiziert.

KRITERIUM IX.2: Zusicherung / Quality of Protection

KRITERIUM IX.2: Zusicherung / Quality of Protection

Bevor ein Nutzer einen Sicherheitsdienst in einer fremden Domäne nutzt, sollte er in die Lage versetzt werden, das Sicherheitsniveau und die Umsetzung des Dienstes nach dem Stand der Technik prüfen bzw. bewerten zu können. Das heißt, die Gast-Domäne muss dem Nutzer Informationen über die Sicherheitsmechanismen und deren technische Details — soweit diese zur Beurtei-

Kategorie	Schadenshöhe
niedrig	Die Auswirkung des Schadens ist tragbar. Schäden dieser Kategorie haben keine oder nur geringe Auswirkungen. Die Beseitigung des Schadens ist entweder nicht erforderlich oder mit einfachen Mitteln möglich.
mittel	Die Auswirkung des Schadens ist überschaubar. Schäden dieser Kategorie können von der Organisation mit Aufwand behoben werden. Eine Behebung des Schadens ist erforderlich.
hoch	Die Auswirkung des Schadens ist erheblich oder nicht tragbar. Schäden dieser Kategorie sind nur mit erheblichem Aufwand zu beseitigen und haben erhebliche Konsequenzen. Eine Behebung des Schadens ist unbedingt und unmittelbar erforderlich.

Tabelle 3.4: Definition des Schadensausmaß

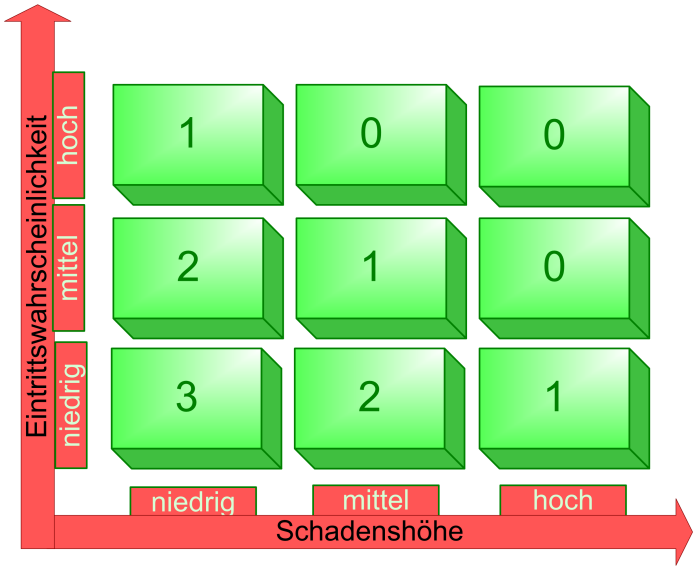

Abbildung 3.3: Bewertung Sicherheitsniveau

lung des Sicherheitsniveaus erforderlich sind — mitteilen. In diesem Zusammenhang wird häufig von Assurance Levels oder von Quality of Protection (QoP) gesprochen.

0 Es sind keine Verfahren vorgesehen, um dem Nutzer Aussagen über das Sicherheitsniveau des verwendeten Dienstes mitteilen zu können.

1 Es werden Informationen über die Sicherheit des Dienstes ausgetauscht. Dies muss jeder Nutzer individuell interpretieren.

2 Eine unabhängige dritte Instanz legt über ein Akkreditierungsverfahren Sicherheits-Levels fest. Jeder Sicherheitsmechanismus kann geprüft und

akkreditiert werden. Der erreichte Sicherheits-Level wird an den Nutzer kommuniziert.

3 Es gibt festgelegte Verfahren, um QoP-Werte des Sicherheitsdienstes zu bestimmen (z.B. auch über eine Akkreditierung) und Protokolle um QoP-Werte zu kommunizieren und automatisiert zu nutzen.

Kapitel 4

Bewertung von Sicherheitskonzepten und -Mechanismen

Inhaltsverzeichnis

Mit den Vorarbeiten aus den vorangegangenen Abschnitten kann in diesem Kapitel eine Bewertung existierender Sicherheitsmechanismen durchgeführt werden. Am Beginn dieses Kapitels werden zuerst die notwendigen Grundlagen der Grid Technologie präsentiert (vgl. Abschnitt 4.1). Das in Abschnitt 2 vorgestellte Rahmenwerk und die dort angegebenen Dienste und ihre Abhängigkeiten dienen dann als strukturbildendes Element für den Rest des Kapitels. Im Rahmenwerk wurden die Sicherheitsdienste in Dienstklassen gruppiert (vgl. Abbildung 4.1). Als erster Schritt wurden die vorgestellten Sicherheitsmechanismen, die Sicherheitsdienste realisieren, innerhalb dieser Dienstklassen gruppiert. Die Dienstklassen bilden also die Struktur der Unterkapitel ab 4.2 für den Rest des Kapitels 4. Innerhalb eines Unterkapitels werden die jeweiligen Sicherheitsmechanismen in Teilabschnitten untersucht. Zu Beginn jedes Teilabschnitts wird der jeweilige Sicherheitsmechanismus und dessen technische Grundlagen kurz vorgestellt. Im Anschluss daran wird der in Kapitel 3 entwickelte Kriterienkatalog angewendet, um den Sicherheitsmechanismus, soweit dies möglich ist, zu bewerten. Für einen schnellen Überblick und für Vergleiche zwischen den Bewertungen verschiedener Mechanismen, werden die Bewertungen mit Hilfe von Netzdiagrammen visualisiert.

Das Kapitel 4 schließt mit einer Defizitanalyse. Dabei werden grundlegende Schwächen von Sicherheitsdiensten in Grids herausgearbeitet.

4.1 Vorbemerkungen zu Grid Technologien

Als technische Basis für viele Sicherheitsmechanismen dienen Grid Middlewares. In diesem Abschnitt werden grundlegende Aspekte dieser Middleware-Technologien eingeführt.

In Abschnitts 4.1.1 werden die grundlegenden Standardisierungsbemühungen sowie die wichtigsten Standards für Grids kurz vorgestellt. Dies ist insofern erforderlich, weil, wie oben dargelegt, Grids in dieser Arbeit als Phänotyp für Virtuelle Organisationen und Föderationen betrachtet werden.

Die technische Umsetzung der Grid-Dienste erfolgt durch eine so genannte Grid Middleware. Die Middleware ist auch in der Lage, die Heterogenität der tatsächlichen Ressourcen und Ausführungsumgebungen zu verschatten. Das heißt, dass die Grid Middleware die Unterschiede in Hardware Plattformen und den verschiedensten Betriebssystemen durch eine standardisierte gemeinsame Dienstzugriffsschnittstelle vereinheitlicht. Im Folgenden werden die wichtigsten Vertreter der Grid Middleware-Technologien, das „Globus Toolkit“ (Abschnitt 4.1.5), „UNICORE“ (Abschnitt 4.1.7) und „LCG/gLite“ (Abschnitt 4.1.9) exemplarisch vorgestellt.

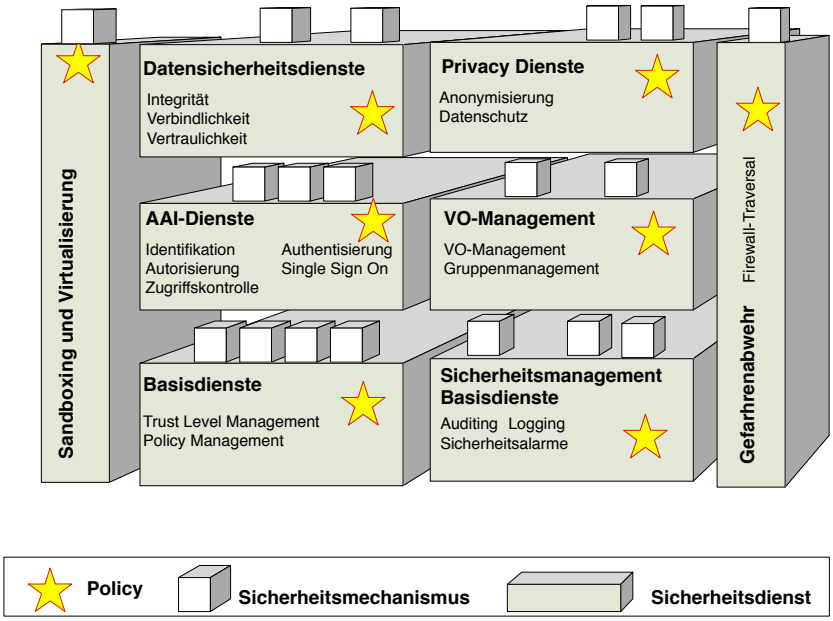

Abbildung 4.1: Rahmenwerk als strukturbildendes Element der Bewertung

Diese drei Middleware-Technologien bilden die Basis für die Sicherheitsdienste und stellen die Plattform für die Implementierung von Sicherheitsmechanismen dar. Die Middleware-Technologien sind jedoch von der Konzeption sehr unterschiedlich und es gibt Sicherheitsmechanismen, die genau auf eine Middleware zugeschnitten sind und auch nur für diese Middleware einsetzbar sind. Deshalb werden die Grundprinzipien der jeweiligen Middleware in diesem Abschnitt vorgestellt.

Der Teilabschnitt schließt mit einer kurzen Beschreibung der Grid Anwendungsszenarios, die diese Arbeit insgesamt geprägt haben.

4.1.1 Standardisierung von Grids

Standardisierung im GGF

Das Global Grid Forum [GGFb] setzt sich aus Nutzern, Entwicklern und Herstellern zusammen. Deren Ziel ist eine weltweite Standardisierung für Grid Computing. Organisatorisch gibt es drei Arten von Gremien innerhalb des GGF: Working Groups zur Entwicklung von Standards, Research Groups zur Diskussion neuer Fragen sowie zur Entwicklung neuer Use Cases und Community Groups, die sich mit bestimmten Funktionsbereichen (z. B. Sicherheit) befassen. Die Working Groups können aus den anderen Gruppen gebildet werden. Das GGF arbeitet ähnlich wie die Internet Engineering Task Force (IETF). In der IETF wird ein Dokument, das als RFC veröffentlicht wird, zum De-facto-Standard. Im GGF heißt das vergleichbare Dokument „proposed recommendation“. Im Juni 2006 haben sich das GGF und die Enterprise Grid Alliance [EGA] zum Open Grid Forum [OGF] zusammengeschlossen. Die EGA, welche überwiegend von Herstellern getragen wurde, konzentrierte sich primär auf die Entwicklung von Grid-Anwendungen. Durch den Zusammenschluss sollen die Interessen gebündelt und gemeinsam vertreten werden.

GGF und EGA bilden OGF

Die wichtigen Standards des GGF, und damit des OGF, sind die Open Grid Service Infrastructure (OGSI) (vgl. Abschnitt 4.1.2) und die Open Grid Service Architecture (OGSA) (vgl. Abschnitt 4.1.4).

Web Services als Basis

Sowohl die Standards als auch viele Middleware-Implementierungen stützen sich auf Web Services ab. Ein Grid Service kann als Web-Service realisiert werden. Ein Web Service ist ein Dienst, der über einen Uniform Resource Identifier (URI) eindeutig identifizierbar und referenzierbar ist. Die Schnittstellen des Dienstes werden mit Hilfe der Extensible Markup Language (XML) spezifiziert. Der Zugriff auf den Dienst erfolgt unter Verwendung von XML-basierten Nachrichten mit Hilfe internet–basierter Protokolle.

Ein Anbieter eines Web Service beschreibt diesen mit Hilfe der Web Service Description Language (WSDL) [CCMW 01], einer XML basierten Spezifikationssprache. Mit Hilfe eines Verzeichnisdienstes, z. B. UDDI (Universal Description, Discovery and Integration) [CHvRR 04] wird der Web Service mit der Dienstbeschreibung und seiner URI beim Verzeichnisdienst registriert. Ein Nutzer kann dann das UDDI-Verzeichnis nach einem für seine Zwecke geeigneten Dienst durchsuchen und diesen dann nutzen. Für den Aufruf des Dienstes wird das Simple Object Access Protocol (SOAP) [Mitr 03, GHM+ 03a, GHM+ 03b] oder XML-RPC (Remote Procedure Call) [Wine 99] verwendet. Beide Protokolle basieren wiederum auf XML. Damit wird einerseits erreicht, dass auf jeder Plattform die Methodenaufrufe dekodiert werden können. Anderseits kann man die verschiedensten Protokolle (z. B. TCP, HTTP, SMTP u. a.) verwenden, um SOAP- oder XML-RPC-Nachrichten zwischen Nutzer und Web-Service zu übertragen.

Die Web Services sind nicht primär für einen menschlichen Nutzer gedacht, sondern für die Nutzung durch andere Web Services oder sonstige Programme, die damit automatisiert Daten austauschen oder Funktionen auf entfernten Rechnern aufrufen. Google bietet beispielsweise Web Services an, um mit

eigenen Programmen automatisiert auf die Suchfunktion von Google zuzugreifen [googleWS].

4.1.2 Open Grid Service Infrastructure (OGSI)

OGSI: Grid Services als Web Service implementieren

Grid Services lassen sich als Web Service implementieren. Die **Open Grid Service Infrastructure (OGSI)** [TCF+ 03] war der erste Versuch dies standardisiert umzusetzen. OGSI beinhaltet eine detailliertere formale Spezifikation von Grid Services. Dazu werden Verhaltensweisen, Schnittstellen und WSDL/XML-Schemata verwendet. Die OGSI erweitert auch die bestehenden Web Services Standards, um deren Defizite im Hinblick auf Grids zu beseitigen. Diese Erweiterungen umfassen die folgenden Punkte:

Erweiterung der WS-Standards

- Zustandsbehaftete und potentiell transiente Dienste. Klassische Web Services sind persistent und zustandslos. Viele Grid Services sind zustandsbehaftet und kurzlebig.
- Mechanismen zum Management des Lebenszyklus von Diensten (Lifecycle Management)
- Dienstinformationen (Zustandsinformationen und Metadaten)
- Asynchrone Benachrichtigungen (Notifications)
- Dienstgruppen
- Erweiterungen der Schnittstellendefinitionen eines Web Services (`portType`-Erweiterungen)
- **Grid Service Handle (GSH)** zur global eindeutigen Identifikation eines Grid-Dienstes und **Grid Service Reference (GSR)** als „Netzwerk-Pointer“, um auf den Grid-Dienst zugreifen zu können. D. h. ein GSH muss auf eine GSR abgebildet werden, um den Dienst verwenden zu können.

Das heißt, ein Grid Service nach OGSI ist ein Web Service mit erweiterten wohldefinierten Schnittstellen. Die Implementierung der erweiterten Schnittstellen erfolgt mittels WSDL `portTypes`.

keine Akzeptanz durch OASIS

Diese Erweiterungen der klassischen Web Services Standards wurden von OGSI proklamiert. Die OGSI wurde vom GGF standardisiert, wohingegen für die Standardisierung der Web Services die OASIS (Organization for the Advancement of Structured Information Standards) [OASIS] verantwortlich zeichnet. Als eine Folge davon wurde OGSI nur von der „Grid Community“ akzeptiert. Aus Sicht der „Web Service Community“ gingen die proprietären Erweiterungen zu weit und wurden nicht akzeptiert. Die vom Global Grid Forum erhoffte Konvergenz zwischen OGSI und den Web Services Standards trat nicht ein. Dies führte auf Seite der Grid Community zur Entwicklung von OGSA (vgl. Abschnitt 4.1.4).

4.1.3 Web Services Resource Framework (WSRF)

WSRF als „echte“ Erweiterung der WS-Standards

Von den Standardisierungsgremien für Web Services wurde, obwohl OGSI als Standard nicht akzeptiert wurde, trotzdem erkannt, dass es einen Bedarf nach den innerhalb von OGSI spezifizierten Erweiterungen gab. Um dennoch eine Konvergenz zwischen der Web Service und der Grid Community zu erreichen, wurde 2004 eine Standardisierungsbemühung ins Leben gerufen. Diese Initiative mündete in eine Gruppe von Standards, die als **Web Services Resource Framework (WSRF)** [OASI 05] bezeichnet werden und von der OASIS standardisiert wurden. Das WSRF ist im Gegensatz zu OGSI integraler Bestandteil der Web Services und kein Aufsatz oder Patch wie OGSI. Funktional betrachtet werden dieselben Erweiterungen wie bereits im Rahmen von OGSI vorgegeben umgesetzt. Mittlerweile hat WSRF eine deutlich höhere Akzeptanz erreicht und OGSI ersetzt.

4.1.4 Open Grid Service Architecture (OGSA)

OGSA: allgemeiner architektureller Rahmen

Die **Open Grid Service Architecture (OGSA)** [FKS+ 05] ist ein allgemeiner architektureller Rahmen zur Beschreibung und Organisation von Grid-Infrastrukturen. Der Standard stellt gewissermaßen ein Modell dar, wie Grids entwickelt werden sollen, gibt aber keinerlei Hinweise, die beispielsweise eine konkrete Umsetzung oder Implementierung betreffen. OGSA verwendet als Basiskonzept eine „Service Oriented Architecture“ (SOA, [CHT 03a, CHT 03b, IBM 05, NiMc]). OGSA adressiert die Standardisierung der wichtigsten Kernkomponenten eines Grids. Das OGSA-Dienst-Framework mit seinen Diensten ist in Abbildung 4.2 dargestellt, wobei die Autoren in [FKS+ 05] bereits anmerken, dass diese Darstellung nicht erschöpfend ist. Die Zylinder repräsentieren individuelle Dienste. Diese Dienste werden nach den Web-Services-Standards implementiert. Dadurch, dass OGSA auf den Standards der Web Services Familie aufbaut, können die spezifizierten Konzepte und Standards wie z. B. SOAP [Mitr 03, GHM+ 03a, GHM+ 03b], WSDL [CCMW 01], WS-Inspection [BBM+ 01], WS-Security [OASI d] usw. sehr einfach integriert werden. Zum Teil werden sie aber auch um Grid-spezifische Aspekte erweitert.

OGSA spezifiziert Dienste, deren Schnittstellen, ihre Semantik bzw. ihr Verhalten und die Interaktionen zwischen den Diensten. Die Interna des Dienstes oder seine Implementierung sind nicht Gegenstand von OGSA. Die Architektur ist weder geschichtet (wie z. B. das OSI Referenzmodell [ISO 7498]) noch objektorientiert, wenn auch viele Konzepte objektorientiert erscheinen. Dienste sind lose gekoppelt und erfüllen ihre Funktion entweder alleine oder in einer Gruppe. Diese Gruppen werden durch Komposition oder über Interaktionen zwischen den Diensten gebildet. Komplexere Dienste können durch eine „Orchestrierung“, d.h. die Kopplung einfacher Dienste mit Hilfe eines Workflows, realisiert werden. Der Standard definiert sieben Dienstklassen:

OGSA Dienstklassen

1. Infrastrukturdienste (Infrastructure Services)

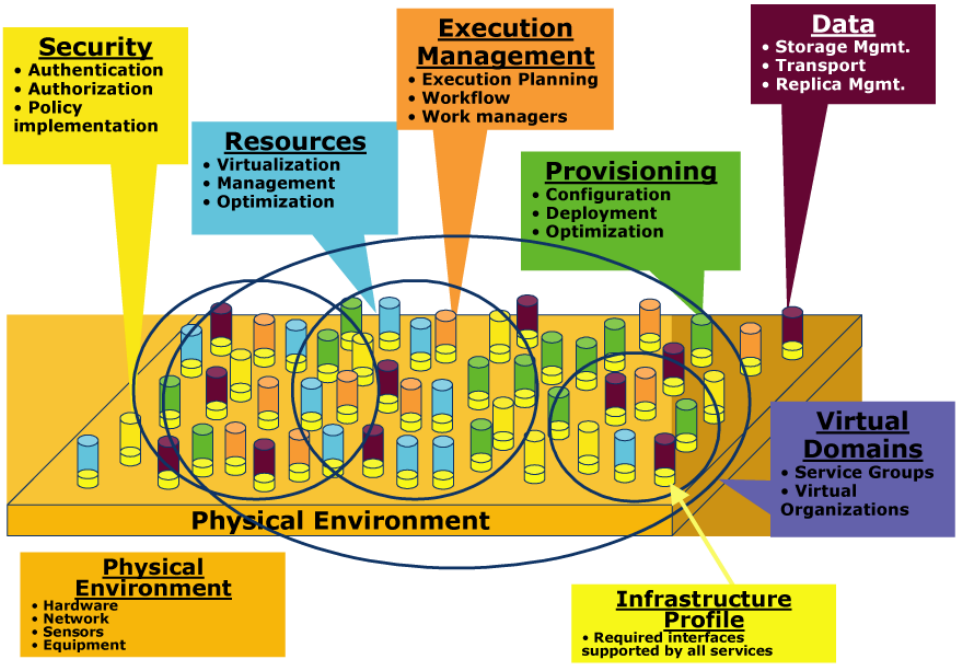

Abbildung 4.2: OGSA-Framework [FKS+ 05]

2. Ausführungsmanagement (Execution Management Services)
3. Datendienste (Data Services)
4. Ressource Management Services
5. Security Services
6. Self Management Services
7. Informationsdienste (Information Services)

Im Rahmen dieser Arbeit sind natürlich insbesondere die Security Services von Interesse.

4.1.5 Globus Toolkit

Die Globus Alliance [globus] leitet die wissenschaftliche Untersuchung und die Entwicklung von Grid-Technologien und will Wissenschaftler und Entwickler zusammenbringen, um Standards ebenso wie die Grid Middleware Globus Toolkit (GT) weiter zu entwickeln und zu verbessern Die wichtigsten Partner dieser Allianz sind das Argonne National Laboratory, die Universitäten von Southern California, Chicago, Edinburgh, das Schwedische Center for Parallel Computers sowie das National Center for Supercomputing Applications (NCSA). Globus Alliance

Historie, Unterschiede zwischen den Globus Versionen

Globus Toolkit 2: pre Web Services

Die erste Version des Globus Toolkit wurde im Herbst 1998 veröffentlicht [Glob 03]. Zu einer Anwendung in breiterem Rahmen kam es aber erst mit der Veröffentlichung von GT2 im Jahre 2002. Mit der Version 3 des Globus Toolkit wurden für die Implementierung der Komponenten Web Services verwendet und es wurde OGSI implementiert. Aus den oben genannten Gründen (vgl. Abschnitt 4.1.2) konnte sich OGSI jedoch außerhalb der Grid Community nicht durchsetzen. Dies führte auch dazu, dass sehr viele Grid-Nutzer nicht bereit waren von GT2 auf GT3 zu wechseln. Deshalb entschloss man sich mit der aktuellen Version GT4, die im Jahr 2005 freigegeben wurde, verstärkt auf das Web Services Ressource Framework zu setzen. GT4 ist auch eine Referenzimplementierung für OGSA.

GT3: OGSI

GT4: OGSA

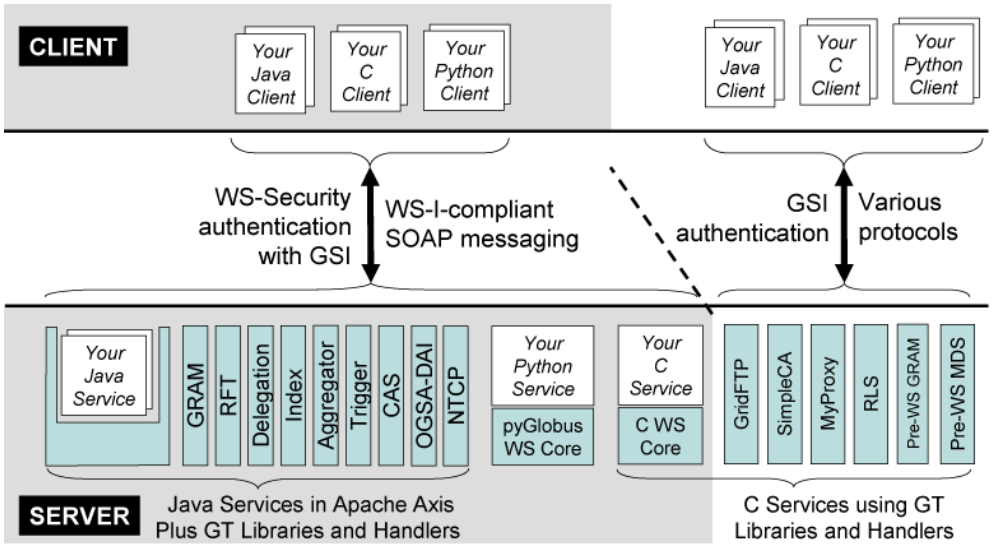

Abbildung 4.3: Kommunikation zwischen Globus Client und Server; Globus Komponenten [Fost 05]

In Abbildung 4.3 sind die Kommunikation zwischen dem Globus Client und verschiedenen Globus Diensten (die auf einem Server laufen) sowie die Komponenten des Globus Toolkit dargestellt. In der Abbildung sind auch die Unterschiede zwischen GT2 und GT3/GT4 verdeutlicht. Die GT2 Komponenten, die noch keine Web Services unterstützen, sind weiß hinterlegt. Die Kommunikation dort erfolgt über Globus-eigene Protokolle. Erst ab Version 3 wurden die Komponenten als Web Services implementiert und die Kommunikation erfolgt hier über Web Service Standards und mittels SOAP.

Globus Web Services

GT4 stützt sich auf verschiedene Web Service Standards ab, d. h. insbesondere, dass Globus Dienste als Web Services implementiert sind. Aus der Sicht eines Clients ist ein Web Service eine über das Netz erreichbare Einheit, die in

der Lage ist SOAP (Simple Object Access Protocol) Nachrichten zu verarbeiten [Mitr 03, GHM+ 03a, GHM+ 03b]. D. h. zur Kommunikation zwischen einem Client und einem Grid Service wird SOAP verwendet, das über verschiedenste Kommunikationsprotokolle (z. B. HTTP, TCP, Mail, u. ä.) übertragen werden kann. Zur Vereinfachung der Implementierung eines Web Services wird unterschieden zwischen:

Container

1. dem Container als Ressourcen–unabhängige Komponente, die in der Lage ist SOAP Nachrichten zu identifizieren und zu empfangen, um dann den für die Bearbeitung notwendigen Programm-Code und evtl. administrative Funktionen aufzurufen.

Web Service Implementierung

2. der eigentlichen Web Service Implementierung, die den Ressourcen–abhängigen Teil enthält, um die Nachricht zu verarbeiten. Der Entwickler muss nur diesen Teil implementieren. Der Container wird von der Middleware bereitgestellt.

Abbildung 4.4 fasst diese Konzepte nochmal zusammen.

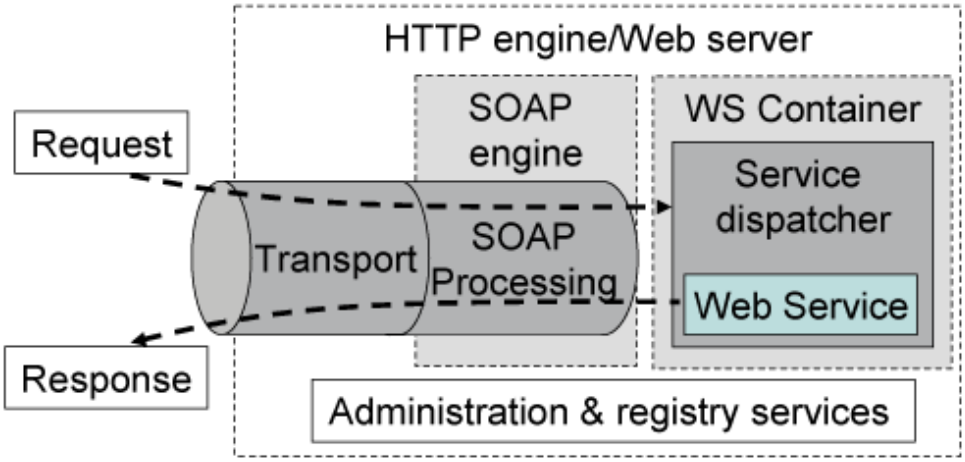

Abbildung 4.4: Web Service Implementierung [Fost 05]

GT4 verwendete Standards

Für die Realisierung stützt sich GT4 auf eine Vielzahl von Standards ab. Diese sind, soweit sie die Realisierung des Containers betreffen, in Abbildung 4.5 zusammengefasst. Globus unterstützt unterschiedliche Container Implementierungen. Ein typischer Anwendungsfall ist es, den Web Service über einen Webserver zugänglich zu machen. Für diesen Fall bietet Globus einen Container an, der auf Tomcat aufsetzt. Daneben gibt es Container für Java und solche für C oder Python.

Globus Komponenten

Globus unterteilt seine Komponenten in fünf Funktionsbereiche: gemeinsame Laufzeitumgebung/Web Service Core, Informationsdienste, Execution- und

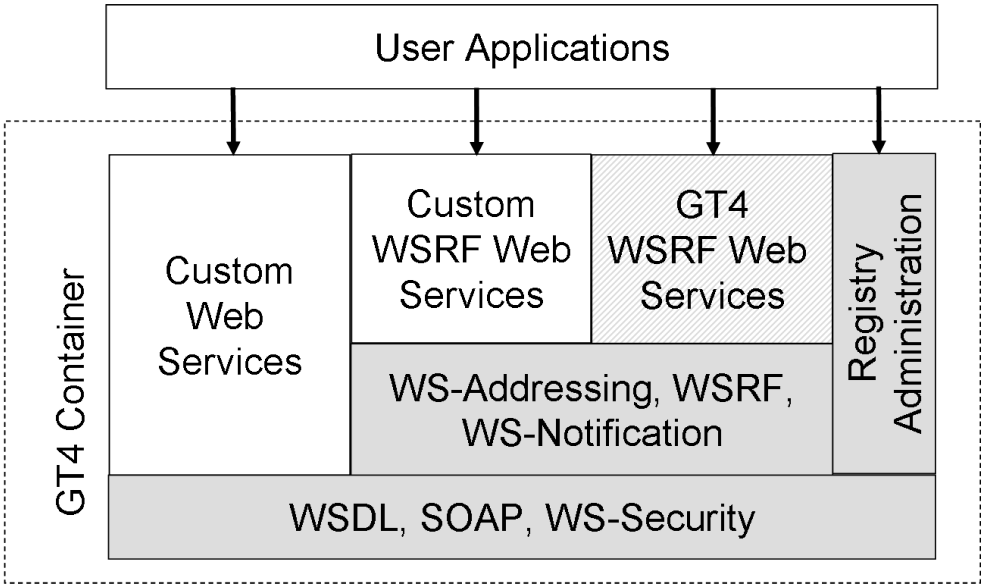

Abbildung 4.5: Globus GT4 Web Service Container und Standards [Scho 06]

Datenmanagement sowie Sicherheit (vgl. auch Abschnitt 4.1.6). Die Abbildung 4.6 beinhaltet alle GT4 Komponenten und ordnet diese den Funktionsbereichen zu. Dabei wird zwischen Core Components, Contribution/Tech Preview und Deprecated Component unterschieden.

Core Components

Die **Core Components** zeichnen sich durch Schnittstellen aus, für welche garantiert wird, dass sie zwischen den inkrementellen Releases (z.B. von GT 4.0 auf GT 4.1) nicht verändern und am besten unterstützt werden.

Contribution/Tech Preview Components

Bei den **Contribution/Tech Preview Components** handelt es sich um solche, die im Moment entwickelt werden und bei denen sich die Schnittstellen auch bei einem inkrementellen Release ändern können.

Die **Deprecated Components** werden nicht mehr unterstützt und in zukünftigen Releases unter Umständen gelöscht.

Von den Globus Komponenten wird im Folgenden GridFTP und RFT vorgestellt.

GridFTP und Reliable File Transfer (RFT)

Übertragung großer Datenmengen

Zur Übertragung großer Datenmengen und zum so genannten Stage In und Stage Out, bei dem Input- und Output-Daten von Berechnungen zwischen Grid-Ressourcen übertragen werden, bietet Globus GridFTP und das darauf aufsetzende Reliable File Transfer (RFT) [MAP 05] die benötigte Funktionalität. Diese Dienste sind aus sicherheitstechnischen Überlegungen insbesondere im Hinblick auf Gefahrenabwehr und Firewall-Fragestellungen von besonderem Interesse (vgl. Abschnitt 4.9 und insbesondere Abschnitt 4.9.3).

RFT bietet, wie der Name schon andeutet, einen zuverlässigen Dienst zur Übertragung von Daten. Zuverlässig meint hier, dass bei RFT Mechanis-

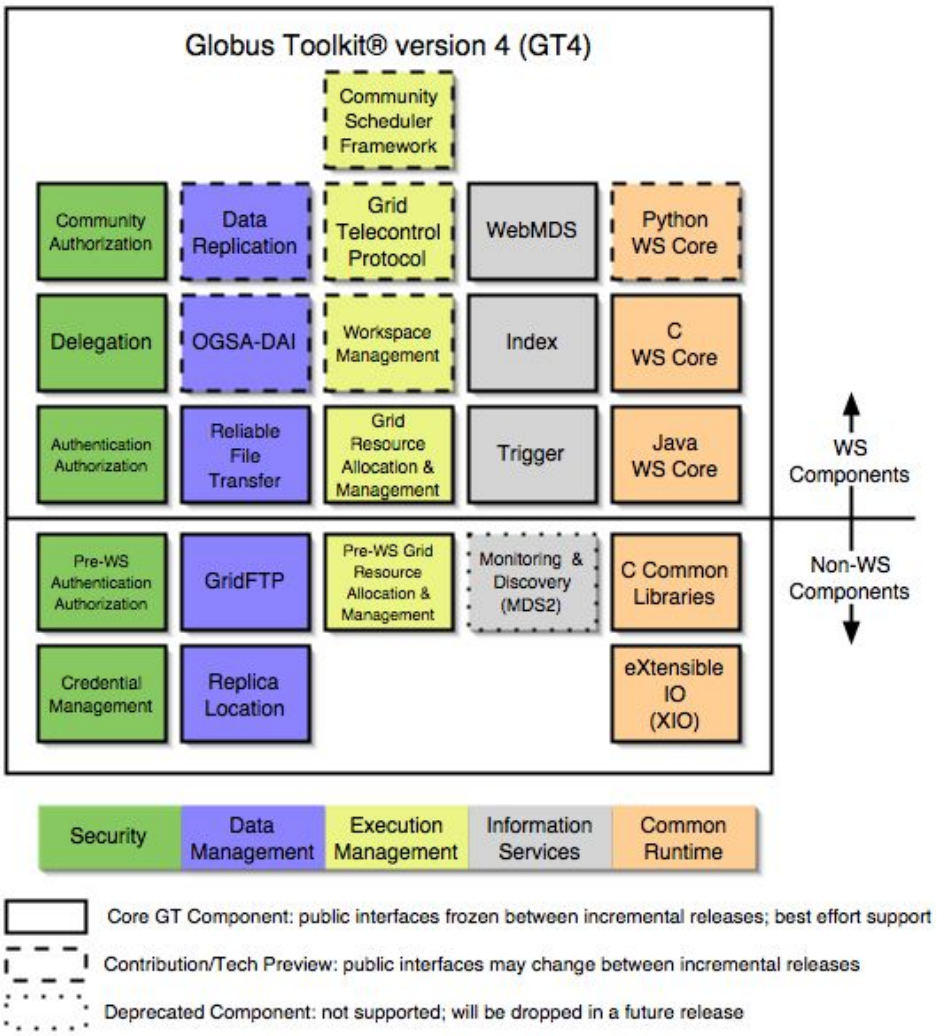

Abbildung 4.6: Komponenten und Funktionsbereiche von GT4 [Glob a]

men implementiert sind, um nach Netzausfällen oder ähnlichen Fehlern die Datenübertragung wieder aufnehmen zu können, ohne die gesamte Datenübertragung von Beginn an neu starten zu müssen. Außerdem ist RFT, im Gegensatz zu GridFTP, Web-Service basiert. Vom sonstigen Funktionsumfang sind GridFTP und RFT äquivalent. Deshalb wird häufig nur von GridFTP gesprochen.

GridFTP ist als Erweiterung des Standard FTP Protokolls [PoRe 85, HoLu 97, Heth 07] konzipiert und unterstützt ebenso wie dieses den Aktiv- und Passiv-Modus. Der Server von GridFTP wird aufgeteilt in ein Front End, das die Kommandos entgegennimmt, und einen oder mehrere Übertragungsknoten, welche die eigentliche Datenübertragung abwickeln. Durch diese konzeptionelle Aufteilung sind weitere Betriebsarten möglich, die bei FTP nicht realisierbar sind (vgl. auch Abbildung 4.7):

GridFTP Betriebsarten

1. Beim **parallelen Datenverkehr** werden für die Übertragung der Daten mehrere TCP-Verbindungen parallel geöffnet.

2. Beim **Stripped Datenverkehr** wird die Datenübertragung gleichzeitig von mehreren Servern durchgeführt, von denen jeder wieder im parallelen Verfahren Daten übertragen kann.

3. Beim **Third Party Transfer** initiiert der Client die Datenübertragung zwischen zwei von ihm unabhängigen Servern. Dazu sendet er GridFTP Kommandos an die beiden Server. Die Datenübertragung erfolgt zwischen den Servern, ohne dass ein Kontrollkanal zwischen ihnen bestehen muss.

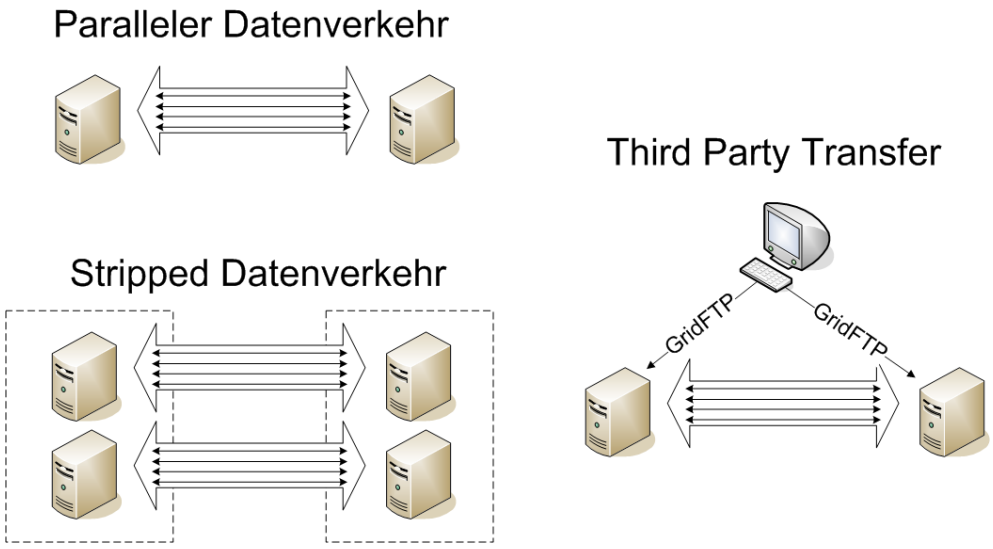

Abbildung 4.7: GridFTP Betriebsarten

GridFTP wurde, wie man auch insbesondere an den ersten beiden Betriebsmodi sieht, primär zur Erhöhung der erreichbaren Übertragungsbandbreite konzipiert. GridFTP ist im Hinblick auf Sicherheitsfragen insbesondere bezüglich der Firewall-Problematik von Interesse. Diese Aspekte werden im Abschnitt 4.9.3 untersucht.

4.1.6 Globus: Grid Security Infrastructure (GSI)

GSI als Sicherheitsinfrastruktur von GT4

Bei den Komponenten und Funktionsbereichen von GT4 (vgl. Abbildung 4.6 auf S. 65) werden für den Funktionsbereich Sicherheit die Teilbereiche Authentisierung, Autorisierung, Community-Authentisierung, Delegation und Credential Management genannt. Die systematischen Überlegungen zum Thema Security gehen bis ins Jahr 1998 zurück [FKTT 98]. In dieser Arbeit wurde bereits das Grundkonzept des User- und Ressource-Proxy vorgestellt, das in 4.2.10 näher beschrieben wird. Im Jahr 2000 beschäftigte man sich mit einer nationalen Authentisierungsinfrastruktur und verwendete das Konzept des User Proxies. In dieser Arbeit war als Ziel die Entwicklung einer Grid Security Infrastructure (GSI) angegeben [BWE+ 00, Glob 02]. Diese

und weitere Überlegungen mündeten dann in die **Grid Security Infrastructure (GSI)**, die weiterentwickelt und auch in GT4 integriert wurde [Welc 05]. GSI beschäftigt sich mit der Vertraulichkeit der Kommunikation, der Authentisierung, der Delegation sowie der Autorisierung. Die verwendeten Standards sind in Abbildung 4.8 dargestellt. Im Folgenden werden die Konzepte und Komponenten der GSI vorgestellt und in den Abschnitten 4.2 ff. werden die Sicherheitsmechanismen von Globus detaillierter dargestellt und bewertet.

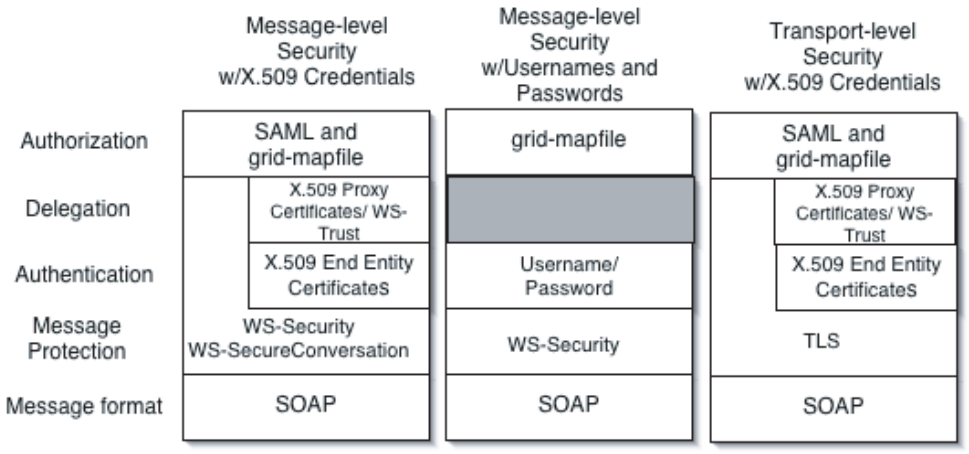

Abbildung 4.8: Sicherheitsstandards in GSI [Welc 05]

4.1.7 UNICORE

Die Basis für UNICORE wurde in einem vom Bundesministerium für Bildung und Forschung geförderten Verbundprojekt gelegt. Die Projektlaufzeit war von August 1997 bis Dezember 1999 [Erwi 00]. Im Anschluss daran wurde die Weiterentwicklung von UNICORE in einer zweiten Phase vom Januar 2000 bis Dezember 2002 als UNICORE Plus gefördert [Erwi 03].

UNICORE steht für Uniform Interface to Computing Ressources und bietet einen nahtlosen, sicheren und intuitiven Zugang zu verteilten Hochleistungsrechnern. Vereinfacht gesagt kann ein Nutzer mit Hilfe von UNICORE Batch Jobs oder ganze Prozessgruppen auf den verschiedenen verteilten Hochleistungsrechnern absetzen.

Architektur und Komponenten

Aus Sicht eines Nutzers stellt sich UNICORE wie eine Drei-Tier Architektur dar. UNICORE besteht aus den folgenden Schichten:

1. Ein UNICORE Client, der auf dem Rechner des Benutzers ausgeführt wird.
2. Eine oder mehrere **UNICORE Grid Sites (Usite)**, zu denen sich der Cli- Usite

ent verbinden kann und die Ressourcen anbietet. Jedes Hochleistungsrechenzentrum bildet eine Usite.

Vsite

3. Innerhalb einer Usite können die Ressourcen so genannten **Virtual Sites (Vsites)** zugeteilt werden.

In Abbildung 4.9 sind die architekturellen Komponenten, die im folgenden näher erläutert werden, dargestellt. Die Abbildung beinhaltet zwei Usites (Site A und B) und drei Vsites (zwei in Site A, eine in B).

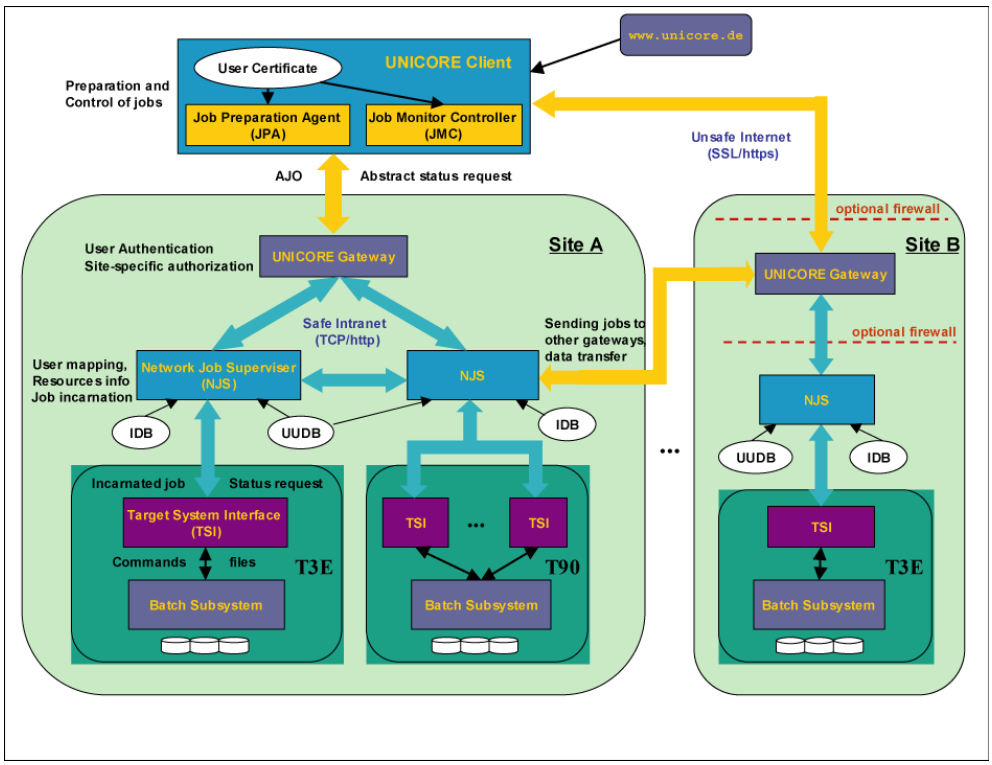

Abbildung 4.9: UNICORE Architektur [Erwi 03]

Client und Job Modell

Der Client ist die Schnittstelle, über die der Nutzer UNICORE bedienen und verwenden kann. Die Jobs werden mit Hilfe des **Job Preparation Agent (JPA)** entwickelt und in Form von **Abstract Job Objects (AJO)** beschrieben und übertragen. Ein AJO kann ein serialisiertes Java Objekt oder eine XML Beschreibung sein. Mit Hilfe des **Job Monitor Controllers (JMC)** kann der Job überwacht werden.

UNICORE Gateway

Das **UNICORE Gateway** stellt den einzigen Zugangspunkt in eine Usite dar. Jede Usite muss ein Gateway betreiben. Das Gateway authentisiert den Nutzer und identifiziert die Netzverbindung zwischen Client und Gateway.

NJS

Eine Vsite wird durch die beiden Komponenten **Network Job Supervisor (NJS)** und dem Target System Interface (TSI) gebildet. NJS autorisiert die Jobs und verwaltet alle UNICORE Jobs der entsprechenden Vsite. Aus dem Abstract Job Object werden die konkreten Jobs für die tatsächliche Ausführungsumgebung (z. B. eine bestimmte Großrechnerplattform) erzeugt.

Job–Zustand	Beschreibung
Successfull	Der Job wurde erfolgreich beendet. In einer Job-Gruppe wurden alle Jobs erfolgreich beendet.
Failed	Ein Job wurde fehlerhaft beendet. In einer Job-Gruppe wurde wenigstens ein Sub Job fehlerhaft beendet.
Pending	Der Job wartet auf Ausführung.
Queued	Der Job wird im Ziel Batch–System in eine Warteschlange (Queue) gestellt.
Executing	Der Job wird ausgeführt.

Tabelle 4.1: UNICORE: mögliche Job-Zustände

IDB

TSI

Dieser Vorgang wird als **Incarnation** bezeichnet. Diese Umsetzung der Jobs aus dem AJO wird vom NJS mit Hilfe der Informationen aus der **Incarnation Data Base (IDB)** durchgeführt. Die Jobs werden dann an das **Target System Interface (TSI)** übergeben. Das TSI ist für die spezifische Hardware Plattform implementiert und steuert die Abarbeitung der inkarnierten Jobs unter Verwendung der lokalen Batch Systeme.

Job-Modell

Job-Gruppen, Jobs, Tasks

In UNICORE besteht ein Job aus einem oder mehreren Tasks. Die Jobs können selbst wieder in Job-Gruppen organisiert werden. Eine Job-Gruppe kann wieder Job-Gruppen oder einfache Sub–Jobs enthalten. Die Jobs in einer Gruppe können entweder unabhängig voneinander ausgeführt werden oder der Benutzer kann zeitliche Abhängigkeiten zwischen den Jobs spezifizieren. Im letzten Fall stellt UNICORE sicher, dass ein Job nur ausgeführt wird, wenn alle seine Vorgänger erfolgreich beendet wurden und alle notwendigen Daten, die er zur Verarbeitung braucht, auch zur Verfügung stehen. Im Standard–Fall werden die Jobs und Job-Gruppen ohne eine bestimmte Ordnung ausgeführt. Wenn beim TSI genügend Ressourcen zur Verfügung stehen, werden die Jobs dann parallel ausgeführt.

Wenn innerhalb von UNICORE von einem „Job“ gesprochen wird, so sind damit sowohl einfache Jobs als auch komplexe Job-Gruppen gemeint. In diesem Sinne sind in Tabelle 4.1 die Zustände der Jobs zusammengefasst.

Zuordnung zu Vsite

keine Ortstransparenz

aber Hardware-Abstraktion durch TSI

Jeder UNICORE Job muss einer bestimmten Vsite zugeordnet werden. Der Job kann Sub–Jobs oder Sub–Job–Gruppen beinhalten, die selbst wieder auf anderen Vsites ausgeführt werden. Trotzdem muss der Nutzer für jeden Job eine Vsite in der Spezifikation festlegen. Das heißt, UNICORE kennt kein Konzept der Orts–Transparenz. Mit Hilfe des TSI wird allerdings eine Abstraktionsschicht zur Verschattung der konkreten Hardware zur Verfügung gestellt.

Für jeden Job wird innerhalb des ausführenden Systems Speicherplatz für temporäre Dateien angelegt. Dieser **Uspace (UNICORE File Space)** besteht nur während der Ausführung des Jobs. Alle Daten, die ein Job benötigt,

müssen aus dem Dateisystem auf dem Zielsystem (**Xspace, Unix File Space**) oder dem des lokalen Client Rechners (**Nspace; No UNICORE Space**) geladen werden.

Ressourcen Modell

Ressourcen Modell aus Höchstleistungsrechnen

Das Ressourcen Modell von UNICORE lehnt sich sehr stark an Hochleistungsrechensysteme mit ihren Batch–Systemen an. Für jede Task innerhalb eines Jobs müssen die benötigten Ressourcen spezifiziert werden. Dabei sind die folgenden Parameter anzugeben:

Spezifikation benötigter Ressourcen

- Anzahl der Nodes
- Anzahl der Prozessoren pro Node
- Speicher (pro Node)
- CPU Zeit (Semantik abhängig vom konkret verwendeten System)
- Plattenplatz
- Priorität
- Anforderungen an bestimmte Software, Bibliotheken oder Anwendungen

Für jede Vsite werden vom Administrator die Ressourcen in der Incarnation Database (IDB) beschrieben. Mit Hilfe dieser Informationen überprüft der UNICORE Client 4.1.7, ob die Anforderungen von einer Vsite erfüllt werden können.

Kommunikationsmodell

eigenes Protokoll: UPL

Die Kommunikation zwischen UNICORE Client und den Server Komponenten erfolgt über ein eigenes Protokoll **UPL (UNICORE Protocol Layer)** mit eigenen Protokollprimitiven. UPL setzt auf SSL auf, um Vertraulichkeit, Integrität sowie eine Authentisierung der Kommunikation zu erreichen.

Job-Migration über serialisierte Java-Objekte

Für die Übertragung von Dateien zwischen Nspace und Uspace oder zwischen zwei Uspaces in verschiedenen Vsites werden serialisierte Byte–Ströme verwendet. Header Informationen der Dateien (wie z. B. Größe, Dateiinhalt, u. ä.) werden als serialisierte Java Objekte gefolgt vom eigentlichen Inhalt, den serialisierten (und ggf. komprimierten) Daten, übertragen.

UNICORE Client

UNICORE Client als einzige Schnittstelle

Die Schnittstelle für den Endbenutzer ist der UNICORE Client. Der Client bietet folgende Funktionalität:

- Entwicklung und Erzeugung von UNICORE Jobs

- Überprüfung, ob ein erzeugter Job korrekt ist und auf einer bestimmten Ziel-Site ausgeführt werden könnte.
- Submission von Jobs und Job-Gruppen
- Überwachung des Lebenszyklus des Jobs
- Abfrage der Ergebnisse von Jobs, die beendet wurden.

In Abbildung 4.10 ist der Client dargestellt. Ganz links sieht man das User Interface für die Job Preparation und darunter für das Job Monitoring. In der Mitte sind die Usites (LRZ und Pallas) sowie die Vsites innerhalb von Pallas angegeben. In der Grafik rechts ist der Abhängigkeitsgraph für die Job-Gruppe dargestellt. Abbildung 4.11 zeigt die Entwicklung eines Job-Scripts

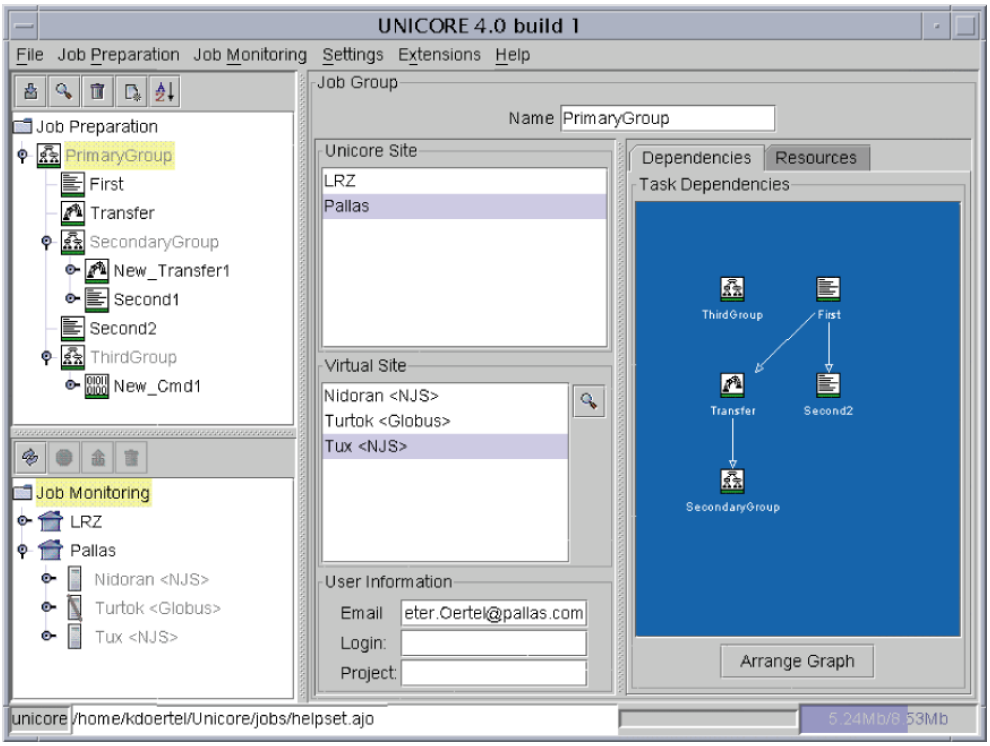

Abbildung 4.10: UNICORE Client: Job-Hierarchie [Erwi 03]

mit Hilfe des UNICORE Clients.

4.1.8 Überblick über Sicherheitsaspekte von UNICORE

Sicherheitsarchitektur als Bestandteil der Entwicklung

Bei der Entwicklung von UNICORE wurden sehr früh Überlegungen zur Sicherheit angestellt. Ein Konzept für eine Sicherheitsarchitektur war Bestandteils des Projektes und elementarer Teil von UNICORE [Erwi 00]. Dabei werden Dienste für die vertrauliche Kommunikation, die zweifelsfreie Authentisierung und die Autorisierung spezifiziert und umgesetzt. Auch die architekturellen Komponenten und deren Platzierung innerhalb des verteilten System

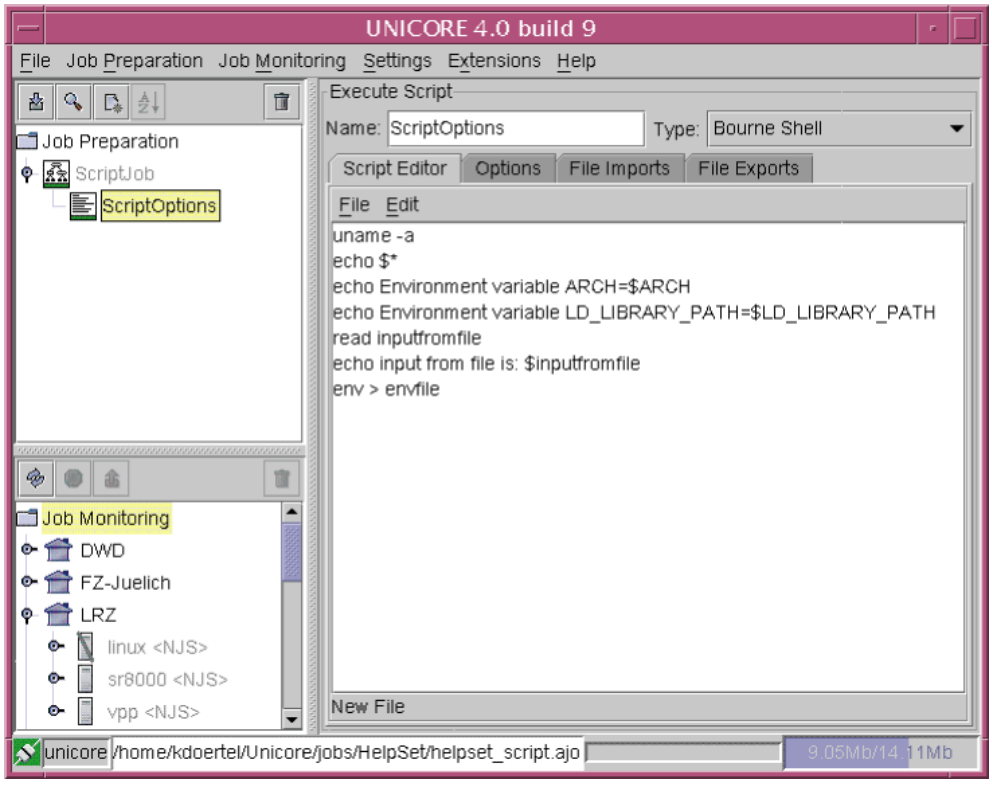

Abbildung 4.11: UNICORE Client: Erzeugung eines Job [Erwi 03]

„Grid“ tragen mit zur Sicherheitsarchitektur bei (vgl. Abb. 4.9). Einziger Zugangspunkt zu einer UNICORE Site ist das UNICORE Gateway. Diese Komponente ist entsprechend zu schützen.

Sehr früh wurden auch bereits Fragen des Perimeterschutzes und Firewalls betrachtet. So kann das Gateway in einer demilitarisierten Zone (DMZ) platziert werden. Das Gateway führt auch die Authentisierung durch. Die NJS sind für die Autorisierung verantwortlich. Durch das Konzept der VSite lassen sich innerhalb einer UNICORE Site weitere Zonen bilden, die auch im Sinne abgestufter Sicherheitszonen genutzt werden können.

In den folgenden Abschnitten (ab Kapitel 4.2) werden die Sicherheitsmechanismen von UNICORE detaillierter dargestellt und bewertet.

4.1.9 LCG / gLite

Large Hadron Collider

Am CERN in Genf wird derzeit ein neuer Beschleuniger, der **Large Hadron Collider (LHC)** in Betrieb genommen. Auf dem LHC sollen vier Gruppen wissenschaftlicher Experimente aus dem Bereich der Elementarteilchenphysik durchgeführt werden (vgl. Abschnitt 4.1.11). Für die Verteilung und Verarbeitung der riesigen dabei anfallenden Datenmengen werden Grid-Infrastrukturen entwickelt und verwendet. Diese Grid-Aktivitäten werden unter dem Begriff des Large Hadron Collider Computing Grid (LCG) zusam-

mengefasst [LHC].

Im Rahmen mehrer Forschungsprojekte wurden verschiedene Grid Middlewares entwickelt. In den Jahren 2002 und 2003 begann im Rahmen des DataGrid [DataGRID] Projektes die Entwicklung der LCG Middleware. In den Jahren 2003 und 2004 wurden LCG-1 und LCG-2 herausgegeben. Als Folgeprojekt des DataGrid begann im Frühjahr 2004 **EGEE (Enabling Grids for E-SciencE)** [EGEEa, EGEEc, EGEE-Tec] und im Jahr 2005 wurde parallel zu LCG die Middleware gLite etabliert. Mit der Veröffentlichung der Version 3.0.0 von gLite im Mai 2006 sollte die parallele Entwicklung von zwei verschiedenen Entwicklungszweigen wieder in einen zusammengefasst werden, d.h. LCG Version 2.x soll in gLite aufgehen. Aus diesem Grund wird im folgenden nur gLite vorgestellt.

LCG und gLite als parallele Entwicklungslinien

Als Basis von gLite wird Globus GT2 (vgl. Abschnitt 4.1.5) verwendet und GT2 wurde um spezifische gLite Dienste erweitert.

Globus GT2 als Basis von gLite

Im April 2006 startete auch EGEE II, das die Entwicklungen aus EGEE verstetigen, die entstandene Infrastruktur erweitern und konsolidieren soll. Außerdem sollen neben der Elementarteilchenphysik weitere Disziplinen erschlossen werden.

gLite Dienste

Abbildung 4.12 stellt die Dienste dar, die von LCG und gLite angeboten werden [EGEE 04a, EGEE 05a, EGEE 04b]. Neben der eigentlichen Nutzfunktionalität in den „Job Management Services“ und den „Data Services“ gibt es Dienste für den Zugriff („Access Services“), für das Monitoring („Information & Monitoring Services“) sowie Sicherheitsdienste („Security Services“).

Grid-Nutzer erhalten über die **Access Services** Zugriff zu den Grid Ressourcen. Die **Sicherheitsdienste** umfassen Authentisierung, Autorisierung und Auditing Dienste sowie Dienste für Vertraulichkeit. Die **Informations- und Monitoring Dienste** bieten Mechanismen um Informationen bereitzustellen und auf diese zuzugreifen, um diese für Monitoring Zwecke zu nutzen, oder, um Informationen über Ressourcen im Grid bekannt zu machen.

Im Kontext von EGEE wurde für alle Dienste ein Gültigkeitsbereich (Scope) für die Durchsetzung der Dienst-Policies festgelegt (vgl. Tabelle 4.2). Es wird zwischen Nutzer (User), Domäne (Site), VO und Global unterschieden. Global meint dabei Multi–VO Szenarien. Kombinationen der Gültigkeitsbereiche sind möglich, so können bspw. Accounting Policies Global, für eine VO oder auch für die jeweilige Domäne spezifiziert werden.

Gültigkeitsbereich für Policies (Scope)

Die unter den Daten– und Job Management Diensten aufgeführten Komponenten werden im folgenden Abschnitt näher erläutert.

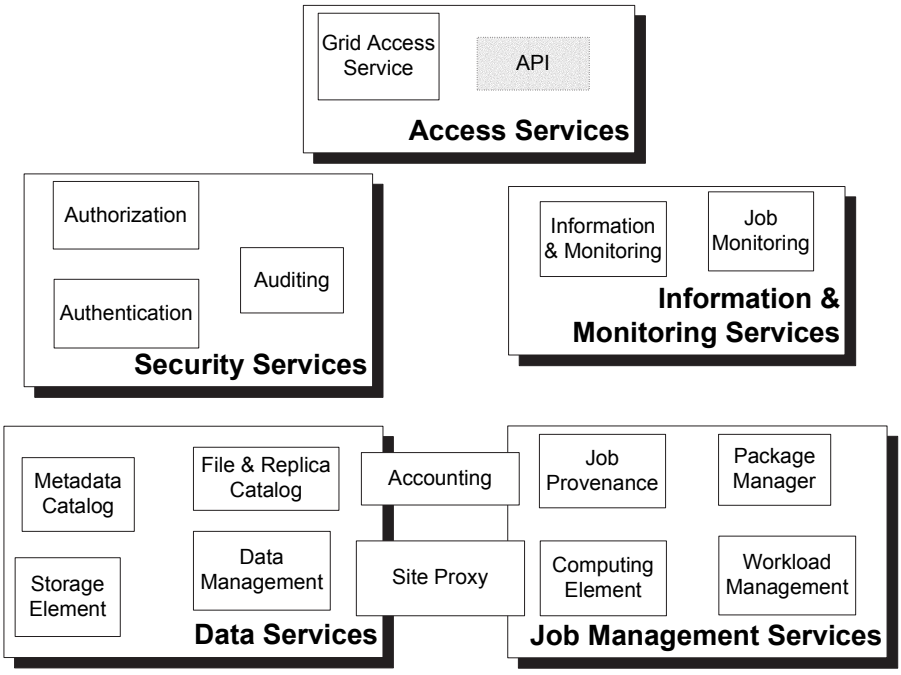

Abbildung 4.12: gLite Dienste [EGEE 04b]

Architekturelle Komponenten

Sowohl LCG als auch gLite basieren auf Globus Version 2 (vgl. Abschnitt 4.1.5) und sind damit weder OGSA-konform noch primär Web Service-basiert. Allerdings soll im Rahmen von gLite eine sanfte Migration hin zu Web Services erfolgen.

Computing Element

Worker Node

Local Ressource Management System

Ressourcen in gLite werden durch **Computing Elements (CE)** repräsentiert. Das Computing Element dient primär dem Job Management, d.h. zur Submission und zur Kontrolle der Jobs und steht stellvertretend für eine Menge oder ein Cluster von Rechenknoten (so genannten **Worker Nodes (WN)**), die mit Hilfe eines **Local Ressource Management System (LRMS)**, z.B. PBSPro [Alta],Torque[Clus] usw., verwaltet werden. Ein CE kann auch heterogene Ressourcen umfassen, die unterschiedliche Hardware oder auch heterogene Software Konfigurationen besitzen. In diesem Fall sorgt ein Ressourcen-Managementsystem dafür, dass die Jobs auf die Ressourcen verteilt werden, die am besten zu den vom Nutzer spezifizierten Anforderungen passen.

Die Job-Submission kann beim CE sowohl im Push- als auch im Pull-Mode durchgeführt werden. Im Push-Mode wird der Job aktiv an das CE übertragen, im Pull-Mode fordert das CE beim Workload Management Service Jobs an und signalisiert damit freie Kapazitäten.

CE beschreibt konkrete Ressource

Neben den Job Management Funktionen liefert ein CE Informationen, welche Eigenschaften des CE beschreiben. Dazu gehören die Charakteristika des CE (z.B. Art und Anzahl der Ressourcen, deren Hardware und Software-Konfiguration), der Status des CE (z.B. Anzahl der freien und genutzten

Service	Scope
Accounting	global, VO, site
Auditing	VO, site
Authentication	global
Authorization	VO, site
Catalogs	VO
Computing Element	VO, site
Data Management	VO, site
Gird Access Service	VO
Information & Monitoring	global, VO, site
Job Monitoring	user, VO
Job Provenance	user, VO
Package Manager	VO
Storage Element	VO, site
Workload Management	VO
Site Proxy	global, VO, site

Tabelle 4.2: Scope der gLite Dienste [EGEE 04a]

Ressourcen, Anzahl der wartenden und rechnenden Jobs) und die Policies, die auf den Ressourcen gelten (z.B. Liste der berechtigten VOs oder Nutzer). Die Informationen werden im Push-Mode mit Hilfe des Information Service bekannt gemacht. Im Pull-Mode werden die Informationen über die CE-Eigenschaften in die Nachricht eingebettet, mit der das CE seine Verfügbarkeit für Rechenaufträge beim Workload Management Service bekannt macht.

Architektur eines CE

Abbildung 4.13 repräsentiert die interne Architektur eines Computing Elements. Das **Computing Element Acceptance (CEA)** nimmt die Job-Submissionen entgegen. Die **AuthZ Auditing** Komponente entscheidet, ob der Nutzer berechtigt ist den Job abzusetzen. Im positiven Fall wird der Job an das **Job Management** übergeben. Hier wird geprüft, ob der Job mit den vorhandenen Ressourcen ausgeführt werden kann, und dann an das LRMS übergeben. Die konkrete Instanz des LRMS wird dabei durch eine Abstraktionsschicht verschattet. Der **CE Monitor (CEMon)** dient der Benachrichtigung der Nutzer und als Schnittstelle für Job Monitoring Systeme.

Workload Management Service

Mit dem **Workload Management Service (WMS)** bietet gLite einen Ressource Broker, der die Verteilung von Jobs auf passende Computing Elements übernimmt. Für diese Entscheidung verwendet WMS die in der Job-Beschreibung spezifizierten Anforderungen und Präferenzen. Die Entscheidung, welche Ressource genutzt werden soll, ist Ergebnis eines Vergleichsprozesses (**Matchmaking Process**) zwischen Submissionsanfragen und verfügbaren Ressourcen. Der WMS implementiert zwei Scheduling Policies. Bei der gierigen Strategie (**Eager Scheduling**) versucht WMS den Job so schnell wie möglich an eine Ressource zu verteilen. Auf der Ressource wird der Job dann in der Regel in eine Queue des LRMS gestellt, die entsprechend der lokalen Queuing Policy abgearbeitet wird. Bei der trägen Strategie

Eager Scheduling

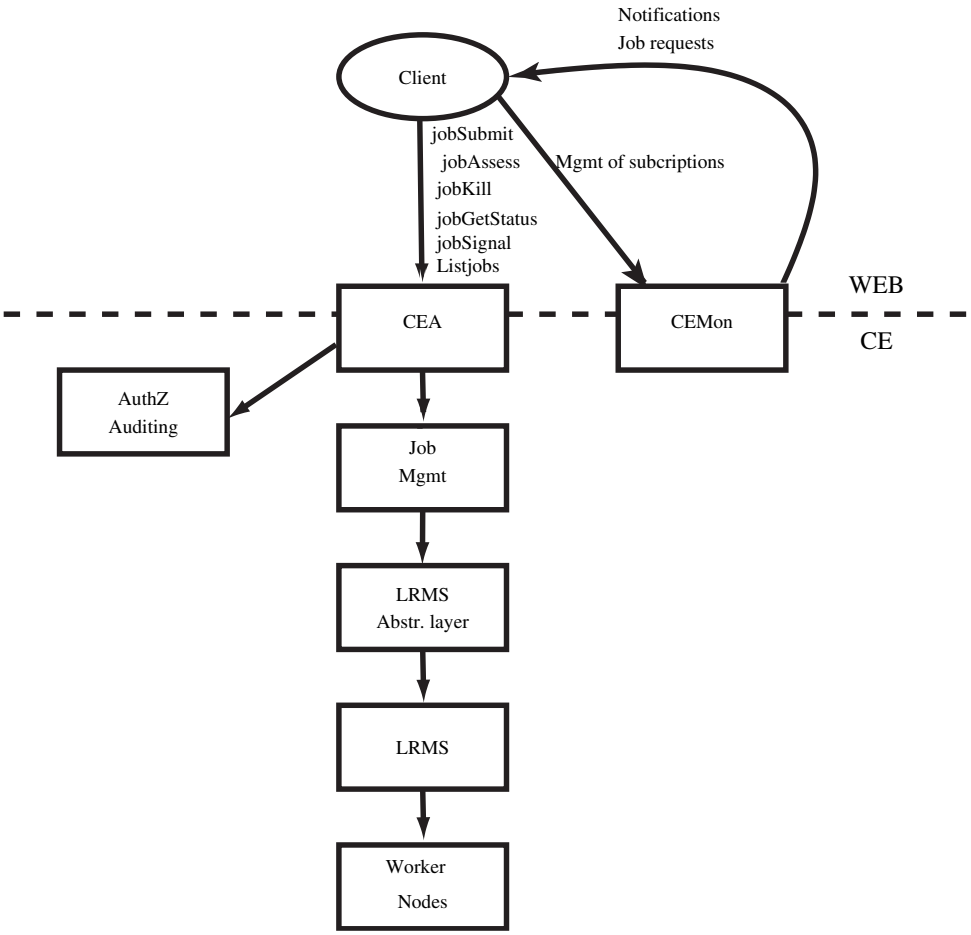

Abbildung 4.13: Architektur des Computing Element [EGEE 05a]

Lazy Scheduling

(**Lazy Scheduling**) behält der WMS den Job so lange bis eine Ressource verfügbar wird, die den Job sofort ausführen kann.

Überwachung durch L&B

Für die Überwachung und Verfolgung von Jobs wird die **Job Logging and Bookkeeping (L&B)** Komponente des WMS verwendet. L&B arbeitet Event basiert. Wichtige Punkte der Job-Ausführung, z.B. Submission des Jobs, Zuweisung zu einem CE, Start und Stopp der Ausführung usw., werden von verschiedenen WMS als Event bei einem L&B Bookkeeping Server verarbeitet und dem Nutzer zur Verfügung gestellt. Der L&B wird in Abschnitt 4.7.1 als Basisdienst für das Sicherheitsmanagement bewertet.

4.1.10 Sicherheitsüberblick für EGEE bzw. LCG

Auch bei EGEE war eine Sicherheitsarchitektur früh ein integraler Bestandteil des Projektes und eine globale Sicherheitsarchitektur wurde spezifiziert [EGEE 04d]. Diese Spezifikation befasst sich mit Fragen der Authentisierung, dem Schlüsselmanagement, der Autorisierung, mit Logging und Auditing und der Netzsicherheit.

Besonders hervorzuheben ist, dass bereits hier Überlegungen zu Sandboxing angestellt wurden (vgl. auch Abschnitt 4.6). Allerding beschränken sich die Ansätze auf Betriebssystem-Container und dynamische Accounts. Diese Ansätze werden in Abschnitt 4.6.2 untersucht und neueren Verfahren gegenüber gestellt. Im Rahmen der Spezifikation der globalen Sicherheitsarchitektur wurde im Abschnitt „Network Isolation“ ein Proxy beschrieben, der in der Lage sein sollte temporär Netzverbindungen zuzulassen. Dazu soll das Autorisierungs-Framework der Sicherheitsarchitektur genutzt werden, um diese Entscheidungen zu treffen und durchzusetzen. Der Dienst sollte auch in der Lage sein über ein Management Interface dynamisch die Konfiguration von Routern zu verändern. Damit hat EGEE bereits sehr früh das Konzept der dynamischen Firewall, wie in Abschnitt 4.9.5 vorgestellt, angedacht.

4.1.11 Grid-Projekte

Die prominentesten Grid-Projekte im nationalen und europäischen wissenschaftlichen Umfeld sind D-Grid, DEISA und LCG. Im Folgenden werden diese Projekte, die auch diese Arbeit maßgeblich mitgeprägt haben, kurz vorgestellt.

D–Grid

e-Science Infrastruktur

Das Bundesministerium für Bildung und Forschung (BMBF) fördert im Rahmen der e–Science–Initiative die Entwicklung neuer Verfahren und Dienstleistungen für Wissenschaft und Forschung. Wissenschaftler sollen künftig unabhängig von der vor Ort vorhandenen Ausstattung mit Rechnern, Programmen, Daten und Informationen komplexe wissenschaftliche Fragestellungen bearbeiten können. In einer ersten Phase, die zum Aufbau einer solchen Infrastruktur führen soll, fördert das BMBF das Grid Computing im so genannten D–Grid Projekt. Das Projekt begann im September 2005.

Abbildung 4.14 zeigt das e-Science-Framework. Wissenschaftler aus den unterschiedlichsten Anwendungsbereichen nutzen die verschiedensten e-Science-Anwendungen, die selbst wieder auf Diensten und Ressourcen aufsetzen. Dabei war in der Vergangenheit die gemeinsame Nutzung von Diensten, Ressourcen und Großgeräten allenfalls für benachbarte Forschungsbereiche gegeben. Im Rahmen des **D–Grid** [NKG 07, D-Grid] soll eine nachhaltig nutzbare Basis-Grid Infrastruktur entstehen und weiterentwickelt werden. Die Förderung des BMBF verteilt sich auf wissenschaftliche Verbundprojekte, die Grids nutzen; die so genannten **Community Projekte (CPs)** und ein Verbundprojekt von Partnern, die Grid-Infrastrukturen entwickeln und betreiben und den Community Projekten zur Verfügung stellen sollen. Das letztgenannte Projekt wird als **D-Grid Integrationsprojekt (DGI)** bezeichnet. Im Jahr 2008 besteht das D-Grid aus knapp 20 CPs die sich mit verschiedensten Forschungsfragestellungen und Anwendungsbereichen beschäftigen, wie z.B. Fragen der Klimaforschung (**C3-Grid**), der Hochenergiephysik (**HEP-Grid**),

Community Projekte (CPs) und Integrationsprojekt (DGI)

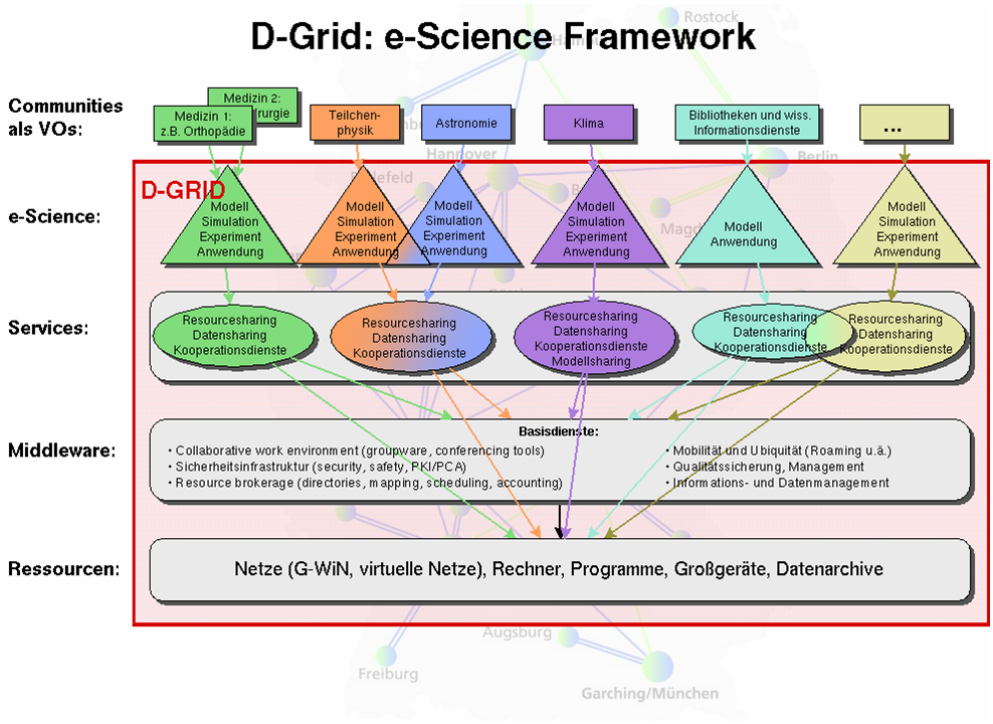

Abbildung 4.14: D-Grid: e–Science Framework [Hege 04]

der Astronomie (**GACG (German Astro Community Grid)**), der Medizin (**Medi-Grid**), den Ingenieurwissenschaften (**IN-Grid**), den Geisteswissenschaften (**Text–Grid**), Fragen der Integration von Grids in Unternehmensverbunde (**Biz2Grid**) und betriebliche Informationssysteme (**BIS-Grid**), bei der Produktentwicklung (**ProGRID**), in der Bauwirtschaft (**BauVOGrid**), im Bankensektor (**FinGrid**), usw.

D-Grid unterstützt explizit mehr als eine Middleware-Technologie. Verwendung finden Globus GT4 (vgl. Abschnitt 4.1.5), UNICORE (vgl. Abschnitt 4.1.7 sowie LCG/gLite (vgl. Abschnitt 4.1.9).

DEISA (Distributed European Infrastructure for Supercomputing Applications)

Koppelung von Supercomputern

Die Distributed European Infrastructure for Supercomputing Applications (DEISA) [DEISA] wird gebildet aus einem Verbund von z. Zt. elf Hochleistungsrechenzentren (vgl. Tabelle 4.3).

Lastverteilung im paneuropäischen Maßstab

globales Dateisystem

Durch eine geeignete Middleware und ein globales Dateisystem, das alle Partner–Sites umfasst, soll mit Hilfe von DEISA eine Lastverteilung zwischen Hochleistungsrechnern im europäischen Maßstab ermöglicht werden. Damit können, durch Verlagerung kleinerer Jobs von einem Hochleistungsrechner auf andere europäische Partner, freie Kapazitäten geschaffen werden. Diese können dann wissenschaftlichen Großprojekten einen frühen Zugang zu sehr großen Rechenressourcen ermöglichen. Als Middleware Technologie in DEI-

DEISA Partner	Lokation
Institut du Développement et des Ressources en Informatique Scientifique — Centre National de la Recherche Scientifique (IDRIS-CNRS)	Orsay, Frankreich
Forschungszentrum Jülich (FZJ)	Jülich, Deutschland
Rechenzentrum Garching der Max Panck Gesellschaft (RZG)	Garching, Deutschland
Consorzio Interuniversitario (CINECA)	Bologna, Italien
Finnish Information Technology Centre of Science (CSC)	Espoo, Finnland
SARA Computing and Networking Services	Amsterdam, Niederlande
Leibniz Rechenzentrum der Bayerischen Akademie der Wissenschaften (LRZ)	Garching, Deutschland
Barcelona Supercomputing Center (BSC)	Barcelona, Spanien
European Centre for Medium-Range Weather Forecasts (ECMWF)	Reading, Großbritannien
Höchstleistungsrechenzentrum Stuttgart, (HLRS)	Stuttgart, Deutschland
Edinburgh Parallel Computing Centre (EPCC)	Daresbury, Großbritannien

Tabelle 4.3: DEISA Partner

SA kommt UNICORE (vgl. Abschnitt 4.1.7) zum Einsatz.

Innerhalb von DEISA wurde die Extreme Computing Initiative ins Leben gerufen. Damit wurden seit Mai 2005 eine kleine Anzahl von „Flaggschiff"–Anwendungen ausgewählt, die sich mit komplexen und innovativen Simulationen beschäftigen, die ohne die DEISA Infrastruktur nicht durchführbar gewesen wären. Als Beispiel seien hier u. a. die Simulation von Turbulenzen in Galaxien und Sternentstehungsgebieten oder Probleme der Strömungsmechanik und Schallemission bei Flugzeugen genannt.

Für DEISA wird eine angepasste UNICORE Infrastruktur verwendet, die auf heterogenen Hochleistungsrechnern installiert wurde. Um die Infrastruktur betreiben und Jobs migrieren zu können, ist eine sehr leistungsfähige und zuverlässige Netzinfrastruktur erforderlich. Das DEISA-Netz ist als durchgängiges 10 Gbit/s-Netz aufgebaut und innerhalb der beteiligten Forschungsnetz–Provider als VPN (virtual private network) bzw. OPN (optical private network) geführt, also in der Anwendung völlig von den öffentlich zugänglichen Forschungsnetzen separiert. Abbildung 4.15 stellt diese Netzinfrastruktur symbolisch dar. Jedes nationale Hochleistungsrechenzentrum ist über ein nationales Forschungsnetz (National Research and Education Network; NREN) mit dem DEISA Backbone verbunden. Die vier deutschen Standorte sind beispielsweise über den DFN (Deutsches Forschungsnetz) angebunden. Damit bietet der Netzverbund die Möglichkeit innerhalb des DEISA-Verbundes große Datenmengen schnell zwischen den Rechenzentren

zu verlagern.

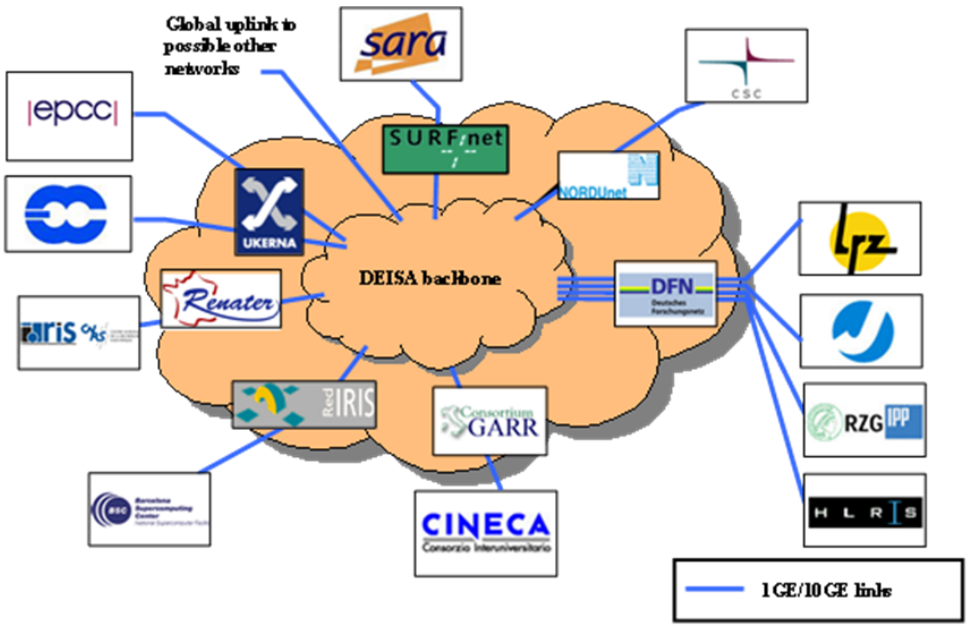

Abbildung 4.15: DEISA Partner und Netzwerk-Infrastruktur [BAR+ 05]

Large Hadron Collider Computing Grid und EGEE

Am CERN in der Nähe von Genf wird der als **Large Hadron Collider (LHC)** bezeichnete Teilchenbeschleuniger gebaut. Der LHC wird der leistungsfähigste Teilchenbeschleuniger der Welt sein. Er ist als Ringbeschleuniger, der in einem Tunnel mit 27 km Umfang, unterhalb der Schweiz und Fankreichs verläuft, konzipiert. In diesem Tunnel werden Protonen oder Bleiionen beschleunigt und zur Kollision gebracht, mit dem Ziel bisher nicht entdeckte Elementarteilchen experimentell nachzuweisen.

Entlang des Ringes sind vier Kammern angeordnet, in denen die Detektoren der sechs verschiedenen Experimente untergebracht sind:

- **ALICE (A Large Ion Collider Experiment)**
- **ATLAS (A Toroidal LHC ApparatuS)**
- **CMS (Compact Muon Solenoid)**
- **LHCb (Large Hadron Collider beauty)**
- **TOTEM (Total Cross Section, Elastic Scattering and Diffraction Dissociation at the LHC)**
- **LHCf (Large Hadron Collider forward)**

LCG: Effizienter Zugriff auf Experimentdaten

Die Experimente erzeugen am Beginn des Betriebes ca. 15 Petabyte an Daten pro Jahr. Das CERN ist zwar in der Lage diese Datenmengen zu speichern,

aber es ist nicht in der Lage den weltweit verteilt arbeitenden Wissenschaftler auch einen effizienten Zugriff auf die Daten zu ermöglichen. Aus diesem Grund wurde eine Grid Infrastruktur und eine Grid Middleware, das **Large Hadron Collider Computing Grid (LHC)**, entwickelt.

Für die Verteilung und dezentrale Verarbeitung der Daten wurde ein Tier-Konzept spezifiziert. Am CERN (Tier 0) fallen die Daten an und werden dort auf Bändern gesichert. Gleichzeitig werden die Daten an 10 weltweite Tier 1 Zentren verteilt. Die Wissenschaftler sitzen im Tier 3 und sind jeweils einem Tier 2 Zentrum zugeordnet (davon gibt es rund 100), die sich die Daten grundsätzlich bei jedem Tier 1 Zentrum abrufen können, die aber im Normalfall das am besten zu erreichende nutzen werden. Die Tier 2 Zentren stellen auch Rechenkapazität für die Datenanalyse bereit. Das LRZ ist Tier 2 Zentrum für die Münchner und bayerischen Physiker. Das Forschungszentrum Karlsruhe ist das deutsche Tier 1 Zentrum.

Die entwickelte Middleware LCG war die Basis für die Arbeiten in EGEE. Im 6. Rahmenprogramm der EU wurde das EGEE (Enabling Grids for E-Science) Projekt gefördert das eine nachhaltige europäische Grid Infrastruktur und mit gLite eine Middleware entwickelt. EGEE wird im 7. Rahmenprogramm ab 2008 weiter gefördert.

In den folgenden Teilkapiteln werden die Sicherheitsdienste und -mechanismen für die vorgestellten Middleware-Technologien untersucht und bewertet.

4.2 AAI-Dienste

Die AAI-Dienste (Authentication, Authorization, Identification) umfassen Mechanismen zu Realisierung von Identifikation, Authentisierung, Single Sign On, Autorisierung und Zugriffskontrolle. In den Abschnitten 4.2.1 bis 4.2.4 werden Mechanismen für die Identifikation, die Authentisierung und Single Sign On vorgestellt. Da die technische Realisierung der verschiedenen Sicherheitsdienste bei allen Middlewares so eng miteinander gekoppelt ist, wird in Abschnitt 4.2.4 eine gemeinsame Bewertung durchgeführt.

Ab Abschnitt 4.2.5 werden Mechanismen zur Autorisierung und Zugriffskontrolle untersucht. In diesem Bereich gibt es viele unterschiedliche Mechanismen, die zum Teil aufeinander aufbauen. Als zwei grundlegende Konzepte zur Rechtevergabe und Durchsetzung in Globus, gLite und UNICORE werden das User Mapping des GSI mit dem Grid Map File (Abschnitt 4.2.5) sowie der UNICORE UUDB (vgl. Abschnitt 4.2.6) vorgestellt. Da beide Mechanismen vom Funktionsumfang sehr ähnlich sind,werden sie zusammen in Abschnitt 4.2.7 bewertet.

Damit eine VO-basierte Autorisierung ermöglicht wird, wurden aufbauend auf dem GSI User Mapping zusätzliche Mechanismen entwickelt. In diesem

Zusammenhang werden CAS und GridShib untersucht und bewertet.

Die Übertragung von Rechten an entfernte Domänen ist ein fundamentales Basiskonzept in Grids. Für die Delegation von Rechten ist MyProxy der wichtigste Stellvertreter, der am Ende dieses Teilabschnitts vorgestellt und bewertet wird.

4.2.1 GSI: End Entity Certificates

End Entity Certificates für persistente Entitäten

Für die Authentisierung verwendet GSI **End Entity Certificates (EEC)**. Diese X.509-Zertifikate [X.509, HPFS 02] werden für persistente Entitäten wie Nutzer, Hosts und Dienste ausgestellt. Das Zertifikat verbindet einen öffentlichen Schlüssel mit einem Identifikator, der die Entität eindeutig identifiziert. Dieses Zertifikat wird von einer Certification Authority (CA) digital signiert. Damit bestätigt die CA, dass der entsprechende Schlüssel zu der im Zertifikat bezeichneten Entität gehört. Dabei kann jede Domäne eine eigene CA betreiben und eigene Identifikatoren vergeben. Es gibt also keine globalen Vereinbarungen über ein einheitliches Namensschemata.

EEC lange Gültigkeit

Die EECs haben eine relativ lange Gültigkeit und aus Sicherheitsgründen soll das entsprechende Schlüsselmaterial nicht für das „Daily Business“ verwendet werden. Beim Login in Globus wird der Benutzer über einen Public Key Mechanismus und mit Hilfe seines EEC authentisiert. Danach erzeugt der Benutzer einen so genannten (User-) Proxy (eine Software Komponente mit dem Namen `myProxy` und ein Proxy Zertifikat [TWE+ 04]. Der Benutzer erzeugt sich sozusagen einen „Stellvertreter“, der in seinem Namen handeln darf. Für alle weiteren Aktionen wird nur noch das kurzfristige Zertifikat des Proxies verwendet. Der Proxy kann sich damit wiederum bei einer weiteren Domäne anmelden. Durch diesen Mechanismus lässt sich ein Single Sign On (SSO) realisieren. Auf die Proxies wird in Abschnitt 4.2.10 noch näher eingegangen.

kurzlebiges Zertifikat im Proxy

In der Mitte der Abbildung 4.8 ist auch noch die Authentisierung mit Benutzername und Passwort angegeben. Diese Möglichkeit wird im WS-Security-Standard vorgegeben, sollte aber für Globus die absolute Ausnahme sein. Die sensiblen Informationen (Username, Passwort) werden zwar durch Message- oder Transport Level Security vor dem Ausspähen geschützt, aber weitergehende Sicherheitsmechanismen der GSI wie Delegation, Integrität oder Replay Prevention werden bei diesem Verfahren nicht unterstützt.

4.2.2 International Grid Trust Federation

Regeln der Zertifizierung durch PMA vorgegeben

Damit die Zertifikate innerhalb einer VO akzeptiert werden, muss eine Vertrauensbeziehung zwischen den lokalen CAs aufgebaut werden. Die Beteiligten müssen sich also auf Policies einigen, mit deren Hilfe die zweifelsfreie Identifikation von Grid-Nutzern und Ressourcen möglich ist. Diese Regeln werden innerhalb einer so genannten **Policy Management Authority (PMA)** festgelegt und alle Nutzer müssen diese Policy akzeptieren. Technisch wird

dies i. d. R. durch eine hierarchische Struktur umgesetzt. Wurzel dieser Struktur ist eine CA, der alle Beteiligten innerhalb der PMA vertrauen. In Deutschland gibt es beispielsweise zwei PMAs mit dazugehörigen CAs: die Grid-KA-CA [GridKA-CA] und die CA des DFN-Verein [DFN-PKI]. Diese nationalen CAs lassen sich selbst wieder seit 2005/6 von der **Europäischen Grid PMA (EUGrid PMA)** [EUGrid PMA] zertifizieren [EU P 04a, EU P 04b].

Im Asiatischen und Pazifischen Raum gibt es mit der **APGrid PMA (Asia Pacific Grid PMA)** [APGrid PMA] eine vergleichbare Policy Management Authority. Für den amerikanischen Kontinent gibt es **The Americas Grid PMA (TAGPMA)** [TAGPMA]. Jede dieser PMAs erstellt Policies, an die sich die untergeordneten CAs und auch die Zertifizierten halten müssen.

GridPMA: weltweite Interoperabilität von Grid Zertifikaten

Im März 2003 haben sich Vertreter dieser drei Organisationen getroffen, um eine **International Grid Trust Federation (IGTF)** zu etablieren [Groe 05]. Dazu einigte man sich auf einen gemeinsamen (Mindest-) Satz von Policies, der von allen akzeptiert und durchgesetzt wird, sowie auf eine regionale Aufteilung der Zuständigkeiten der einzelnen PMAs (vgl. Abbildung 4.16). Durch diese Konstruktion, die als Grid PMA [GridPMA] bezeichnet wird, ist es einfach möglich einen weltweiten Grid-Verbund oder eine weltumspannende VO zu etablieren.

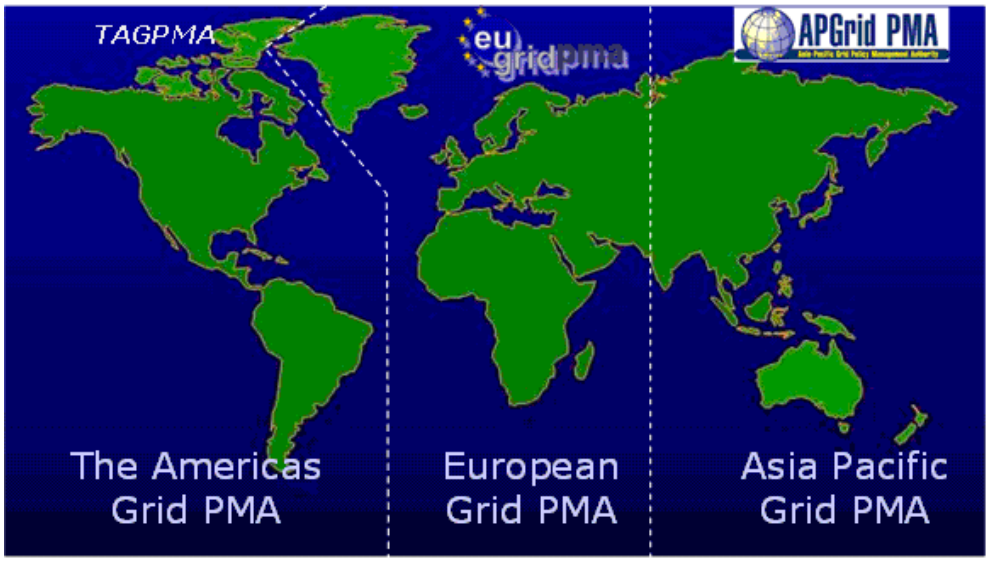

Abbildung 4.16: International Grid Trust Federation [Groe , GridPMA]

Identifizierung durch Registration Authorities

Die Identifizierung der Nutzer erfolgt durch so genannte **Registration Authorities (RA)**, die im Idealfall möglichst nahe beim Nutzer lokalisiert sind. Im D-Grid beispielsweise kann jedes wissenschaftliche Institut eine RA betreiben. Zur Identifizierung muss der Nutzer mit einem amtlichen Ausweisdokument persönlich bei der RA erscheinen und dort einen Antrag auf Erteilung eines Grid-Zertifikates unterschreiben. Das Schlüsselpaar wird in der Regel vom Nutzer selbst erzeugt. Es gibt aber auch RAs, die Schlüssel für die Nutzer erzeugen. Der Certification Request, der den öffentlichen Schlüssel

enthält, wird dann an die CA geschickt und der Nutzer erhält sein Zertifikat per Mail oder WWW.

4.2.3 Endorser, Consigner Modell bei UNICORE

X.509 Zertifikate

Das Sicherheitsmodell von UNICORE basiert, ebenso wie das von Globus, an vielen Stellen auf X.509-Zertifikaten [Erwi 00], mit denen sich sowohl Nutzer als auch Software-Komponenten authentisieren lassen.

Zur Authentisierung und Autorisierung verwendet UNICORE ein so genanntes Consigner/Endorser Modell [Erwi 03].

Endorser signiert Job

- Ein **Endorser** ist eine Entität (Nutzer oder Software Komponente), die einen Job oder eine beliebige andere Anfrage erzeugt und diese mit ihrem privaten Schlüssel signiert.

Consigner submittiert Job

- Der **Consigner** ist die Entität, die einen Job oder eine Anfrage submittiert.

Bei einem einfachen Job fallen Consigner und Endorser zusammen. Der Nutzer erzeugt den Job und submittiert ihn später dann. Erst bei hierarchischen Job-Gruppen, die verschiedene Vsites verwenden, fallen die Rollen auseinander. Der Nutzer ist Endorser, er erzeugt und signiert die Job-Gruppe mit seinem privaten Schlüssel. Der primäre Network Job Supervisor (NJS) tritt als Consigner auf. Er signiert die (Sub-) Jobs, die über ein Gateway zu einem anderen NJS übertragen werden.

NJS als Consigner

Das UNICORE Gateway authentisiert seine Kommunikationspartner auf der Basis des Consigners. Technisch wird dazu die SSL/TLS Schicht verwendet, um den Kommunikationspartner mit Hilfe des Zertifikates des Consigners zu authentisieren. Der NJS hingegen authentisiert sowohl Consigner als auch Endorser. Die Consigner Informationen, d. h. das entsprechende Zertifikat, wird vom Gateway an den NJS weitergeleitet. Der Endorser wird über seine digitale Signatur, als Teil des (Sub-) Jobs, authentisiert.

Sowohl Gateway als auch NJS besitzen selbst Zertifikate, um als Consigner auftreten zu können.

optional Site-Specific Security Objects

Für Usites, die keine zertifikatsbasierte Authentisierung erlauben oder dieser nicht trauen, gibt es **Site-Specific Security Objects (SSO)**, die zusätzliche Authentisierungsinformationen (z. B. SecureID, o. ä.) enthalten. Das SSO wird dann zusammen mit dem Job übertragen und vom entsprechenden Gateway oder NJS ausgewertet.

4.2.4 Bewertung Identifikation und Authentisierung

In Abschnitt 3 wurde ein szenario-unabhängiges Bewertungsschema in Form eines Kriterienkataloges entwickelt. Der Kriterienkatalog lässt sich als Baum darstellen (vgl. Abbildung 3.2), an dessen Spitze die Gesamtbewertung des

jeweiligen Mechanismus steht. Im folgenden wird dieser Kriterienkatalog zur Bewertung der Indentifikations- und Authentisierungsdiensten verwendet. Dabei werden alle Kriterien der Ebene 2 und 3 angewendet und die Mechanismen, die AAI-Dienste realisieren, damit bewertet. Für Kriterien der Ebene 1, die keine Teilkriterien beinhalten, erfolgt die Bewertung zwischen den Kriterien der Ebene 2. Die Bewertungszahlen für die Kriterien der Ebene 1 werden in Form eines Netzdiagramms (vgl. Abbildung 4.17) zusammengefasst präsentiert. Dabei werden sowohl die abgeleiteten (d.h. aus Teilkriterien berechneten) als auch die unmittelbaren Kriterien der Ebene 1 berücksichtigt. Das Netzdiagramm ermöglicht einen schnellen Überblick über die Bewertungen der Mechanismen. Die Gesamtbewertungen (d.h. die Bewertungszahl an der Wurzel des Kriterienkatalogs) aller in Kapitel 4 bewerteten Mechanismen, werden in Abschnitt 4.10 zusammengefasst und verglichen.

- **Middleware-Integration:**
  Identifikation und Authentisierung mittels X.509-Zertifikaten sind voll in alle drei Middlewares integriert (Bewertung: 3).
- **Ressourcen-Integration:**
  Auf den lokalen Systemen müssen die Identitäten aus den X.509-Zertifikaten auf systemlokale IDs abgebildet werden. Eine direkte Verwendung der Zertifikate zur Authentisierung auch auf lokalen Systemen ist i.d.R. nicht möglich. Für jede Klasse von lokalen Ressourcen (z.B. unterschiedliche Betriebssysteme) und für jeden Nutzer ist eine solche Abbildung zu definieren. Die lokale Integration ist deshalb aufwändig (Bewertung: 1).
- **Erweiterbarkeit:**
  Die Authentisierung des Nutzers erfolgt über sein Grid-Zertifikat und eine entsprechende digitale Signatur mit seinem privaten Schlüssel. Der private Schlüssel wird durch ein mit Passwort-Verfahren gesichertes Verschlüsselungsverfahren vor unberechtigtem Zugriff gesichert. Setzt eine Organisation für die lokale Authentisierung andere Verfahren ein, z.B. Hardware-Tokens, und sollen diese Verfahren auch für die Grid-Authentisierung verwendet werden, müssten Middleware Komponenten (z.B. myProxy, vgl. Abschnitt 4.2.10) geändert werden. Diese Änderungen müssten in die offiziellen Releases eingebracht werden (Bewertung 1).
- **Middleware übergreifende Interoperabilität:**
  Über den Verbund der verschiedenen GridPMAs in den weltweiten IGTS ist auch eine weltweite Anerkennung der Zertifikate gegeben. Zertifikatsbasierte Verfahren werden von allen gängigen Middlewares unterstützt (Bewertung 3).
- **Interoperabilität auf Policy-Ebene:**
  Die Policies der verschiedenen CAs und PMAs werden nicht formalisiert. Möglichkeiten zum automatisierten Austausch sind nicht realisierbar (Bewertung: 1).

- **Interoperabilität beim ID-Management:**
  Die technische Realisierung von Grid-Namensräumen erfolgt über die Festlegung auf Distinguished Names (DNs), die in den Standards [ITU- 93, X.509] festgelegt sind. Auf dieser technischen Ebene ist die Interoperabilität hoch. Allerdings erfolgt keine Abstimmung bezüglich der Attribut-Belegungen innerhalb eines DNs. Eine Abstimmung auf Grid-Ebene existiert allenfalls in Ansätzen (Bewertung 1).
- **Trust Management; Formalisierung:**
  Die PMAs gründen über die gegenseitige Anerkennung ihrer PMA Policies und die damit verbundene Akzeptanz abhängiger Zertifikate ein implizites Vertrauensmodell. Die Vertrauenswerte sind binär repräsentiert (Bewertung 1).
- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Anpassung existieren nicht. Es gibt auch keine formalisierten Reputationsmechanismen für den Fall, dass eine CA sich nicht Policy-konform verhält. Die Festlegung der für alle gültigen Mindest-Policy erfolgt zentral durch die International Grid Trust Federation. Eine Verschärfung durch untergeordnete CAs ist möglich (Bewertung: 0).
- **Trust Management; Granularität:**
  Die Granularität des Trust Level Management ist global definiert. Eine Organisation kann IGTS Zertifikate akzeptieren oder ablehnen; eine feinere Abstufung ist nicht vorgesehen (Bewertung: 0).
- **Rechtedelegation; Granularität und Sicherheit:**
  Die Struktur der CAs besteht in der Regel aus der CA selbst und vielen RAs, die die Identifizierung der Nutzer bzw. der Systeme übernehmen. Hier ist eine klare, aber statische Delegation des Identifikationsrechts realisiert. Das Recht zur Zertifizierung verbleibt bei der CA. Der Widerruf des Identifikationsrechts lässt sich relativ leicht und vor allen Dingen zentral durch die CA umsetzen (Bewertung 2).
- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Die Delegation erfolgt ausschließlich von der CA zur RA und eine erweiterte Delegation ist in diesem Konzept nicht vorgesehen (Bewertung: 0).
- **Delegation von Policies:**
  Die Policies bezüglich Identifikation werden von der CA, in Form von beschreibenden Texten ohne Formalisierung, vorgegeben und müssen von den RAs verpflichtend umgesetzt werden (Bewertung: 1).
- **Delegation von Aufgaben:**
  Die Delegation von Aufgaben bei der Identifizierung beschränken sich auf vorher fest vorgegebenen Teilschritte, wie z.B. der Überprüfung des Ausweisdokumentes durch die RA. Die Delegation selbst erfolgt einmalig und ist weder formalisiert noch automatisiert (Bewertung 1).

- **Mapping:**
  Es ist nicht vorgesehen, dass die Heimat-Domäne eigene Policies spezifiziert, die Festlegungen bezügliche Identifikation oder Authentisierung beinhalten. Dementsprechend sind auch keine Mechanismen zum Mapping vorgesehen (Bewertung: 0).

- **Skalierbarkeit:**
  CA-Infrastrukturen sind eine lange etablierte Technik und sehr ausgereift. Eine starke Zunahme der Nutzer oder der beteiligten Organisationen und eine starke räumliche Verteilung führt zu keinem überproportionalen Aufwand. Auch im Hinblick auf Hardware-Skalierbarkeit sind CA-Lösungen als hoch zu bewerten (Bewertung: 3).

- **Flexibilität; Entitätenvielfalt:**
  Die Zertifikate sind primär für die Authentisierung von Nutzern gedacht. Die Zugehörigkeit zu einer Organisation lässt sich über die Attribute Organization (O) bzw. Organizational Unit (OU) abbilden. Allerdings setzt dies eine Grid-weite Abstimmung über die Belegung dieser Attribute voraus, die i.d.R. nicht gegeben ist (Bewertung: 1).

- **Flexibilität; Organisationsflexibilität:**
  Die Struktur der CAs ist sehr hierarchisch und zentralistisch konzipiert. Aus Gründen der Skalierung und der Benutzernähe werden organisationsnahe RAs eingesetzt. Eine lokale oder föderierte Umsetzung ist nicht vorgesehen (Bewertung: 0).

- **Administrierbarkeit:**
  Der Betrieb einer Grid-weiten CA setzt eine komplexe Infrastruktur und komplexe Prozesse voraus. Dies gilt sowohl auf Ebene der Grid-CA und den angeschlossenen RAs als auch auf Ebene der lokalen Organisationen. Die hierarchische Struktur aus RAs und CAs und die CA selbst sind administrativ sehr aufwändig. In der Praxis kommt es bei der Beantragung eines Zertifikates durch die administrativen Vorschriften bei RA und CA, deren notwendigem Zusammenspiel bei gleichzeitiger organisatorischer und räumlicher Trennung, zu sehr langen Laufzeiten. Auch die Verpflichtung zur persönlichen Identifikation durch Besuch bei der RA stellt für den Zertifizierten eine hohe administrative Hürde dar. Für den Betrieb der Infrastruktur und zur Prozeßunterstützung gibt es sehr wenig Management-Unterstützung durch Werkzeuge. Dementsprechend ist der Betriebsaufwand als hoch bzw. sehr hoch einzustufen (Bewertung: 1).

- **Sicherheit; Sicherheitsniveau:**
  Das Sicherheitsniveau der Authentisierung ist abhängig vom Sicherheitsniveau bei der Schlüsselerzeugung, der Speicherung der privaten Schlüssel sowie dem Sicherheitsniveau beim Prozess zur Erzeugung der Zertifikate. Der Prozess zur Zertifikatserzeugung ist durch verschiedenste Policies relativ stark reglementiert. Geht man davon aus, dass diese Policies umgesetzt bzw. eingehalten werden, ist das Sicherheitsniveau

hoch. Schlüsselgenerierung und -speicherung erfolgen in der Regel organisationslokal und auf lokalen und sogar mobilen Endsystemen des Nutzers. Daher kann bestenfalls von einem mittleren Sicherheitsniveau bei der Speicherung der Schlüssel ausgegangen werden. Je größer die Anzahl der ausgegebenen Zertifikate, desto größer ist auch die Wahrscheinlichkeit, dass ein privater Schlüssel kompromittiert und damit die Identität des Eigentümers übernommen werden kann. Die EGEE Policies [Join 06a, Join 06b, Dani 03] tragen diesem Problem Rechnung indem sie den Besitzer des Schlüsselpaares für alle Aktionen haftbar machen, die mit seinem Schlüssel unmittelbar oder mittelbar authentisiert oder autorisiert wurden. Ein expliziter Zusatz besagt: „*Users may be held responsible for* **all** *actions using their credentials, wether carried out personally or not.*“ [Join 06b]. Die Eintrittswahrscheinlichkeit, dass ein Angreifer an den privaten Schlüssel eines berechtigten Grid-Nutzers kommen und damit dessen Identität übernehmen kann, ist als hoch einzustufen. Der Schaden, der dadurch entstehen kann, muss mindestens als Mittel eingestuft werden. Damit liegt die Bewertung des Sicherheitsniveaus nach Abbildung 3.3 bei 0.

- **Sicherheit; Zusicherung, QoP:**
  Eine explizite Festlegung auf Sicherheitsklassen bei der Identifikation oder Authentisierung erfolgt nicht. Allerdings werden im Rahmen der Policies der PMAs [Groe 06a, Geno 05, Geno 06, BuGe 03] algorithmische Vorgaben, Vorgaben über den technischen Betrieb der Zertifizierungskomponenten, die Verwendung von Hardware Security Modulen, von Schlüssellängen, deren maximale Gültigkeitsdauer usw. gemacht. Das heißt, jede CA, die diese Policies ratifiziert und sich bei der entsprechenden PMA akkreditiert [EU P 04a], verpflichtet sich die Vorgaben auch einzuhalten. Damit existieren zwar keine Mechanismen, mit deren Hilfe QoP-Parameter festgelegt und deren Einhaltung und Umsetzung verifiziert werden könnten, aber es liegen zumindest implizite Informationen vor, die eine Bewertung der Sicherheit des Dienstes zulassen. Aus den Informationen aus diesen Policies kann sich jeder Nutzer individuell ein mehr oder weniger umfassendes Bild über das Sicherheitsniveau des Dienstes machen (Bewertung: 1).

Abbildung 4.17 fasst die Bewertung der Kriterien auf Ebene 1 des Kriterienbaumes zusammen und stellt sie in einem Netzplan graphisch dar.

4.2.5 GSI: User Mapping, Grid Map File

kein global eindeutiges Namensschema

Über ein Zertifikat lässt sich eine Entität zweifelsfrei bestimmen und es können Rechte an diese Entität vergeben werden. Allerdings gibt es kein global eindeutiges Namensschema, sondern jede Domäne kann in gewohnter Weise Namen oder Kennungen vergeben. Diese Kennungen werden erst durch die zusätzlichen Informationen über die Heimatdomäne eindeutig.

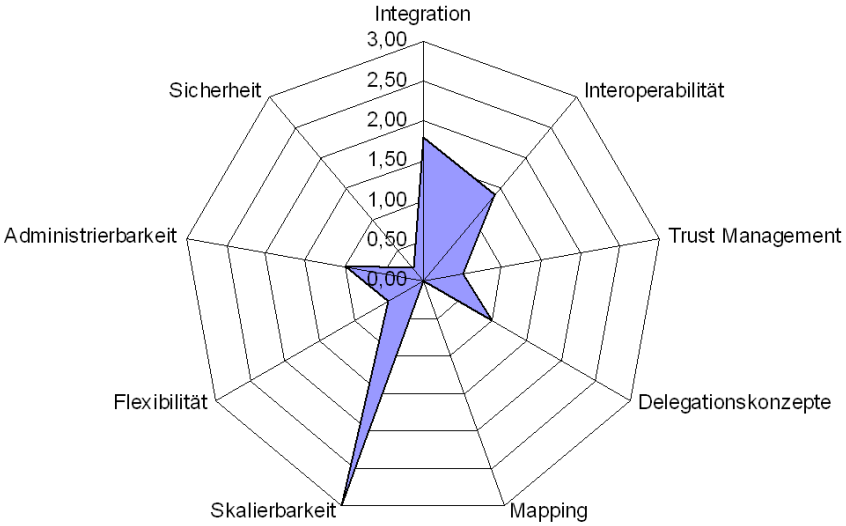

Abbildung 4.17: Bewertung Identifikations- und Authentisierungsmechanismen

Eine Eindeutigkeit der Kennungen (d. h. ohne Information über die Heimatdomäne) ist damit nicht erreichbar. Da aber viele Systeme ausschließlich in der Lage sind, Rechte an eine User-ID oder eine Kennung zu knüpfen, entsteht ein Problem. Globus reagiert darauf, indem es in jeder Domäne ein so genanntes Grid Map File vorschreibt.

Grid Map File: Abbildung auf lokale Kennung

Das **Grid Map File** muss vom Globus-Administrator in jeder Domäne erstellt und gepflegt werden. Es bildet globale Grid-IDs auf lokale Kennungen ab. D. h., der Distinguished Name (DN) aus einem Zertifikat wird auf eine lokale Kennung abgebildet (vgl. Abbildung 4.18). Die Autorisierung erfolgt mittelbar durch diese Abbildung. Auf den verschiedenen Zielsystemen werden der entsprechenden lokalen Kennung Rechte zugeteilt. In der Domäne wird dann für weitere Autorisierungsentscheidungen ausschließlich diese lokale Kennung und die lokale Zugriffskontroll-Policy verwendet.

```
# Distinguished Name                          local User ID

/C=DE /O=BSI /OU=Globus /CN=John Doe          jdoe
/C=US /O=USC /CN=Karl Kesselmann              a282419
```

Abbildung 4.18: Grid Map File (vereinfachtes Beispiel)

Im Umfeld von LCG ist es üblich mehrere Nutzer auf so genannte Pool-Accounts abzubilden (vgl. Abb. 4.19).

Pool-Account

Ein **Pool-Account** repräsentiert eine Menge von lokalen Nutzergruppen, die

```
# Distinguished Name                              local User ID
/C=CH /O=CERN /OU=LHC-Grid /CN=Tim Berners-Lee       .WEB
/C=CH /O=CERN /OU=LHC-Grid /CN=Jon Doe               .WEB
```

Abbildung 4.19: Beispiel eines Mapping auf einen Pool-Account)

alle dieselben Rechte besitzen. Für einen solche Pool-Account (gekennzeichnet durch den Punkt vor der local User ID) werden im Beispiel von Abbildung 4.19 dynamisch die lokalen Kennungen WEB0 bis WEBn erzeugt und die Nutzer diesen Pool-Accounts dynamisch zugeordnet. Damit wird die Zurechenbarkeit einer Aktion zu einem bestimmten Nutzer erheblich erschwert und im schlimmsten Fall sogar unmöglich gemacht. Ein Pool-Account darf, nachdem der Nutzer sich abgemeldet hat, wieder einem anderen Nutzer zugeteilt werden. Dieses „Account-Sharing“ führt dazu, dass der neue Nutzer Zugriff auf alle Daten hat, die der vorherige Nutzer auf dem System erzeugt und evtl. zu löschen vergessen hat. Der neue Nutzer „erbt“ alle Daten und Programme des vorherigen Nutzers. Bei den Programmen ist dieses Erbe noch fataler. Zum einen weiß der Nutzer i.d.R. gar nicht, dass er Programme geerbt hat und kennt weder den Namen noch die Funktionalität des einzelnen Programms. Zum anderen ist er aber für die Konsequenzen der Ausführung dieser Programme verantwortlich (*„You are liable for the consequences of any violation by you of these conditions of use.*“ [Join 06a]). Ein böswilliger Vornutzer des Pool-Accounts kann sehr einfach ein Schadprogramm erzeugen, dies unter einem Namen eines häufig genutzten Werkzeugs abspeichern und das Löschen „vergessen“.

4.2.6 UNICORE UUDB

lokale Rechtevergabe

Jeder Request (z. B. Status Request, Job Submission oder der Job selbst) wird vom Endorser digital signiert. Das Gateway gibt das Endorser– und Consigner–Zertifikat an den NJS weiter. Der NJS entscheidet dann, welche Rechte der Request in der lokalen Vsite erhält. Dazu wird, ähnlich wie bei Globus, ein Mapping auf eine lokale Kennung durchgeführt. Die entsprechenden Mapping-Regeln sind in einer **UNICORE User Databse (UUDB)** gespeichert. Die Zugriffskontrollentscheidung wird anhand der Rechte der lokalen ID getroffen.

Möglichkeit Usites auszuschließen

Durch die doppelte Authentisierung und Autorisierung von Endorser und Consigner ist es möglich, bestimmte Usites auszuschließen. Beispielsweise kann ein bestimmter Benutzer alle Ressourcen nutzen. Wenn sein Request jedoch über einen NJS einer bestimmten Usite abgesetzt wurde, darf er keine Ressourcen mehr nutzen. Durch diesen Mechanismus ist es möglich kompromittierte Usites vom UNICORE Grid auszuschließen.

4.2.7 Bewertung Grid Map File und UUDB

Für die Autorisierung nutzen die verschiedenen Middleware-Technologien sehr unterschiedliche Konzepte. Globus und gLite verwenden Grid Map Files, UNICORE die UUDB. Zusätzlich dazu gibt es Erweiterungen, wie z.B. den in Abschnitt 4.2.8 präsentierten CAS oder Delegationsmechanismen wie MyProxy (vgl. Abschnitt 4.2.10). Es existiert also kein einheitlicher Mechanismus zur Autorisierung und da die Mechanismen sehr unterschiedlich sind, kann auch keine gemeinsame Bewertung erfolgen. Lediglich die beiden Mechanismen Grid Map File und UUDB werden gemeinsam mit den Kriterien des Kataloges bewertet. Abbildung 4.20 stellt die Ergebnisse beider Mechanismen in einem vergleichenden Netzdiagramm dar und dabei zeigt sich, dass die Stärken und Schwächen beider Mechanismen sehr ähnlich sind.

- **Middleware-Integration:**
  Der Mechanismus des Grid Map Files ist voll in die Middlewares Globus und gLite integriert. UNICORE nutzt einen äquivalenten Mechanismus mit Hilfe seiner UUDB, es besteht aber keine Möglicheit die UUDB in Globus oder das Grid Map File in UNICORE zu verwenden. Die Integration des Grid Map Files in die spezifische Middlewares (Globus und gLite) ist jeweils mit 3 zu bewerten, die in UNICORE mit 0. Für die UUDB gilt eine Bewertung von 0 für Globus und gLite und eine 3 für UNICORE. Daraus folgt eine Gesamtbewertung für das Grid Map File von 2 und für die UUDB von 1.
- **Ressourcen-Integration:**
  Durch die Umsetzung auf lokale und Ressourcen-spezifische Kennungen ist auch eine vollkommene Integration gegeben (Bewertung: 3).
- **Erweiterbarkeit:**
  Die Erweiterung um andere Mechanismen wie bspw. eine Capability-orientierte Autorisierung ist bei keiner Middleware vorgesehen (Bewertung: 0).
- **Middleware-übergreifende Ineroperabilität:**
  Die Realisierung des User Mapping mittels Mapping File wird von Globus und gLite unterstützt (Mapping File; Bewertung: 1). Die UUDB wird nur in UNICORE verwendet (UUDB; Bewertung: 0).
- **Inertoperabilität auf Policy-Ebene:**
  Bei der Autorisierung muss die Interoperabilität auf Ebene der Policies die Autorisierungsentscheidungen und die Rechte betrachten, die natürlich auch systemunabhängig in Form abstrakterer Policies spezifiziert und damit austauschbar gemacht werden könnten. Die vorgestellten Konzepte, die sich voll auf die lokalen Autorisierungs- und Zugriffskontroll-Mechanismen stützen, bieten keine unabhängige Policy-basierte Spezifikation von Rechten (Bewertung: 0)
- **Interoperabilität beim ID-Management:**
  Mit Hilfe der UUDB und des Mapping Files lassen sich Benutzer ei-

nem individuellen Account zuordnen und sind daher grundsätzlich mit 2 zu bewerten. Die Interoperabilität des Grid Map Files erscheint durch die Verschattung der konkreten System-Spezifika gegeben. Jedes VO–Mitglied hat ein Zertifikat und wird in jeder beliebigen Domäne auf eine lokale ID abgebildet. Innerhalb einer lokalen Domäne bleibt das Problem der Interoperabilität aber sehr wohl bestehen. Gibt es verschiedene Systeme, die nicht in der Lage sind die gleichen IDs zu verwenden, kann es bereits innerhalb einer lokalen Domäne verschiedene Mappings für den gleichen Nutzer geben. Je nachdem welche konkrete Ressource ein Job des Nutzers verwendet, ändert sich dessen lokales Mapping. Für das Grid Map File insbesondere bei gLite, in der Form, wie es im LCG verwendet wird, muss auf Grund der Pool-Accounts die Bewertung auf 1 reduziert werden (UUDB: Bewertung: 2) (Mapping File: Bewertung: 1).

- **Trust Management; Formalisierung:**
  Die Vertrauenswerte zwischen lokalen Ressourcen-Providern und dem Nutzer erfolgen einzig über den impliziten Vertrauenswert, der an das Zertifikat gebunden ist. Es gibt kein formalisiertes Modell (Bewertung: 1).

- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).

- **Trust Management; Granularität:**
  Die Granularität des Trust Management ist global definiert. Der Nutzer vertraut „dem Grid“ bzw. der VO und der Ressource-Provider vertraut der Mitgliedschaft innerhalb der VO oder der Heimat-Organisation, aus der der Nutzer kommt (Bewertung: 0).

- **Rechtedelegation; Granularität und Sicherheit:**
  Alle betrachteten Autorisierungsverfahren gehen, zumindest implizit, von der Prämisse der lokalen Entscheidung als letzter Instanz aus. Eine Delegation von Rechten zur Autorisierung ist nicht vorgesehen. (Bewertung: 0)

- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Weder erweiterte Delegation noch die Beschränkung des Delegationsrechtes sind vorgesehen (Bewertung: 0).

- **Delegation von Policies:**
  Autorisierungs-Policies werden lokal festgelegt und im Normalfall auch nicht kommuniziert. Eine Delegation von Policies von der VO zu den Ressourcen-Providern oder zwischen den einzelnen Organisationen ist nicht vorgesehen. Für die Autorisierungsverfahren mittels UUDB und Grid Map File sind keine Möglichkeiten vorgesehen, mit denen die Quell-Domäne über Policies Einfluss auf die Gewährung von Rechten nehmen könnte. Es sind keine Konzepte zur Delegation von Autorisierungspolicies vorgesehen (Bewertung: 0).

- **Delegation von Aufgaben:**
  Da die Festlegung der Autorisierungs-Policy lokal erfolgt, gibt es keine Konzepte, um Autorisierungsaufgaben zu verteilen oder zu delegieren (Bewertung 0).

- **Mapping:**
  Da weder Konzepte zur Delegation von Rechten noch von Policies vorgesehen sind, ist das Mapping mit 0 zu bewerten.

- **Skalierbarkeit:**
  Das Konzept des lokalen Mappings skaliert bei Hinzunahme neuer Ressourcen (Hardware und Software) gut, da im Normalfall das Mapping bereits bestehender Ressourcen derselben Klasse übernommen werden. Das Konzept ist, durch die Fokussierung auf das lokale Mapping, unabhängig von der räumlichen Verteilung der Grid Ressourcen. Die Hinzunahme neuer Nutzer oder einer neuen Organisation erfordert im schlimmsten Fall bei allen beteiligten Domänen und allen Ressourcen (-Klassen) eine lokale Anpassung bzw. Ergänzung. Eine organisatorische Skalierbarkeit ist damit nicht gegeben. Das Konzept skaliert in zwei Dimensionen. (Bewertung: 2).

- **Flexibilität; Entitätenvielfalt:**
  Für die Autorisierung lassen sich Informationen aus dem Zertifikat nutzen. Hiermit können Nutzer und Organisation abgebildet werden. Über die EECs an Rechenressourcen ließen sich auch Ressourcen selbst abbilden, allerdings finden diese beim Grid Map File keinerlei Berücksichtigung bei der Spezifikation von Rechten (Grid Map File: Bewertung: 1).
  Der UNICORE Mechanismus bildet hier mit seinem Endorser, Consigner Modell eine Ausnahme. Hier können Ressourcen in Form von NJS in die Autorisierungsentscheidung mit einbezogen werden (UNICORE UUDB: Bewertung: 2).

- **Flexibilität; Organisationsflexibilität:**
  Dem Ansatz des Grid Map File liegt die implizite Prämisse zugrunde, dass alle Entscheidungen bezüglich der Autorisierung ausschließlich organisationslokal getroffen werden. Damit ist eine Autorisierung durch die VO nicht ohne weiteres möglich. Diese Einschränkung wurde auch relativ schnell erkannt und es wurde versucht diese mit verschiedenen Konzepten aufzuheben (vgl. Abschnitte 4.2.8 und 4.3.1). Ein föderierter Ansatz ist nicht vorgesehen (Bewertung: 0).

- **Administrierbarkeit:**
  Problematisch an diesem Ansatz ist, dass für jeden Nutzer im Grid in jeder Domäne ein eigenes Mapping definiert, angelegt, aktualisiert und verwaltet werden muss. Der Aufwand dafür beträgt $O(n * m)$ bei n Benutzern und m Domänen. Klar ist auch, dass die Erstellung und Aktualisierung dieser lokalen Mappings einen erheblichen administrativen Aufwand für die lokale Domäne bedeutet. Ändert sich bspw. der Status

eines VO-Mitglieds, muss in allen Domänen das lokale Mapping angepasst werden. Auch die Ausstellung eines neuen Zertifikates oder der Widerruf eines bestehenden erfordert ggf. lokale Reaktionen. Eine flexible und hoch dynamische Nutzer- und Rechteverwaltung ist mit diesem Ansatz nicht möglich. Je größer die VO, umso schwieriger wird es, die Rechteverwaltung auf Basis individueller Nutzer zu pflegen (Bewertung: 0).

- **Sicherheit; Sicherheitsniveau:**
  Durch die verteilte, dezentrale und zum Teil redundante Administration von Rechten und ID-Informationen kommt es zwangsläufig zu Daten-Inkonsistenzen. Eine Übersicht, wer in einer VO welche Rechte besitzt, ist bei diesem Konzept nur sehr schwer zu erstellen. Es sind keine Mechanismen vorgesehen, um Informationen über das lokale Mapping bzw. die lokalen Rechte an eine VO zu propagieren. Die Wahrscheinlichkeit, dass ein Nutzer durch Inkonsistenzen unberechtigten Zugang zu Ressourcen erhält oder das Löschen eines Accounts vergessen wird, ist nicht zu vernachlässigen. Die in Abschnitt 4.2.5 dargestellten Sicherheitsprobleme führen zu einer Eintrittswahrscheinlichkeit, die als hoch einzustufen ist. Der Schaden der durch eine missbräuchliche Übernahme einer fremden Identität (**Impersonation**) oder durch das „Unterschieben“ von Schadprogrammen entstehen kann, ist mindestens mit Mittel zu bewerten (Bewertung: 0). Bei Verwendung von Pool-Accounts ist ein individualisiertes Auditing, Logging oder die Nachvollziehbarkeit unmöglich.

- **Sicherheit; Zusicherung, QoP:**
  Ein Austausch von Informationen über die Mapping-Regeln, d.h. die Abbildungsregeln einer ID mit der ihr zugeordneten Rechte, von der Ziel-Domäne zum Nutzer ist weder beim Grid Map File noch bei der UUDB vorgesehen. Im schlimmsten Fall kennt der Nutzer seine Rechte in der Ziel-Domäne überhaupt nicht. Ebenso fehlen Informationen über die technische Umsetzung bzw. Durchsetzung von Rechten. Mechanismen zum Austausch von Informationen über das Sicherheitsniveau der Autorisierung sind nicht vorgesehen (Bewertung: 0).

Die Abbildung 4.20 stellt die Bewertung der beiden Authentisierungsmechanismen Grid Map File und UUDB gegenüber.

4.2.8 Community Authorization Service (CAS)

Im Normalfall verwendet Globus ein Autorisierungsmodell, das voll auf den Ressourcen-Provider fokussiert ist. Nur der Ressourcen-Provider kann die Rechte an seinen lokalen Ressourcen vergeben. Dazu muss er das Mapping von Grid-Zertifikaten und User-IDs auf lokale Accounts durchführen. Wie bereits in Abschnitt 4.2.7 dargestellt ist die Administrierbarkeit und Sicherheit dieses Ansatzes sehr schlecht. Es entsteht erheblicher zusätzlicher Aufwand bei der Erfassung und Pflege dieser Daten. Eine für die gesamte VO gültige

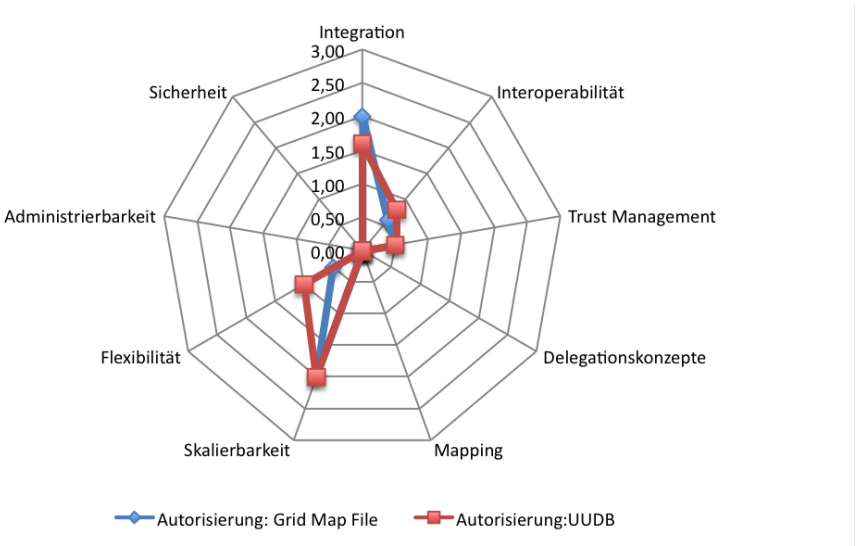

Abbildung 4.20: Bewertung von Grid Map Files und UUDB

Policy, z.B. bezüglich der Vergabe von Ressourcen, zu etablieren und durchzusetzen ist mit diesem Modell nicht trivial.

Rechte werden an CAS übertragen

Aus diesen Gründen wurde ein **Community Authorization Service (CAS)** als weitere Komponente von Globus entwickelt [PWF+ 02]. Die Grundidee des CAS ist die Autorisierung an einen vertrauenswürdigen Dritten (Trusted Third Party), d. h. den CAS, zu übertragen. Der CAS beinhaltet Informationen über vertrauenswürdige CAs, über Benutzer und Ressourcen, die eine VO bilden. Der CAS beinhaltet auch Policy-Regeln, die spezifizieren, wer (welcher Nutzer oder welche Gruppe) für welche Ressourcen welche Rechte besitzt. Der Nutzer bzw. der Dienst greift nicht mehr direkt auf die Ressource zu. Der Zugriff wird über den CAS vermittelt (vgl. Abbildung 4.21). Der Nutzer meldet sich beim CAS, dieser überprüft anhand seiner Policies, ob der Nutzer auf die Ressource zugreifen darf. In diesem Fall stellt der CAS ein Proxy–Zertifikat (Capability) aus, mit dem der Nutzer den Request bei der Ressource absetzen kann (vgl. Abbildung 4.22). Die lokale Zugriffskontrolle entscheidet, ob die Rechte in dem Capability ausreichen und ob der CAS überhaupt berechtigt ist diese Rechte zu erteilen.

CAS autorisiert VO-weit

Restricted Proxy

Mit dem CAS wurde auch die Möglichkeit eingeführt, innerhalb des Proxies eine Policy-Spezifikation anzugeben und damit die Rechte des Nutzers weiter zu beschränken. Auf diese Weise wird der **Restricted Proxy** realisiert. In Abbildung 4.22 ist ein Beispiel einer solchen Beschränkung angegeben. Das Credential erlaubt nur über GridFTP lesend auf alle Verzeichnisse und Dateien unterhalb von `/myhost/mydir/` und schreibend auf `/myhost/myfile` zuzugreifen. Grundsätzlich werden hier die verschiedensten Policy-Sprachen zugelassen. Allerdings müssen alle lokalen Ressourcen in der Lage sein, diese Policy-Sprache zu interpretieren. Falls in einer Domäne die Policy nicht interpretiert werden kann oder der Dienst nicht in der Lage ist die Policy durchzusetzen, muss er das Proxy-Zertifikat ablehnen. Der CAS kann auch völlig

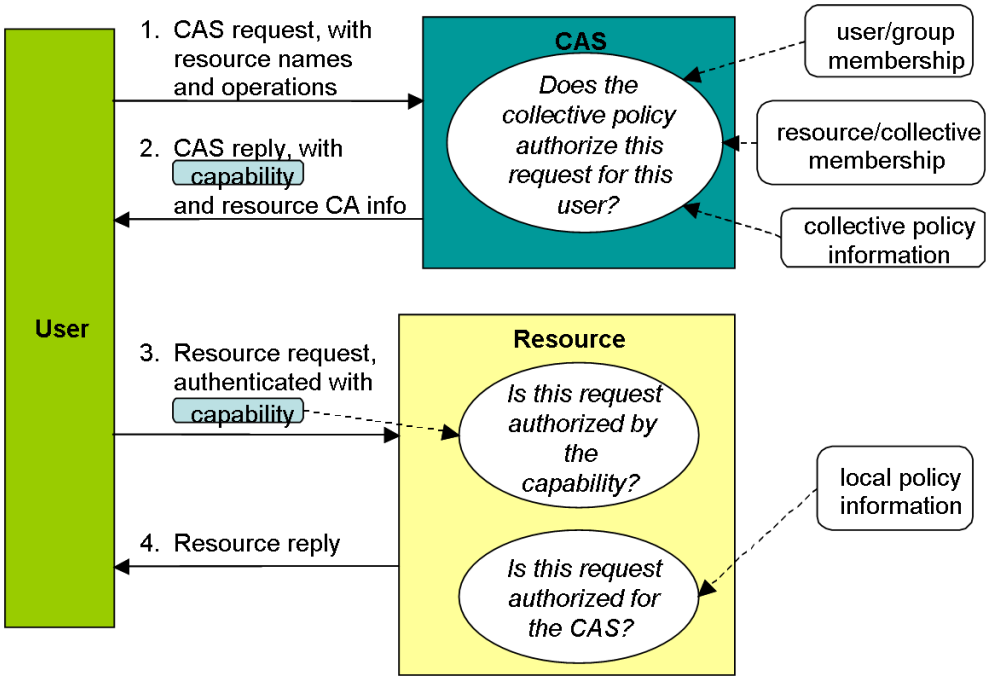

Abbildung 4.21: Ressourcen-Zugriff mittels CAS [Tuec 01]

ohne Policy-Einschränkungen betrieben werden. Das heißt, in diesen Fällen erfolgt ein Rückfall auf die in Abschnitt 4.2.5 beschriebenen Verfahren ohne Einschränkung der Rechte.

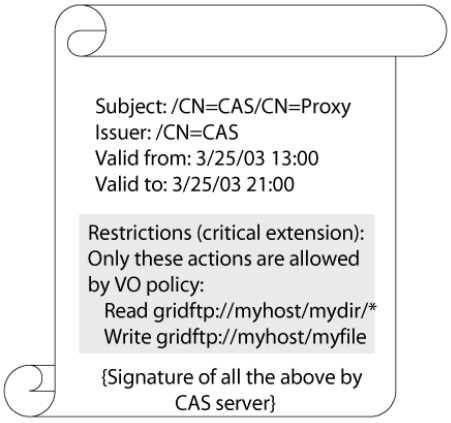

Abbildung 4.22: CAS Capabiltity ohne User Bezug [PKW+ 03]

lokales Mapping wg. Abwärts-kompatibilität

Nachdem der Ansatz „abwärtskompatibel“ mit der bisherigen Autorisierung sein muss, wird weiterhin ein lokales Mapping durchgeführt. In einer Domäne muss nur noch eine Mapping-Regel für den CAS eingepflegt werden. Dadurch wird die Komplexität auf $O(n + m)$ reduziert (vgl. auch Abschnitt 4.2.7). Der erste Prototyp wurde im Jahr 2002 freigegeben. Bei der Anwendung in der Praxis wurden Mängel identifiziert, die zu einer Änderung des Konzeptes führten. Das Capability gab nur Auskunft über die Mitgliedschaft in einer

VO, der konkrete Nutzer, der sich mit dem CAS-Capability bei der Ressource anmeldete, war nicht identifizierbar. Damit implementierte der CAS im Vergleich zu GSI bezüglich Autorisierung das andere Extrem: Alle Rechte an den Ressourcen mussten quasi an den CAS übertragen werden.

Überarbeitung von CAS: zus. Benutzer-Info

Im darauf folgenden Jahr wurde CAS grundlegend überarbeitet [PKW+ 03], um diese Mängel zu beseitigen Das CAS Zertifikat wurde um Benutzerinformationen ergänzt. Außerdem wird das CAS Zertifikat als Erweiterung in einem normalen User Proxy eingebettet (vgl. Abbildung 4.23). Damit besteht volle Abwärtskompatibilität zu GSI. Anwendungen, die weder Policies noch CAS unterstützen, werten die CAS Erweiterungen im User Proxy nicht aus. Zur Autorisierung werden dann nur die Nutzerinformationen im User Proxy verwendet.

Auch die Rechtevergabe kann feingranularer erfolgen. Im alten Ansatz bestimmten sich die Rechte eines Nutzers aus den Rechten, die eine Domäne der VO, d. h. dem CAS übertragen hatte, und den Rechten, die CAS dem Nutzer per CAS Policy zugeteilt hatte. Im neuen Ansatz kann jede Domäne zusätzlich noch eigene Rechte an den Nutzer vergeben. In diesem Fall steigt die Komplexität allerdings wieder auf $O(n * m)$.

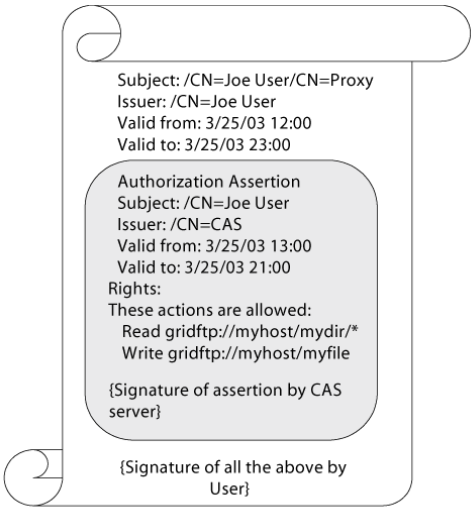

Abbildung 4.23: CAS Capability als Teil des User Proxy

Der CAS kann auch dazu verwendet werden ein Gruppen- und Rollenkonzept einzuführen. Dazu wurde in einer Arbeit von Cannon et al [CCO+ 03] vorgeschlagen die Policies zu erweitern. Eine Policy besteht aus Aktionen, die auf einem Target durchgeführt werden dürfen (vgl. Abbildung 4.23). Die Autoren schlagen als neue Aktion „member of“ und als neues Target eine Gruppen- oder Rollenspezifikation vor.

Ein Problem, das beim Einsatz von CAS auftreten kann, ist die Kompromittierung des CAS selbst. Dann kann nur durch eine Änderung der lokalen Mapping-Regeln Schaden von der einzelnen lokalen Domäne abgewendet werden.

4.2.9 Bewertung CAS

Der Community Authorization Service (CAS) ist als Erweiterung des Grid Map Ansatzes zu verstehen und zu diesem auch abwärtskompatibel. Im folgenden wird CAS mit den Kriterien des Kriterienkataloges (aus Abschnitt 3.2) bewertet.

- **Middleware-Integration:**
  CAS ist als Zusatzmechanismus in Globus integrierbar, d.h die Integration in Globus ist mit 2 zu bewerten. Nachdem CAS von UNICORE und gLite nicht unterstützt wird, ist hier mit 0 zu bewerten. Als Gesamtbewertung ergibt sich damit ein Wert von 0,66.

- **Ressourcen-Integration:**
  Durch die Ergänzung der User-Proxies um CAS-Policies und die Abwärtskompatibilität ist eine Rückfallmöglichkeit auf das in Abschnitt 4.2.7 bewertete User Mapping möglich. Da von CAS allerdings keine Vorgaben zu den verwendeten Policies gemacht werden, kann theoretisch jede lokal verwendete Policy-Sprache auch beim CAS verwendet werden. Damit ist die lokale Integration sehr einfach möglich (Bewertung: 2).

- **Erweiterbarkeit:**
  CAS lässt sich durch die Verwendung von sehr generischen Policies um Sicherheitsmechanismen erweitern, die mit Hilfe der verwendeten Policies spezifiziert und durch entsprechende Implementierungen realisiert werden. Ein Beispiel einer solchen Erweiterung ist das von Cannon et al [CCO+ 03] vorgeschlagene Gruppen und Rollenkonzept (Bewertung: 2).

- **Middleware-übergreifende Interoperabilität:** CAS wird nur von Globus unterstützt (Bewertung: 0).

- **Interoperabilität auf Policy-Ebene:**
  Die Interoperabilität auf Ebene der Policies ist nur bedingt gegeben, da keine Vorgaben über die zu verwendende Policy-Sprache gemacht werden. Policies werden vom CAS spezifiziert, müssen aber von den Ressource-Providern nicht verstanden werden. In diesem Fall erfolgt ein Rückfall auf die Standard-Mechanismen der GSI (Bewertung 2).

- **Interoperabilität beim ID-Management:**
  Bei den in Abbildung 4.22 dargestellten CAS Capabilities ohne User Bezug ist nur ein zentrales Identitätsmanagement möglich. Folglich ist diese Betriebsart des CAS mit 0 zu bewerten. Mit Hilfe der CAS-Erweiterungen des User Proxy (vgl. Abbildung 4.23) lassen sich individuelle Benutzer identifizieren und es bestünde die Möglichkeit diese auf individuelle Accounts abzubilden (CAS ohne User-Bezug; Bewertung 0; CAS im User-Proxy; Bewertung 2).

- **Trust Management; Formalisierung:**
  Die lokalen Ressource-Provider treten die Autorisierung an den CAS ab. Das Vertrauensverhältnis zwischen CAS und lokalen Ressourcen-Providern ist binär und gilt für alle Anwendungsfälle (Bewertung: 1).

- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).

- **Trust Management; Granularität:**
  Die Vertrauenswerte werden lediglich implizit auf Ebene der Institutionen (Ressource-Provider und CAS) festgelegt (Bewertung: 0).

- **Rechtedelegation; Granularität und Sicherheit:**
  Bei der vom CAS vorgesehenen Betriebsart übertragen die Mitglieder einer VO alle Autorisierungsentscheidungen und damit alle ihre Rechte auf den CAS. Sollen dem CAS diese Rechte entzogen werden, müssen lokal die entsprechenden Einträge für den CAS in den Mapping Files geändert werden, dies führt dazu, dass kein CAS Zertifikat mehr akzeptiert wird. Soll ein Recht für eine bestimmte Ressource entzogen werden, ist nur auf dieser Ressource der Eintrag zu ändern. CAS wurde auch eingeführt um den Administrationsaufwand bei der Pflege der Mapping Files zu reduzieren. Das heißt, im Normalfall werden beim Einsatz von CAS auch keine lokalen User-Mapping-Einträge gepflegt. Werden nun dem CAS Rechte entzogen, können damit auch alle Mitglieder der VO, die über diesen CAS versorgt werden, die Ressourcen nicht mehr nutzen (Bewertung: 1).

- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Eine Weiterdelegation der Rechte vom CAS an andere ist nicht vorgesehen (Bewertung 0).

- **Delegation von Policies:**
  Für die Festlegung von Autorisierungspolicies bzw. die Delegation lokaler Policies bei den Ressourcen-Providern an den CAS sind im CAS keine Mechanismen vorgesehen. Das Konzept geht davon aus, dass Autorisierungspolicies zentral festgelegt werden (Bewertung: 0).

- **Delegation von Aufgaben:**
  Ebensowenig gibt es Konzepte, um Autorisierungsaufgaben zu verteilen oder zu delegieren (Bewertung: 0).

- **Mapping:**
  Beim CAS gibt es grundsätzlich die Möglichkeit VO-globale Policies und Rechte festzulegen. Da die verwendeten Sprachen nicht mit festgelegt wurden bzw. der Rückfall auf das klassische Grid-Map File spezifiziert wurde, wird ein möglicher Konflikt in diesem Fall nicht durch das Ordnungsschema (VO-Policies sind höherwertig) sondern durch lokale Regeln der Ziel-Domäne gelöst. Für den Nutzer ist dies

nicht transparent (Bewertung: 1).

- **Skalierbarkeit:**
  Die räumliche Verteilung der Ressourcen hat kaum Einfluss auf den CAS. Werden dem Grid neue Ressourcen hinzugefügt, müssen für diese die Autorisierungspolicies beim CAS spezifiziert werden. Hier ist von einem linearen Zusammenhang auszugehen. Bei der Zunahme organisatorischer Einheiten gilt im Prinzip dasselbe. CAS skaliert also in allen drei Dimensionen (Bewertung: 3).

- **Flexibilität; Entitätenvielfalt:**
  Für die Autorisierungsentscheidung können zentral beim CAS beliebige Policies zum Einsatz kommen. Damit lassen sich auch prinzipiell Nutzer, Organisationen, VO, Gruppen und Rollen abbilden. Bei einer sehr engen Kooperation zwischen Ressource-Provider und CAS lassen sich auch Sicherheitsattribute auf Ebene der Ressourcen vergeben (Bewertung: 3).

- **Organisationsflexibilität:**
  Die Autorisierungsentscheidung wird ausschließlich vom CAS, d.h. auf zentraler Ebene getroffen. Ein Rückgriff des Ressourcen-Providers auf Informationen des User Proxy und die damit verbundene lokale Autorisierungsentscheidung ist grundsätzlich möglich, im CAS Konzept aber nur für Ausnahmefälle vorgesehen und bedeutet einen erheblichen zusätzlichen Administrationsaufwand. Ein föderativer Ansatz ist nicht vorgesehen (Bewertung 1).

- **Administrierbarkeit:**
  Der Betriebsaufwand ist durch den zentralistischen Ansatz eigentlich gering. Im Normalbetrieb sind nur an einer Stelle Anpassungen nötig. Der Aufwand beträgt $O(n+m)$ bei n Benutzern und m Domänen. Allerdings führt die Unterstützung beliebiger Policy-Sprachen zu zusätzlichem Aufwand. Entweder einigt sich die gesamte Föderation auf eine zu verwendende Policy-Sprache oder es müssen alle bei den verschiedenen Ressourcen-Providern verwendeten Policy-Sprachen unterstützt werden (Bewertung: 1).

- **Sicherheit; Sicherheitsniveau:**
  Bei einer zentralen Komponente, in der alle Rechte verwaltet werden, ist die Schadenshöhe bei einem erfolgreichen Angriff auf diese Komponenten naturgemäß sehr hoch. Der Angreifer kann sich damit Rechte auf allen Ressourcen beschaffen. Allerdings bietet die zentrale Komponente den Vorteil, dass diese sehr gezielt und mit sehr effektiven Mitteln geschützt werden kann. Unter der Annahme, dass für den CAS-Server besondere Sicherheitsmechanismen entwickelt werden, darf die Eintrittswahrscheinlichkeit für einen Angriff als niedrig angesehen werden (Bewertung: 1).

- **Sicherheit; Zusicherung, QoP:**
  Im standardmäßig vorgesehenen Betriebsmodus übertragen alle lokalen

Domänen ihre Rechte zur Autorisierung an den CAS als zentrale Komponente. Hier wäre es besonders wichtig, dass sich die Nutzer über die Einhaltung von Sicherheitsmechanismen bei dieser sehr kritischen Infrastrukturkomponente informieren könnten. In den Spezifikationen zum CAS werden keine Vorgaben für einen sicheren Betrieb der CAS Server gemacht. Die Infrastruktur wird als Trusted Third Party proklamiert, es sind aber keine Mechanismen vorgesehen, um das Sicherheitsniveau prüfen zu können oder über QoP Parameter zu kommunizieren (Bewertung: 0).

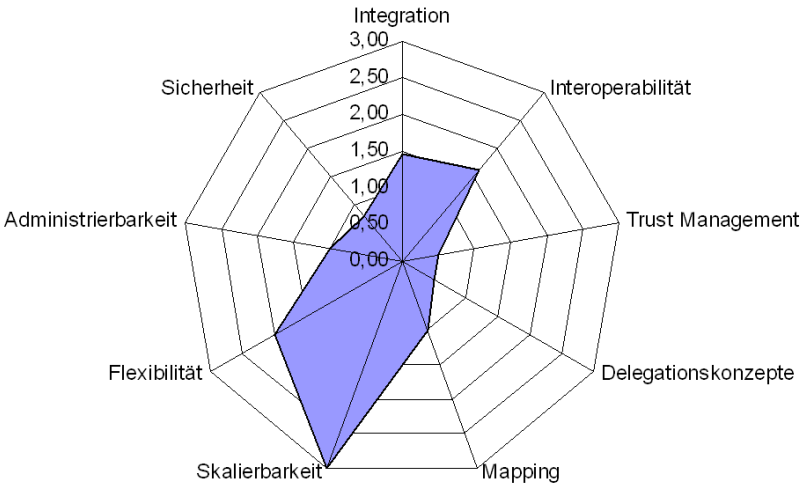

Abbildung 4.24: Bewertung des CAS

Abbildung 4.24 stellt die Bewertung der Kriterien der Ebene 1 des Kriterienkataloges in einem Netzdiagramm dar. CAS, als Erweiterung des Grid Map Files, kann seine Bewertungen im Vergleich zum Grid Map File bei einigen Kriterien verbessern (z.B. bei der Skalierbarkeit, der Flexibilität oder der Integration). Ein Vergleich der beiden Netzdiagramme in Abbildung 4.24 und Abbildung 4.20 zeigt dies sehr deutlich.

4.2.10 MyProxy

Wie bereits in Abschnitt 4.2.1 dargestellt, erzeugt sich der Benutzer bei der Authentisierung einen Proxy, der in seinem Namen und mit seinen Rechten handelt. Dieses Konzept lässt sich verallgemeinern und zur Delegation von Rechten in entfernte Domänen oder an Dienste verwenden. In der Literatur wird in diesem Zusammenhang häufig von **Proxy Credentials** oder Proxy-Zertifikat [TWE+ 04] gesprochen. In Abbildung 4.25 wird das Prinzip des Proxy–Zertifikates dargestellt. Der linke Teil der Abbildung symbolisiert die Heimat-Domäne des Benutzers, der rechte Teil die Gastdomäne.

Proxy Credential

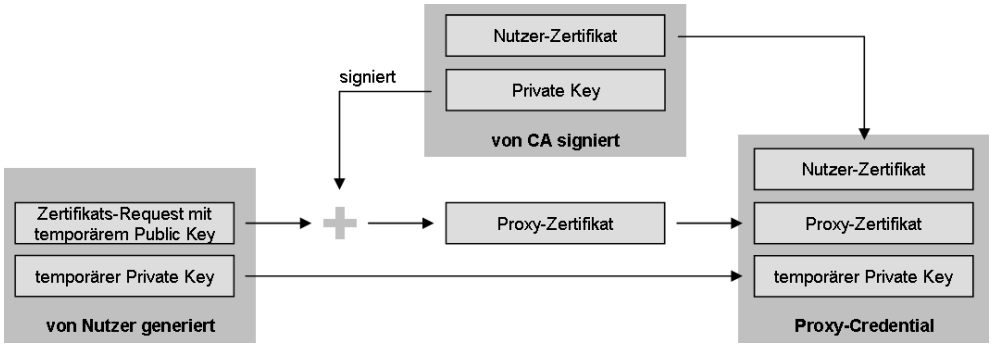

Abbildung 4.25: GSI Proxy Zertifikate [GPR 06]

Proxy signiert mit EEC

Der Benutzer erzeugt lokal ein Schlüsselpaar aus privatem und öffentlichem Schlüssel. Mit dem öffentlichen Schlüssel erzeugt er ein so genanntes Proxy Zertifikat mit beschränkter Gültigkeitsdauer. Dieses Zertifikat signiert er mit seinem, zum EEC gehörenden, privaten Schlüssel (der eine relativ lange Gültigkeit besitzt). Das Proxy Zertifikat überträgt er an seinen Proxy in die Gastdomäne. Damit dieser das Zertifikat auch nutzen kann, um stellvertretend für den Nutzer zu handeln, bedarf der Proxy-Prozess noch des temporären privaten Schlüssels. Dieser wird, ebenso wie das Proxy-Zertifikat selbst, über eine verschlüsselte Verbindung an den Proxy in der Gastdomäne übertragen.

Proxy Zert. und privater Schlüssel an Gast-Domäne

Der private Schlüssel wird im Dateisystem der Gastdomäne abgelegt und über normale Dateisystemrechte (gebunden an die Benutzer-ID) gesichert. Damit kann ein Proxy-Prozess die Schlüssel auch ohne Benutzerinteraktion verwenden. Da die Zertifikate und Schlüssel damit in potentiell unsicheren Umgebungen genutzt werden, ist ihre Gültigkeit relativ kurz. In der Globus Implementierung von MyProxy ist der Standardwert für die Lebenszeit 12 Stunden. Danach werden die Zertifikate, wenn sie nicht verlängert werden, ungültig. Jedes Proxy Zertifikat hat eine eigene Identität (`Subject` Feld im Zertifikat), die vom Ersteller des Zertifikates vergeben wird. Nach [TWE+ 04] muss diese ID das Feld `Issuer`, d.h. den Common Name des Ausstellers, enthalten und um eine CN Komponente für den Proxy erweitert werden. Diese Erweiterung soll so gewählt werden, dass jedes Proxy Zertifikat eines Ausstellers einen global eindeutigen Namen erhält.

Dienst kann weiteren Proxy erzeugen

Mit Hilfe dieses Konzeptes lassen sich auch Schlüsselpaare und Proxies für entfernte Dienste erzeugen und diesen damit die Rechte oder Teile der Rechte des Nutzers übertragen. In diesem Fall spricht man von einem **Ressource-Proxy**. Der Dienst kann das Zertifikat dann nutzen, um selbst wieder (quasi im Auftrag) des Nutzers weitere Dienste aufzurufen oder neue zu erzeugen und weitere Ressource-Proxies zu generieren.

In Abbildung 4.26 wird ein Anwendungsbeispiel vorgestellt. Der Benutzer erzeugt beim Grid Login seinen User Proxy und möchte dann in den Domänen A und B jeweils Prozesse erzeugen, die miteinander kommunizieren sollen. Gleichzeitig soll der Prozess in A auf Daten in Domäne C zugreifen. Der User–Proxy erzeugt dazu jeweils einen Proxy in A und B. Dabei wird die

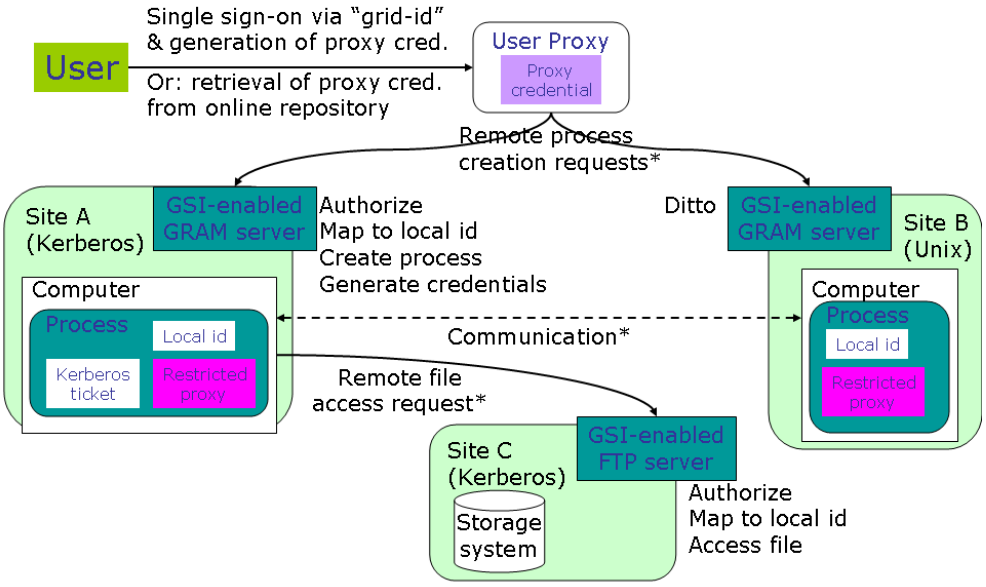

Abbildung 4.26: Beispiel einer Delegation: Prozesse in Domäne A und B die kommunizieren und Daten von Domäne C nutzen [Kim 02]

User ID bzw. der Common Name aus dem Zertifikat auf jeweils eine lokale Identität abgebildet (mittels Map File vgl. Abschnitt 4.2.5). In der Abbildung ist auch symbolisiert, dass der User Proxy die Proxies in A und B in ihren Rechten beschränken kann, indem er Restricted Proxies erzeugt. Der Proxy aus A greift auf Daten in C zu. Zur Authentisierung verwendet er sein Proxy–Zertifikat. Nachdem auch in C ein Mapping durchgeführt und die Rechte überprüft wurden, werden die Daten nach A zurückgeliefert.

Auch für die Kommunikation zwischen A und B müssen sich die beiden Proxies gegenseitig authentisieren.

Erweiterungen

Mit dem RFC 3820 [TWE+ 04] wurden Erweiterungen für Proxy Zertifikate eingeführt, die eine Beschränkung der Länge der Proxy-Zertifikatskette und eine Proxy Policy einführen. Abbildung 4.27 fasst die Erweiterungen zusammen.

Längenbeschränkung: wie weit kann delegiert werden

Das Attribut zur Längenbeschränkung (`pCPathLenConstraint`) kann auch für eine einfache Beschränkung des Delegationsrechts verwendet werden. Die Länge `pCPathLenConstraint` gibt die maximale Tiefe des Zertifizierungspfades eines Proxy Zertifikates an. Falls $\texttt{pCPathLenConstraint} = 0$ ist, kann mit diesem Zertifikat kein Proxy Zertifikat erzeugt werden. Mit diesem Attribut kann damit angegeben werden, wie „weit“ ein Proxy weiterdelegieren kann.

Policy für Restricted Proxy

Die Proxy Policy ist zur Autorisierung und damit zur Beschränkung der Rechte zu verwenden (Restricted Proxy). Eine detaillierte Policy kann di-

```
ProxyCertInfo ::= SEQUENCE {
     pCPathLenConstraint INTEGER (0..MAX) OPTIONAL,
     proxyPolicy          ProxyPolicy }

ProxyPolicy ::= SEQUENCE {
     policyLanguage        OBJECT IDENTIFIER,
     policy          OCTET STRING OPTIONAL }
```

Abbildung 4.27: Policy-Erweiterungen für Proxy Zertifikate [TWE+ 04]

rekt im Feld `policy` kodiert werden. Dazu kann grundsätzlich jede beliebige Policy-Sprache verwendet werden, die über einen OID im Attribut `policyLanguage` definiert wird. Der Standard legt sich auf keine Policy-Sprache fest. Im RFC sind zwei Policies vordefiniert, die über das Feld `policyLanguage` spezifiziert werden.

Impersonation Policy

- Die Policy `id-ppl-inheritAll` mit dem OID 1.3.6.1.5.5.7.1.14.21.1 besagt, dass alle Rechte des Ausstellers an den Proxy übertragen werden (Impersonation).

Policy für Proxy ohne Rechte

- Die Policy `id-ppl-independent` mit dem OID 1.3.6.1.5.5.7.1.14.21.2 bedeutet, dass der Proxy überhaupt keine Rechte des Ausstellers bekommt. Der entsprechende Proxy ist von der Ziel-Domäne als unabhängige Entität zu betrachten. Nur wenn dem Proxy in der Ziel-Domäne explizit Rechte erteilt werden, kann er überhaupt Aufgaben ausführen.

Quell-Domäne spezifiziert; Ziel-Domäne muss umsetzen

Die Policies und die entsprechenden Rechte werden in einer Quell-Domäne spezifiziert, aber nur die Ziel-Domäne ist überhaupt in der Lage die Policy auch durchzusetzen. Die Autorisierung erfolgt in der Quell-, die Zugriffskontrolle in der Ziel-Domäne. Damit muss sich die Quelle auf die korrekte Umsetzung der Policy verlassen. Damit dieses Zusammenspiel überhaupt funktionieren kann, muss die Ziel-Domäne in der Lage sein, die Policy zu interpretieren und die Policy-Sprache zu verstehen. Falls dies nicht der Fall ist, muss die Ziel-Domäne die Anfrage ablehnen oder sie kann das Proxy Zertifikat bzw. die Kette ignorieren und Rechte in Abhängigkeit des EEC erteilen (d.h. die Ziel-Domäne verhält sich, als ob sie die Proxy-Zertifikatskette nie gesehen hätte).

Proxy kann weitere Rechte erhalten

Da ein Proxy als eigenständige Entität auftritt, kann er auch Rechte von Anderen außer dem ursprünglichen Aussteller des Proxy-Zertifikates bzw. der Proxy-Policy erhalten. In Abbildung 4.28 ist ein Beispiel hierfür angegeben. Der Proxy erhält im Rahmen der Rechtedelegation Nutzerrechte, zusätzlich erhält er, abhängig von den Nutzerrechten und den VO-Policies, Rechte aus der VO. Auch die Ziel-Domäne besitzt eigene Policies über die Nutzung ihrer Ressourcen und kann dem Proxy Rechte erteilen. Der Proxy besitzt letztendlich die Vereinigungsmenge der Rechte des Ausstellers und anderer Rechte, die er von Dritten erhalten hat.

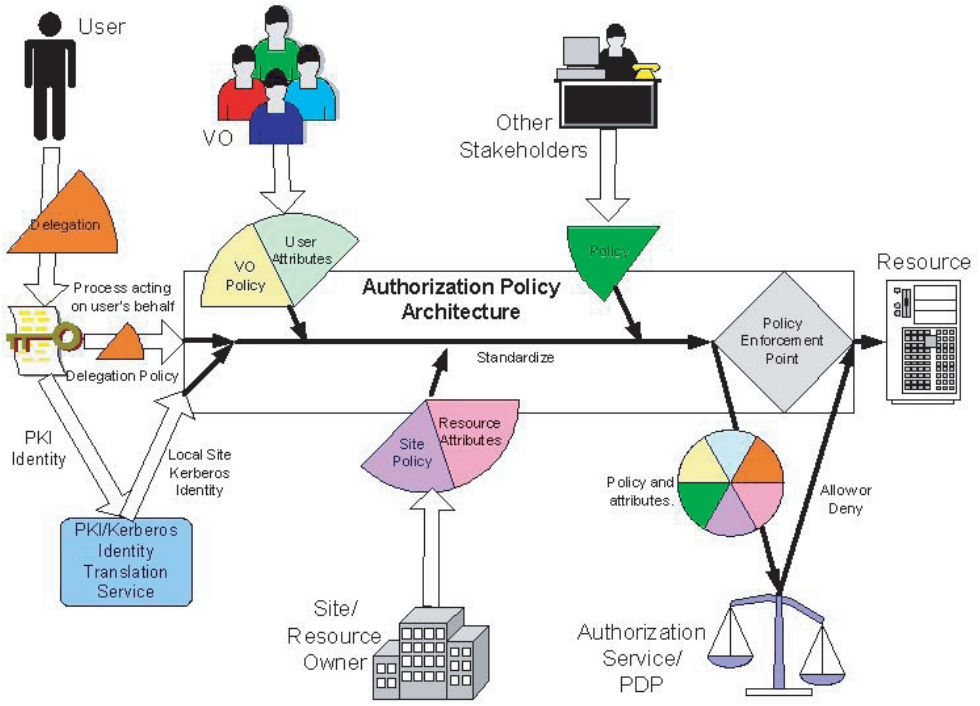

Abbildung 4.28: Rechte als Vereinigungsmenge von Teilrechten; Beispiel aus [EGEE 05b]

fortgesetzte Delegation erzeugt Zertifikatsketten

Wie oben bereits erwähnt, entstehen durch die fortgesetzte Delegation Ketten von Proxy-Zertifikaten. Bei der Zugriffskontrolle in der Domäne n muss das Proxy-Zertifikat n verifiziert werden, d.h. aber, es ist nicht mehr nur das einzelne Zertifikat zu verifizieren, sondern die gesamte Kette ist zu überprüfen (vgl. auch Abschnitt 5.2.2):

Verifikation von Zertifikatsketten

1. Die Kette muss mit einem EEC erzeugt worden sein. Das heißt, das 1. Zertifikat ist ein Proxy-Zertifikat, das mit dem privaten Schlüssel des EEC signiert wurde.

2. Für alle Zertifikate x in $[1, \ldots, n-1]$ muss x der Aussteller des Zertifikates $x+1$ sein. Für die Verifikation bedeutet dies, dass alle digitalen Signaturen und alle Zertifikate verifiziert werden müssen.

3. Alle x in $[1, \ldots, n]$ müssen noch gültig sein, d.h. der Gültigkeitszeitraum darf noch nicht überschritten worden sein.

4. Die Länge der Zertifikatskette, darf `pCPathLenConstraint` nicht überschreiten, falls dieses Attribut belegt wurde.

Diese Verifikation ist sehr rechenintensiv und damit für lange Ketten sehr teuer. In Abschnitt 5.2.2 wird mit Implanted Chain Certificates ein deutlich effizienteres, alternatives Verfahren für Zertifikatsketten vorgeschlagen.

4.2.11 Bewertung MyProxy

MyProxy ist ein Standard-Verfahren in Globus und gLite, um Rechte weiter zu delegieren und den Proxy gewissermaßen im Auftrag seines „Erzeugers“ tätig werden zu lassen. Der Mechanismus ist sehr weit verbreitet und darf deshalb nicht unberücksichtigt und unbewertet bleiben. Im Folgenden werden wieder die verschiedenen Kriterien aus dem Kriterienkatalog auf MyProxy angewendet.

- **Middleware-Integration:**
  MyProxy ist integraler Bestandteil von Globus und gLite und deshalb jeweils mit 3 zu bewerten. UNICORE unterstützt MyProxy nicht (UNICORE; Bewertung:0) (Middleware-Integration; Bewertung: 2).

- **Ressourcen-Integration:**
  MyProxy selbst ist als Teil der Middleware gut in die Ressourcen integriert. Problematischer stellt sich die Integration bzw. die Umsetzung von Restricted Proxies dar. Die Quell-Domäne kann grundsätzlich beliebige Policies spezifizieren. Bei den Ziel-Domänen kann aber nicht davon ausgegangen werden, dass diese Rechte auf den vorhandenen Ressourcen auch technisch umsetzbar sind. Dadurch, dass weder Policy-Sprache noch eine Menge spezifizierbarer Rechte in [TWE+ 04] vorgegeben wurden, kann eine Integration in alle Ressourcen schwierig werden (Bewertung: 1).

- **Erweiterbarkeit:**
  Diese fehlende Festlegung bzw. die grundsätzliche Unterstützung beliebiger Sprachen macht den Ansatz natürlich sehr flexibel und damit gut erweiterbar (Bewertung: 3).

- **Middleware-übergreifende Interoperabilität:**
  MyProxy wird von den zwei Middleware-Technologien Globus und gLite unterstützt (Bewertung: 1).

- **Interoperabilität auf Policy-Ebene:**
  MyProxy ermöglicht mit der Erzeugung eines Restricted Proxy die Verwendung von Policies. Allerdings werden im Standard keinerlei Vorgaben bezüglich Syntax und Semantik der Policies gemacht. Das heißt, jede VO muss sich in einem nicht trivialen Abstimmungsprozess auf Policy-Sprache, Semantik und eine technische Umsetzung auf Seiten der Ressourcen einigen. Versteht ein Ressource-Provider die Policy nicht, muss er entweder den Proxy ablehnen oder Rechte entsprechend dem EEC vergeben (Rückfall auf GSI-Mechanismen) (Bewertung: 2).

- **Interoperabilität beim ID-Management:**
  Mit Hilfe des DN lassen sich individuelle Nutzer identifizieren. Durch die Übernahme des DN aus dem EEC bzw. des DN aus dem Proxy Zertifikat in das Issuer Feld des Proxies und eine Erweiterung um eine eindeutige Namenskomponente wird jeder Proxy eindeutig identifizierbar.

Theoretisch ließen sich damit Nutzer und auch Proxies auf individuelle Accounts abbilden. Letzteres setzt aber eine Einigung über die Namenserweiterung des Proxies voraus, die den Proxy eindeutig identifizierbar macht. Im Standard sind dazu keine Regelungen getroffen. Eine Abstimmung der Namensschemata ist nicht vorgesehen. Jeder Aussteller kann die Namenserweiterung seines Proxies selbst definieren und festlegen. In diesem Fall bleibt der Gastdomäne wieder nur die Information aus dem EEC, um Abbildungen auf lokale Accounts durchzuführen (Bewertung: 2).

- **Trust Management; Formalisierung:**
  Die Festlegung der Vertrauenswerte erfolgt implizit. Die Ressourcen-Provider vertrauen dem Inhaber des EEC oder dessen Aussteller, d.h. der entsprechenden CA. Die Vertrauenswerte sind binär repräsentiert (Bewertung: 1).
- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).
- **Trust Management; Granularität:**
  Theoretisch könnten Vertrauenswerte in Abhängigkeit jeder Entität und damit auch an Proxies vergeben werden. Praktisch werden Vertrauenswerte lediglich implizit auf Ebene der Institutionen (ausstellende CA; Heimat-Organisation des Inhabers des EEC) oder vielleicht noch auf Ebene der Nutzer festgelegt (Bewertung: 0).
- **Rechtedelegation; Granularität und Sicherheit:**
  Grundsätzlich nutzt auch MyProxy die Mechanismen des Mapping Files, d.h. eine Vergabe der Rechte in Abhängigkeit des lokalen Accounts des Inhabers des EEC ist immer möglich. Zusätzlich kann jeder Aussteller eines Proxy Zertifikates über eine Policy einen Restricted Proxy erzeugen und damit dessen Rechte einschränken. Zur Umsetzung muss sich der Aussteller und Delegierende jedoch auf die technische Durchsetzung innerhalb der Ziel-Domäne verlassen (können). Theoretisch lassen sich mit Hilfe der Policy abgestufte Rechte realisieren, praktisch werden Restricted Proxies wegen des erheblichen technischen und organisatorischen Aufwands kaum eingesetzt. Ein Widerruf einmal vergebener Rechte ist nicht vorgesehen. Hier wird mit der kurzen Lebenszeit (i.d.R. 12 bis 24 Stunden) argumentiert (Bewertung: 2).
- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Mit Hilfe des Attributs `pCPathLenConstraint` lässt sich die „Reichweite“ der Delegation einschränken. Mit Hilfe einer geeigneten Policy ließen sich auch andere Beschränkungen des Delegationsrechtes realisieren, z.B. um festzulegen, welche einzelnen Rechte weiter delegiert werden dürfen. Im Standard ist dafür allerdings nichts vorgegeben, d.h. dies müsste durch globale Vereinbarungen und eine eigene technische Umsetzung sichergestellt werden (Bewertung: 2).

- **Delegation von Policies:**
  Für die Delegation von Policies wird die Zertifikatserweiterung `ProxyPolicy` (vgl. Abbildung 4.27) spezifiziert. Damit ist es grundsätzlich möglich, dass der Aussteller eines Proxy Zertifikates eigene Policies spezifiziert und an die Ziel-Domäne übermittelt. Problematisch ist allerdings, dass diese Festlegung zu allgemein gefasst wurde. Es kann jede beliebige Policy-Sprache verwendet werden, solange sie von der Ziel-Domäne interpretierbar und technisch umsetzbar ist (Bewertung: 1).

- **Delegation von Aufgaben:**
  In MyProxy sind keine Mechanismen zur Delegation von Aufgaben vorgesehen (Bewertung: 0).

- **Mapping:**
  MyProxy bietet zwar die technischen Möglichkeiten Policies zu delegieren, allerdings werden keinerlei Festlegungen bezüglich eines Policy Mappings getroffen. Das Problem konkurrierender Policies wird nicht thematisiert (Bewertung: 0).

- **Skalierbarkeit:**
  Bei MyProxy handelt es sich um eine lokale, dezentrale Komponente, die grundsätzlich Teil jeder Middleware-Instanz ist. Insofern führt eine Vergrößerung des Gesamtsystems zu einer linearen Zunahme an MyProxy Diensten (Bewertung: 3).

- **Flexibilität; Entitätenvielfalt:**
  Im Subject Feld des DN eines X.509-Zertifikates sind Attribute für die Organisation (`/O`), Abteilung (Organizational Unit `/OU`) und Identität des Zertifizierten (Common Name `CN`) vorgesehen. Das Attribut `/OU` kann mehrfach angegeben werden. Der Common Name wird zur Identifikation von Nutzern und Ressourcen verwendet, daneben lässt sich die Organisation abbilden. Bei den Deutschen GridCAs wird unter Organization einheitlich `GridGermany` eingetragen und die organisatorische Zugehörigkeit wird als Organizational Unit kodiert [GridKA-CA, LRZ-RA, DFN-PKI]. (Bewertung: 2)

- **Flexibilität; Organisationsflexibilität:**
  MyProxy ist ausschließlich für die lokale Nutzung gedacht. (Bewertung: 0).

- **Administrierbarkeit:**
  Der Betriebsaufwand für MyProxy ist gering. Sollen Policies und Restricted Proxies eingesetzt werden, entsteht erheblicher technischer und organisatorischer Aufwand für die Umsetzung der Rechtebeschränkung. Die VO muss sich auf eine Syntax und Semantik einer Policy-Sprache einigen und es muss die Menge der delegierbaren bzw. beschränkbaren Rechte definiert werden. Diese Rechte bzw. die Beschränkungen der Rechte müssen dann auf den heterogenen Ressourcen auch technisch umsetzbar sein. Der erhebliche Aufwand, der bei der Einführung und

dem Betrieb von Restricted Proxies entsteht, zeigt sich auch in der nur sehr geringen Verbreitung dieses Konzeptes (Bewertung: 1).

- **Sicherheit; Sicherheitsniveau:**
  Für Proxy-Zertifikate existiert kein Widerrufsmechanismus vorgesehen, stattdessen wird die Lebenszeit der Proxies auf einen kurzen Zeitraum (i.d.R. 12 - 24 Stunden) festgelegt. Im Normalfall werden die Proxies als Impersonation Proxies, d.h. mit allen Rechten des Ausstellers, betrieben. Die Proxies werden in einer fremden Domäne betrieben und für einen böswilligen Administrator in dieser Domäne ist es relativ einfach möglich einen Proxy eines Benutzers zu übernehmen, den zugehörigen privaten Schlüssel zu stehlen und damit dann im Namen des Ausstellers zu handeln. Die Eintrittswahrscheinlichkeit ist deshalb als hoch, die Schadenshöhe durchaus als mittel einzustufen (Bewertung: 0).

- **Sicherheit; Zusicherung, QoP:**
  Der Nutzer erzeugt mit MyProxy einen kurzlebigen Stellvertreter, den er im Extremfall mit allen seinen Rechten ausstattet. Den dafür nötigen privaten Schlüssel, den der Proxy zur Authentisierung und als Nachweis seiner Rechte verwendet, überträgt der Nutzer in die Ziel-Domäne. Der Proxy kann den Schlüssel auch nutzen, um weitere Proxies zu erzeugen und diesen Rechte zu erteilen (vgl. Abbildung 4.26). Wünschenswert für den Nutzer wäre mindestens eine Festlegung im Sinne eines QoP Parameters, was die Ziel-Domäne für die Sicherung des kurzlebigen privaten Schlüssels tut, bzw. wie eine Proxy-Übernahme und damit ein Rechtediebstahl verhindert werden kann. Gegenwärtig ist jedoch nichts dergleichen vorgesehen (Bewertung: 0).

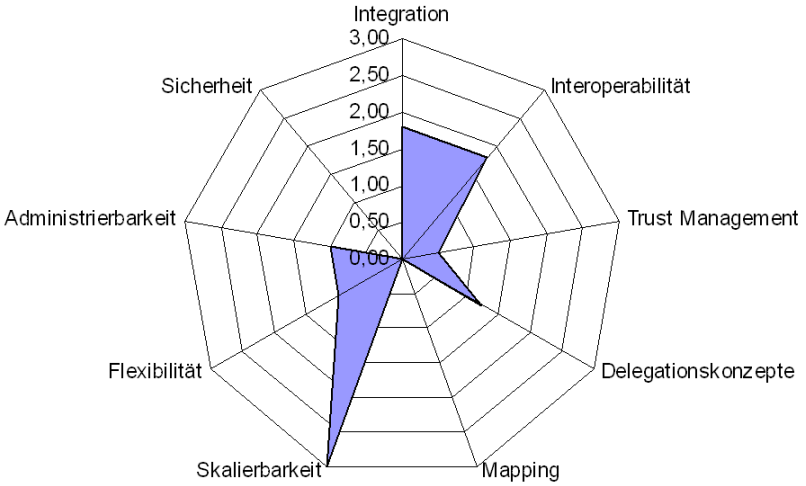

Abbildung 4.29: Bewertung MyProxy

Das Netzdiagramm in Abbildung 4.29 fasst die Ergebnisse der Bewertung von MyProxy zusammen.

4.3 VO-Management

Die virtuelle Organisation ist ein zentrales Konzept in Grids. Dementsprechend umfasst das VO-Management sehr viele verschiedene Managementaufgaben, z.B.: Unterstützung bei der Bildung von VOs, Instantiierung einer VO, Administration der VO selbst, Aufnahme und Löschen von Mitgliederung, Hinzufügen, Löschen und Ändern von Rollen, Hinzufügen und Entfernen von Ressourcen, SLA-Management u.a. Dabei ist es wichtig den gesamten VO-Lebenszyklus und nicht nur die Betriebsphase zu unterstützen [Schi 07].

Im Rahmen dieser Arbeit werden nicht alle Aufgaben des VO-Managements betrachtet, sondern nur diejenigen, die mit sicherheitsrelevanten Fragestellungen zusammenhängen. Dies betrifft zum einen natürlich die enge Koppelung mit Authentisierung und Autorisierung, zum anderen die zweifelsfreie Zuordnung von Nutzern und ggf. auch Ressourcen zu VOs. Die Vergabe von Rechten an eine VO und die Nutzung dieser Rechte durch VO-Mitglieder ist ebenfalls eine Frage des VO-Managements. In diesem Abschnitt werden VO-Management Mechanismen vorgestellt. Diese werden eingeführt und erläutert, um sie dann mit Hilfe des in Kapitel 3 vorgestellten Kriterienkatalogs zu bewerten. Eine Defizit- und Schwachstellenanalyse aller bewerteter Mechanismen erfolgt in Abschnitt 4.10.

Im Abschnitt 4.3.1 wird der Virtual Organisatzion Membership (VOMS) Service als ein weit verbreitetes Verfahren zur Verwaltung von VO-Informationen vorgestellt. VOMS ist ein zentralistischer Ansatz, d.h. für jede VO im Grid ist ein VOMS Server zu betreiben.

Kapitel 4.3.3 stellt Konzepte vor, die versprechen dezentral einsetzbar zu sein. Ein besonderer Aspekt wird dabei auf die Nutzung von Techniken des föderierten Identitätsmanagements (FIM) gelegt. Mit Hilfe von FIM kann die Heimatdomäne als (alleinige) autoritative Quelle für Nutzerinformationen verwendet werden. Es werden Shibboleth als grundlegende FIM Technologie sowie GridShib, das Shibboleth verwendet und an die Gegebenheiten des Grid adaptiert, eingeführt. Mit MyVocs wird ein Konzept vorgestellt, das wiederum GridShib nutzt, um ein VO-Management System zu realisieren.

4.3.1 Virtual Organization Membership Service (VOMS)

Der **Virtual Organization Membership Service (VOMS)** [ACC+ 04, ACC+ 05, Czyz 06] wurde im Rahmen des DataGRID Projektes (auch als

European DataGrid (EDG) bezeichnet) [DataGRID] entwickelt, um eine adäquate Lösung zur Bildung von VOs zu haben. In Globus wurde bis dahin eine VO allenfalls implizit über Zertifikate und den Distinguished Name (DN) darin, gebildet.

VOMS als Attribute Authority

Jede VO betreibt einen (oder mehrere) zentrale VOMS Server, die als **Attribute Authority (AA)** fungieren. Der VOMS Server verbindet einen Nutzer mit seinen VO–Attributen.

Gruppenstruktur als Graph

Der VO–Begriff, der VOMS zugrunde liegt, geht von einer hierarchischen Struktur aus. Das heißt, eine VO wird aus Gruppen und Subgruppen gebildet. Eine VO mit ihren Gruppen und Subgruppen lässt sich als gerichteter azyklischer Graph darstellen. Die Gruppen werden durch Knoten und die Gruppen–Subgruppen–Relation durch gerichtete Kanten dargestellt. Ein Nutzer kann in mehreren Gruppen Mitglied sein. Die Mitgliedschaft in einer Gruppe impliziert die Mitgliedschaft in allen in der Hierarchie höherstehenden Gruppen. Innerhalb einer Gruppe kann ein Nutzer unterschiedliche Rollen einnehmen. Dabei werden Rollen auf Subgruppen vererbt. Ein Nutzer in einer Gruppe G_i mit einer bestimmten Rolle R hat diese Rolle in allen Subgruppen G_j mit $j > i$ inne. Der Umkehrschluss gilt im Allgemeinen nicht. Zur Erhöhung der Flexibilität wird auch noch die Möglichkeit gegeben spezielle Charakteristiken oder allgemeine Informationen über den Nutzer in einer so genannten Capability zu beschreiben. Der Begriff Capability ist unglücklich gewählt. Ein **Capability** bezeichnet in der üblichen Verwendung eine Datenstruktur, die Rechte enthält, fälschungssicher ist und über unsichere Kommunikationskanäle ausgetauscht werden kann. Im Zusammenhang mit VOMS ist hier ein beliebiger Freitext (ohne Leerzeichen) gemeint. Diese Charakterisierung des Nutzers wird verwendet, um auf Seite des Ressource-Providers feingranular Rechte auf Grund dieser Capability vergeben zu können.

FQAN: Datenstruktur für VO-Mitgliedschaft

Die Mitgliedschaft eines Nutzers in einer VO wird durch das folgende Tupel beschrieben: (VO-Bezeichner, Gruppe, Rolle, Capability), das durch einen **Fully Qualified Attribute Name (FQAN)** technisch umgesetzt wird [Cias 04, FrCi 04]. Der FQAN hat folgende Form [ACC$^+$ 05]:

/VO[/group[/subgroup(s)]][/Role=role][Capability=cap]

Ein Benutzer kann Mitglied in mehreren VOs und in mehreren Gruppen und Subgruppen innerhalb einer VO sein. Ein Nutzer kann auch mehrere Rollen und Capabilities haben. Diese Informationen werden durch mehrere FQANs repräsentiert. Abbildung 4.30 stellt ein fiktives Beispiel für einen Nutzer der AstroGrid Community dar, der in der Gruppe IVOA (International Virtual Observatory Alliance) und in der Subgruppe `production` Mitglied ist.

technische Realisierung

Die VO-, Gruppen- und Rollenzugehörigkeit wird in Verzeichnis-Diensten oder Datenbanken gespeichert. Die Mitgliedschaft in einer VO wird vom VOMS Server technisch durch ein **Attributzertifikat** nach [FaHo 02, HPFS 02] realisiert, das dem Benutzer seine Attribute zuweist. Ein Attributzertifikat ist die Erweiterung eines Standard X.509-Zertifikates um zusätzliche Datenstrukturen mit ggf. eigener Signatur. Für VOMS wurden

```
/astrogrid–d.d–grid.de
/astrogrid–d.d–grid.de/IVOA
/astrogrid–d.d–grid.de/IVOA/production
/astrogrid–d.d–grid.de/Role=VO–Admin
/astrogrid–d.d–grid.de/IVOA/production/Role=Admin
/astrogrid–d.d–grid.de/IVOA/Capability=long–jobs
/astrogrid–d.d–grid.de/Capability=large–space
```

Abbildung 4.30: Beispiel für VOMS Attribute eines Nutzers

VOMS Erweiterungen

zwei nicht kritische Erweiterungen definiert, d.h. eine Anwendung, welche die VOMS Erweiterungen nicht interpretieren kann, ist in der Lage das Proxy-Zertifikat in der gewohnten Weise zu nutzen. Damit ist eine Abwärtskompatibilität sichergestellt. Die Extension 1 ist in Tabelle 4.4 dargestellt und definiert das eigentliche VOMS Attributzertifikat. Der Datentyp n bezeichnet dabei eine String–Repräsentation einer Zahl, s steht für einen String und t für eine ASN.1 Repräsentation eines Zeitpunktes. Ein Attributzertifikat kann mehrere Struktur–Blöcke (Datenbereich von SIGLEN bis DATA) beinhalten. Damit kann der Nutzer verschiedene VOMS Server kontaktieren und alle Daten gesammelt in seinem Proxy–Zertifikat zusammenfassen.

Benutzer-eigene Daten im Zertifikat

Die Extension 2 (vgl. Tabelle 4.5) soll es dem Benutzer ermöglichen seinem Proxy–Zertifikat eigene Daten hinzuzufügen. In dieser Erweiterung könnte bspw. ein Kerberos Ticket abgelegt werden, mit dem sich der Nutzer bei einer bestimmten Domäne authentisieren kann. Der Inhalt dieses Erweiterungsfeldes ist kein Ergebnis einer VOMS–Abfrage, sondern wird vom Nutzer selbst eingefügt.

architekturelle Komponenten

Abbildung 4.31 stellt die architekturellen Komponenten von VOMS Server und Client dar. Der VOMS Server unterteilt sich in administrative Komponenten und solche für den Endbenutzer. Der Benutzer wird über die Standard GSI-Mechanismen (d.h. ein Proxy–Zertifikat; vgl. Abschnitt 4.1.6 und 4.2.10) beim VOMS Server authentisiert und erhält von diesem einen VOMS–Proxy mit dem entsprechenden Attribut–Zertifikat. Die beiden anderen Komponenten (`edg-voms-admin` und `voms-httpd`) dienen der Administration des VOMS Servers und der Aktualisierung der lokalen Grid Map Files.

Auf Client–Seite dient das Kommando `voms-proxy-init` der Erzeugung eines Proxies und der Authentisierung. Der Administrator des VOMS Servers hat entweder ein Kommandozeilenwerkzeug (`edg-voms-admin`) zur Administration des Servers oder er kann einen Web-Browser benutzen [Froh 04a, Froh 04b].

Zur lokalen Autorisierung werden auch bei VOMS Grid Map Files genutzt. Mit dem Kommando `edg-mkgridmap` können die notwendigen Daten beim VOMS Server abgefragt und automatisch ein Grid Map File erzeugt werden. Die Daten werden also auch hier lokal administriert. Allerdings be-

Attribut	Datentyp bzw. Belegung	Erläuterung
Name:	voms-attribute	Schlüsselwort für VOMS Erweiterung
OID:	1.3.6.1.4.1.8005. 100.100.1	Object ID für VOMS im Internet–Registrierungsbaum (1.3.6.1.4.1.8005 ist der Enterprise Teilbaum des EDG)
Struktur		
SIGLEN:	n	Länge der VOMS Signatur in Bytes
SIGNATURE:	s	VOMS Signatur
USER:	s	DN aus dem Zertifikat des Benutzers
UCA:	s	DN der CA, die das Nutzer–Zertifikat ausgestellt hat
SERVER:	s	DN des VOMS Server Zertifikats
SCA:	s	DN der CA, die das Server Zertifikat ausgestellt hat
VO:	s	Name der VO zu welcher der Server gehört
TIME1:	t	Startzeitpunkt der Gültigkeit der VOMS Attribute
TIME2:	t	Endzeitpunkt der Gültigkeit der VOMS Attribute
DATALEN:	n	Läge des Datenfeldes
DATA		Attribut–Daten, in der Form `<Attributname> : <Attributbelegung>`, z. B. GROUP: s ROLE: s CAP: s

Tabelle 4.4: VOMS Extension 1 für Proxy–Zertifikate (nach [FrCi 04])

steht mit dem VOMS Server die Möglichkeit ein zentrales Grid Map File zu verteilen. Durch die Abwärtskompatibilität zum Grid Map File ist keine feingranulare Autorisierung möglich.

Aus Sicht des Benutzers sind für die Autorisierung folgende Schritte zu durchlaufen: Nutzer-Sicht

1. Nutzer und VOMS Server authentisieren sich gegenseitig mit Hilfe ihrer Zertifikate. Der Nutzer verwendet hierzu das Kommando `voms-proxy-init` welches das Kommando `grid-proxy-init` ersetzt.

2. Der Nutzer schickt eine signierte Anfrage an den VOMS Server.

3. Dieser verifiziert die Identität des Nutzers und erzeugt ein vom VOMS Server signiertes Attributzertifikat mit den Daten des Nutzer.

4. Der Nutzer prüft die Gültigkeit dieser Daten.

Attribut	Datentyp bzw. Belegung	Erläuterung
Name:	IncFile	Schlüsselwort für VOMS Erweiterung
OID:	1.3.6.1.4.1.8005. 100.100.2	Object ID für VOMS Extension 2
Struktur		
	Byte–Sequenz	

Tabelle 4.5: VOMS Extension 2 für Proxy–Zertifikate (nach [FrCi 04])

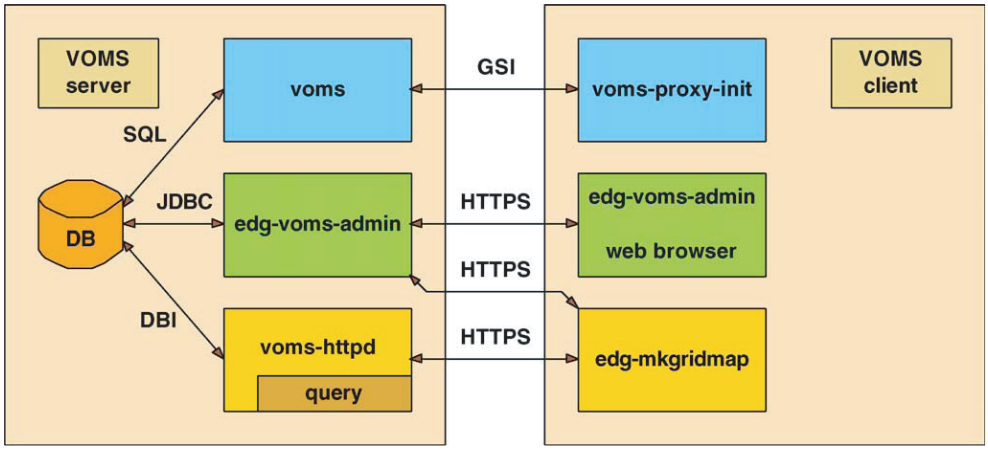

Abbildung 4.31: VOMS — Architekturkomponenten [ACC+ 05]

5. Der Nutzer kann diesen Prozess mit mehreren VOMS Servern wiederholen.

6. Aus allen, von den VOMS Servern erhaltenen Attributzertifikaten erzeugt der Nutzer ein Proxy Zertifikat, das als unkritische Erweiterung die VOMS Zertifikate enthält.

Auch bei VOMS gibt es nach wie vor die Möglichkeit verschiedene Nutzer dynamischen Pool Accounts zuzuordnen. Wird diese Möglichkeit genutzt, gelten für VOMS dieselben Probleme bezüglich Zurechenbarkeit, Datensicherheit und mangelnder Abschottung der Nutzer voreinander wie im Abschnitt 4.2.5 beschrieben.

4.3.2 Bewertung VOMS

VOMS stellt ein weit verbreitetes Verfahren zum VO-Management dar, das im folgenden Abschnitt entsprechend der Kriterien aus dem Kriterienkatalog (vgl. Abschnitt 3.2) bewertet wird.

- **Middleware-Integration:**
  VOMS ist als Zusatzmechanismus in Globus und gLite integrierbar und

deshalb jeweils mit 2 zu bewerten. UNICORE unterstützt VOMS nicht (UNICORE; Bewertung:0) (Middleware-Integration; Bewertung: 1,33).

- **Ressourcen-Integration:**
  VOMS führt zur Spezifikation von Nutzer-Attributen den FQAN ein. Eine Umsetzung des FQAN in Autorisierungspolicies des lokalen Systems ist in der Regel möglich. Für die Umsetzung stützt sich VOMS voll auf den in Abschnitt 4.2.7 bewerteten Mechanismus des Grid Map Files ab. Für den Fall, dass die VOMS-Erweiterungen des X.509-Zertifikates vom lokalen System nicht verstanden werden, erfolgt ein Rückfall auf die Mechanismen des Grid Map Files (Ressourcen-Integration; Bewertung: 2)
- **Erweiterbarkeit:**
  Die Extension 2 (vgl. Tabelle 4.5) kann vom Nutzer verwendet werden, um seinem Proxy Zertifikat beliebige Daten hinzuzufügen. Das in der Literatur verwendete Beispiel für eine derartige Erweiterungen sind Kerberos Tickets zur Autorisierung. Die Erweiterbarkeit ist damit sehr einfach möglich. Allerdings erfordert jede Erweiterung über diesen Mechanismus eine VO-weite Abstimmung und ggf. Erweiterungen oder Ergänzungen der lokalen Ressourcen (Bewertung: 3).
- **Middleware-übergreifende Interoperabilität:**
  VOMS wird von Globus und gLite unterstützt (Bewertung: 1).
- **Interoperabilität auf Policy-Ebene:**
  VOMS spezifiziert das Format des FQAN, der als Basis für die Spezifikation von Autorisierungspolicies verwendet werden kann. Die Spezifikation der lokalen Autorisierungspolicies erfolgt über das Mapping File, dessen Erstellung allerdings über den VOMS Server zentralisiert werden kann. Das Capability Attribut des FQAN kann, wie die Beispiele aus der Spezifikation [FrCi 04] zeigen, auch zum Zweck der Autorisierung verwendet werden. Allerdings werden hier keinerlei Vorgaben bezüglich Syntax und Semantik der Attribut-Belegung gemacht. Darauf muss sich jede VO in einem nicht trivialen Abstimmungsprozess einigen (Bewertung: 2).
- **Interoperabilität beim ID-Management:**
  Mit Hilfe des FQAN von VOMS lassen sich individuelle Nutzer identifizieren und auf individuelle Accounts abbilden. Die Abbildung von lokalen Accounts auf Grid Accounts erfolgt zentral auf Ebene des VOMS (Bewertung: 2).
- **Trust Management; Formalisierung:**
  Die Festlegung der Vertrauenswerte erfolgt implizit. Innerhalb einer VO vertrauen alle Entitäten dem VOMS Server. Die Vertrauenswerte sind binär repräsentiert (Bewertung: 1).
- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).

- **Trust Management; Granularität:**
  Vertrauenswerte werden lediglich implizit auf Ebene der Institutionen (Ressource-Provider und VOMS Server) festgelegt (Bewertung: 0).

- **Rechtedelegation; Granularität und Sicherheit:**
  Bei der Betriebsart des VOMS, die standardmäßig vorgesehen ist, erzeugt der VOMS Server die Grid Map Files für alle Ressourcen-Provider, d.h. diese übertragen alle Rechte an ihren Ressourcen auf den VOMS Server. Sollen dem VOMS Server die Rechte zur Autorisierung der lokalen Ressourcen wieder entzogen werden, muss der Ressourcen-Provider den Mechanismus zur Übernahme des vom VOMS Server generierten Grid Map Files deaktivieren und ein eigenes Grid Map File anlegen. Im Gegensatz zum CAS (vgl. Abschnitt 4.2.9) können die Mitglieder der VO Ressourcen weiter nutzen auch wenn dem VOMS Server die Rechte zur Autorisierung entzogen wurden. Aber auch beim VOMS Server ist eine feingranularere Delegation abgestufter Rechte nicht vorgesehen (Bewertung: 1).

- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Eine Weiterdelegation des Rechte des VOMS an andere ist nicht vorgesehen (Bewertung 0).

- **Delegation von Policies:**
  Für die Delegation lokaler Policies von den Ressourcen-Providern an den VOMS Server sind keine Mechanismen vorgesehen. Es besteht entweder die Möglichkeit die Autorisierungsentscheidungen zentral treffen zu lassen oder die Informationen des VOMS Server für eine lokale Umsetzung der Autorisierung zu verwenden. Eine Delegationsmechanismus ist nicht vorgesehen (Bewertung: 0).

- **Delegation von Aufgaben:**
  Bei VOMS kann die Erzeugung des Grid Map Files von den Ressource-Providern an den VOMS Server delegiert werden. Die Delegation ist aber nur für diese eine Aufgabe möglich (Bewertung: 1).

- **Mapping:**
  Da weder Konzepte zur Delegation von Rechten noch von Policies vorgesehen sind, ist das Mapping mit 0 zu bewerten.

- **Skalierbarkeit:**
  Die räumliche Verteilung der Ressourcen hat kaum Einfluss auf VOMS. Werden dem Grid neue Ressourcen hinzugefügt, müssen für diese nur dann Autorisierungspolicies beim VOMS Server spezifiziert werden, wenn nicht bereits Ressourcen der selben Klasse existieren. Bei der Zunahme organisatorischer Einheiten kann von einem linearen Zusammenhang ausgegangen werden. Das Konzept des VOMS Servers sieht eine Lastverteilung auf mehrere physische Server bereits vor (Bewertung: 3).

- **Flexibilität; Entitätenvielfalt:**
  Mit Hilfe des FQAN lassen sich VOs, Gruppen, Subgruppen und Rollen

abbilden. Durch die Integration der VOMS Erweiterungen in ein Standard X.509-Zertifikat lassen sich auch Informationen über Nutzer und Organisationen abbilden (Bewertung: 3).

- **Flexibilität; Organisationsflexibilität:**
  Der VOMS Server ist für den zentralen Betrieb gedacht. Wird der Dienst zur Erzeugung des Grid Map Files nicht genutzt, sondern eine lokale Umsetzung realisiert, kann der Ressourcen-Provider auch eine rein lokale Umsetzung realisieren. Beide Mechanismen lassen sich in einer VO parallel betreiben. Jeder Ressourcen-Provider muss sich festlegen, ob er den zentralen Dienst nutzen oder seine Entscheidungen lokal treffen will (Bewertung: 2).

- **Administrierbarkeit:**
  Der Betriebsaufwand ist durch den zentralistischen Ansatz eigentlich gering. Im Normalbetrieb sind nur beim VOMS Server Anpassungen nötig und es sind bei Änderungen neue Mapping Files zu generieren. Bei einer zentralen Erzeugung der Mapping-Dateien entsteht aber erheblicher Aufwand durch die Abstimmung der lokalen Kennungen, auf die abgebildet werden soll. Hier besteht das Problem, dass globale Schemata, die von der VO bzw. VOMS vorgegeben werden, den lokalen Richtlinien und den lokalen I&AM Lösungen widersprechen und deshalb nicht unterstützt werden können (Bewertung: 1).

- **Sicherheit; Sicherheitsniveau:**
  Ein Angreifer, der in der Lage ist eigene VOMS Attribute zu erstellen, kann damit unberechtigt erhebliche Rechte innerhalb der gesamten VO erlangen. Die Schadenshöhe ist deshalb als hoch einzustufen. Das Konzept des zentralen Servers bietet aber auch den Vorteil, dass dieser sehr gezielt mit sehr effektiven Mitteln geschützt werden kann. Unter der Annahme, dass der VOMS Server besonders geschützt wird, kann die Eintrittswahrscheinlichkeit für einen Angriff als niedrig angenommen werde (Bewertung: 1).

- **Sicherheit; Zusicherung, QoP:**
  Der VOMS Server arbeitet als Attribute Authority für eine oder mehrere VOs. Er vergibt als CA Attribut-Zertifikate, welche die VO-, Gruppen- und Subgruppen- sowie die Rollenzugehörigkeit festlegen können. Für diese zentrale und kritische Infrastrukturkomponente wäre eine Überprüfung oder Zusicherung der Einhaltung eines bestimmten Sicherheitsniveaus sehr wichtig. Allerdings sind hierfür keinerlei Mechanismen vorgesehen und in den Spezifikationen zum VOMS werden dazu auch keine Aussagen gemacht (Bewertung: 0).

Abbildung 4.32 fasst die Ergebnisse der Bewertung von VOMS für die 1. Ebene der Kriterien des Kriterienkatalogs (vgl. Abbildung 3.2) in einem Netzdiagramm zusammen.

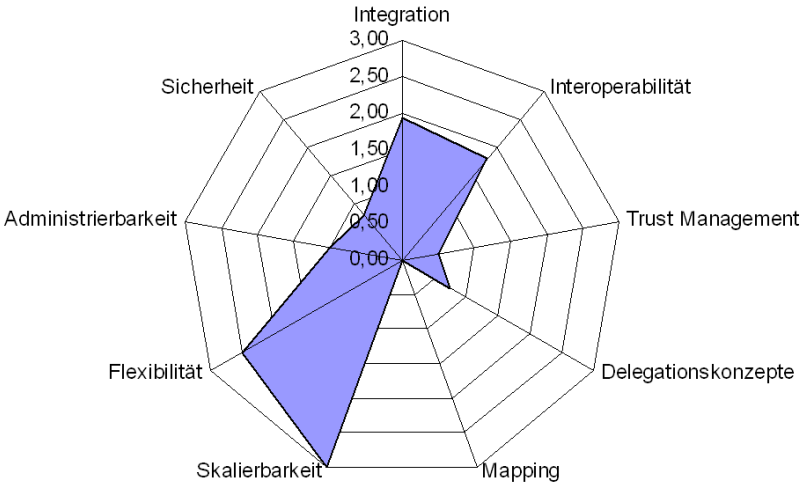

Abbildung 4.32: Bewertung VOMS

4.3.3 VO-Management mit FIM Techniken

GridShib ist ein Globus-Projekt, um das föderierte Identitätsmanagementsystem (FIM) Shibboleth in die Grid Middleware Globus zu integrieren [GridShib]. Das Grundprinzip von FIM ist es die Heimatdomäne eines Nutzers als autoritative Quelle für Identitätsinformationen, aber auch für Autorisierungsinformationen zu nutzen [Homm 07].

Im folgenden werden die Grundprinzipien von Shibboleth und GridShib dargestellt, um dann die Nutzung als VO-Management System zu beschreiben und zu bewerten.

Shibboleth

Shibboleth wurde im Rahmen von Internet2 als Single Sign On und Autorisierungslösung für webbasierte Anwendungen entwickelt. Das Organisationsmodell von Shibboleth unterscheidet Service- und Identitätsprovider.

Identitätsprovider als autoritative Quelle für Nutzerinformation

Der **Service Provider (SP)** stellt einen Dienst für Nutzer bereit. Der **Identitätsprovider (IdP)** ist verantwortlich, die Nutzer aus seiner Organisation zweifelsfrei zu authentisieren und dem SP Identitäts- und ggf. Autorisierungsinformationen zu übermittlen. Der IdP liefert als **Attribute Authority (AA)** Benutzerattribute an den SP.

Ablauf einer FIM basierte Authentisierung

Abbildung 4.33 stellt den (vereinfachten) Ablauf einer Browser-basierten, durch Shibboleth authentisierten Dienstnutzung dar. Anhand dieser Darstellung werden die für Shibboleth notwendigen Komponenten eingeführt:

1 Der Benutzer ruft einen Web-Dienst des SP auf.

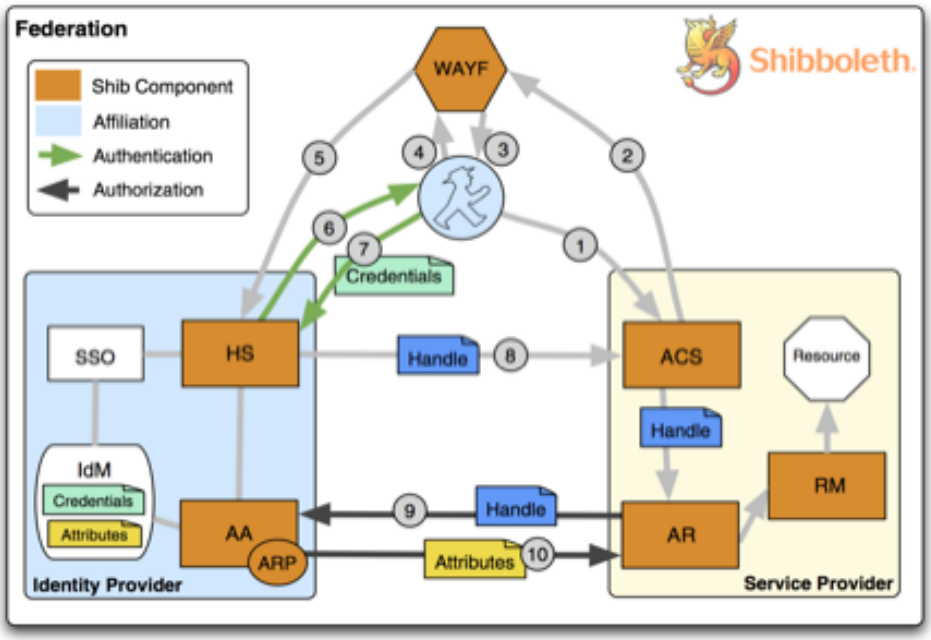

Abbildung 4.33: Shibboleth Transaktionen [MSZ 07]

2 – 4 Die Anfrage wird an einen, für die Föderation zentralen, **Where Are You From (WAYF)** Dienst weitergeleitet. Der WAYF dient der Ermittlung der Heimatdomäne des Benutzers und bietet Mechanismen, um den Benutzer zur Authentisierung mit seiner Heimatdomäne zu verbinden. Die Auswahl der Heimatdomäne erfolgt i.d.R. durch eine Benutzereingabe beim WAYF.

5 Der Nutzer wird zum **Handle Service (HS)** seiner Heimatdomäne umgeleitet. Der Handle Service ist für die lokale Authentisierung zuständig und nutzt dafür lokale Identity Managementsysteme (IdM).

6 – 7 Der Nutzer wird auf sichere Art und Weise über die in seiner Heimatdomäne eingesetzten lokalen Authentisierungsverfahren authentisiert.

8 Der Handle Service erzeugt ein eindeutiges Handle für den Benutzer und übermittelt dieses direkt an den Service Provider. Das Handle beinhaltet Informationen wie der SP die Attribute Authority (AA) des IDP kontaktieren kann. Der **Assertion Consumer Service (ACS)** überprüft das Handle und übergibt es an den **Attribute Requestor (AR)**

9 Der AR verwendet das Handle um den AA des IdP zu kontaktieren und Attribute des Nutzers abzufragen.

10 Die AA liefert unter Berücksichtigung der Attribute Release Policy (ARP) eine Attribute Assertion mit den Informationen an den AR zurück. Mit Hilfe dieser Attribute kann der **Ressource Manager (RM)** dann entscheiden, ob der Benutzer den angeforderten Dienst benutzen darf.

Für die technische Umsetzung des Handles und Assertions wird die Security Assertion Markup Language (SAML) verwendet.

GridShib

GridShib [GridShib] wurde als Plugin für Globus Toolkit 4 (GT4) entwickelt, um Shibboleth Mechanismen in die Globus Middleware integrieren zu können. Dabei wurde aus Kompatibilitätsgründen auf die weitestgehende Beibehaltung der Zertifikatsbasierten GSI (vgl. Abschnitt 4.1.6) geachtet.

Beibehaltung der zertifikatsbasierten Authentisierung

Globus wird um eine Attribute Request Komponente zur Abfrage einer AA beim IdP und einem Policy Decision Point (PDP) zur Auswertung der Informationen aus den Assertions, erweitert. Durch die Beibehaltung der Zertifikatsbasierten Authentisierung kann nicht unmittelbar das im vorigen Abschnitt beschriebene Verfahren der Authentisierung beim Dienstaufruf verwendet werden. Assertion Consumer Service und Attribute Requestor können auseinanderfallen. Stattdessen wird eine so genannte **GridShib CA** eingeführt, die als ACS in Kooperation mit dem IdP des Nutzers, für den Nutzer kurzlebige X.509 Zertifikate, so genannte **Short Lived Credentials (SLC)** ausstellt, die dann vom Nutzer für die Authentisierung beim Globus Dienst verwendet werden. Die SLCs haben eine maximale Gültigkeitsdauer von einer Million Sekunden, was rund 11 Tagen entspricht.

GridShib CA erstellt Short Lived Credentials

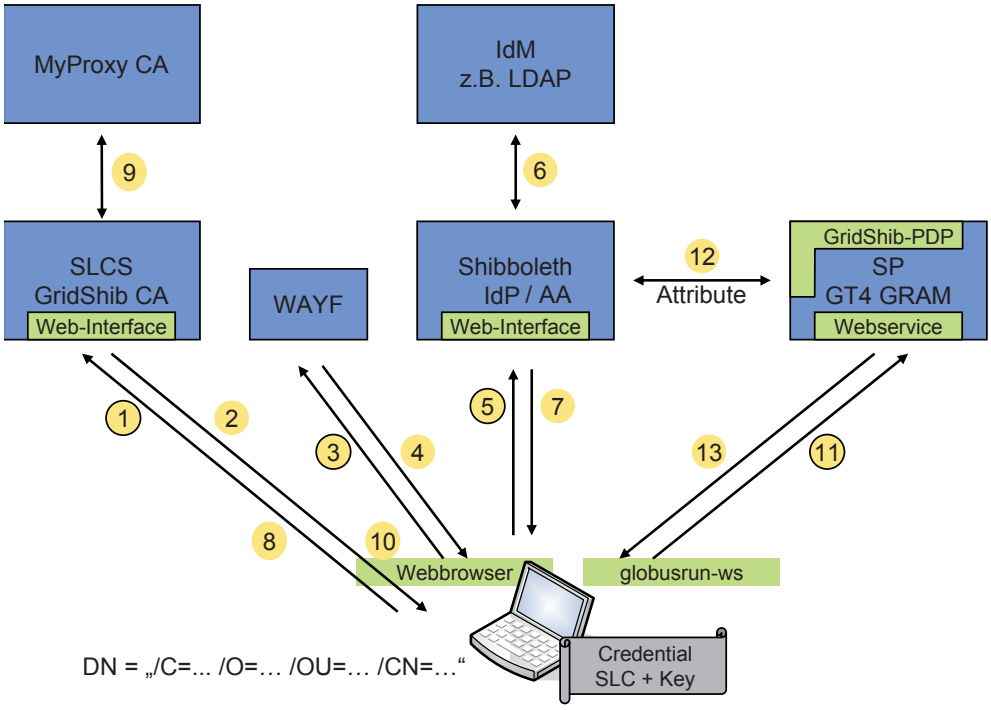

Abbildung 4.34: Komponenten und Ablauf einer GridShib basierten Authentisierung [GrGr 02]

Abbildung 4.34 fasst die Komponenten und den Ablauf einer GridShib basierten Authentisierung zusammen.

1-5 Der Nutzer lässt sich von der GridShib CA, welche die ACS Komponente darstellt, ein SLC ausstellen. Die Schritte 1 bis 5 aus der Abbildung entsprechen denselben Schritten aus dem allgemeinen Shibboleth Ablauf, der in Abbildung 4.33 dargestellt ist. Das heißt, der Nutzer wird über den WAYF zu seiner Heimatdomäne bzw. seinem IdP verwiesen.

6 Bei seinem IDP authentisiert sich der Nutzer über lokale Mechanismen.

7 Der IdP erzeugt ein Handle das aber wegen der möglichen Trennung von ACS und AR nicht direkt an den ACS (vgl. Schritte 8 und 9 in Abb. 4.33) sondern an den Nutzer zurückgegeben wird (Schritt 7 in Abb. 4.34).

8-10 Der Nutzer verwendet das Handle um sich von der GridShib CA ein SLC ausstellen zu lassen.

11 Mit Hilfe dieses SLC kann er den eigentlichen Dienst nutzen.

12 Der Service Provider kann über den AA des IdP ggf. weitere Attribute des Nutzers abfragen und diese für die Zugriffskontrollentscheidung im GridShib PDP zu nutzen.

Trennung von ACS und AR: AA muss lokalisiert werden

Damit der SP den AA finden kann, benötigt er Informationen, die er bei Shibboleth direkt aus dem Handle des ACS entnehmen kann (vgl. Abb. 4.33). Da in GridShib dieses Handle aber an die GridShib CA übertragen wird, müssen die Informationen über das SLC übermittelt werden. Hier werden zwei Alternativen vorgeschlagen:

1. Die Kodierung des AA innerhalb des DN des Nutzers.
2. Die Übertragung einer SAML `authn` Assertion im SLC.

Beide Alternativen stellen eine private Zertifikatserweiterung (Extension) des X.509 Zertifikates dar, die einer Abstimmung bedürfen. Die zweite Alternative ließe sich leichter umsetzen, da die SLCs eine deutlich kürzere Lebensdauer als Nutzerzertifikate haben und eine X.509 Extension für `authn` einfacher und auch schneller einzufügen wäre. Die erste Alternative stellt eine Verwendung des DN dar, die eigentlich im Standard gar nicht so vorgesehen ist. In einem bestehenden Grid müssten sich alle Partner auf das Format der Kodierung der AA im DN einigen, und alle Nutzer müssten dann neue Zertifikate beantragen. Dies ist für große produktive Grids nicht realistisch.

MyVocs für das VO-Management

GridShib ist primär als System zum föderierten Identitätsmanagement und nicht vorrangig als VO-Management System entwickelt worden. Es gibt jedoch Projekte, die basierend auf GridShib versuchen einen VO-Management Dienst zu realisieren. Im folgenden soll **MyVocs** [MyVOCS, GRSW 06] als exemplarischer Vertreter kurz erläutert werden.

Die Grundidee von MyVocs ist die Heimatdomäne als IdP, und zwar ausschließlich zur Authentisierung des Nutzers zu verwenden. Als AA für eine bestimmte VO wird ein MyVocs Server verwendet, d.h. die GridShib CA

erhält die benötigten VO-Informationen nicht vom IdP, sondern von MyVocs Server. Der MyVocs Server wird durch einen VO-Admin verwaltet und beinhaltet alle VO relevanten Informationen der VO-Mitglieder.

Sowohl GridShib und insbesondere MyVocs sind noch sehr junge Projekte und dementsprechend noch am Anfang ihrer Entwicklung. Die architekturellen Grundkonzepte sind spezifiziert allerdings fehlt für einige Komponenten die Feinspezifikation und die technische Umsetzung.

4.3.4 Bewertung GridShib

allgemeine Schwächen von Shibboleth

Shibboleth wurde ursprünglich als Single Sign On Lösung für Web-basierte Anwendungen entwickelt und hat als solches einige Schwachstellen (vgl. [HoRe 05a, HoRe 05b]), die auch bei der Integration ins Grid mittels GridShib bestehen bleiben:

- Syntax und Semantik von Attributen zugrundeliegender Datenschemata sind verschieden und es ist deshalb eine Grid-weite Abstimmung auf ein föderationsweites Datenschema erforderlich.
- Datenaktualität: Der Austausch von Attributen zwischen AA und AR ist nur während der Dienstnutzung möglich. Eine Rückfrage zu einem späteren Zeitpunkt ist nicht vorgesehen. Sollen die Daten des Nutzers auch für spätere Abrechnungszwecke verwendet werden, kann der Fall eintreten, dass mit veralteten Daten gearbeitet wird und der Ressourcen-Provier keine Möglichkeit hat sich über einen Attribute Request aktuelle Daten zu beschaffen.
- Berücksichtigung von VO-Informationen: Shibboleth und auch GridShib gehen von einem Organisationsmodell aus, das Service Provider und Identity Provider beinhaltet. Die Bildung und Verwaltung von VOs ist nicht vorgesehen und in den Schema-Definitionen auch nicht unterstützt. Durch Erweiterungen wie MyVocs und die Einigung auf global eindeutige Schema Definitionen lässt sich dieser Mangel beheben. Allerdings sind in dieser Richtung bisher kaum Anstrengungen unternommen worden und deshalb auch keine Erfahrungen bekannt.

Im folgenden wird GridShib anhand des Kriterienkataloges aus Kapitel 3 bewertet.

- **Middleware-Integration:**
  GridShib wird derzeit als Globus GT4 Projekt als Middleware-Erweiterung entwickelt. Eine spätere Integration direkt in die Middleware ist möglich, derzeit aber noch nicht abzusehen (Globus; Bewertung: 2). Die anderen Middlewares werden nicht unterstützt (gLite und UNICORE, Bewertung: 0; Middleware-Integration; Bewertung: 0,66).
- **Ressourcen-Integration:**
  GridShib verwendet zur Authentisierung SLCs, die sich strukturell nur

durch ihre kürzere Gültigkeitsdauer von GSI Zertifikaten unterscheiden. Eine Verwendung von Attribute Assertions zur Autorisierung ist zwar vorgesehen, wie dieses Konzept mit Hilfe von Globus und lokalen Ressourcen auf Zugriffskontrollentscheidung auf Ebene der Ressourcen umgesetzt werden soll, ist aber noch nicht geklärt. Eine Verwendung des in Abschnitt 4.2.7 vorgestellten Grid Map Files ist aus Gründen der Abwärtskompatibilität vorgesehen. Shibboleth und damit GridShib beschränken sich bisher auf Web Services. Für Legacy Systeme, die nicht als Web Service implementiert sind, ist eine Anbindung derzeit schwierig (Bewertung: 1).

- **Erweiterbarkeit:**
  Durch die föderativen Mechanismen und das Konzept der Übertragung beliebiger Attribute mit Hilfe von Attribute Assertion sind wohldefinierte Schnittstellen für die Integration weiterer Mechanismen gegeben (Bewertung 3).

- **Middleware übergreifende Interoperabilität:**
  GridShib wird derzeit nur von Globus Version 4 unterstützt. Im Rahmen des D-Grid Projektes und hier insbesondere vom Teilprojekt IVOM [ZiGr 06, IVOM] wurde ein Konzept für die Integration von Shibboleth in die anderen Middlewares entwickelt [GGG+ 07], aber derzeit gibt es dafür noch keine konkrete Realisierung (Bewertung 0).

- **Interoperabilität auf Policy-Ebene:**
  Shibboleth stützt sich für die Formalisierung der Handles auf SAML. Die Sprache. mit der Handles erstellt werden, ist formalisiert. Allerdings wurde in [HoRe 05b, HoRe 05a] gezeigt, dass für eine global eindeutige Nutzung eigentlich ein gemeinsames Daten-Schema, bzw. Profile notwendig wäre. Dazu ist jedoch eine Abstimmung bezüglich Syntax und Semantik der Daten-Schemata aller Beteiligter nötig, die eine Anpassung auch der Attribute der lokalen Identity Managementsysteme erforderlich machen würden. Da ein solches Vorgehen in der Praxis nicht umsetzbar ist, wurde in [HoRe 05] ein Attribute Converter vorgeschlagen. Mit Hilfe dieser Komponente ließen sich global eindeutige Handles realisieren (Bewertung: 3).

- **Interoperabilität beim ID-Management:**
  GridShib stützt sich auf Shibboleth als den de-fakto-Standard für föderiertes Identitätsmanagement. Die Heimatdomäne ist die einzige autoritative Quelle für Benutzerinformationen (Bewertung: 3).

- **Trust Management; Formalisierung:**
  Die Festlegung der Vertrauenswerte erfolgt implizit innerhalb der Föderation. Die Teilnehmer der Föderation vertrauen den verschiedenen IdP Provider im Hinblick auf die Richtigkeit der Identitätsinformationen (Bewertung: 1).

- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**

Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).

- **Trust Management; Granularität:**
  Vertrauenswerte werden lediglich auf Ebene der IdPs festgelegt (Bewertung: 0).

- **Rechtedelegation; Granularität und Sicherheit:**
  Shibboleth sieht vor, die Heimatdomäne eines Nutzer mit in die Autorisierung einzubeziehen. Mit Hilfe der Attribute Assertions kann ein IdP oder eine andere Entität in der Rolle der Attribute Authority einem Nutzer dedizierte Rechte zuweisen. Obwohl diese Teilkonzepte in den präsentierten Lösungen noch nicht umgesetzt sind, besteht die Möglichkeit damit abgestufte Reche zu vergeben. Insofern kann das Konzept mit zwei bewertet werden, auch wenn es noch keine Realisierung existiert. Eine Möglichkeit zum Widerruf der über Assertions delegierten Rechte ist allerdings in Shibboleth nicht vorgesehen. Die Assertions besitzen eine Gültigkeitsdauer aber keinen Widerrufsmechanismus. (Bewertung 2).

- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Eine Weiterdelegation von Rechten ist nicht vorgesehen (Bewertung: 0).

- **Delegation von Policies:**
  Shibboleth und auch GridShib sieht vor, dass der Nutzer über Attribute Release Policies festlegen kann, welche Daten über ihn an welchen Service Provider übermittelt oder eben nicht übermittelt werden dürfen. Analog dazu besteht die Möglichkeit des Service Providers Attribute Acceptance Policies zu spezifizieren, die beispielsweise das Mindestmaß an Attributen enthalten, die der Service Provider für eine Zugriffskontrollentscheidung benötigt. Eine Sprache zur Beschreibung der Policies wird nicht vorgeschrieben. Es können hierfür bspw. XACML (eXtensible Access Control Markup Language) [Homm 05a, Mose 05, OASI a] oder das im Rahmen des P3P Projektes (Platform for Privacy Preferences) entstandene APPEL (A P3P Preference Exchange Language) [CLM 02] verwendet werden (Bewertung: 3).

- **Delegation von Aufgaben:**
  Bei GridShib kann die Aufgabe der Attribute Authority an (de)zentrale Entitäten delegiert werden. Nach diesem Prinzip arbeitet das beschriebene MyVocs. Diese Delegation erfolgt jedoch statisch (Bewertung 1).

- **Mapping:**
  Mechanismen zur Konfliktauflösung, falls es bspw. zu einem Policy Konflikt zwischen ARP und AAP kommt, sind nicht vorgesehen. Diese Regeln können derzeitig allenfalls lokal festgelegt werden (Bewertung: 1).

- **Skalierbarkeit:**
  Die räumliche Verteilung der Ressourcen hat kaum Einfluss auf GridShib. Werden dem Grid neue Nutzer hinzugefügt, sind diese Nutzer beim

entsprechenden IdP bereits bekannt und über Shibboleth basierende Mechanismen im Grid sofort handlungsfähig. Bei der Zunahme organisatorischer Einheiten muss die neue Organisation eine IdP Komponente aufsetzen, d.h. es besteht ein linearer Zusammenhang, d.h. GridShib skaliert in allen drei Dimensionen (Bewertung: 3).

- **Flexibilität; Entitätenvielfalt:**
  Mit Hilfe von Shibboleth lassen sich beliebige Entitäten abbilden. Allerdings bedarf es im Grid dazu eine Abstimmung bezüglich des Datenschemas für die Handles (Bewertung: 3).

- **Flexibilität; Organisationsflexibilität:**
  Mit Hilfe von GridShib lassen sich Entscheidungen, z.B. bezüglich Autorisierung, lokal aber auch föderiert treffen. Werden logisch zentrale VO-Management Systeme eingesetzt, wie z.B. MyVocs können die Rechte zur Autorisierung auch an diesen Server abgetreten werden, d.h. es werden alle Dimensionen unterstützt (Bewertung: 3).

- **Administrierbarkeit:**
  Durch die Verlagerung der Authentisierung und ggf. auch der Autorisierung auf die Heimatdomäne des Nutzers wird der Aufwand der für die Pflege der Daten verringert; es können die Daten verwendet werden, die sowieso im lokalen Identity Management System gepflegt werden müssen. Problematisch ist im Hinblick auf die Administrierbarkeit das Problem, dass ein durchgehendes einheitliches VO-weites Datenschema nicht umsetzbar ist und daher Abbildungsregeln von lokalen auf globale Schemata erforderlich sind und auch gepflegt werden müssen (Bewertung: 1).

- **Sicherheit; Sicherheitsniveau:**
  Durch die dezentrale Struktur und die interorganisationale Realisierung der IdPs und AAs entstehen viele sicherheitskritische Komponenten. Ein Angreifer kann versuchen einen beliebigen IdP anzugreifen, um Zugang zum Grid zu erhalten. Eine Absicherung dieser vielen Komponenten ist auch schwieriger als die Abschottung eines zentralen Systems. Die Eintrittswahrscheinlichkeit ist deshalb mit hoch anzunehmen. Wird auch die Autorisierungsentscheidung zumindest teilweise auf den IdP verlagert, kann sich ein Angreifer ggf. umfangreiche Rechte einräumen und dadurch einen hohen Schaden verursachen (Bewertung: 0).

- **Sicherheit; Zusicherung, QoP:**
  Da jede Domäne mit seinem IdP oder seiner AA zur Gesamtsicherheit des Systems beiträgt und die Sicherheit des gesamten Grids auch von der Sicherheit des einzelnen IdP/AA abhängen kann, wäre eine Zusicherung von Sicherheitseigenschaften des IdP bzw. der AA ausgesprochen wichtig. Sicherheitseigenschaften oder QoP ist jedoch nicht vorgesehen (Bewertung: 0).

Abbildung 4.35 fasst die Bewertung von GridShib zusammen.

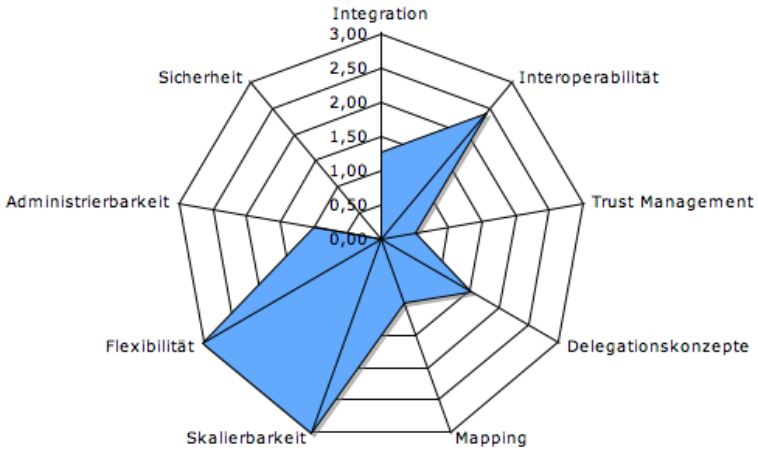

Abbildung 4.35: Bewertung GridShib

4.3.5 Gruppenmanagement

Gruppenkonzepte unterhalb der VO erforderlich

In den vorangegangenen Abschnitten wurden Mechanismen zum VO-Management betrachtet. In sehr großen Forschungsprojekten mit sehr vielen beteiligten Organisationen, Nutzern oder unterschiedlichen Experimenten ist die Etablierung einer VO für die tägliche Arbeit u.U. zu grobgranular. Es werden unterhalb der VO Gruppenkonzepte benötigt.

Ein Beispiel aus der Klima-Community soll dies verdeutlichen. Von März 2007 bis März 2009 wird das Forschungsprogramm „International Polar Year (IPY)“ durchgeführt [IPY]. Dabei werden Forschungsfragestellungen aus dem Bereich Klimaforschung für Arktis und Antarktis untersucht. Das IPY umfasst mehr als 200 Projekte mit tausenden von Wissenschaftlern aus mehr als 60 Nationen. Das C3-Grid [C3-Grid] als Community Projekt im D-Grid nimmt ebenfalls am IPY teil. C3 bildet innerhalb des D-Grid eine VO. Allerdings beschäftigen sich bereits im C3-Grid verschiedene Gruppen mit abgeschlossenen Forschungsfragestellungen des IPY (z.B. Untersuchung von Eisbohrkernen, Meeresströmungen, Solare Einstrahlung, u.v.a.). Diese Projekte setzen sich aus Forschern zusammen die Gruppen innerhalb der C3-VO bilden. Diese Gruppen verwenden für ihre Forschung spezifische Ressourcen und arbeiten die meiste Zeit innerhalb ihres Gruppen-Kontextes. Um schnell und effizient arbeiten zu können, besteht der Bedarf nach einem eigenen dynamischen und dezentralen Gruppenmanagement, das in eingeschränkter Weise die Aufgaben des VO-Managements und der Authentisierung und Autorisierung im „kleineren Maßstab“ innerhalb der Gruppe nachbildet. Folgende Aufgaben sollen innerhalb der Gruppe für die Gruppe durchführbar sein:

Aufgaben des Gruppenmanagement

- Etablierung von Subgruppen
- Autonome Pflege der Gruppenmitgliedschaft (Aufnehmen neuer Mit-

glieder, Entfernen von Mitgliedern, u.ä.)

- Zuordnung von Ressourcen zu Gruppe oder Subgruppen
- Vergabe von gruppenspezifischen Rechten

Die in den Abschnitten 4.2 und 4.3 untersuchten Systeme basieren auf X.509 Zertifikaten. Eine Gruppenmitgliedschaft kann hier kodiert werden.

Im EGEE und LCG werden bspw. individuelle Nutzer mit Hilfe des Mapping-Files beim Ressourcen-Provider auf sogenannte Pool-Accounts (vgl. Abbildung 4.19) abgebildet, die einer Gruppenkennung entsprechen können. Eine weitere Untergliederung oder systematische Managementverfahren existieren jedoch nicht.

Bei dem in Abschnitt 4.3.1 vorgestellten VOMS wird dies über den FQAN im Zertifikat gelöst. Er enthält Bezeichner für Gruppen und Subgruppen (vgl. Beispiel in Abbildung 4.30). Bei diesem Ansatz ist eine Selbstveraltung der Gruppe nicht umsetzbar.

Selbstverwaltung von Gruppen nicht möglich

Wichtige Gruppenmanagement-Aufgaben, wie die Etablierung von Subgruppen oder die Aufnahme neuer Gruppenmitglieder, müssen vom VO-Administrator durchgeführt werden. Auch wenn diese Informationen nur innerhalb der Gruppe bekannt sein müssen und verwendet werden, bedarf es einer VO-weiten Festlegung und einer VO-weiten Pflege.

Bestehende Konzepte lassen eine dezentrale und dynamische Selbstverwaltung von Gruppen nicht zu. Ein möglicher Lösungsansatz für diese Problemstellung wird in Abschnitt 5.2.2 vorgeschlagen.

4.4 Datensicherheitsdienste

Datensicherheitsdienste umfassen Sicherheitsmechanismen für Integrität, Vertraulichkeit und Verbindlichkeit. Dabei ist jeweils zwischen den Sicherheitsmechanismen zum Schutz der Nachrichten bei der Kommunikation und dem Schutz gespeicherter Daten oder Daten, die bei Berechnungen auf lokalen Systemen anfallen, zu unterscheiden. Dementsprechend werden in Abschnitt 4.4.1 Mechanismen zur Integritätssicherung, in Abschnitt 4.4.2 Verfahren zur Sicherung der Verbindlichkeit und in Abschnitt 4.4.3 solche zur Gewährleistung der Vertraulichkeit untersucht, um sie dann mit Hilfe des Kriterienkataloges aus Kapitel 3 zu bewerten.

Bei der Vertraulichkeit gibt es im Grid die zusätzliche Forderung nach der Vertraulichkeit der Ressourcen- und Dienstnutzung. Dieser Spezialfall wird im Abschnitt 4.4.4 vorgestellt; entsprechende Mechanismen werden in Abschnitt 4.6 untersucht.

4.4.1 Integrität

Die Integritätssicherung von Daten gewährleistet, dass Veränderungen an den Daten erkannt werden können. Wie bereits erläutert, werden Mechanismen zur Sicherung der Kommunikations- und Datenintegrität unterschieden.

Integrität der Kommunikation

Die Integration der Kommunikationsbeziehung ist gewährleistet, sobald Änderungen von übertragenen Nachrichten erkannt werden können. Alle betrachteten Middlewares setzen zur Sicherung der Kommunikationsbeziehungen auf SSL bzw. TLS auf [DiRe 06, DiAl 99, Holl 04, BWNH+ 06, IETF].

SSL/TLS als Mechanismus

TLS berechnet über die zu übertragende Nachricht eine kryptographische Prüfsumme, genannt **Message Authentication Code (MAC**). Dieser MAC wird mit der Nachricht an den Empfänger übertragen. Dieser wiederum berechnet seinerseits den MAC der empfangenen Nachricht und vergleicht ihn mit dem übertragenen MAC-Wert. Sind beide Werte gleich, ist die Integrität der Nachricht sichergestellt, andernfalls wurde die Nachricht manipuliert. Um die gleichzeitige Veränderung von Nachricht und übertragenem MAC zu verhindern, verwendet TLS den in Abbildung 4.36 dargestellten **HMAC** [KBC 97].

kryptographische Prüfsumme HMAC

```
HMAC_hash(MAC_write_secret, seq_num + TLSCompressed.type +
          TLSCompressed.version + TLSCompressed.length +
          TLSCompressed.fragment));
```

Abbildung 4.36: Berechnung des HMAC bei TLS [DiRe 06]

Dabei wird der MAC-Algorithmus mit einem zusätzlichen geheimen Schüssel (`MAC_write_secret`) parametrisiert. Das heißt, ein Angreifer müsste für die Manipulation des HMAC auch diesen geheimen Schlüssel kennen. Die Sequenznummer (`seq_num`) stellt sicher, dass das Löschen ganzer Nachrichten, die Änderung der Reihenfolge oder deren Verzögerung sowie das Widereinspielen von Nachrichten erkannt werden können.

Manipulationssicherheit

TLS unterteilt den Klartext in Fragmente mit einer Größe kleiner gleich 2^{14} Bytes. Die Klartextblöcke werden dann komprimiert. TLS arbeitet grundsätzlich mit komprimierten Nachrichten. Die Informationen über die Art der Komprimierung (`TLSCompressed.type`), die Version des verwendeten Protokolls (`.version`) und die komprimierten Daten selbst (`.fragment`) werden mit dem Schlüssel und der Sequenznummer konkateniert (”+”).

Vor der eigentlichen Kommunikation wird ein TSL Handshake Protokoll durchgeführt, das in Abbildung 4.37 mit den dabei zu übertragenen Nachrichten dargestellt ist. Optionale Nachrichten sind kursiv gesetzt. Außerdem

ist anzumerken, dass der Initiator der Verbindung als Client und derjenige, der die Verbindung annimmt, als Server bezeichnet werden. Im Grid-Kontext kann jede Entität sowohl als „Client“ als auch als „Server“ im Sinne von TLS auftreten.

TLS Handshake Protokoll zur Aushandlung der verwendeten Verfahren

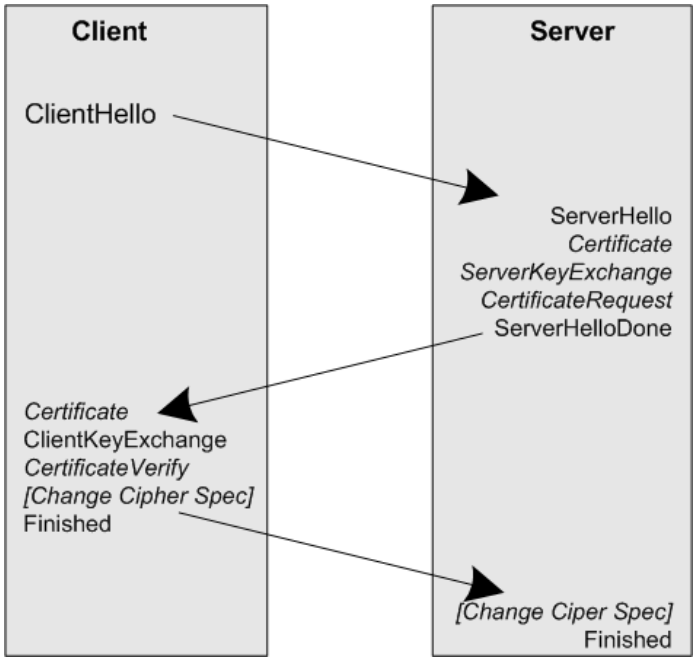

Abbildung 4.37: TLS Handshake Protokoll nach [DiRe 06]

Für die Integritätssicherung sind dabei die beiden Hello-Nachrichten (`ClientHello`, `ServerHello`) von Bedeutung. Die Hello-Nachricht des Initiators der Kommunikation enthält eine Liste von möglichen Algorithmen, die der Client unterstützen kann. Die Liste ist nach absteigender Präferenz sortiert. Das heißt, in seiner Hello-Nachricht macht der Client einen Vorschlag über zu verwendende MAC-, Kompressions- und Verschlüsselungsalgorithmen. Derjenige, der den Kommunikationswunsch entgegennimmt, führt die eigentliche Auswahl der konkreten Algorithmen, unter Berücksichtigung der Prioritätenliste des Clients und eigener Präferenzen, durch. Sind die beiden Listen nicht in Übereinstimmung zu bringen, antwortet der Server mit einem `handshake failure alert` und beide Partner beenden die Verbindung.

Bewertung Integrität der Kommunikation

Da die Mechanismen zur Sicherung der Integrität für Kommunikationsdaten sich deutlich von denen für gespeicherte bzw. lokal verarbeiteten Daten unterscheiden, sind die Mechanismen auch gesondert zu bewerten. Im folgenden wird mit SSL/TLS der wohl am häufigsten eingesetzte Mechanismus zur Integritätssicherung von Kommunikationsdaten bewertet.

- **Middleware-Integration:**
  SSL/TLS ist voll in alle Middlewares integriert (Bewertung: 3).
- **Ressourcen-Integration:**
  SSL/TLS Bibliotheken sind für nahezu alle Betriebssysteme verfügbar und damit ist eine nahtlose Integration möglich (Bewertung: 3).
- **Erweiterbarkeit:**
  Die Protokolle SSL und TLS sind sehr umfassend dokumentiert und spezifiziert. Die Protokolle schreiben auch sehr strikt die zu verwendenden kryptographischen Verfahren vor. Eine Anpassung ist nur vorgesehen und möglich durch die Konfiguration bzw. Auswahl aus verschiedenen vorgegebenen kryptographischen Algorithmen (vgl. Anmerkungen zur Hello-Nachricht im vorangegangenen Abschnitt). Für die Integritätssicherung werden MD5 [Rive 92] und SHA [EaJo 01] angeboten. Schnittstellen für eine Erweiterung oder Integration zusätzlicher Komponenten sind nicht vorgesehen. Auch bei den Middlewares selbst wäre für eine Ergänzung oder Erweiterung der Protokolle zur Integritätssicherung eine aufwändige internationale Abstimmung erforderlich (Bewertung: 1).
- **Middleware übergreifende Interoperabilität:**
  Die Integritätssicherung über SSL/TLS ist sehr weit verbreitet und darf als einer der Quasi-Standards bei der Sicherung von Kommunikationsbeziehungen betrachtet werden. Dementsprechend wird er von sehr vielen Middlewares unterstützt (Bewertung 3).
- **Interoperabilität auf Policy-Ebene:**
  Die weiter oben bereits erwähnten engen Vorgaben der Protokollspezifikation erweisen sich in Hinblick auf die Interoperabilität auf Policy Ebene als Vorteil. Die Integritätssicherung kann lediglich über die Auswahl des Message Authentication Code Verfahrens (und des Kompressionsverfahren) individuell angepasst werden, d.h. die Policy-Spezifikationen von Quell- und Ziel-Domöne betreffen auch nur diesen Aspekt. Die Festlegung der Verfahren ist im Protokoll formalisiert und global eindeutig (Bewertung: 3).
- **Interoperabilität beim ID-Management:**
  Für die Integritätssicherung sind keine Informationen über die Identitäten der Nutzer erforderlich, dementsprechend ist dieses Kriterium nicht anwendbar.
- **Trust Management; Formalisierung:**
  Die Vertrauenswerte zwischen den Partnern erfolgen einzig über den impliziten Vertrauenswert, der an das Zertifikat gebunden ist. Es gibt kein formalisiertes Modell (Bewertung: 1).
- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).

- **Trust Management; Granularität:**
  Die Granularität des Trust Management ist global definiert. Der Nutzer vertraut „dem Grid“ bzw. der VO und der Ressource-Provider vertraut der Mitgliedschaft innerhalb der VO oder der Heimat-Organisation, aus der der Nutzer kommt (Bewertung: 0).

- **Rechtedelegation; Granularität und Sicherheit:**
  Konzepte zur Delegation von Rechten sind nicht vorgesehen (Bewertung: 0).

- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Eine erweiterte Delegation oder die Beschränkung des Delegationsrechts ist nicht vorgesehen (Bewertung: 0).

- **Delegation von Policies:**
  Auf Protokollebene gibt es im Rahmen der Spezifikation der Hello-Nachrichten des TLS Handshake Protokolls (vgl. Abb. 4.37) klare Festlegungen, wie die Auswahl der zugrundeliegenden Verfahren zu erfolgen hat (Bewertung: 3).

- **Delegation von Aufgaben:**
  Eine Delegation von Aufgaben ist nicht vorgesehen (Bewertung: 0).

- **Mapping:**
  Auch die Konfliktauflösung für den Fall, dass Quelle und Ziel verschiedene Verfahren verwenden wollen, ist bereits im Protokoll festgelegt. Der Initiator der Verbindung macht einen Vorschlag über zu verwendende Verfahren, derjenige, der den Verbindungsaufbauwunsch entgegennimmt, wählt das Verfahren unter Berücksichtigung der beiden Prioritätenlisten aus. Ist keine Übereinstimmung zu finden, wird die Verbindung beendet. (Bewertung: 3).

- **Skalierbarkeit:**
  Die räumliche Verteilung der Partner hat keinen Einfluss auf die Performance von TLS. Durch die Hinzunahme neuer Partner oder neuer Ressourcen entsteht ebenfalls kein zusätzlicher Aufwand, d.h. der Mechanismus skaliert in allen drei Dimensionen (Bewertung: 3).

- **Flexibilität; Entitätenvielfalt:**
  Für die Integritätssicherung ist es nicht erforderlich, die Entitäten der beteiligten Kommunikationspartner zu kennen; damit ist dieses Kriterium nicht anwendbar.

- **Flexibilität; Organisationsflexibilität:**
  Die Integritätssicherung lässt sich grundsätzlich sowohl in zentralen, lokalen, aber auch in föderierten Umgebungen ohne Anpassungen einsetzen (Bewertung: 3).

- **Administrierbarkeit:**
  Für die Integritätssicherungkomponente von TLS entsteht kein zusätzlicher Administrationsaufwand. Der abhängig vom Einsatz des

Protokolls zu verwendende Sitzungsschlüssel wird während des Handshake Protokolls automatisch ausgehandelt (Bewertung: 3).

- **Sicherheit; Sicherheitsniveau:**
  Die zur Integritätssicherung verwendeten MAC-Algorithmen sind sehr gut untersucht und bisher sind keine schlüssigen und umfassenden Verfahren bekannt, die es möglich erscheinen lassen, dass die verwendeten Verfahren gebrochen werden könnten. Für die Integritätssicherung gibt es sehr viele kryptographische Algorithmen. Sollten für einzelne MAC-Algorithmen Verfahren bekannt werden, um diese zu brechen, kann innerhalb von SSL/TLS auf ein anderes Verfahren gewechselt werden. Es darf auch angenommen werden, dass in diesem Fall auch schnell neue Verfahren mit in das Protokoll aufgenommen würden. Die Eintrittswahrscheinlichkeit darf deshalb als niedrig angenommen werden.

  Für den Fall, dass es einem Angreifer gelingen sollte, die Integritätssicherung zu brechen, wäre er in die Lage versetzt, Nachrichten zwischen zwei Partnern zu manipulieren. Dazu muss er aber innerhalb des Kommunikationskanals die Möglichkeit zur Manipulation haben, d.h. er muss zusätzlich die Kontrolle über z.B. ein Vermittlungssystem zwischen den Partnern erlangen. Für eine konkret angegriffene Kommunikationsbeziehung kann durch die Verletzung der Integritätssicherung durchaus erheblicher Schaden entstehen. Ein Angriff erfolgt aber sehr lokal, eine Ausweitung auf das gesamte Grid dürfte kaum umsetzbar sein. Die Schadenshöhe wird deshalb insgesamt als Mittel bewertet. Damit ergibt sich eine Gesamtbewertung von 2.

- **Sicherheit; Zusicherung, QoP:**
  Neben der Übermittlung der verwendeten MAC-Verfahren gibt es bei TLS keinerlei Mechanismen für die Festlegung oder Zusicherung von QoP Werten (Bewertung: 0).

Abbildung 4.38 fasst die Ergebnisse der Kriterienbewertung für die Integritätssicherung bei der Kommunikation nochmal in einem Netzdiagramm zusammen.

Integrität der Daten

Im vorangegangenen Abschnitt wurde lediglich die Integrität der Kommunikationsdaten betrachtet. Im Grid-Umfeld ist es jedoch auch von entscheidender Bedeutung, dass die Integrität von Daten, die gespeichert oder auf Ressourcen bearbeitet werden (z.B. Zwischenergebnisse), gesichert ist. Im Fall der Integrität der Kommunikation kann davon ausgegangen werden, dass ein externer Dritter den Angriff durchführt, indem er versucht, die Nachrichten auf Vermittlungssystemen zwischen Sender und Empfänger zu manipulieren. Die Prämisse des „externen Dritten“ kann für die Integrität der Daten auf Ressourcen nicht aufrecht erhalten werden. Wenn die Integrität von gespeicherten oder auf Ressourcen bearbeiteten Daten sichergestellt werden soll, muss

Schutz vor Veränderung auf fremdem System

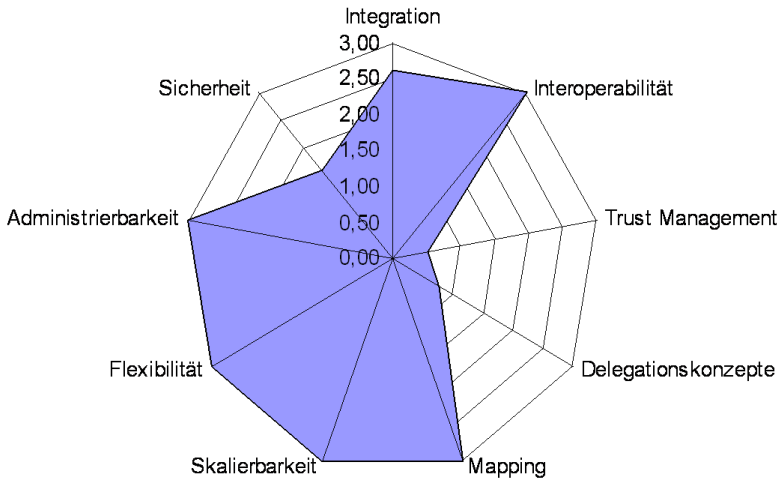

Abbildung 4.38: Bewertung Integritätssicherung der Kommunikation

zwangsweise eine Veränderung der Daten auf fremden Speichersystemen oder den verarbeitenden Ressourcen erkennbar sein. Dieses Problem ist natürlich deutlich schwieriger, denn in diesem Fall sollte auch berücksichtigt werden, dass ein fremder Administrator in einer kooperierenden Organisation (dem man eigentlich vertrauen können sollte) die Daten manipuliert.

analoges Problem: Schutz Mobiler Agenten

Das Problem der Integritätssicherung von Daten auf fremden Systemen oder während der Verarbeitung auf Systemen anderer Organisationseinheiten ist vergleichbar mit dem Problem der Integritätssicherung Mobiler Agenten (MA). Ein MA ist eine Software-Komponente zur Lösung bestimmter Aufgaben, die Möglichkeit zur Migration bzw. zur Mobilität besitzt. Das heißt, ein so genannter Multi-Hop MA kann eine Menge von heterogenen Systemen in verschiedenen Domänen aufsuchen, um dort seine Aufgaben auszuführen.

Die Integritätssicherung des MA sollte auch Veränderungen durch das fremde Gast-System erkennen lassen, d.h. Veränderungen bspw. am Programm Code des MA oder Manipulationen von Zwischen- oder Endergebnissen, die der MA dann in seine Heimatdomäne kommuniziert. Der Programm-Code eines MA in einem Agentensystem unterscheidet sich in Bezug auf Integritätsfragen nicht von einer Job-Spezifikation im Grid. Gleiches gilt für Zwischenergebnisse von Berechnungen. In [Reis 01] wurde gezeigt, dass die Integritätsfrage für MAs, die auf einem fremden System ausgeführt werden und damit unter der vollständigen Kontrolle des fremden Systems stehen, nicht in Allgemeinheit lösbar ist. Für Spezialfälle, bei denen ein Agent mit seinen Daten mehrere Ausführungsumgebungen besucht oder mit seinen Daten immer wieder zu seiner Quelle zurückkehrt, gibt es Spezialverfahren zur Sicherung der Integrität.

Problem in Allgemeinheit nicht lösbar

Für einen MA, der sich dauerhaft auf einem fremden System befindet und nur

mit der Quelle Ergebnisse austauscht, ist es in Allgemeinheit nicht möglich, Manipulationen am MA oder dessen Ergebnissen zu erkennen. Diese Erkenntnis gilt auch für Daten im Grid, die auf fremden Systemen gespeichert oder verarbeitet werden. In Spezialfällen wäre eine Integritätssicherung auch im Grid denkbar.

Problem für Spezialfälle, z.B. Replikate lösbar

Das primäre Ziel von Daten-Grids ist die ausfallsichere Speicherung von Daten sowie einen effizienten Zugriff auf diese Daten zu realisieren (vgl. Abschnitt 2.1.2). Hierzu werden Replikate der Daten auf unterschiedlichen Systemen gehalten (bspw. durch eine Replica Management Funktion wie in OGSA spezifiziert, vgl. Abschnitt 4.1.4). Sobald dieselben Daten auf mehreren unabhängige Systemen in unterschiedlichen Domänen vorliegen, lassen sich Kopien zum Vergleich nutzen, um damit Veränderungen an einzelnen Datensätzen zu erkennen.

Ein anderer Fall ist die bloße redundante Speicherung von Daten, ohne diese auf fremden Systemen weiter zu verarbeiten. Die Daten müssten auf einem sicheren (eigenen) System erzeugt werden, danach wäre eine kryptographische Prüfsumme zu berechnen und diese verschlüsselt mit den Daten zu speichern. Dieser Datensatz kann dann auf fremde Systeme zum ausschließlichen Zweck der Speicherung übertragen werden, eine Veränderung der Daten durch Berechnungen ist nicht möglich, denn diese würden die Prüfsumme ändern. Manipulationen wären nach der Rückübertragung auf eigene Systeme erkennbar. In der Praxis sind diese Beispiele zur Integritätssicherung jedoch mit massiven Einschränkungen im Hinblick auf die Nutzbarkeit des Grid verbunden. Wenn bspw. die Integrität nur gewährleistet werden kann, wenn Veränderungen an den Daten nur auf eigenen „sicheren“ Systemen zulässig sind, kann nicht mehr wirklich von einer Grid-Nutzung gesprochen werden.

In heutigen Middlewares oder Replikatsmanagementsystemen sind, vermutlich aus den angeführten Gründen, keine Mechanismen zur Integritätssicherung lokaler Daten implementiert. Eine Bewertung anhand des Kriterienkataloges aus Abschnitt 3 kann deshalb nicht durchgeführt werden.

4.4.2 Verbindlichkeit

Die Verbindlichkeit stellt sicher, dass ein eingetretenes Ereignis oder eine durchgeführte Aktion zu einem späteren Zeitpunkt nicht geleugnet werden kann. Im Grid-Kontext wäre eine typische Aktion die Submission eines Jobs. Derjenige, der den Job abgeschickt hat, könnte anschließend die Submission nicht leugnen. Andererseits könnte auch der Ressourcen-Provider nicht behaupten, den Job nie erhalten zu haben. Dies setzt voraus, dass entsprechende Belege und Nachweise für die Job-Submission erzeugt werden und diese Belege zweifelsfrei einem Verursacher zurechenbar sind. Die Belege müssen sicher gespeichert werden und zu einem späteren Zeitpunkt zugänglich sein. Tritt eine Kontroverse über eine Aktion oder ein Ereignis ein, müssen diese Belege auch ohne Mithilfe der beteiligten Partner, verifizierbar sein.

In derzeitigen Grids werden Informationen über Ereignisse und Aktionen in

der Regel lediglich auf lokalen Systemen protokolliert. Weitere Komponenten, die zumindest Informationen über Jobs besitzen, sind die verschiedenen Monitoring-Systeme der Grids. Allerdings steht bei diesen Systemen häufig die Job-Anzahl, die Auslastung der Systeme und der Zustand der Jobs im Vordergrund. Der Eigentümer des Jobs kann über das Job-Monitoring und -Management den Zustand seiner Jobs überwachen und diese administrieren. Beim Grid-weiten Job-Monitoring wird jedoch häufig die Information über den Job-Eigner überhaupt nicht mehr zur Verfügung gestellt. Das heißt, derzeit ist eine Relation zwischen Job-Eigner und entsprechenden Jobs allenfalls lokal realisiert. Die entsprechenden Einträge in den Log-Dateien sind jedoch keine zweifelsfreien Belege im Sinne der Verbindlichkeit, da ein böswilliger lokaler Systemadministrator die Log-Dateien natürlich manipulieren kann. Bei einer Kontroverse zwischen Job-Eigner und Ressourcen-Provider über die Ausführung eine bestimmten Jobs könnten die Logs — von einem unabhängigen Dritten — nicht zum zweifelsfreien Nachweis der Aktion genutzt werden.

Die oben geforderten Belege müssten dazu eine fälschungssichere Verknüpfung von Job-Eigner zu einer bestimmten Job-Submission realisieren. Dies ließe sich bspw. dadurch realisieren, dass der Job-Eigner bei der Submission des Jobs einen Beleg generiert und diesen mit seinem privaten Schlüssel digital signiert. Dieser Beleg müsste sicher von einem unabhängigen Dritten gespeichert werden. Das heißt, dass der Beleg nicht nur beim Ressource-Provider verbleiben darf. Den dieser könnte den Beleg löschen.

Derzeit keine Verbindlichkeits-mechanismen

Falls eine solche Infrastruktur existieren würde, wäre zu einem späteren Zeitpunkt ein zweifelsfreier Nachweis der Aktion möglich. Derzeit gibt es in Grid Middlewares keinerlei Mechanismen, um ein derartiges System umsetzen zu können. Eine Anwendung des Kriterienkataloges ist deshalb nicht möglich.

4.4.3 Vertraulichkeit

Die Vertraulichkeit ist gewährleistet, wenn nur Berechtigte Zugang zu Daten besitzen und diese auch auswerten können. Dies betrifft sowohl Daten während der Übertragung als auch während der Verarbeitung auf Endsystemen. Dementsprechend wird in diesem Abschnitt, analog zur Integritätssicherung (Abschnitt 4.4.1), zwischen Vertraulichkeit der Kommunikation und Vertraulichkeit der Daten unterschieden. Oftmals ist es auch erforderlich die Vertraulichkeit der Ressourcen- und Dienstnutzung sicherzustellen. Diese Fragestellung wird im Abschnitt 4.6 nochmal aufgegriffen.

Vertraulichkeit der Kommunikation

Für die Vertraulichkeit der Kommunikation, d. h. den Schutz der Daten, die mit Hilfe von Middlewares ausgetauscht werden, ist sicherzustellen, dass ein Angreifer, der Zugriff auf den Kommunikationskanal zwischen Sender und

Empfänger besitzt, keine Nachrichten mitlesen kann. Hierzu werden Verschlüsselungsverfahren eingesetzt.

Die GSI von Globus sieht hierzu zwei unterschiedliche Mechanismen vor:

Message Level Security: Sicherung der einzelnen Nachricht

1. Bei der **Message Level Security** werden die SOAP-Nachrichten geschützt. Dazu stützt sich GSI auf WS-Security und die WS-SecureConversation-Spezifikationen [ABB+ 05] ab. Diese Spezifikationen sind in Bezug auf Zertifikate neutral formuliert. GSI unterstützt allerdings nur X.509-Zertifikate (vgl. Abschnitt 4.2.1). Mit den WS-Standards ist es möglich, einen Sicherheitskontext zwischen zwei Kommunikationspartnern zu etablieren und diesen für die Sicherung des nachfolgenden Verkehrs zu verwenden. Message Level Security liefert als Schutzmechanismen Integrität (d. h. ein Empfänger kann erkennen, ob die Nachricht verändert wurde), Vertraulichkeit durch Verschlüsselung und Replay Prevention (d. h. der Empfänger kann erkennen, ob er dieselbe Nachricht schon einmal erhalten hat).

Transport Level Security: Sicherung durch vertraulichen Kanal

2. Bei der **Transport Level Security** werden nicht einzelne (SOAP-) Nachrichten geschützt, sondern Sender und Empfänger bauen mit Hilfe von TLS [DiAl 99, DiRe 06] einen vertraulichen Kanal auf, über den dann alle Nachrichten übertragen werden. (SOAP over TLS). Da es bei Verwendung von Message Level Security zu massiven Performance-Problemen kommt, wird ein der Praxis in der Regel nur Transport Level Security eingesetzt.

Auch die Sicherheit der Kommunikationsbeziehungen in UNICORE und gLite stützen sich voll auf SSL/TLS ab.

TLS als Mechanismus

TLS arbeitet transparent zwischen der Transport- und der Anwendungsschicht. Es besteht selbst aus zwei (Teil-) Schichten und fünf (Teil-) Protokollen (vgl. Abbildung 4.39).

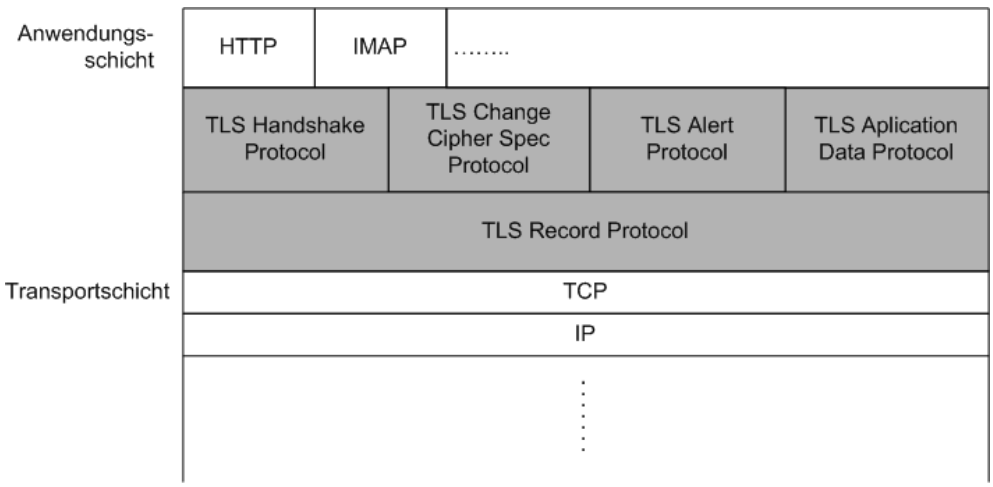

Abbildung 4.39: TLS Einordnung in den TCP/IP Protokollstack

TLS Record Layer Protocol stellt verschlüsselten Kanal bereit

Das TLS Handshake Protokoll dient zum Verbindungsaufbau, zur Authentisierung der Partner, zum Schlüsselaustausch und zur Aushandlung der zu verwendenden kryptographischen Verfahren. Das TLS Record Layer Protokoll stellt dann einen transparenten verschlüsselten Ende-zu-Ende Kanal zwischen Sender um Empfänger bereit. Die Verschlüsselung erfolgt mit Hilfe von symmetrischen Verschlüsselungsverfahren. Unterstützt werden die folgenden Algorithmen: RC4 (Ron's Code 4), [Schn 96] IDEA (International Date Encryption Algorithm) [Schn 96], DES (Data Encryption Standard) [FIPS 46-2], Tripple-DES,[DiRe 06] AES (Advanced Encryption Standard) [Chow 02, FIPS 197], Kerberos [MeHu 99], Camelia [MKK 05] sowie Elliptische Kurven (Elliptic Curve Cryptography (ECC)) [BWBG+ 06].

Mit Hilfe des Handshake Protokolls wird — auf sichere Art und Weise — ein symmetrischer Sitzungsschlüssel vereinbart, der dann für das ausgehandelte Verfahren verwendet wird. Der Schlüssel und auch die verwendeten Verfahren können von jedem Kommunikationspartner jederzeit mit Hilfe einer `Change Cipher Spec` Nachricht geändert werden. Damit kann sichergestellt werden, dass ein Schlüssel nicht über einen zu langen Zeitraum benutzt wird.

Bewertung Vertraulichkeit der Kommunikation

Der in Abschnitt 4.4.1 bereits im Hinblick auf Integrität bewertete Mechanismus SSL/TLS lässt sich auch zur Sicherung der Vertraulichkeit verwenden. In diesem Teilabschnitt wird der Kriterienkatalog aus Abschnitt 3.2 im Hinblick auf die Sicherung der Vertraulichkeit der Kommunikation angewendet und SSL/TLS bewertet.

- **Middleware-Integration:**
  SSL/TLS ist voll in alle Middlewares integriert (Bewertung: 3).

- **Ressourcen-Integration:**
  TLS Bibliotheken sind für nahezu alle Betriebssysteme verfügbar und damit ist eine nahtlose Integration möglich (Bewertung: 3).

- **Erweiterbarkeit:**
  Die Protokolle SSL und TLS sind sehr umfassend dokumentiert und spezifiziert. Die Protokolle schreiben auch sehr strikt die zu verwendenden kryptographischen Verfahren vor. Eine Anpassung ist nur vorgesehen und möglich durch die Konfiguration bzw. Auswahl aus verschiedenen vorgegebenen kryptographischen Algorithmen (vgl. Anmerkungen zur Hello-Nachricht im vorangegangenen Abschnitt). Für die Sicherung der Vertraulichkeit werden derzeit acht verschiedene Verschlüsselungsverfahren angeboten. Schnittstellen für eine Erweiterung oder Integration zusätzlicher Komponenten sind nicht vorgesehen. Auch bei den Middlewares selbst wäre für eine Ergänzung oder Erweiterung der Protokolle zur Sicherung der Vertraulichkeit eine aufwändige internationale Abstimmung erforderlich (Bewertung: 1).

- **Middleware übergreifende Interoperabilität:**
  Die Sicherung der Vertraulichkeit über SSL/TLS ist sehr weit verbreitet und darf als einer der Quasi-Standards bei der Sicherung von Kommunikationsbeziehungen betrachtet werden. Dementsprechend wird er von sehr vielen Middlewares unterstützt (Bewertung 3).

- **Interoperabilität auf Policy-Ebene:**
  Die weiter oben bereits erwähnten engen Vorgaben der Protokollspezifikation erweisen sich in Hinblick auf die Interoperabilität auf Policy Ebene als Vorteil. Die Sicherung der Vertraulichkeit kann lediglich über die Auswahl der Verschüsselungs-, der Message Authentication Code- und des Kompressionsverfahren individuell angepasst werden, d.h. die Policy-Spezifikationen von Quell- und Ziel-Domöne betrifft auch nur diese Aspekte. Die Festlegung der Verfahren ist im Protokoll formalisiert und global eindeutig (Bewertung: 3).

- **Interoperabilität beim ID-Management:**
  Im Handshake Protokoll besteht die Möglichkeit zur zweiseitigen Authentisierung der Partner. Der Schlüssel für die Sicherung der Vertraulichkeit durch ein symmetrisches Verschlüsselungsverfahren wird während des Handshake ausgehandelt und kann jederzeit (durch eine `Change Cipher Spec` Nachricht gewechselt werden. Die Sicherung der Vertraulichkeit stützt sich, in Bezug auf Identitätsinformationen, voll auf die Authentisierung über X.509 Zertifikate ab. Insofern muss hier die Bewertung aus Abschnitt 4.2.4 übernommen werden (Bewertung: 1).

- **Trust Management; Formalisierung:**
  Die Vertrauenswerte zwischen den Partnern erfolgen einzig über den impliziten Vertrauenswert, der an das Zertifikat gebunden ist. Es gibt kein formalisiertes Modell (Bewertung: 1).

- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).

- **Trust Management; Granularität:**
  Die Granularität des Trust Management ist global definiert. Der Nutzer vertraut „dem Grid“ bzw. der VO und der Ressource-Provider vertraut der Mitgliedschaft innerhalb der VO oder der Heimat-Organisation, aus der der Nutzer kommt (Bewertung: 0).

- **Rechtedelegation; Granularität und Sicherheit:**
  Konzepte zur Delegation von Rechten sind nicht vorgesehen (Bewertung: 0).

- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Eine erweiterte Delegation oder die Beschränkung des Delegationsrechts ist nicht vorgesehen (Bewertung: 0).

- **Delegation von Policies:**
  Auf Protokollebene gibt es im Rahmen der Spezifikation der Hello-Nachrichten des TLS Handshake Protokolls (vgl. Abb. 4.37) klare Festlegungen, wie die Auswahl der zugrundeliegenden Algorithmen zu erfolgen hat (Bewertung: 3).

- **Delegation von Aufgaben:**
  Eine Delegation von Aufgaben ist nicht vorgesehen (Bewertung: 0).

- **Mapping:**
  Auch die Konfliktauflösung für den Fall, dass Quelle und Ziel verschiedene Verfahren verwenden wollen, ist bereits im Protokoll festgelegt. Der Initiator der Verbindung macht einen Vorschlag über zu verwendende Verfahren, derjenige, der den Verbindungsaufbauwunsch entgegennimmt, wählt das Verfahren unter Berücksichtigung der beiden Prioritätenlisten aus. Ist keine Übereinstimmung zu finden wird die Verbindung beendet. (Bewertung: 3).

- **Skalierbarkeit:**
  Die räumliche Verteilung der Partner hat keinen Einfluss auf die Performance von TLS. Durch die Hinzunahme neuer Partner oder neuer Ressourcen entsteht ebenfalls kein zusätzlicher Aufwand, d.h. der Mechanismus skaliert in allen drei Dimensionen (Bewertung: 3).

- **Flexibilität; Entitätenvielfalt:**
  TLS ist für die Sicherung der Kommunikation zwischen zwei Kommunikationspartnern entwickelt worden. Eine vertrauliche Kommunikation auf Gruppen- oder gar auf VO-Ebene ist damit nicht möglich. Insofern treten in TLS als Entitäten nur Nutzer und Ressourcen auf (Bewertung: 1).

- **Flexibilität; Organisationsflexibilität:**
  Die Sicherung der Vertraulichkeit lässt sich grundsätzlich sowohl in zentralen, lokalen, aber auch in föderierten Umgebungen ohne Anpassungen einsetzen (Bewertung: 3).

- **Administrierbarkeit:**
  Für die Komponente von TLS zur Sicherung der Vertraulichkeit entsteht kein zusätzlicher Administrationsaufwand. Der abhängig vom Einsatz des Protokolls zu verwendende Sitzungsschlüssel wird während des Handshake Protokolls automatisch ausgehandelt, verwendete Zertifikate werden über den Authentisierungsmechanismus zur Verfügung gestellt (Bewertung: 3).

- **Sicherheit; Sicherheitsniveau:**
  Die zur Sicherung der Vertraulichkeit verwendeten Algorithmen sind sehr gut untersucht und bisher sind keine schlüssigen und umfassenden Verfahren bekannt, die es möglich erscheinen lassen, dass die verwendeten Verfahren gebrochen werden könnten. Es werden sehr viele kryptographische Algorithmen unterstützt. Sollten für einzelne Algorithmen

Verfahren bekannt werden, um diese zu brechen, kann innerhalb von SSL/TLS auf ein anderes Verfahren gewechselt werden. Die Eintrittswahrscheinlichkeit darf deshalb als niedrig angenommen werden.

Für den Fall, dass es einem Angreifer gelingen sollte, die Sicherung der Vertraulichkeit zu brechen, wäre er in die Lage versetzt, Nachrichten zwischen zwei Partnern mitzulesen. Dazu muss er aber innerhalb des Kommunikationskanals die Möglichkeit zum Zugriff haben, d.h. er muss zusätzlich die Kontrolle über z.B. ein Vermittlungssystem zwischen den Partnern erlangen. Für eine konkret angegriffene Kommunikationsbeziehung kann durch die Verletzung der Vertraulichkeit durchaus erheblicher Schaden entstehen. Ein Angriff erfolgt aber sehr lokal, eine Ausweitung auf das gesamte Grid dürfte kaum umsetzbar sein. Die Schadenshöhe wird deshalb insgesamt als Mittel bewertet. Damit ergibt sich eine Gesamtbewertung von 2.

- **Sicherheit; Zusicherung, QoP:**
  Neben der Übermittlung der verwendeten MAC-Verfahren gibt es bei TLS keinerlei Mechanismen für die Festlegung oder Zusicherung von QoP Werten (Bewertung: 0).

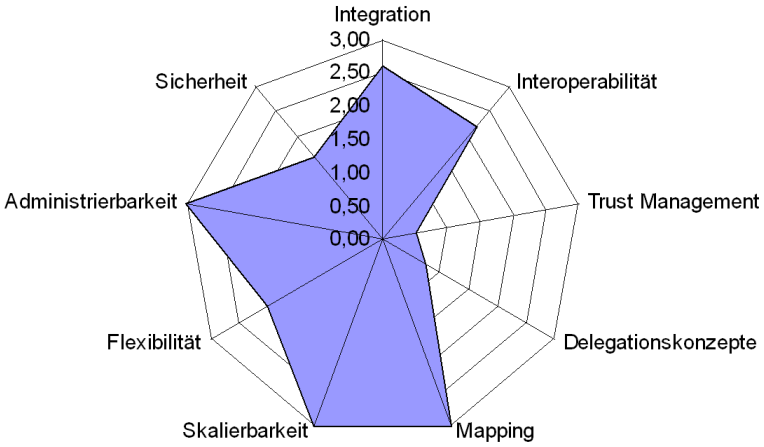

Abbildung 4.40: Bewertung Vertraulichkeit der Kommunikation

Das Netzdiagramm in Abbildung 4.38 fasst die Bewertungen der Kriterien der Ebene 1 des Kriterienkatalogs (vgl. Abbildung 3.2) für die Mechanismen zur Sicherung der Vertraulichkeit der Kommunikation zusammen.

Vertraulichkeit der Daten

Mit Hilfe der Vertraulichkeit von Daten sind zwei Fälle unberechtigten Erlangens von Informationen aus den Daten abzusichern.

1. Es ist zu verhindern, dass unberechtigte Dritte Zugang zu gespeicherten Daten erhalten.

2. Es ist zu verhindern, dass unberechtigte Dritte während der Ausführung von Jobs Zugang zu Input- und Output-Daten und Zwischenergebnissen erhalten.

fremder Administrator als potentieller Angreifer

Im Grid-Umfeld kann nicht davon ausgegangen werden, dass die Speicherung und Verarbeitung auf vertrauenswürdigen eigenen Systemen erfolgt. Der Regelfall ist, dass die Daten auf „fremden“ Systemen gespeichert werden. Es kann auch nicht per se davon ausgegangen werden, dass die Administratoren dieser Systeme vertrauenswürdig sind. Für eine Risikoanalyse muss der fremde Administrator, mit allen seinen technischen Möglichkeiten, mit in die Liste der potentiellen Angreifer aufgenommen werden.

keine allgemeingültige Lösung verfügbar

Für die oben angegebene Alternative 1 kann die Vertraulichkeit dadurch sichergestellt werden, dass die Daten nur verschlüsselt auf fremden Systemen gespeichert werden. Die Verarbeitung und die Verschlüsselung muss dabei auf sicheren eigenen Systemen erfolgen. Dies bedeutet im Grid-Kontext natürlich eine erhebliche Einschränkung im Hinblick auf die Nutzbarkeit. Grids sollen für die Nutzer die Daten transparent speichern und bei der Verarbeitung und für Berechnungen die optimalen Ressourcen zuteilen. Unter den angegebenen Prämissen ist eine derartige gewollte Transparenz nicht mehr möglich und Berechnungen wären nur auf eigenen Systemen erlaubt. In diesem Fall kann, insbesondere im Hinblick auf Berechnungen mit den Daten, nicht mehr von einer echten Grid-Nutzung gesprochen werden. Gibt man diese strengen Vorgaben auf, muss der oben angegebene Fall 2 für die Sicherung der Vertraulichkeit technisch umgesetzt werden. Hierbei gilt, analog zur Integritätssicherung (vgl. S. 132), dass dieses Problem im Allgemeinen nicht lösbar ist. In den Middlewares werden derzeit auch keine Mechanismen angeboten, um Daten vor der Speicherung transparent zu verschlüsseln.

Eine Bewertung anhand des Kriterienkataloges (vgl. Abschnitt 3) ist nicht möglich.

4.4.4 Vertraulichkeit bei der Ressourcen- und Dienstnutzung

Die Vertraulichkeit bei der Ressourcen- und Dienstnutzung ist dann gegeben, wenn ein Benutzer im Grid bzw. auf einer Ressource keine Informationen über die Ressourcen- und Dienstnutzung eines anderen Nutzers erlangen kann. Ein Beispiel aus der Entwicklung in der Industrie kann diese Forderung verdeutlichen. Für den Fall, dass zwei konkurrierende Unternehmen, die ähnliche Produkte entwickeln, dieselbe spezielle Ressource (z.B. einen Höchstleistungsrechner) nutzen, sollen sie keine Informationen übereinander erlangen können.

Ein Unternehmen muss nicht die exakten Jobs der Konkurrenz kennen, denn

bereits über die Anzahl und Größe der Jobs können Rückschlüsse über den Status im Lebenszyklus der Produktentwicklung getroffen werden. Falls von einem Unternehmen A sehr viele kleine Jobs, vom anderen Unternehmen B sehr wenige, dafür aber sehr große Jobs gerechnet werden, könnte man in der Automobilindustrie folgern, dass A ein Fahrzeug entwickelt,dass sich gerade in der Design-Phase befindet. Bei B hingegen befindet sich das Fahrzeug bereits im Crash-Test.

An diesem stark vereinfachten Beispiel wird bereits deutlich, dass es einen Bedarf für Vertraulichkeit bei der Ressourcen- und Dienstnutzung gibt und verschiedenen Nutzer auf den Systemen völlig voreinander abgeschottet werden müssen. Technisch kann dies durch Sandboxing bzw. Virtualisierung realisiert werden. Diese Aspekte werden in Abschnitt 4.6 näher untersucht.

4.5 Privacy-Dienste

Unter Privacy-Diensten werden alle Techniken subsumiert, die in der Lage sind die unterschiedlichen Datenschutzanforderungen umszusetzen.

Recht auf informationelle Selbstbestimmung

Das Bundesdatenschutzgesetz [BDSG] ist die gesetzliche Grundlage für den Datenschutz. „*Zweck [des] Gesetzes ist es, den Einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird.*“ (§ 1 Abs. 1 BDSG) Für jeden Einzelnen ist das Recht auf **informationelle Selbstbestimmung** sicherzustellen, d.h. jeder kann selbst über die Preisgabe und Verwendung seiner personenbezogenen Daten entscheiden und verfügen. „**Personenbezogene Daten** *sind Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener).*“ (§ 3 Abs. 1 BDSG)

Personenbezogene Daten

Weitere Anforderungen des Datenschutzes sind Methoden zum Anonymisieren bzw. Pseudonymisieren. „**Anonymisierung** *ist das Verändern personenbezogener Daten derart, dass die Einzelangaben über persönliche oder sachliche Verhältnisse nicht mehr oder nur mit einem unverhältnismäßig großen Aufwand an Zeit, Kosten und Arbeitskraft einer bestimmten oder bestimmbaren natürlichen Person zugeordnet werden können.*“ (§3 Abs. 6 BDSG) „**Pseudonymisieren** *ist das Ersetzen des Namens und anderer Identifikationsmerkmale durch ein Kennzeichen zu dem Zweck, die Bestimmung des Betroffenen auszuschließen oder wesentlich zu erschweren.*“ (§ 3 Abs. 6a) Das heißt, beim Pseudonymisieren können die verbleibenden Angaben einem Einzelnen, ohne Kenntnis der Zuordnungskritieren, jedoch nicht mehr einer bestimmten natürlichen Person zugeordnet werden. Pseudonymisierung wird dort eingesetzt, wo die wesentlich stärkeren Maßnahmen zur Anonymisierung nicht möglich sind.

Eine besondere Bedeutung hat die Frage der Vertraulichkeit und des Datenschutzes in der Medizinischen Forschung und hier insbesondere bei multizen-

trischen Forschungsprojekten [Sax 06]. Für diese Projekte wurde im Rahmen von MediGrid [MediGRID] die Verwendung von Grid-Technologien untersucht [SMVR 06]. Die speziellen Anforderungen werden im Abschnitt 4.5.1 vorgestellt.

Besondere Anforderungen in der Medizin

In den folgenden Abschnitten wird untersucht, inwieweit Datenschutzmechanismen technisch in Grid umgesetzt werden können.

4.5.1 Informationelle Selbstbestimmung in Grids

Datenschutzdienste im Grid müssen jeden Nutzer in die Lage versetzen über die Verwendung seiner personenbezogenen Daten zu entscheiden und zu verfügen. Das heißt bspw., dass der Nutzer bestimmen kann, wer Zugriff auf seine Daten erhält und wann diese Daten wieder zu löschen sind. Es müssen also technische Verfahren existieren, mit Hilfe derer der Nutzer diese Festlegungen (Policies) spezifizieren kann sowie Verfahren, welche diese Spezifikationen auch umsetzen.

Die derzeit wohl umfassendste Möglichkeit zur technischen Umsetzung der informationellen Selbstbestimmung bietet das im Abschnitt 4.3.3 vorgestellte Shibboleth bzw. GridShib. Shibboleth bietet dem Nutzer die Möglichkeit über Attribute Release Policies (ARPs) zu spezifizieren, wem welche Teile seiner Daten übermittelt werden dürfen. Die Attribute Authority überprüft vor der Erstellung einer Attribute Assertion den Service Provider (Empfänger der Daten) und gleicht dessen Datenanforderungen mit der ARP ab. Der Service Provider erhält nur die für ihn freigegebenen Daten.

Attribute Release Policies spezifizieren Datenfreigabe

Die Erstellung einer ARP ist für einen gewöhnlichen Grid-Nutzer nicht ohne technische Unterstützung möglich. Es kann nicht davon ausgegangen werden, dass der Nutzer in der Lage ist entsprechende Policies direkt in der jeweils verwendeten Policy Sprache zu spezifizieren. Aus diesem Grund wurden Werkzeuge zur Unterstützung des Nutzers entwickelt [MaMa 07], die zwei grundlegende Ansätze verfolgen:

Werkzeuge zur Spezifikation von ARPs

1. Rückfrage beim Nutzer: Sobald ein SP das erste Mal Daten über den Nutzer abruft, wird der Nutzer gefragt, ob er der Übertragung der Daten zustimmt. Ab diesem Zeitpunkt kann der SP solange diese Daten ohne explizite Zustimmung des Nutzers abrufen, bis dieser der Freigabe für diesen SP widerspricht.

2. ARP Editor: In so genannten Attribute Release Policy (ARP) Editoren (z.B. [Autograph]) werden dem Nutzer alle über ihn verfügbaren Attribute gezeigt und er kann für jeden SP bestimmen ob dieser ein Datum sehen darf. Der ARP Editor erzeugt dann die entsprechende Policy.

Im Grid Kontext erscheint nur der zweite Fall sinnvoll. Bei der ersten Alternative würde bspw. bei einer Job-Migration zu einem anderen Ressourcen-Provider, die auch automatisch durch die Middleware oder einen Scheduler ausgelöst werden kann, versucht beim Nutzer rückzufragen. Da Grid Anwen-

dungen nicht immer interaktiv sind und sehr lange dauernde Laufzeiten haben können, würde dieses Vorgehen zu unnötigen Verzögerungen im Ablauf führen.

Auch im Umfeld der Grid Middleware-Entwicklung, und hier insbesondere bei Globus, wurde erkannt, dass sich Shibboleth gut für Datenschutzaspekte verwenden lässt [WBKS 05, BBF+ 06]. Allerdings ist GridShib noch in einer relativ frühen Entwicklungsphase und als Globus Projekt auch noch nicht integraler Bestandteil von Globus.

Datenschutz für Grid-Nutzer: GridShib und ARPs

GridShib und ARPs sind ein geeigneter Mechanismus, um Datenschutz für Grid-Nutzer während der Grid-Nutzung zu realisieren. Die ARPs beziehen sich auf Daten, die während der Authentisierung, der Autorisierung und ggf. zu Abrechnungszwecken zwischen der Heimat-Domäne und dem Ressource-Provider ausgetauscht werden. Es existieren aber noch weitere Datenschutzanforderungen, die mit ARPs kaum oder überhaupt nicht realisierbar sind.

In der medizinischen Forschung mit Patientendaten sind die Anforderungen an den Datenschutz besonders hoch [MSVR 07]. Hier wird gefordert:

Strenge Anforderungen in der Medizin

- Es darf nur mit ausdrücklicher Einwilligung des Patienten oder mit pseudonymisierten Daten gearbeitet werden [DSMS 06].
- Der Patient kann in seiner Einwilligung auch den Personenkreis oder den Kreis der Institutionen, die Zugriff auf seine Daten erhalten sollen, einschränken.
- Der Patient kann jederzeit und auch teilweise die Nutzung seiner Daten widerrufen.
- Der Patient kann die sofortige Löschung von Daten fordern und auch einen Nachweis verlangen, der belegt, dass die Daten gelöscht wurden.
- Dem Patienten ist auf Verlangen zu jedem Zeitpunkt nachzuweisen, auf welchen Systemen und von wem seine Daten bearbeitet bzw. genutzt wurden (Auditing und Tracking) [Moha 06].

Um dies zu realisieren und einen Löschungsnachweis erstellen zu können, muss eine Auditing und Tracking Komponente sicher mitprotokollieren, wann und auf welchen Systemen die Daten verarbeitet, gespeichert sowie ggf. gelöscht wurden und wer Zugriff auf die Daten hatte.

Spezielle medizinische Datenformate werden im Grid nicht unterstützt

Außerhalb des Grids wurden Systeme entwickelt, die auf der Trennung der Daten in identifizierende Daten (IDAT) und medizinische Daten (MDAT) basieren [Sax 06]. Es wurden auch spezifische Datenformate mit entsprechender Zugangssoftware entwickelt, die Möglichkeiten der Authentifizierung und eines autorisierten und gesicherten Zugriffs, auch auf Teile dieser Dokumente, ermöglichen (z.B. Digital Imaging and Communications in Medicine (DICOM) [DIC , DICO 07]) [Stei 06]. Damit ist eine Trennung von Patienten- und Falldaten möglich. Durch die enge Koppelung von Datenformat und Zugangssoftware kann eine strenge und feingranulare Zugriffskon-

trolle auch auf Teildaten sowie eine sichere Protokollierung und entsprechende Löschungsnachweise erstellt werden.

Derzeit ist allerdings keine Middleware in der Lage diese Standards zu unterstützen oder ähnliche Charakteristiken umzusetzen.

Nachvollziehbarkeit konfliktär mit Transparenz

Werden medizinische Daten in Grids verarbeitet, widerspricht das Gebot der Nachvollziehbarkeit (Auditing und Tracking) den Grundsatz der Transparenz in Grids. Bei der Speicherung im Grid soll ein Replikatsdienst den schnellen Zugriff auf die Daten ermöglichen und durch verteilte Speicherung von Replikaten die Gefahr von Datenverlust durch Komponentenausfall minimieren. Die Systeme sind dabei so ausgelegt, dass die Speicherorte für die Replikate und deren dynamische Verlagerung für den Nutzer grundsätzlich transparent erfolgen. Für diese Fälle würde ein System zur Nachvollziehbarkeit der Datenströme eine erhebliche Änderung und Einschränkung der bisherigen Konzepte zur Speicherung von Daten in Grids bedeuten.

Schutz von Ein-, Ausgabedaten und Job-Informationen schwierig

Ein weiterer Datenschutzaspekt, der bei der bisherigen Anwendung von ARPs vollkommen unberücksichtigt bleibt und rein technisch sehr schwer umzusetzen ist, betrifft Ein- und Ausgabedaten sowie den Programm-Code von Jobs. Die Jobs werden auf fremden Systemen ausgeführt, die unter vollständiger Kontrolle eines Ressource-Providers sind und der damit potentiell sowohl Einsicht in Ein- und Ausgabedaten des jeweiligen Jobs nehmen kann als auch den Programm-Code der Jobs analysieren kann.

4.5.2 Anonymisieren und Pseudonymisieren in Grids

In Kommunikationsnetzen werden drei Arten der Anonymität unterschieden: Sender- und Empfängeranonymität sowie Unverkettbarkeit (Unlinkability) [PfWa 87, PfHa 07].

Arten der Anonymität

Bei der **Senderanonymität** ist es nicht möglich durch die Beobachtung eine Nachricht des Seders auf dessen Identität zu schließen. Auch für den Empfänger der Nachricht darf die Identität des Senders nicht ableitbar sein. Analog gilt dies für den Empfänger bei der **Empfängeranonymität**. Die **Unverkettbarkeit** ist gegeben, wenn ein Angreifer die Kommunikationsbeziehung zwischen Sender und Empfänger nicht erkennen kann (Verkehrsflussanalyse), auch wenn er in der Lage ist, den gesamten Netzverkehr abzuhören. Die Unverkettbarkeit muss auch ohne Empfänger- bzw. Senderanonymität realisierbar sein.

Diese Arten der Anonymität werden auch in Client-Server Systemen genutzt, der Client wird hier dem Sender, der Server dem Empfänger gleichgesetzt. Eine Empfängeranonymität wird in der Praxis bei P2P oder auch für Systeme zur anonymen Publikation gefordert. Auf den Grid-Kontext übertragen würde dies einer Anonymität des Ressourcen-Providers entsprechen. Eine Anonymität des Ressourcen-Providers wird in der Praxis nicht benötigt und kann deshalb unberücksichtigt bleiben. Allerdings wird im Grid häufig eine Anonymität der Ressourcen-Nutzung verlangt. Für einen Außenstehenden soll

nicht erkennbar sein, welche Ressourcen ein bestimmter Benutzer in welchem Umfang nutzt. Das heißt, im Grid Kontext bleiben die Forderungen nach

- Anonymität des Nutzers (Senderanonymität)
- Unverkettbarkeit
- Anonymität der Ressourcen-Nutzung.

Im folgenden wird dargestellt, wie diese Anforderungen im Grid umgesetzt sind oder sich umsetzen ließen.

Bei der vollkommenen Anonymisierung des Nutzers liegt eine nicht nur technische, sondern auch eine inhaltliche absolut irreversible Abtrennung der Daten von den dahinter stehenden Personen vor [DSMS 06]. Für Anwendungsfelder, die nicht durch rechtliche Vorschriften zur vollkommenen Anonymisierung verpflichtet sind, wird anstelle der Anonymisierung die Pseudonymisierung verwendet.

Nutzeranonymität: Pseudonymisierungsdienst

Im Rahmen von EGEE wird eine Pseudonymisierungsdienst vorgeschlagen. Zusätzlich wird gefordert, dass ein Dritter nicht in der Lage sein soll, die Aktivitäten eines Nutzers (z.B. Umfang der Ressourcen-Nutzung, welche Ressourcen werden genutzt, welche Anwendungen werden benutzt, usw.) abzuleiten [EGEE 05b]. Es wird aber davon ausgegangen, dass die wahre Identität vor der Mitgliedern derselben VO nicht geheim gehalten werden muss.

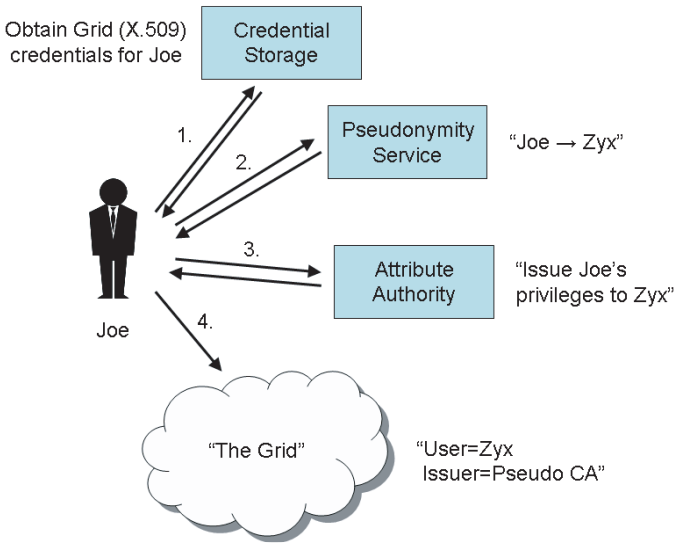

Abbildung 4.41: EGEE: Möglicher Ablauf einer Pseudonymisierung [EGEE 05b]

Die Pseudonymisierung (vgl. Abbildung 4.41) wird wie folgt durchgeführt:

1. Der Nutzer erhält ein Zertifikat mit seiner echten Benutzerkennung (im Bsp. Joe).
2. Mit diesem Zertifikat authentisiert sich der Nutzer beim Pseudonymisierungsdienst, der für ihn ein Pseudonym (im Bsp. Zyx) erstellt.

3. Der Nutzer authentisiert sich mit beiden Zertifikaten bei der Attribute Authority, die dann die Rechte von Joe an das Pseudonym Zyx binden kann.

4. Der Benutzer besitzt nun ein Zertifikat unter einem Pseudonym mit allen seinen ursprünglichen Rechten und kann damit die Grid–Dienste nutzen.

Pseudonymisierung muss vertrauenswürdig sein

Der Pseudonymisierungsdienst ist für alle Mitglieder einer VO ebenso wie die CA und die Attribute Authority (AA) eine vertrauenswürdige Komponente. Jeder Angreifer der Zugang zum Pseudonymisierungsdienst oder der AA erlangen kann, ist in der Lage die Pseudonyme wieder aufzulösen und im Beispiel von Abbildung 4.41 die Identität von Joe aufzudecken. Dementsprechend sind diese Komponenten besonders zu schützen. Die Pseudonymisierung stößt allerdings an ihre Grenzen, wenn aus den sonstigen Daten, die mit dem Pseudonym verknüpft werden, ein Rückschluss auf die individuelle Identität möglich wird. Im vorgestellten Ansatz werden von der AA Rechte an dass Pseudonym-Zertifikat gebunden. Je feingranularer das Autorisierungsschema und damit die Rechte sind, umso größer ist natürlich auch die Gefahr, dass Joe aufgrund seiner „einzigartigen“ Rechtekombination identifizierbar bleibt. Der Pseudonymisierungsdienst von EGEE nutzt nur den Grid-Nutzern. Für die Verarbeitung personenbezogener Daten innerhalb des Grids müssen diese natürlich vor der Submission ins Grid pseudonymisiert werden.

Problem: Rückschluss auf Identität über sonstige Daten möglich

Anonymität der Ressourcen-Nutzung

Die Pseudonymisierungsichert auch die Anonymität der Ressourcen-Nutzung. Selbst für den Ressource-Provider ist die Identität des Nutzers seiner Ressourcen nicht mehr erkennbar, er kann, unter den oben genannten Annahmen bezüglich der Rechte, nicht von Zyx auf Joe zurückschließen.

Es gibt auch noch weitergehende Forderungen bezüglich der Anonymität der Ressourcen-Nutzung. In einigen Anwendungsgebieten wird verlangt, dass verschiedene Nutzer, die auf der selben Ressource rechnen, nichts über die Ressourcen-Nutzung (Art und Umfang) eines anderen in Erfahrung bringen können. Dieser Spezialfall der Anonymisierung wird in Abschnitt 4.6 bei Sandboxing- und Virtualisierungstechniken betrachtet.

Unverkettbarkeit: Mixes oder Onion-Routing

Die Unverkettbarkeit, d.h. die Verschleierung von Kommunikationsbeziehungen, wird bei normalen Kommunikationsdiensten durch sogenannte Mixing-Dienste oder durch Onion-Routing realisiert. Im folgenden wird kurz das Konzept der (umkodierenden) Mixe, das auf eine Arbeit von David Chaum [Chau 85] zurückgeht, vorgestellt. Dabei werden die Nachrichten nicht direkt vom Sender zum Empfänger übertragen, sondern über mehrere Zwischenstationen — die als Mixes oder als Mix-Kaskade bezeichnet werden — geleitet. Ein Mix vermittelt also Nachrichten. Der Mix speichert die Nachrichten zwischen, führt eine kryptographische Transformation durch und schickt die Nachricht an das nächste Ziel (nächster Mix oder Empfänger). Durch das Store-and-Forward und eine stochastische Verzögerung wird erreicht, dass die vom Mix empfangenen Nachrichten nicht mit den versandten Nachrichten in Verbindung gebracht werden können. Durch Padding werden die Nachrichten oder Nachrichtenteile auf eine konstante Länge auf-

Verschlüsselung, Store and Forward, stochastische Verzögerung verschleiern Kommunikationsbeziehungen

gefüllt und damit der Umfang der Kommunikation verschleiert. Durch das Verschicken von Dummy-Nachrichten kann zusätzlich auch noch die Senderate verschleiert werden.

Die Nachrichten werden mit Hilfe von Public-Key Verfahren geschützt. Wenn ein Sender A eine Nachricht M an B verschicken und dazu den Mix m verwenden möchte erzeugt er die folgende Nachricht:

$$K_m(R_1, K_B(R_0, M), B)$$

K_m bezeichnet den öffentlichen Schlüssel des Mix m, K_B entsprechend den von B. R_i sind Zufallszahlen und B bezeichnet die Adresse von B. Die Nachricht wird an den Mix m geschickt, der den „Umschlag“ mit seinem privaten Schlüssel öffnen kann, aber selbst keinen Zugriff auf die mit dem Schlüssel von B verschlüsselte Nachricht hat. Er kann dann die innere Nachricht $(K_B(R_0, M), B)$ an B übermitteln. Im Normalfall wird allerdings nicht ein einzelner Mix, sondern eine Mix-Kaskade verwendet. Entsprechend wird bei n Mixes von A die folgende Nachricht erzeugt

$$K_n(R_n, K_{n-1}(R_{n-1}, \ldots, K_2(R_2, K_1(R_1, K_B(R_0, M), B)) \cdots))$$

und an den Mix m_n aus dem Mix-Netz gesendet, der seinen Umschlag auspackt, die Nachricht stochastisch verzögert und an Mix $n-1$ weiterleitet.

Unverkettbarkeit derzeit im Grid nicht im Einsatz

Verfahren zur Unverkettbarkeit und zur Verschleierung von Kommunikationsbeziehungen werden im Grid derzeit nicht eingesetzt. Dies liegt vor allem daran, dass durch die stochastischen Verzögerungen und die Mix-Kaskade zum einen eine erhebliche Verzögerung der Nachricht, zum anderen aber auch eine erhebliche Zunahme des Datenvolumens resultiert. Das Konzept der Mixe wurde ursprünglich für E-Mail Verkehr entwickelt [Chau 85, Mixminion], mittlerweile aber verallgemeinert und es gibt Dienste zur anonymen Nutzung des Internet [JAP, Tor]. Alle diese Anonymisierungsdienste haben aber das Problem der Verzögerung, schlechter Antwortzeiten und damit eine zum Teil erhebliche Einschränkung der Performance. Bisher war man im Grid nicht bereit diese Performance-Einbußen hinzunehmen.

4.5.3 Bewertung Privacy-Dienste: Pseudonymisierung

Im Grid Kontext besteht, wie auf Seite 146 dargestellt, die Forderung nach Nutzeranonymität, Anonymität der Ressourcen-Nutzung und Unverkettbarkeit. Vollständige Anonymität wird im Grid derzeit nicht realisiert. Ein Pseudonymisierungsdienst ist aber in der Lage eine gewisse Anonymität der Ressourcen-Nutzung und eine Nutzer-Pseudonymisierung zu realisieren. Da auch die Unverkettbarkeit derzeit überhaupt nicht umgesetzt wird, erfolgt nur eine Bewertung des Pseudonymisierungsdienstes nach dem Kriterienkatalog.

- **Middleware-Integration:**
  Privacy-Dienste wurden zwar vereinzelt spezifiziert, sind aber bisher

nicht im produktiven Einsatz oder in eine Middleware integriert (Bewertung: 0).

- **Ressourcen-Integration:**
  Im Hinblick auf die Ressourcen-Nutzung macht die Verwendung von Pseudonymen keinen Unterschied zur Nutzung der Ressourcen mit der echten Identität des Nutzers. Es sind auch keine Anpassungen auf lokalen Ressourcen durchzuführen. Insofern ist der Pseudonymisierungsdienst nahtlos auf den Ressourcen integrierbar (Bewertung: 3).

- **Erweiterbarkeit:**
  Nachdem der Pseudonymisierungsdienst nur im Rahmen von EGEE in einer sehr abstrakten Form spezifiziert, aber bisher nicht in den Middlewares umgesetzt wurde, kann eine mögliche Erweiterbarkeit derzeit nicht beswertet werden.

- **Middleware übergreifende Interoperabilität:**
  Die Spezifikation des Pseudonymisierungsdienstes lässt eine Middleware-unabhängige Implementierung grundsätzlich zu. Trotzdem wird dieses Konzept derzeit nicht produktiv eingesetzt und auch von keiner Middleware unterstützt (Bewertung 0).

- **Interoperabilität auf Policy-Ebene:**
  Der Nutzer oder eine VO kann derzeit nicht spezifizieren, wie die Pseudonymisierung zu erfolgen hat. Es gibt auch keine Policies auf Seite des Dienstbetreibers, die darüber Auskunft geben würde. Es ist kein Policy Konzept vorgesehen (Bewertung: 0).

- **Interoperabilität beim ID-Management:**
  Was die Abbildung von ID-Informationen auf Pseudonyme betrifft, existiert aufgrund des zentralen Pseudonymisierungsdienstes ein global eindeutiges Verfahren. Allerdings müssen die Ressourcen-Provider, wenn sie bisher die Identitätsinformation genutzt haben, um mit Pseudonymen umgehen zu können, die lokalen ID-Management Systeme anpassen. Eine Abstimmung über Namensräume ist zwar möglich derzeit aber allenfalls auf Ebene des Grid bzw. der VO vorgesehen (Bewertung: 1).

- **Trust Management; Formalisierung:**
  Der Pseudonymisierungsdienst ist eine zentrale Komponente, der ebenso wie einer CA vertraut werden muss. Alle Nutzer und alle Ressourcen-Provider müssen dem Dienst vertrauen. Das Vertrauensverhältnis ist binär und gilt global (Bewertung: 1).

- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Es existieren keine Verfahren, um Vertrauenswerte dynamisch zu berechnen oder abzuleiten (Bewertung: 0).

- **Trust Management; Granularität:**
  Die Vertrauenswerte werden lediglich implizit und Grid-global festgelegt (Bewertung: 0).

- **Rechtedelegation; Granularität und Sicherheit:**
  Bei der Pseudonymisierung überträgt der Nutzer das Recht zur Umsetzung seiner Identitätsinformation vollständig auf den Dienst. Ein Widerruf ist dadurch realisierbar, dass der Nutzer das ausgestellte Pseudonym-Zertifikat bzw. den gesamten Dienst nicht mehr nutzt (Bewertung: 1).
- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Eine Delegation der Pseudonymisierung ist nicht vorgesehen (Bewertung: 0).
- **Delegation von Policies:**
  Nachdem keine Verfahren zur Spezifikation von Policies vorgesehen sind, existiert auch kein Konzept zur Delegation (Bewertung: 0).
- **Delegation von Aufgaben:**
  Die Delegation von Aufgaben ist nicht vorgesehen (Bewertung: 0).
- **Mapping:**
  Es sind keine Mechanismen zur Abstimmung und zur Auflösung von Policy-Konflikten vorgesehen (Bewertung: 0).
- **Skalierbarkeit:**
  Die räumliche Verteilung des Grid hat genausowenig Einfluss auf die Skalierbarkeit des Pseudonymisierungsdienstes wie die Anzahl der Ressourcen (Hardware und Software) im Grid. Allerdings ist die Leistungsfähigkeit des Dienstes von der Anzahl der Nutzer abhängig (Bewertung: 2).
- **Flexibilität; Entitätenvielfalt:**
  In der gegenwärtigen Form unterstützt der Pseudonymisierungsdienst nur Endnutzer auf Ebene der VOs. Es ist also möglich Nutzer zwischen verschiedenen VOs zu pseudonymisieren. EGEE geht davon aus, dass innerhalb einer VO eine Pseudonymisierung nicht erforderlich ist. Auch die Pseudonymisierung von Organisationen, VOs oder Gruppen ist nicht vorgesehen. Auch die in Abschnitt 4.5.1 geforderte Sicherung von Ein- und Ausgabedaten sowie von Programm-Code und Jobs im Hinblick auf Datenschutzaspekte, lässt sich nicht realisieren. Mechanismen dafür sind weder beim Pseudonymisierungsdienst und den ARPs der Attribute Authority noch auf Seiten der Ressource-Provider vorgesehen (Bewertung: 1).
- **Flexibilität; Organisationsflexibilität:**
  Die Struktur des Dienstes ist grundsätzlich zentralistisch angelegt. Über eine Verbindung von Pseudonymisierungsdienst und Attribute Authority (z.B. von GridShib) ließe sich auch ein föderiertes Konzept umsetzen. Das heißt, der Identitätsprovider wäre gleichzeitig auch Betreiber des Pseudonymisierungsdienstes für seine Nutzer. In diesem Fall gäbe es aber keine Möglichkeit die Heimat-Organisation des Nutzers, d.h. die seines IDPs, zu verschleiern. Es wäre immer nachvollziehbar, aus welcher Organisation das Pseudonym stammt. Dies könnte, je nach Anwendungsfall — wenn z.B. immer nur sehr wenige Nutzer pro Organisation

als Grid-Nutzer auftreten — die Pseudonymisierung unwirksam machen (Bewertung: 1).

- **Administrierbarkeit:**
  Der Administrationsaufwand für den Pseudonymisierungsdienst sowohl auf Seiten des Dienstbetreibers als auch auf Seite des Ressourcen-Providers ist gering (Bewertung: 2).

- **Sicherheit; Sicherheitsniveau:**
  Bei einer zentralen Komponente, die alle Abbildungen von IDs auf Pseudonyme verwaltet, ist die Schadenshöhe bei einem erfolgreichen Angriff auf diese Komponente naturgemäß sehr hoch. Allerdings bietet die zentrale Komponente die Möglichkeit, dass diese sehr gezielt und mit sehr effektiven Mitteln geschützt werden kann. Damit kann die Eintrittswahrscheinlichkeit als niedrig angesehen werden (Bewertung: 1).

  Unter der Annahme, dass eine Auflösung von Pseudonymen, z.B. für Accounting-Zwecke o.ä., nicht gebraucht wird, wäre eine Speicherung der Pseudonyme überhaupt nicht mehr erforderlich. Der Dienst könnte ein Pseudonym erteilen und sofort wieder „vergessen“. Damit könnten auch bei einem erfolgreichen Angriff auf den Pseudonymisierungsdienst keine bereits vergebenen Pseudonyme mehr rekonstruiert, sondern nur noch aktuell zu vergebende mitprotokolliert werden. Die Schadenshöhe eines erfolgreichen Angriffs wäre mittel und die Bewertung des Sicherheitsniveaus würde auf 2 steigen.

  Für die strengen gesetzlichen Regelungen bei der Verarbeitung medizinischer und insbesondere von Patientendaten und die in Abschnitt 4.5.2 beschriebenen strengen Anforderungen an den Datenschutz sind die präsentierten Mechanismen nicht geeignet. Eine Einschränkung der Institutionen die Daten verarbeiten dürfen, wäre über ARPs eventuell noch umsetzbar, eine Beschränkung auf einzelne Nutzer mit GridShib jedoch nicht. Mechanismen, die revisionssichere Nachweise über die Datennutzung und Löschung (Auditing und Tracking) sowie eine schnelle, sichere und umfassende Umsetzung einer Forderung nach Löschung von Daten durch den Patienten, sind nicht vorgesehen. Die Nutzung und Unterstützung spezieller medizinischer Datenformate ist nicht vorgesehen.

- **Sicherheit; Zusicherung, QoP:**
  Die Spezifikation des Pseudonymisierungsdienstes gibt keine Anhaltspunkte wie der Dienst sicher zu implementieren wäre. Es sind auch keine Mechanismen vorgesehen, um das Sicherheitsniveau prüfen oder bewerten zu können (Bewertung: 0).

Abbildung 4.42 fasst die Ergebnisse für die Bewertung des Pseudonymisierungsdienstes zusammen.

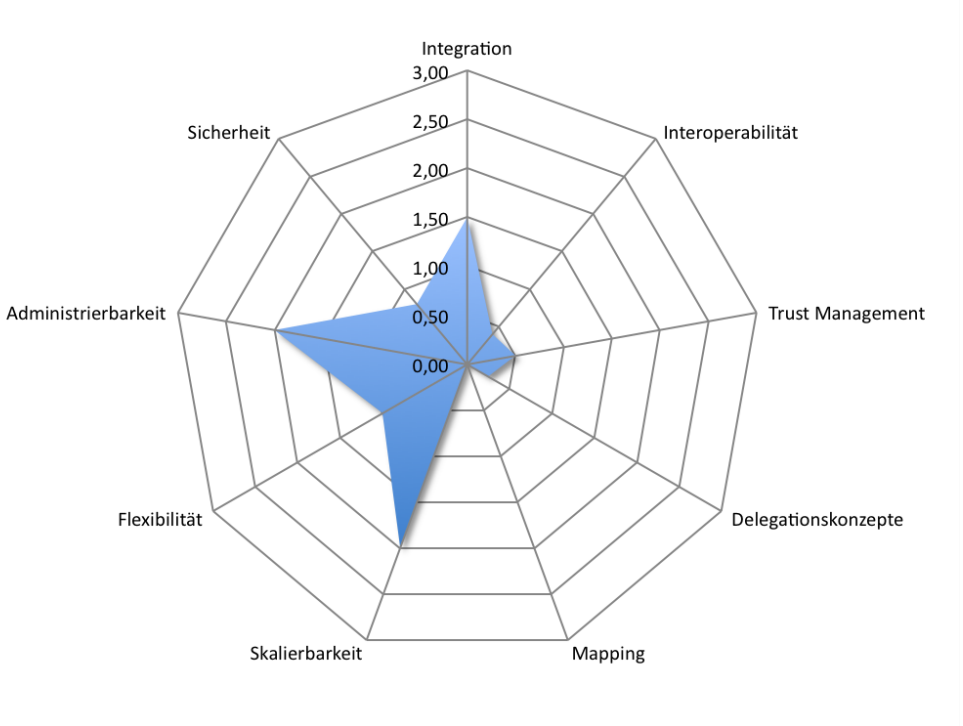

Abbildung 4.42: Bewertung von Privacy Diensten

4.6 Sandboxing und Virtualisierung

Der Einsatz von Sandboxing [CIK 00] und Virtualisierungstechniken werden im Grid aufgrund zweier Anforderungen diskutiert:

Beschränkung nicht vertrauenswürdiger Programme

1. Es sollen nicht vertrauenswürdige Programme bzw. Code innerhalb einer sicheren isolierten Umgebung (der Sandbox) ausgeführt werden. Das Ziel dabei ist, eventuellen Schaden an den Ressourcen auf die Sandbox zu beschränken und damit zu minimieren. Die Sandbox garantiert dabei, dass die unterliegende Maschine nicht negativ beeinflusst wird.

Nutzer voreinander verschatten

2. Unterschiedliche Nutzer sollen auf derselben Ressource voreinander verschattet werden. Ein Nutzer A soll keinerlei Informationen über einen Nutzer B erlangen können. Im Idealfall soll gar nicht erkennbar sein, dass ein anderer Nutzer die Ressource verwendet.

Der Begriff Sandbox wird insbesondere im Zusammenhang mit Java und der Java Virtual Machine (JVM) verwendet. Im folgenden Abschnitt wird die Java Sandbox und deren Verwendung im Grid vorgestellt. Im Anschluss daran werden Virtualisierungstechniken eingeführt. Da sich bei der Ressourcen- bzw. Betriebssystemvirtualisierung noch keine einheitliche Begriffsbildung durchgesetzt hat, werden die unterschiedlichen interessierenden Verfahren und die verwendeten Begriffe definiert.

4.6.1 Java Sandbox

Die Java Virtual Maschine stellt eine in Software realisierte Stack-Maschine dar. Java bietet eine mächtige Sicherheitsarchitektur [Oaks 01, SecArch1.2] mit den zentralen Komponenten Security Manager und Classloader.

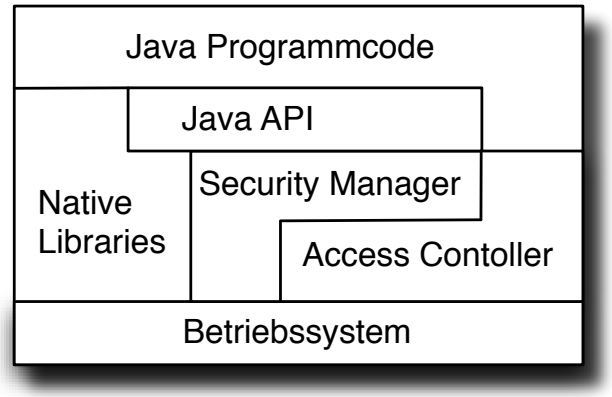

Abbildung 4.43: Komponenten der Java Sicherheitsarchitektur

Sandbox = Security Manager + Access Controler

Der **Security Manager** realisiert in Kooperation mit dem **Access Controler** die Sandbox (vgl. Abbildung 4.43). Basierend auf Policies kontrolliert und beschränkt der Security Manager, unter Zuhilfenahme von Methoden des Access Controlers, den Zugriff auf System-Ressourcen. Das heißt, für Java Programme kann z.B. der Zugriff auf das Dateisystem, die Netzwerkverbindungen usw. beschränkt oder ganz unterbunden werden. Das Java Programm wird in eine Sandbox eingeschlossen, die es nur an wohldefinierten, per Policy gesteuerten Schnittstellen verlassen kann.

Classloader definiert Namensraum für Objekte

Der **Classloader** kann mehrfach instantiiert werden und definiert einen eindeutigen Namensraum. Alle Objekte, die von einem Classloader geladen wurden, bilden einen in sich abgeschlossenen Namensraum und sind von Objekten, die von anderen Classloadern instantiiert wurden, weder sichtbar noch adressierbar. Damit lassen sich voneinander abgeschottete Ausführungsumgebungen realisieren.

Da die gängigen Grid Middlewares in Java programmiert sind, können die Java Sicherheitskonzepte für Grid Jobs verwendet werden. Grid Dienste oder Grid Jobs, die voreinander abgeschottet werden sollen, müssen von jeweils eigenen Classloadern geladen werden. Für den Zugriff auf Ressourcen können mit Hilfe der Security Manager Policies jeweils eigene Sandboxes realisiert werden. Mit dieser Vorgehensweise lassen sich beide Forderungen an Sandboxing und Virtualisierung (vgl. Seite 152) realisieren.

Grenzen der Java Sandbox im Grid

Allerdings sind mit der Verwendung der Java Sandbox im Grid-Kontext zwei Defizite oder Schwächen verbunden:

1. Änderung bzw. Anpassung der Grid Middleware:

Die Anwendung verschiedener Classloader erfordert in der Regel eine Änderung in der Implementierung der Grid Middleware. In Globus Toolkit 4 werden bspw. alle Dienste durch ein und denselben Classloader geladen.

2. Sicherheitskonzept nur für reine Java Implementierungen wirksam:
   Damit bestehende Legacy-Anwendungen aus Java und der JVM heraus genutzt werden können, wird das **Java Native Interface (JNI)** angeboten (vgl. Abbildung 4.43). Mit Hilfe von JNI wird der Aufruf von nativen Betriebssystem-Routinen und Bibliotheksfunktionen, die bspw. in C oder C++ realisiert sind, möglich. Der Aufruf von Routinen über JNI erfolgt außerhalb des Sandbox, d.h. die Sandbox-Restriktionen sind hierfür nicht anwendbar.

Insbesondere im Grid-Umfeld ist der letzte Punkt von entscheidender Bedeutung. Da für viele Spezialressourcen Legacy Code vorhanden ist und häufig genutzt werden muss, ist die Grundannahme der Java Sicherheitsarchitektur, dass alle Grid-Dienste in Java implementiert werden, nicht haltbar. Die Java Sandbox als alleiniger Mechanismus ist damit nicht ausreichend.

4.6.2 Betriebssystemvirtualisierung mittels Betriebssystem-Container

`chroot`-Umgebung

Beschränkung der Prozesse auf Teilbaum im Dateisystem

Relativ alte Ansätze, um Betriebssystem-Umgebungen voreinander abzuschotten, sind `chroot`- bzw. jail-Umgebungen. Change Root (`chroot`) ist ein Systemaufruf, um das Wurzelverzeichnis im Dateisystem (root-Verzeichnis) zu ändern. Der Systemaufruf wirkt auf den aktuellen Prozess und alle seine Kindprozesse. Es wird im Dateisystem ein Teilbereich als `chroot`-Umgebung angelegt und es werden dort Kopien der Systembibliotheken usw. abgelegt. Nach dem Systemaufruf `chroot` betrachten alle (Kind-) Prozesse diesen Teilbereich als ihr Wurzelverzeichnis. Insbesondere sollte dieser Teilbaum nicht verlassen und auf andere Verzeichnisse außerhalb zugegriffen werden können. Trotzdem werden Informationen über andere Prozesse aus fremden Umgebungen bei der Anzeige von Prozessinformationen mit angezeigt. Der Systemaufruf wird von Linux und einigen anderen Unix-Varianten implementiert. Damit wird eine einfache Sandbox realisiert.

Jails: Unterstützung durch zusätzliche Kernel-Strukturen

Jails [jails] sind ein Konzept, das von dem Betriebssystem FreeBSD unterstützt wird. Vom Prinzip arbeitet es wie `chroot`, allerdings werden Jails zusätzlich durch Kernel-Datenstrukturen unterstützt, die eine Interaktion zwischen Prozessen aus verschiedenen Jails unterbinden. So erhält bspw. jede Jail-Umgebung eine eigene IP-Adresse und kann bestimmte Systemaufrufe nicht nutzen.

Das Problem von `chroot` und Jails ist die fehlende Systemunabhängigkeit. Beide Konzepte werden nur von bestimmten Betriebssystemen unterstützt und widersprechen somit dem Virtualisierungsprinzip in Grids. Eine Verschattung der Nutzer voreinander (Forderung 2 auf Seite 152) ist damit

kaum realisierbar. Es ist auch nicht möglich damit virtualisierte Hardware-Ressourcen zu realisieren.

Ausbrechen aus `chroot` möglich

Für `chroot`-Umgebungen sind mittlerweile Verfahren bekannt geworden, die ein Verlassen der `chroot`-Umgebung möglich machen [chrbreak, chrjail].

4.6.3 Paravirtualisierung

Host-Betriebssystem und Hypervisor kontrollieren Zugriffe auf Ressourcen

Bei der **Paravirtualisierung** läuft im Host-Betriebssystem der Ressource ein **Virtual Machine Monitor (VMM)**, oft auch als **Hypervisor** bezeichnet, der eine angepasste API anbietet, die von den virtualisierten Gast-Betriebssystemen genutzt werden kann. Ein Systemaufruf aus dem Gast-Betriebssystem zu diesem Hypervisor wird deshalb auch häufig als hypercall bezeichnet. Das Host-Betriebssystem und der VMM werden auf Kernel Level (Ring 0), die Gast-Systeme auf Ebene der Gerätetreiber (Ring 1) ausgeführt. Auf dem Hypervisor können mehrere Betriebssysteminstanzen gleichzeitig ausgeführt werden. Aus Sicht eines Gast-Systems entsteht der Eindruck einen vollständigen Rechner nutzen zu können. Fremde Gast-Betriebssysteme, der Hypervisor oder das Host-Betriebssystem sind vom Gast-System aus weder sichtbar noch manipulierbar. Das Konzept der Paravirtualisierung mit seiner angepassten API ist nicht vollständig transparent für das Gast-Betriebssystem und dementsprechend muss das Gast-System an den Hypervisor angepasst und entsprechend verändert werden. Daher kommen nur Betriebssysteme als Gast in Frage, von denen der Quell-Code zugänglich und veränderbar ist. Ein Beispiel für Paravirtualisierung ist Xen (bis zur Version 2.0) [Univ 07]. Auf Xen können verschiedenste Linux Derivate sowie Sun Solaris betrieben werden.

Anpassung der Gast-Betriebssysteme notwendig

4.6.4 Hardware-unterstützte Virtualisierung

Die Prozessor-Hersteller Intel und AMD unterstützen mit ihren Technologien VT-x und VT-i [NSL+ 06] bzw. AMD-V [Zeic 06a, Zeic 06b, Whit 07] Virtualisierungstechniken. Damit wird es virtualisierten Gast-Betriebssystemen ermöglicht Systemaufrufe (z.B. für I/O u.ä) „direkt“ auf Prozessorebene aufzurufen. Damit entfällt der Zwang das Gast-Betriebssystem für einen Hypervisor und dessen API anzupassen, d.h. es können damit auch kommerzielle Betriebssysteme, für die kein Zugriff auf den Quell-Code gegeben ist, virtualisiert werden. Um die virtuellen Gastsysteme voreinander und vor dem Host-Betriebssystem zu schützen, kennen die Prozessoren mit Virtualisierungstechnik zwei unterschiedliche Ausführungsmodi, den so genannten **root mode** und den **non-root mode**. In beiden Modi werden Ring 0 bis Ring 3 Systemaufrufe unterstützt. Die Ringe unterscheiden dabei die Zugriffsrechte auf die Hardware. Je höher die Zahl, umso geringer sind die Rechte. Ring 0 bezeichnet den Kernel oder System-

Hardware unterstützt Systemaufrufe von virtualisierten Systemen

Zwei Ausführungsmodi: root mode für Host-OS non-root mode für Gast-OS

modus, Ring 1 und Ring 2 sind Ringe für Gerätetreiber und Ring 3 ist dem Anwendungsprogramm vorbehalten.

Der VMM läuft bei dieser hardware-unterstützten Virtualisierung im root mode, mit vollem Zugriff auf die reale Hardware. Die Gast-Betriebssysteme werden im non-root mode ausgeführt und der Hardware Zugriff durch Systemaufrufe in Ring 0 bis Ring 2 wird durch den VMM im root mode kontrolliert und können entsprechend beschränkt werden. Mit dieser Technik wird es ermöglicht Betriebssysteme unverändert zu virtualisieren. Beispiele für Hardware-unterstützte VMM sind Xen 3.0, VMWare oder Parallels [Univ 07, VMware, Parallels].

Mit den letzten beiden Virtualisierungstechniken lassen sich mehrere virtuelle Rechner auf einer physischen Hardware-Plattform betreiben und sicher voreinander abschotten. Die für den einzelnen virtuellen Rechner zur Verfügung stehenden Hardware Ressourcen können durch die Virtualisierung beschränkt werden.

4.6.5 Virtualisierung im Grid

Für eine sehr einfache Version einer „Sandbox“ werden in Globus und gLite dynamische Accounts empfohlen und verwendet [EGEE 05b, KWL+ 04, KeWe 02]. In gLite ist bspw. LCAS/LCMAPS (Local Centre Authorization Service / Local Credential Mapping Service) [CJK+ 04, ACC+ 03] in der Lage die VOMS-Attribute (vgl. Abschnitt 4.3.1) zu parsen. Aufgrund der Attribute werden für die Nutzer lokale User- und dynamische Gruppen-IDs vergeben. Die Separation der Nutzer erfolgt über die Unix-Accounts, eigene home-Verzeichnisse mit entsprechend gesetzten Zugriffsrechten und den `setuid`- und `setgid`–Mechanismen. Die Systemaufrufe `setuid` und `setgid` können die tatsächliche sowie die effektive Benutzer ID von Prozessen ändern. Die Prozesse, die von der Middleware stellvertretend für einen Nutzer erzeugt werden, erhalten über `setuid` und `setgid` die lokal zugeteilte und nur lokal gültige User-ID und Group-ID des jeweiligen Nutzers. Die von den Prozessen verwendeten und erzeugten Daten werden über normale Unix Dateisystemrechte vor dem Zugriff durch andere Nutzer geschützt. Mit diesen Verfahren wird aber keine der beiden Anforderungen von Seite 152 erfüllt. Es kann also weder von einer Sandbox noch von einer Virtualisierungslösung gesprochen werden. Ein weiteres Problem dieses Ansatzes ist die Verwendung von anonymen Gruppenkennungen (Pool Accounts, vgl. Abschnitt 4.2) und die Wiederverwendung von User-ID und Group-ID. Damit besteht die Möglichkeit, dass ein späterer Nutzer auf nicht gelöschte Daten eines früheren Nutzers — der zufällig dieselbe ID hatte — zugreifen kann. Aus diesen Gründer wird dieser Ansatz nicht weiter betrachtet und bewertet. Da die beschriebenen Verfahren vielfach eingesetzt und mit dem Begriff Sandbox in Verbindung gebracht werden, wurden sie hier vorgestellt.

Im Grid-Kontext werden alle drei Virtualisierungsalternativen diskutiert. Ein breiteres Einsatzfeld haben bisher lediglich Betriebssystem-Container auf Ba-

sis von `chroot` erlangt. Diese sind aber, wie bereits im Abschnitt 4.6.2 diskutiert, nicht in der Lage, die verschiedenen Container sauber und vollständig voreinander zu verschatten und ein Ausbrechen aus der `chroot`-Umgebung sicher zu verhindern.

`chroot` findet breites Einsatzfeld

Eine weitere Alternative, die in der Literatur diskutiert wird [SFEF 06, SFE+ 06, SEADL 05], sind virtualisierte Grid-Knoten auf Basis von Paravirtualisierung oder Hardware-unterstützter Virtualisierung. Hierbei wird für jeden Benutzer bzw. für Benutzergruppen ein Grid-Knoten als virtualisierte Instanz erzeugt. Das heißt, es wird eine komplette virtuelle Maschine inklusive der Grid-Middleware und der notwendigen Anwendungssoftware erzeugt. Mit den angesprochenen Virtualisierungslösungen lassen sich Images von Virtuellen Maschinen erzeugen, die auf Hintergrundspeichern abgelegt werden können. Bei einigen Virtualisierungslösungen besteht auch die Möglichkeit dieses Image zur Laufzeit zu erzeugen, d.h. die virtuelle Maschine „einzufrieren“. Mit dieser Suspend/Resume Technik kann das Image auf ein anderes System verlagert und dort wieder gestartet werden.

virtualisierte Grid Knoten

Abbildung 4.44 stellt einen exemplarischen Deployment Prozess für einen virtualisierten Grid-Knoten dar. In einem ersten Schritt werden vom Benutzer seine Anforderungen ermittelt (z.B. benötigte Bibliotheken, Anwendungsprogramme u.ä.). Basierend auf diesen Daten erzeugt der Grid-Nutzer selbst oder ein unabhängiger Lösungs-Provider ein Xen-Image, das ein Linux-Betriebssystem, die verwendete Middleware und notwendige Anwendungsprogramme enthält (vgl. Nr. 2 und 3 in Abbildung 4.44). Dieses Xen-Image mit der Spezifikation der Anforderungen bezüglich CPU, Haupt- und Hintergrundspeicher wird über das Netz zu einem Ressource-Provider übertragen, der diese Anforderungen erfüllen kann (Schritt 4). Der Ressource-Provider erstellt ein Konfigurationsdatei für die Xen-Instanz (genannt XenU) und erzeugt die Instanz (Schritt 5). Im Schritt 6 wird das übertragene Image gebootet und schließlich der Grid Job ausgeführt. Nach der Ausführung des Jobs kann das Image gelöscht oder für eine spätere Wiederverwendung gespeichert werden.

In [SFEF 06] werden verschiedene Alternativen der Integration von Virtualisierungs- und Java Sandbox Techniken diskutiert. Abbildung 4.45 stellt die beiden wichtigsten Alternativen dar.

virtuelle Knoten: Integrationsalternativen

In beiden Alternativen werden für jeden Nutzer bzw. für jede Nutzergruppe virtualisierte Betriebssystem-Instanzen erzeugt, die in einer Java Sandbox die vollständige Grid Middleware enthalten. Die nativen Bibliotheken und Legacy-Anwendungen werden innerhalb des virtuellen Systems gekapselt. Als zentraler Zugangspunkt wird in der Alternative links in Abbildung 4.45 eine minimale Middleware Instanz eingesetzt, die vollständig innerhalb einer Java Sandbox gekapselt ist. Durch die Kapselung in der Sandbox wird verhindert, dass das Host-Betriebssystem angegriffen werden kann. Diese Middleware Instanz hat nur die Aufgabe virtualisierte Systeme zu instantiieren. Obwohl für das Zugangssystem nur ein minimaler Grid Middleware Stack verwendet wird, werden trotzdem sehr viele Komponenten benötigt um dies zu realisieren (z.B. Tomcat, Axis, GT4, usw.). Diese Komponen-

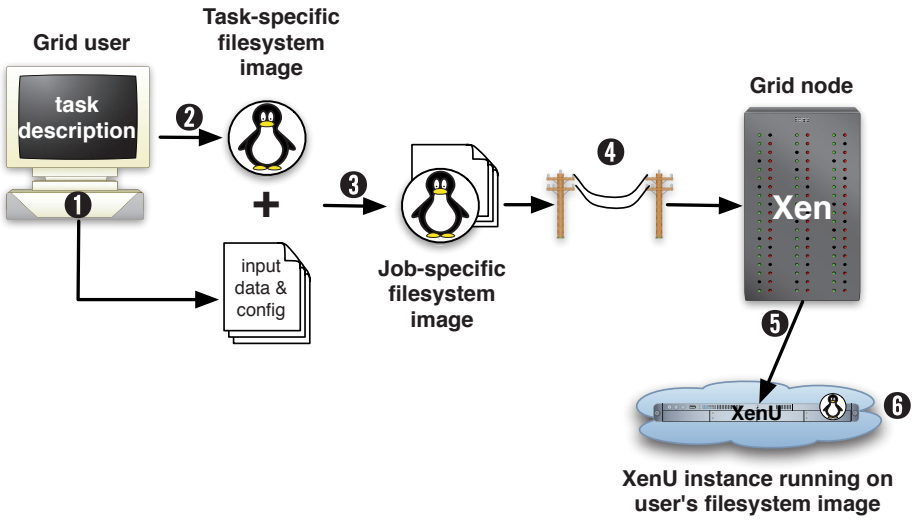

Abbildung 4.44: Beispiel eines Deployment Prozesses für einen virtualisierten Grid-Knoten basierend auf Xen [SFE+ 06]

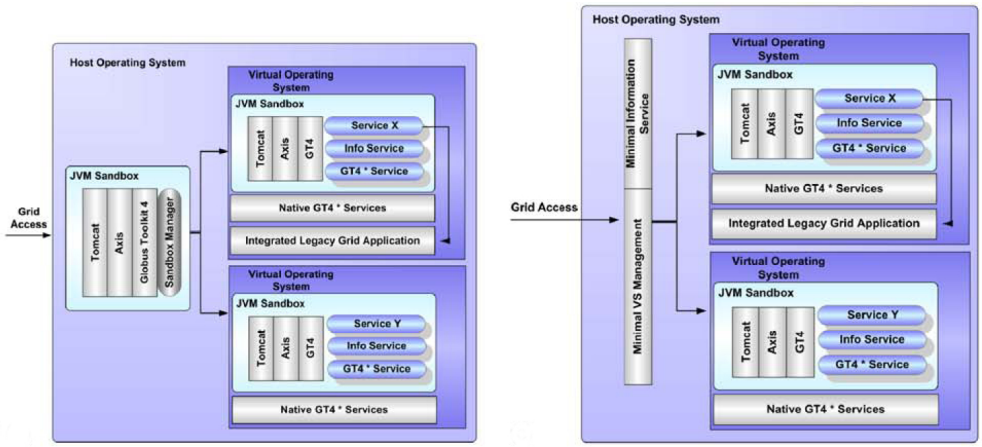

Abbildung 4.45: Möglichkeiten der Integration virtualisierter Grid-Knoten [SFEF 06]

ten können natürlich Implementierungsschwachstellen beinhalten, die von einem Angreifer ausgenutzt werden könnten. Das heißt, falls eine Schwachstelle z.B. im Tomcat bekannt würde, könnte ein Angreifer darüber das Host-Betriebssystem angreifen und dann alle virtuellen Systeme kontrollieren und manipulieren.

Um die Angriffsfläche möglichst klein zu halten, wird in der Alternative rechts in Abbildung 4.45 ein sehr schlankes Management Interface vorgeschlagen, das als einzige Aufgabe die Verwaltung virtualisierter Maschinen hat. Das Problem bei dieser Lösung liegt in der mangelnden Unterstützung existierender Grid-Dienste durch dieses schlanke Management Interface. Bei-

spielsweise kann diese Komponente keine VO-Management Dienste direkt nutzen. Es muss zuerst ein virtualisierter Grid Knoten erzeugt werden, der dann seinerseits VO-Management Dienste nutzt. Dies führt zu einer weiteren Komplexitätssteigerung im Grid Workflow und damit zu einer Anpassung der Nutzungsmuster.

4.6.6 Bewertung Sandboxing und Virtualisierung im Grid

Im folgenden Teilabschnitt werden die sinnvoll im Grid einsetzbaren Virtualisierungsmechanismen der Paravirtualisierung bzw. der Hardware-unterstützen Virtualisierung bewertet. Die in gLite massiv eingesetzten „Sandbox“ Technik der dynamischen Accounts in Verbindung mit `setuid` Mechanismen oder Betriebssystem-Containern werden nicht bewertet. Diese Verfahren werden als nicht sicher angesehen (vgl. Abschnitt 4.6.2) und die Basisanforderungen an Virtualisierungsmechanismen im Grid (vgl. S. 152) werden von diesen Mechanismen nicht erfüllt.

- **Middleware-Integration:**
  Alle Virtualisierungslösungen sind unabhängig von der konkret verwendeten Middleware implementiert. Eine Integration ist nicht vorgesehen. Die Virtualisierung liefert die Ausführungsumgebung, in der die Middleware installiert wird (Bewertung: 0).
- **Ressourcen-Integration:**
  Die Virtualisierungstechniken werden entweder von bestimmten Betriebssystemen unterstützt oder sind auf bestimmte Hardware-Architekturen angewiesen. Für diese Systeme ist eine nahtlose Integration gegeben (Bewertung: 3).
- **Erweiterbarkeit:**
  Eine Erweiterung der Virtualisierungssysteme um weitere Sicherheitsmechanismen ist möglich [SFEF 06], erfordert aber eine Grid-weite Abstimmung (Bewertung 2).
- **Middleware übergreifende Interoperabilität:**
  Die virtualisierte Plattform dient als Basis um eine Middleware aufzubauen. Der Mechanismus ist dementsprechend in der Lage jede Middleware zu unterstützen (Bewertung 3).
- **Interoperabilität auf Policy-Ebene:**
  Für die Spezifikation von Policies, welche die Instantiierung und die Verwendung von virtualisierten Systemen regeln, sind keine Konzepte vorgesehen. Dies wäre aber dringend erforderlich, um beispielsweise den in Abbildung 4.44 vorgestellten Deployment-Prozess für virtualisierte Grid-Knoten überhaupt automatisieren zu können. (Bewertung: 0).
- **Interoperabilität beim ID-Management:**
  Die Virtualisierungslösungen unterstützen, bis auf jails und `chroot`,

keine ID-Management Mechanismen. Eine Anbindung an globale oder lokale ID-Managementsysteme müsste entwickelt werden. (Bewertung 0).

- **Trust Management; Formalisierung:**
  Den Virtualisierungslösungen liegt ein implizites Vertrauensmodell zugrunde. Der Nutzer muss dem Ressource-Provider bzw. dem Betreiber der Virtualisierungslösung vertrauen. Dieser wiederum vertraut dem Nutzer nicht. (Bewertung 1).
- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).
- **Trust Management; Granularität:**
  Die Granularität des Trust Managements beschränkt sich auf sehr grobe Klassen. Eine feingliedrige Unterteilung wird nicht unterstützt und müsste selbst implementiert werden (Bewertung: 0).
- **Delegation:**
  Bei der Virtualisierung wird ein virtualisiertes Ausführungssystem zur Verfügung gestellt, innerhalb dessen die eigentlichen Grid Anwendungen laufen. Eine Delegation von Rechten, Policies und Aufgaben findet nicht statt (Bewertung 0).
- **Mapping:**
  Für die Instantiierung von virtuellen Maschinen und deren Eigenschaften sind keine Mechanismen vorgesehen, um ein Policy-Mapping zwischen Quell- und Zieldomäne durchzuführen. Es gibt beispielsweise keine Mechanismen um einen Konflikt aufzulösen, der durch die Anforderungen des Kunden bezüglich virtualiserter Hardware (CPU, Speicher, usw.) induziert wird (vgl. auch Abbildung 4.44 (Bewertung: 0).
- **Skalierbarkeit:**
  Virtualisierten Grid-Knoten müssen vor der Instantiierung bestimmte Anteile der realen Hardware in Form von virtualisierten Ressourcen zugewiesen werden, die das Gast-System bzw. der virtuelle Grid-Knoten nutzen kann. Durch die virtualisierten Grid-Knoten entsteht ein Verteilungs- und Zuteilungsproblem für die realen Ressourcen zu virtuellen Ressourcen. Insbesonders bei hoch dynamischen oder stark wechselnden Anforderungen führt dies zu Skalierungsproblemen. Auch die organisatorische Skalierbarkeit ist fraglich. Ein Einsatz eines virtuellen Grid-Knotens für jeden Nutzer einer VO erscheint kaum umsetzbar. Die räumliche Ausdehnung des Grids hat jedoch keinen Einfluss auf die Skalierbarkeit (Bewertung: 1).
- **Flexibilität; Entitätenvielfalt:**
  Die Virtualisierungslösungen sind nicht in der Lage mit Grid-Entitäten direkt umzugehen. Hierfür wären Erweiterungen (wie in Abschnitt 4.6.5 diskutiert) notwendig. (Bewertung: 0).

- **Flexibilität; Organisationsflexibilität:**
  Der Sicherheitsmechanismus der Virtualisierung ist aus dem Blickwinkel des Schutzes der Ressourcen vor böswilligen Nutzern entstanden. Dementsprechend liegt die gesamte Entscheidungsbefugnis über die virtuellen Systeme beim Ressourcen-Provider (Bewertung: 0).

- **Administrierbarkeit:**
  Die Bereitstellung, Pflege, Verwaltung, Speicherung und Aktualisierung von Images sowie das Scheduling bei der Instantiierung dieser Instanzen ist mit einem erheblichen administrativen Aufwand verbunden, auch wenn bei dem Deployment Modell in Abbildung 4.44 dieser Aufwand vollständig auf den Nutzer ausgelagert wird. Bei diesem Modell stellt der Grid Provider nur die Xen basierte Ausführungsumgebung bereit, der Kunde muss seinen Grid Knoten selbst konfigurieren und ein Xen Image erzeugen (Bewertung: 0).

- **Sicherheit; Sicherheitsniveau:**
  Mit Hilfe der Virtualisierung lässt sich ein hohes Sicherheitsniveau erreichen. Durch die virtualisierte Maschine ist die Auswirkung eines Angriffs auf diese VM-Instanz beschränkt. Die Schadenshöhe kann als niedrig angenommen werden, solange der Angriff auf die VM-Instanz beschränkt bleibt. Wichtig ist es in diesem Zusammenhang aber auch, die Eintrittswahrscheinlichkeit für einen Angriff gegen den Hypervisor, die Virtualisierungslösung und damit gegen das Host-Betriebssystem zu betrachten. Hier ist von einer niedrigen Eintrittswahrscheinlichkeit auszugehen. (Bewertung: 3).

- **Sicherheit; Zusicherung, QoP:**
  Es sind keine Verfahren vorgesehen, um den Nutzer über das Sicherheitsniveau oder die Realisierung der Virtualisierung zu informieren. (Bewertung: 0).

Abbildung 4.46 fasst die Bewertung der Virtualisierungstechniken zusammen.

4.7 Sicherheitsmanagement: Basisdienste

Eine wichtige Aufgabe des Sicherheitsmanagement ist die Überprüfung und Überwachung des Grid auf Sicherheitsverletzungen oder auf die Degradierung des Sicherheitsniveaus. Diese Aufgabe wird dem Auditing zugeordnet.

Um diese Überprüfungen überhaupt durchführen zu können, müssen die Dienste der Middleware und insbesondere die Sicherheitsdienste diese Daten protokollieren. Die entsprechenden Logging-Mechanismen werden im Abschnitt 4.7.1 betrachtet. Damit der Sicherheitsadministrator zeitnah Kenntnis

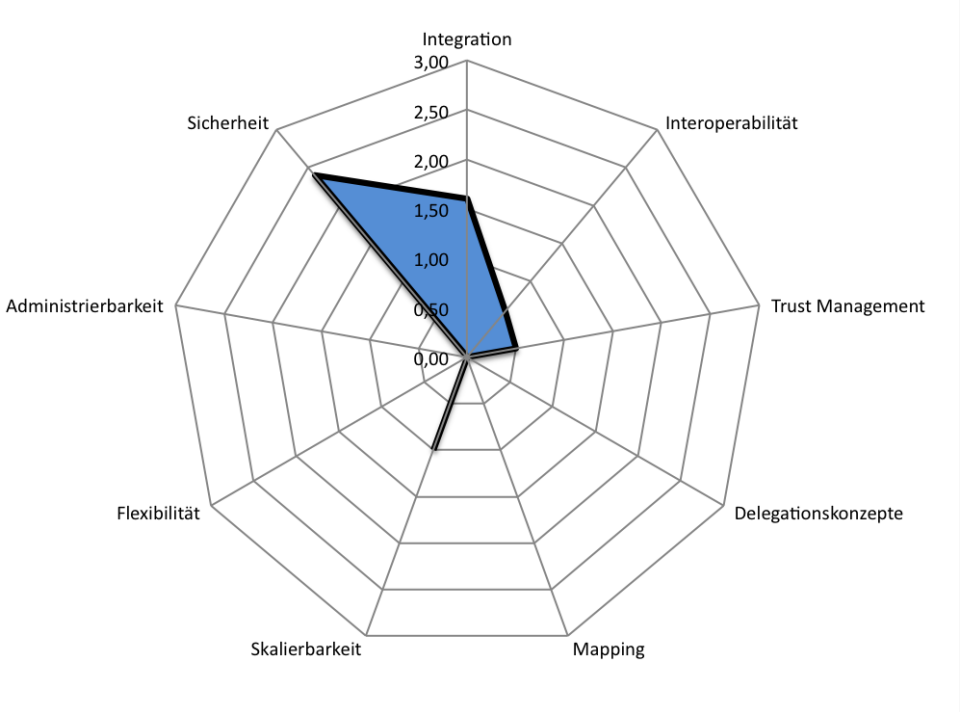

Abbildung 4.46: Bewertung Virtualisierung

von Sicherheitsvorfällen erlangen kann, werden Alarmmechanismen verwendet. Diese werden in Abschnitt 4.7.3 untersucht.

4.7.1 Logging

sichere und verlässliche Protokollierung

Ein sicheres Logging aller Dienste und insbesondere der Sicherheitsdienste umfasst die Erzeugung von Log-Records, d.h. Nutz- und Zugriffsdaten oder sonstiger Ereignisse werden protokolliert. Der Begriff umfasst die verlässliche und richtige Protokollierung, d.h. das Logging darf nicht unterbrechbar oder im Nachhinein änderbar sein. Auch ein erfolgreicher Angriff auf einem System darf nicht dazu führen, dass Logging Daten manipuliert werden können. Da die Logging Daten auf den Ressourcen anfallen, aber nicht auf jeder Ressource separat ausgewertet werden sollen, gibt es den Bedarf nach einem oder mehreren zentralen Log-Servern. Die lokale Log-Systeme sollten die Möglichkeiten bieten die Log-Records auf einen solchen Server weiterzuleiten. Für den Sicherheitsadministrator soll es einfach möglich sein eigene Logging Events zu definieren und Log Quellen festzulegen.

Ein sicheres Logging kann bspw. als Basisdienst für die Verbindlichkeit genutzt werden. Insbesondere im interorganisationalen Umfeld ist die Weitergabe von Logging-Daten, z.B. an die VO (für Abrechnungszwecke o.ä.) ein wichtiger Aspekt. Hierbei sind natürlich Fragen des Datenschutzes und Policies zur Weitergabe von Protokollierungsinformationen zu berücksichtigen.

gLite: Logging und Bookkeeping Service

In gLite gibt es für das Logging einen eigenen Dienst, den **Logging and Bookkeeping Service (L&B)**. Dieser Service dient primär dazu mit Hilfe des Workload Management Systems (WMS) Job-Informationen zu protokollieren und über eine Abfragesprache zugänglich zu machen [Kren 05]. Für jeden erzeugten Job wird eine eindeutige nicht wiederverwendbare ID vergeben, mit der die Job Informationen zweifelsfrei zugeordnet werden können. Alle wichtigen Ereignisse beim Prozessablauf erzeugen Events mit spezfischen Attributen. Diese Events werden von einer lokalen Logging-Komponenten (`locallogger`) auf der Ressource gesammelt und lokal gespeichert. Eine weitere Komponente (`interlogger`) leitet die Events an einen oder mehrere L&B Server (vgl. Abbildung 4.47) weiter.

Protokollierung von Job-Infos

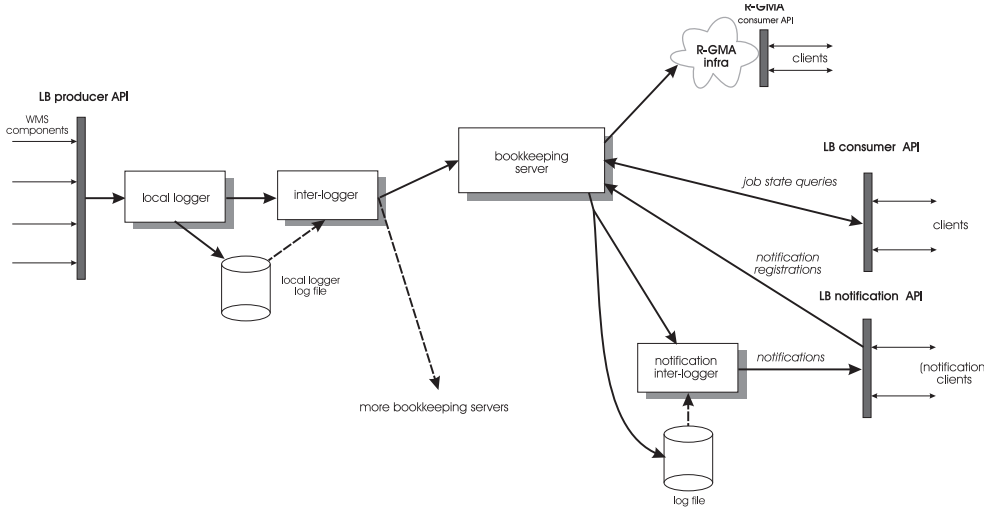

Abbildung 4.47: Architektur des Logging and Bookkeeping Service [Kren 05]

Der L&B Server speichert die Events und bietet einen High-Level-View auf die Prozesse und deren Zustand (z.B. *Submitted*, *Running*, *Done*), aber auch eine Schnittstelle, um gezielt nach Events zu suchen.

Abfrageschnittstelle beim L&B Server

Es gibt auch eine Notification-API, über die sich ein Nutzer für bestimmte Events registrieren kann und dann beim Eintritt eines entsprechenden Events benachrichtigt wird. Der L&B Service besitzt eine Authentisierungs- und Autorisierungskomponente, um den Zugriff auf die Events zu kontrollieren. Standardmäßig darf nur der Eigentümer des Jobs Informationen über seinen Job einsehen. Er kann allerdings über die Spezifikation von Access Control Lists (ACLs) auch Dritten (z.B. VO-Mitgliedern) Zugriff auf die Daten ermöglichen.

Notifikationsmechanismus für Events

Die Events, die an den L&B Service weitergeleitet werden oder für die sich ein Benutzer beim Notification Dienst registrieren kann, sind fest vordefiniert. Eine Ausweitung auf andere Informationsquellen oder Event-Arten ist nicht ohne weiteres möglich.

Events fest vordefiniert; Erweiterung schwierig

Probleme des L&B

Im User Guide für den L&B Service [Kren 05] wird auf einige mögliche Schwächen und Probleme hingewiesen: Da Abfragen für den L&B sehr komplex sein oder sehr große Datenmengen zurückliefern können, werden die Nutzer darauf hingewiesen, dass es durch die Abfrage zu einer Server-Überlastung kommen kann. Desweiteren sollten die verschiedenen Ressourcen zeitlich synchronisiert werden. Andernfalls besteht die Gefahr, dass die Reihenfolge der Events inkorrekt dargestellt wird.

Globus Logging und GRAM Audit Logging

GT4 verwendet `log4j`

Die Globus Middleware GT4 verwendet für das Logging ein ursprünglich für Java entwickeltes Logging- und Debugging-Paket (`log4j`, [log4j]), das im Rahmen eines Projektes für den Apache Webserver entstanden ist.

Mit Hilfe von `log4j` ist es möglich unterschiedliche Logging Ziele anzugeben. Es ist relativ einfach möglich neue Logging-Events zu definieren und in den eigenen Quell-Code zu integrieren (vgl. z.B. [Stel 05]). Das Logging kennt sechs verschiedene Log-Level: Trace, Debug, Info, Warning, Error und Fatal. Das Logging über `log4j` ist sehr flexibel anpassbar. Dies liegt auch an der modularen Architektur des Paketes; `log4j` besteht aus den drei Hauptkomponenten: Logger, Appernders und Layouts [Ceki 02].

Flexibel anpassbar

Logger erzeut Log-Nachrichten

Ein **Logger** ist ein einem Software-Objekt zugeordnetes Logging-Objekt, d.h. der Logger erzeugt die eigentliche Logging Nachricht. Die Logger Objekte werden über eine Objekthierarchie an die Objekte gebunden und die entsprechenden Vererbungsbeziehungen auf der Objektebene gelten auch für die Logger und die entsprechenden Log-Level. Das heißt, falls einem Logger nicht explizit ein Log-Level zugeordnet wird, erbt er den Log-Level von dem nächsten Vorfahren in der Hierarchie, für den ein Log-Level gesetzt ist.

Appender implementiert Log Ausgabe

Entferntes Logging möglich

Ein **Appender** definiert und implementiert die Ausgabe für ein bestimmtes Ziel. Es sind bereits zahlreiche Appender vorhanden, z.B. für die Ausgabe auf einer Konsole, in eine Datei, auf der GUI, für die entfernte Protokollierung über Sockets, Java Messaging Service (JMS), über Windows Event Log oder über den syslog Mechanismus von Unix. Es ist möglich Logging Events gleichzeitig an mehrere Appender zu schicken. Auch asynchrones Logging ist möglich.

Wie die Log-Events in der Ausgabe dann formatiert werden, ist Aufgabe der **Layouts**-Komponente. Das heisst, das Layout der Logging-Nachricht ist nicht statisch, sondern kann an individuelle Bedürfnisse angepasst werden.

`log4j` ist in die Middleware integriert und das Logging lässt sich über Properties-Dateien aktivieren und konfigurieren [Glob c]. Ursprünglich für Debugging-Zwecke entwickelt, gibt es bereits vordefinierte Verfahren um SOAP-Nachrichten und die HTTP-Nachrichten auf Client-Seite mitzuprotokollieren. Auch für das Globus Autorisierungssystem gibt es solche vordefinierten Debugging-Mechanismen. Abbildung 4.48 zeigt, wie diese Verfahren in der Konfigurationsdatei von Globus aktiviert werden.

```
# Enable SOAP message logging
log4j.category.org.globus.wsrf.handlers.MessageLoggingHandler=DEBUG

# Enable on-the-wire logging
log4j.logger.httpclient.wire=DEBUG

# Log every authorization xsdecision the notification consumer makes.
log4j.category.org.globus.wsrf.impl.security.authorization.AuthorizationHandler=WARN

# Enable to see raw wire messages as written or read by the client
log4j.logger.httpclient.wire.content=DEBUG
```

Abbildung 4.48: Aktivierung von vordefinierten Logging-Verfahren in Globus

GRAM Audit Logging zur Job-Protokollierung

Neben dem Globus Logging gibt es seit Version 4.0.5 von Globus das so genannte **GRAM Audit Logging** [Glob f]. Damit ist es möglich, mittelbar über den Grid Ressource Allocation Manager (GRAM), Job-Informationen aus lokalen Batch-Systemen wie z.B. PBS (Portable Batch System), LSF oder Condor zu erhalten. GRAM Audit Loging wurde für Accounting und Auditing entwickelt. Im Lebenszyklus eines Jobs werden drei Audit bzw. Accounting Records erzeugt: bei der Instantiierung des Jobs, bei der Submission auf eine lokale Ressource und beim Job-Ende bzw. -Abbruch. Jeder Record enthält die in [Glob f] angegebenen Job-Informationen. Die Records werden in eine Datenbank geschrieben und können dann über Abfragen auf dieser Datenbank (z.B. mittels OGSA-DAI [Glob d]) abgefragt werden.

UNCORE Logging

Logging lokal

Bei UNICORE existieren sowohl auf Seite des Client als auch auf Server-Seite Mechanismen zur Protokollierung. Das Client-seitige Logging ist als Debugging-Mechanismus für die Entwickler gedacht und schreibt Debugging-Nachrichten in eine lokale Datei (`clientlog.txt` bzw. `clientlog.xml`).

Nicht flexibel anpassbar; kein entferntes Logging

Auf Server-Seite kann für jede Server-Komponenten, d.h. für Gateway, NJS und TSI (vgl. auch Abschnitt 4.1.7) das Logging gesondert aktiviert werden. In UNICORE sind sechs Log-Level definiert: **S**evere, **W**arning, **I**nformation, **C**onfiguration, **T**alk und **D**ebug [vdB 06]. In den Konfigurationsdateien von UNICORE kann man für jede Komponenten den Log-Level konfigurieren. Es werden alle Nachrichten des entsprechenden Levels und der höherliegenden Levels protokolliert. Der Default Wert ist C d.h. es werden auch die Nachrichten der Levels I, W und S protokolliert. Die Log-Nachrichten und -Events sind fest vordefiniert und nicht erweiterbar oder anpassbar. Die Log-Nachrichten werden in lokale Dateien geschrieben. Ein entferntes Logging oder eine Weiterleitung an einen zentralen Log-Server ist nicht vorgesehen.

4.7.2 Bewertung Logging

Bei den Logging-Mechanismen verfolgt jede Middleware ihren eigenen Ansatz. Im folgenden werden die Kriterien des Kriterienkataloges auf die ver-

schiedenen Logging-Mechanismen angewendet. Soweit nichts angegeben ist, gilt die Bewertung für alle vier vorgestellten Mechanismen (L&B, `log4j`, GRAM Audit Logging und UNICORE Logging). Falls sich die Bewertungen unterscheiden, wird der entsprechende Mechanismus gesondert aufgeführt.

- **Middleware-Integration:**
  Für jede Middleware ist der Logging-Mechanismus Teil der Middleware und als solches in der jeweiligen Middleware mit 3 zu bewerten. Für fremde Middlewares ist die Integration nicht möglich, d.h. hier ist die Integration in fremde Middlewares jeweils mit 0 zu bewerten. Daraus ergibt sich eine Gesamtbewertung von 1 (Bewertung: 1).

- **Ressourcen-Integration:**
  Die Logging-Mechanismen sind Teil der Middleware werden aber von den Ressourcen nicht direkt unterstützt (Bewertung: 0).

- **Erweiterbarkeit:**
  Die vorgestellten Logging-Verfahren legen ihren Fokus voll auf Ressourcen und Job-Aktivitäten. Für diese Zwecke sind Logging-Events definiert. Ein echtes Sicherheits-Logging oder eine Protokollierung bei den Sicherheitsmechanismen ist kaum, allenfalls in Form von Debugging-Mechanismen bei Globus, vorgesehen. Eine Erweiterung des Logging um neue Logging-Events ist für UNICORE nur über eine Änderung der offiziellen Quellen des Programm-Code möglich. Auch beim L&B Service von gLite müsste die entsprechende API geändert werden. Beide Mechanismen sind deshalb mit 1 zu bewerten. Lediglich in Globus gibt es vordefinierte Schnittstellen um eigene Logging-Events im eigenen Quellcode zu integrieren. Das Logging von Globus ist deshalb mit 3 zu bewerten.

- **Middleware übergreifende Interoperabilität:**
  Die Logging Mechanismen wurden individuell für die jeweilige Middleware entwickelt. Ein wohldefiniertes Logging Interface, das unabhängig von der Middleware funktionieren würde, ist nicht vorgesehen. Jede Middleware verwendet hier eigene Ansätze. Eine Middleware übergreifende Interoperabilität ist weder gegeben noch vorgesehen (Bewertung 0).

- **Interoperabilität auf Policy-Ebene:**
  Bei keinem der Mechanismen sind Policy-Konzepte vorgesehen (Bewertung: 0).

- **Interoperabilität beim ID-Management:**
  Die Logging-Mechanismen sind unabhängig vom ID-Management. Insofern ist dieses Kriterium nicht anwendbar.

- **Trust Management; Formalisierung:**
  Ein formalisiertes Trust Management existiert nicht. UNICORE protokolliert alle Nachrichten lokal; insofern ist aus Sicht des Server-Betreibers kein Vertrauensverhältnis erforderlich. Allenfalls bei gLite

und Globus, die auch entferntes Logging ermöglichen, wird ein implizites Vertrauensverhältnis gegenüber der Nachrichtensenke angenommen. Mechanismen zur Bewertung der Vertrauenswürdigkeit der Logging-Daten sind nicht vorgesehen (Bewertung 0).

- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Es gibt keine Verfahren, um Vertrauen zu berechnen oder abzuleiten (Bewertung: 0).

- **Trust Management; Granularität:**
  Nachdem weder ein Trust Management noch Verfahren zur Ableitung von Vertrauenswerten vorhanden sind, kann dieses Kriterium nicht angewendet werden.

- **Delegationskonzepte:**
  Die Kriterien zur Bewertung der Delegationskonzepte sind für das Logging nicht anwendbar. Die Aufgabe des Logging muss Ressourcen-nah erfolgen eine Delegation und damit auch eine Bewertung sind nicht möglich.

- **Mapping:**
  Da es keine Mechanismen gibt, um unterschiedliche Policies zu spezifizieren, kann dieses Kriterium nicht angewendet werden.

- **Skalierbarkeit:**
  Das lokale Logging skaliert mit Hard- und Software. Auch eine organisatorische oder räumliche Ausdehnung des Grid hat in diesem Fall keine Auswirkung auf die Skalierbarkeit, d.h. hier ist eine Bewertung von 3 zu vergeben. Im Falle von entferntem Logging, wie in Globus möglich und beim L&B Service explizit vorgesehen, hängt die Leistungsfähigkeit sehr stark von der organisatorischen Größe des Grid ab. Auch die Hinzunahme neuer Hard- und Software-Komponenten kann zu überproportionalem Aufwand führen. In [Kren 05] wird bereits auf Skalierungsprobleme beim L&B Service hingewiesen. Das Logging-System skaliert in diesem Fall nur in einer Dimension und ist mit 1 zu bewerten. Als Gesamtbewertung wird der Mittelwert dieser beiden Werte verwendet. (Bewertung: 2).

- **Flexibilität; Entitätenvielfalt:**
  Das Logging muss nicht alle Entitäten abbilden können, daher ist dieses Kriterium nicht anwendbar.

- **Flexibilität; Organisationsflexibilität:**
  Die präsentierten Logging-Mechanismen sind primär für das lokale Logging konzipiert. Durch die Weiterleitung an entfernte oder zentrale Logging-Server ist eine zentrale Auswertung möglich. D.h. die Mechanismen lassen sich — nach Anpassungen — für zwei Dimensionen nutzen (Bewertung: 2).

- **Administrierbarkeit:**
  Der Aufwand für den Betrieb einer Logging Infrastruktur ist gering (Bewertung: 2).

- **Sicherheit; Sicherheitsniveau:**
  Bei keinem der präsentierten Mechanismen werden die Logging Nachrichten, z.B. durch digitale Signatur, gesichert. Für einen Angreifer, der in der Lage ist ein lokales System zu kompromittieren, ist es sehr einfach die lokalen Log-Einträge zu manipulieren oder eigene Log-Einträge zu erzeugen. Um die nachträgliche Manipulation von Logs zu verhindern, bzw. zu erschweren, werden entfernte Log-Server verwendet. Dies ist bei gLite und Globus möglich. Ein Angreifer, der ein lokale System kompromittiert, kann aber durch die Manipulation und Weiterleitung der lokalen Logs seine Spuren verwischen. Das heißt, die Eintrittswahrscheinlichkeit für eine Manipulation der Logs ist mit hoch und die dadurch entstehenden Schadenshöhe mindestens mit Mittel anzunehmen. Damit ist das resultierende Sicherheitsniveau niedrig (Bewertung: 0).

- **Sicherheit; Zusicherung, QoP:**
  Es sind keine Verfahren vorgesehen, um Aussagen über das Sicherheitsniveau des Logging-Dienstes mitteilen zu können (Bewertung: 0).

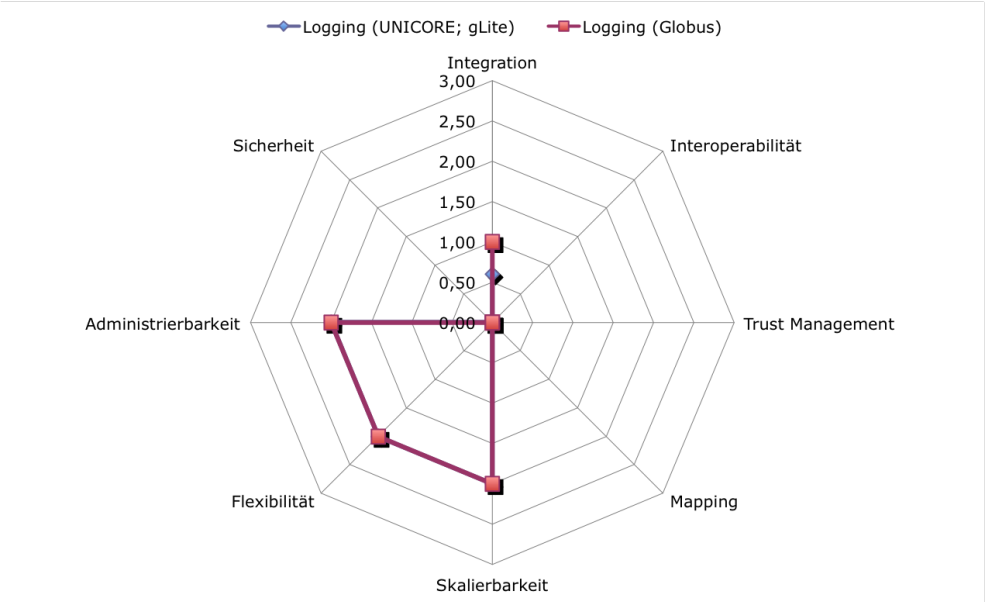

Abbildung 4.49: Berwertung Logging

Obwohl die präsentierten Verfahren sehr unterschiedlich konzipiert und genutzt werden, zeigt sich im Vergleich der Bewertungszahlen im Netzdiagramm in Abbildung 4.49, dass die Bewertung sehr ähnlich ausfällt und es eine hohe Überdeckung bei Stärken und Schwächen gibt.

4.7.3 Sicherheitsalarme

Ein Grundprinzip für jede Art von Security Information Management (SIM) sind Sicherheitsalarme und entsprechende Notifikationsmechanismen. Der Sicherheitsadministrator definiert Regeln, Zustandsänderungen oder Schwellwerte, die beim Eintritt bzw. Über- oder Unterschreitung einen Alarm auslösen und eine Notifikation generieren, die dann an eine konfigurierbare Informationssenke, z.B. eine SIM-Konsole oder einen anderen Alarmierungsmechanismus, übermittelt wird. Für Sicherheitsalarme im Grid bedarf es deshalb mindestens der beiden folgenden Voraussetzungen: Es existiert ein Notifikationsmechanismus und es besteht die Möglichkeit der freien Definition von Alarm-Regeln.

Notifikationsmechanismus vorhanden

Definition von Alarmregeln

Beim L&B Service von gLite gibt es eine Notification API, über die der Nutzer sich für bestimmte Events registrieren kann. Allerdings sind die Events hier vordefiniert und umfassen nur Informationen über die Job-Verarbeitung. Eine Definition eigener Alarmierungsregeln ist hier nicht möglich.

Vordefinierte Regeln bei gLite

Globus unterstützt seit Version 4 den Web-Service Notification Standard [OASI b, GHM 06, ChLi 06, VGN 06]. Damit können interessierende Eigenschaften, die als WS-Topics [VGN 06] bezeichnet werden, definiert und von einem Web-Service veröffentlicht werden. WS-Notification geht von einem Producer Consumer Modell aus. Das heißt, ein Web-Service veröffentlicht als Producer WS-Topics und ein Notification-Consumer kann sich mit Hilfe einer subscribe-Operation für Topics registrieren. Wenn sich dann der Zustand des Topics verändert, wird vom Producer eine Notifikation erzeugt und an alle registrierten Consumer übermittelt. Dies geschieht über den Aufruf einer notify-Operation beim Consumer. GT4 implementiert zwar die WS-Notification Spezifikationen nicht vollständig, aber das oben beschriebene Producer Consumer Modell ist implementiert und die Definition eigener Topics ist möglich. Die Eigenschaften, die an ein Topic geknüpft sind, müssen in der Implementierung des Web-Service integriert und über eine Subscribe Methode im der WSDL-Datei des Web-Service veröffentlicht werden [Soto 05].

Globus nutzt WS-Notification

Definition eigener Regeln möglich

Die Notifikationsmechanismen sind natürlich nur für Web-Services verwendbar. In Globus sind damit die Basismechanismen zwar vorhanden, sie adressieren aber primär neu zu entwickelnde Web-Services, die auf der Middleware aufsetzen. Vordefinierte Topics für Sicherheitsalarme aus der Middleware heraus existieren jedoch derzeit nicht. Für die Definition und Implementierung solcher Topics in der Middleware wäre eine internationale Abstimmung in der Globus Entwickler-Community und eine Integration in die offiziellen Quellen erforderlich. Auch die Diskussion, welche Grid-spezifischen Angriffe es gibt und welche Alarme definiert werden müssen, steht noch am Anfang.

Es gibt derzeit also kein sicherheitsrelevantes Alarming, weder in Globus noch in gLite oder UNICORE. Damit können die Mechanismen auch nicht bewertet werden.

Derzeit kein sicherheitsrelevantes Alarming

4.7.4 Sicherheitsaudit

Audit überprüft Sicherheitsrichtlinien und -prozesse

Unter einem Sicherheitsaudit wird der Prozess verstanden, mit dem überprüft, dokumentiert und sichergestellt wird, ob und wie effektiv ein Sicherheitssystem in einem bestimmten Bereich (z.B. einer Organisation) umgesetzt wird. Als Basis für ein Sicherheitsaudit müssen Sicherheitsrichtlinien und -prozesse definiert sein. Das Audit überprüft dann die Umsetzung der Prozesse und die Einhaltung der Sicherheitsrichtlinien. Ein Ziel eines Audit Prozesses kann auch die Dokumentation des Audit-Ergebnisses in Form eines Audit-Zertifikates sein. Man unterscheidet das interne und externe Audit. Beim internen Audit wird der Audit Prozess eigenverantwortlich, z.B. innerhalb der Organisation, durchgeführt. Beim externen Audit führt die Prüfung ein unabhängiger Dritter (Auditor) durch.

IT-Grundschutz definiert Richtlinien

Vom Bundesamt für Sicherheit in der Informationstechnik (BSI) werden im Rahmen des IT-Grundschutzes [Bund] Standards [Bund 05c, Bund 05a, Bund 05d, Bund 08a] und Grundschutz-Kataloge (früher als Grundschutz-Handbuch bezeichnet) [Bund 07] zur IT-Sicherheit herausgegeben. Der IT-Grundschutz befasst sich mit Standardsicherheitsmaßnahmen und einem Vorgehensmodell, um zu einem sicheren IT-System zu kommen.

ISO/IEC 27001 als internationaler Sicherheitstandard

Auditverfahren definiert

Im internationalen Bereich legt die Norm ISO/IEC 27001 [ISO 270001] die Anforderungen für die Realisierung, den Betrieb, die Überwachung und die kontinuierliche Verbesserung eines Informationsicherheitssystems fest. Die Einhaltung des IT-Grundschutzes bzw. des Standards ISO/IEC 270001 und deren sachgemäße Umsetzung lassen sich im Rahmen eines externen Audit-Verfahren, geregelt in entsprechenden Prüfschemata [Bund 08b], zertifizieren.

Sicherheitsaudit für Spezialbereiche der IT

Neben diesen sehr umfassenden Standards, die sich ganz allgemein mit IT-Systemen befassen, gibt es auch Sicherheitsaudits für ganz spezielle Bereiche der Informationsverarbeitung. So hat beispielsweise die Kreditkartenindustrie (Payment Card Industry, PCI) einen Sicherheitstandard (Data Security Standard, DSS) [Paym 06a] herausgegeben, der alleinig dem Schutz von Karten- und Kundendaten (d.h. Name des Kunden, Servicecode und Ablaufdatum der Kreditkarte) bei der Verarbeitung von Kreditkarten gilt. Der DSS beschäftigt sich mit dem Sicherheitsmanagement, Richtlinien und Prozeduren, Netzarchitekturen, Software Design und anderen Schutzmechanismen zum Schutz dieser Daten. Die Richtlinien, die der Standard festlegt, lassen sich in einem externen Audit-Verfahren überprüfen. Dazu wurde vom PCI ein Security Audit Prozess spezifiziert, der ein umfassendes und gut dokumentiertes Testverfahren und einen Fragenkatalog beinhaltet [Paym 06b] und auch Richtlinien für Sicherheit-Scans [Paym 06c] definiert.

Im Grid keine Richtlinien und Prozesse festgelegt

Im Grid-Umfeld existieren derzeit weder Beschreibungen oder Festlegungen, die vergleichbar mit ISO/IEC 270001 oder IT-Grundschutz, Sicherheitsverfahren oder Prozesse festlegen, noch interne oder externe Auditierungs- oder Zertifizierungsrichtlinien. Eine zweifelsfreie, normierte Überprüfung der implementierten Sicherheitsprozesse, -dienste oder -mechanismen ist nicht möglich. Dies gilt sowohl für die interne Überprüfung innerhalb einer Or-

Kein Audit definiert

ganisation, als auch für eine externes Audit, z.B. der VO-Mitglieder oder der Ressourcen-Provider durch die VO.

4.8 Basisdienste

Die Basisdienste Trust- und Policy Management sind Sicherheitsdienste, von denen viele andere Sicherheitsdienste abhängen. Für die ganzen AAI, VO- sowie Zugriffskontrollmechanismen u.a. sind die Etablierung von Vertrauensbeziehungen, deren Überprüfung und Anpassung essentiell für die Sicherheit und Verlässlichkeit der davon abhängigen Sicherheitsmechanismen. Ein ähnlicher Zusammenhang existiert zwischen dem Policy Management und davon abhängigen Sicherheitsmechanismen (vgl. auch Abschnitt 2.3). Die Nutzung, Konfiguration und das erreichbare Sicherheitsniveau von Sicherheitsmechanismen lässt sich über Regeln, d.h. Policies, festlegen. Für unterschiedliche Entitäten im Grid können diese Regeln natürlich völlig unterschiedlich aussehen. Um trotzdem im Grid zusammenarbeiten zu können, ist ein Abgleich oder ein Einigungsprozess (Policy-Mapping) erforderlich.

In diesem Abschnitt werden deshalb bestehende Verfahren zum Trust- (Abschnitt 4.8.3) und zum Policy Management (Abschnitt 4.8.1) behandelt.

4.8.1 Policy Management

Das Policy Management muss sich mit der Spezifikation, der Administration und der Aktualisierung sowie mit Fragen der Interoperabilität von Policies der verschiedenen Entitäten im Grid befassen. Es sind Verfahren festzulegen, wie die Policies der verschiedenen Partner kommuniziert und ggf. delegiert werden können und wie Policy Konflikte erkannt und aufgelöst werden können (Policy Mapping).

In existierenden Grids gibt es zwei Bereiche, in denen schriftlich fixierte Policies existieren:

1. CA Policies
2. Acceptable Use Policies (AUP)

Daneben gibt es technische Policies, die als Teil von Algorithmen in Sicherheitsmechanismen selbst Verwendung finden, um z.B. zu verwendende Verfahren oder das Schlüsselmaterial festzulegen (vgl. z.B. TLS in Abschnitt 4.4.3).

CA-Policies

Bei den CA Policies (vgl. auch Abschnitt 4.2.2) existiert unter dem Dach der International Grid Trust Federation (IGTF) eine weltweite Hierarchie von Po-

CA Hierarchie unterhalb der IGTF

licy Management Authorities (PMA) und CAs. Die IGTF zertifiziert die drei regionalen PMAs EUGridPMA, APGrid PMA und TAGPMA, die ihrerseits wiederum lokale CAs wie z.B. die DFN-PKI [DFN-PKI] oder die GridKA-CA [GridKA-CA] auf die Einhaltung gewisser Richtlinien verpflichten. Die IGTF hat sich und ihren Mitgliedern eine Charta gegeben [Groe 05], die als IGTF Policy verstanden werden kann. Dieses Dokument legt die Mitgliedschaft fest. Es bestimmt die allgemeine Architektur der CA-Föderation, d.h. die Mitglieds-PMAs zertifizieren nur CAs, die selbst wieder End Entity Certificates (EECs) oder ID-Zertifikate ausstellen. Die PMAs stellen selbst keine solche Zertifikate aus.

IGTF Charta als Policy für Mitglieder

Die Charta legt die grundsätzlichen betrieblichen Anforderungen fest, z.B. dass jede PMA und CA einen eineindeutigen Namen haben muss, welche Kommunikationsmechanismen, welche Kontaktmöglichkeiten existieren müssen und welche Informationen zur Verfügung zu stellen sind. Die Policy bestimmt, dass jede akkreditierte CA seine Sicherheitsmechanismen dokumentieren muss und eine Zertifizierungs-Policy (**Certification Policy (CP)**) sowie Policies zu den Betriebsregeln der CA (ein **Certification Practice Statement (CPS)**) aufzustellen hat [CFS+ 03]. Außerdem werden Vertraulichkeitsbestimmungen und allgemeine Mechanismen zum Desaster Recovery festgelegt. Die letzten Abschnitte befassen sich mit der Verwaltung der Föderation.

Certification Policy (CP) und Certification Practice Statement (CPS)

Die IGTF bzw. die Mitglieds-PMAs verwalten eine Menge von Authentication Profiles (APs), die technische Anforderungen und technische Policies für ID-Zertifikate und CAs festlegen. Die Liste der unterstützten APs wird von der IGTF geführt. Wobei die Verwaltung der APs und deren kontinuierliche Weiterentwicklung Aufgabe einer beauftragten PMA ist.

Akkreditierungsrichtlinien für CAs

Derzeit verwaltet die EUGrid PMA die Akkreditierungsrichtlinien [EU P 04a], den Entwurf des High Level CA Profile [JeGr 06] sowie das Authentication Profile [Groe 06b, Groe 06a]. TAGPMA verwaltet das Short Lived Certificate Service Profile [Geno 05] und den Entwurf des Member Integred Credential Service [Murr 07, Geno 06]. APGrid PMA verwaltet die Regeln für Experimental CAs [Asia 05].

Die Akkreditierungsrichtlinien [EU P 04a] legen fest, wie die CAs von PMAs zertifiziert werden können. Dazu ist ein Registrierungs- und Review-Prozess erforderlich. Der Entwurf des High Level CA Profiles [JeGr 06] hat das Ziel die minimalen Anforderungen für PMAs festzulegen, die selbst lediglich CAs zertifizieren.

Authentication Profile legt CA Anforderungen fest

Die wichtigste Policy ist das Authentication Profile [Groe 06b, Groe 06a], welches die minimalen Anforderungen an CAs festlegt, die längerfristig gültige X.509 Zertifikate (d.h. mehr als 1 Million Sekunden Gültigkeitsdauer) ausstellen (z.B. Nutzer- oder Server-Zertifikate). Dementsprechend beschreibt das Short Lived Certificate Profile [Geno 05] die Anforderungen an einen Short Lived Credential Service (SLCS), der kurzfristig gültige X.509 Zertifikate (d.h. bis zu 1 Million Sekunden Gültigkeitsdauer) ausstellt. Die Policy für Member Integrated Credential Services [Murr 07, Geno 06] fasst die Richtli-

nien für sogenannte **Member Integrated Credential Services (MICS)** zusammen. Ein MICS ist ein Dienst, der X.509 Zertifikate auf Basis von existierenden Identity Managementsystemen ausstellt, die von der jeweiligen Organisation bereits betrieben werden (Legacy Systeme). Damit wird es ermöglicht Zertifikate für Entitäten auszustellen, die sich über ein lokales ID-Managementsystem authentisiert haben. Die Regeln dieses Profiles sind natürlich von den lokalen ID-Managementsystemen einzuhalten.

Die Policy für Experimental CAs [Asia 05] legt die Richtlinien für Online CAs fest. Eine **Online CAs** dient zur Erstellung dynamischer Zertifikate, wie z.B. im GridShib (vgl. Abschnitt 4.3.3) genutzt.

Zertifizierungs-Policy (CP) und Betriebsregeln (CPS) der CA

Für die Akkreditierung nach [EU P 04a] muss jede CA eine CP sowie eine CPS festlegen und bei der Akkreditierung einem Review unterziehen. Der DFN hat bspw. beide Policies in einem Dokument zusammengefasst [DFN- 06], das festlegt, wie die Entitäten zu identifizieren und zu authentisieren sind, wie der Lebenszyklus der Zertifikate definiert ist, wie die Sicherheit der CA Infrastruktur (inklusive Facility Management und operative Kontrolle) realisiert und wie die Zertifikate veröffentlicht und zugänglich gemacht werden.

Acceptable Use Policies

Acceptable Use Policies verpflichten Grid Nutzer

Auf die Grid **Acceptable Use Policies (AUP)** haben die VOs ihre Nutzer zu verpflichten. Von jedem Grid Nutzer soll angenommen werden können, dass er die AUP kennt und akzeptiert.

Die erste Grid AUP hat das EGEE aufgestellt [Join 06a]. Diese wurde von DEISA unverändert übernommen und um die zwei Punkte 8. und 9. ergänzt [RSW 07]. Die DEISA AUP umfasst damit folgende Regeln:

EGGE; DEISA AUP

1. *You shall only use the GRID to perform work, or transmit or store data consistent with the stated goals and policies of the VO of which you are a member and in compliance with these conditions of use.*
2. *You shall not use the GRID for any unlawful purpose and not (attempt to) breach or circumvent any GRID administrative or security controls. You shall respect copyright and confidentiality agreements and protect your GRID credentials (e.g. private keys, passwords), sensitive data and files.*
3. *You shall immediately report any known or suspected security breach or misuse of the GRID or GRID credentials to the incident reporting locations specified by the VO and to the relevant credential issuing authorities.*
4. *Use of the GRID is at your own risk. There is no guarantee that the GRID will be available at any time or that it will suit any purpose.*
5. *Logged information, including information provided by you for registration purposes, shall be used for administrative, operational, accounting,*

monitoring and security purposes only. This information may be disclosed, via secured mechanisms, only for the same purposes and only as far as necessary to other organizations cooperating with DEISA. Although efforts are made to maintain confidentiality, no guarantees are given.

6. *The Resource Providers, the VOs and the GRID operators are entitled to regulate and terminate access for administrative, operational and security purposes and you shall immediately comply with their instructions.*

7. *You are liable for the consequences of any violation by you of these conditions of use.*

8. *You must notify your VO administrator about any changes in your contact information.*

9. *In case that some or parts of these conditions are invalid or impracticable or if they become invalid or impracticable after the contract is signed, the remainder of the conditions is not affected. Only affected conditions may be substituted by conditions which are most close to the intended ones. This includes also incomplete conditions.*

Unterscheidung zwischen User- und Ressource-AUP

Auch im D-Grid gibt es einen AUP Vorschlag [D-Gr 07]. Im D-Grid plant man zwischen einer User- sowie einer Ressourcen-AUP zu unterscheiden [BDE+ 07]. Die Ressourcen-AUP soll die Richtlinien des jeweiligen Ressourcen Providers enthalten, welche die Randbedingungen für die Ressourcen-Nutzung umfasst. Der Nutzer soll dann auf die jeweilige Ressourcen-AUP verpflichtet werden. Allerdings existiert derzeit noch keine Muster Ressourcen-AUP.

4.8.2 Bewertung Policy Management

zentralistische Struktur

Beide vorgestellten Policy-Klassen (IGTF Policies und AUPs) sind sehr stark zentralistisch strukturiert und den Entities bleibt nur die Möglichkeit diese vollständig oder überhaupt nicht zu akzeptieren. Eine Berücksichtigung lokaler Policies ist weder vorgesehen noch möglich. Lediglich die Charta der IGTF sieht in Abschnitt 12.4 einen Passus zur Konfliktlösung vor [Groe 05]. Wobei dort nur festgelegt wird, dass Dispute per Email beim IGTF oder einer anderen Mitglieds-PMA einzureichen sind. Die Konflikte werden dann von der betroffenen PMA gelöst.

strategische Policies erfordern aufwändige Abstimmung

keine Formalisierung

Diese langfristigen angelegten strategischen Policies erfordern eine aufwändige internationale Abstimmung, die für die elementarsten Fragen der Identifikation, Authentisierung, Ausstellung von Zertifikaten und die Nutzungsrichtlinien auch durchgeführt werden. Die Untersuchung der verschiedenen Regeln im Umfeld der IGTF, die ja immer noch nicht abgeschlossen sind, verdeutlichen, welch aufwändige Abstimmungsprozesse sich dahinter verbergen. Ein weiterer Aspekt der existierenden Policies ist die mangelnde Formalisierung. Alle Dokumente sind in Prosa.

Bei den D-Grid AUPs zeigt sich bereits der Versuch den einzelnen Ressourcen-Providern durch individuelle Ressourcen-AUPs die Möglichkeit zu geben, die Nutzer auf eigene spezielle Policies bezüglich der Nutzung der eigenen Ressourcen zu verpflichten. Dies hat natürlich den Preis eines deutlich erhöhten administrativen Aufwands, da es bisher kaum Mechanismen gibt, um diese Verpflichtung des individuellen Nutzers technisch unterstützt zu erhalten. Auch bei den „normalen“ AUPs ist es derzeit i.d.R. so, dass Unterschriften auf Dokumenten verlangt werden und dass die Heimatdomäne oder das Heimat-Rechenzentrum den Nutzer bei der Unterschrift der eigenen Nutzungsrichtlinien auch gleich auf diverse Grid-AUPs verpflichtet. Dieser Ansatz skaliert natürlich bei sich dynamisch ändernden VOs und sich ändernden AUPs nicht.

Policy Raum zwischen strategischen und technischen Policies bleibt leer

Bei den sehr technischen Policies, die bspw. für die Konfiguration von Algorithmen verwendet werden und schon im Algorithmus spezifiziert werden (z.B. zur Festlegung der kryptographischen Algorithmen bei TLS; vgl. Abschnitt 4.4) ist die Verwendung, die Interoperabilität und Möglichkeit der Delegation und Konfliktauflösung bereits im Verfahren festgelegt. Das Problem ist der Policy-Raum zwischen den beiden Extremen der strategischen und sehr technischen Policies. In diesem Bereich existieren i.d.R. keine Policies. Es gibt informelle Vereinbarungen, aber kein wirkliches Policy Management.

Obwohl dies ein Basisdienst für fast alle anderen Sicherheitsdienste darstellt, ist er nur sehr unzureichend behandelt. Dies zeigt sich auch bei der Anwendung des Kriterienkatalogs bei existierenden Sicherheitsmechanismen. Im Kriterienkatalog (vgl. Abschnitt 3.2) gibt es die drei Blattkriterien: Interoperabilität auf Policy-Ebene, Delegationsmechanismen für Policies und Mapping-Konzepte.

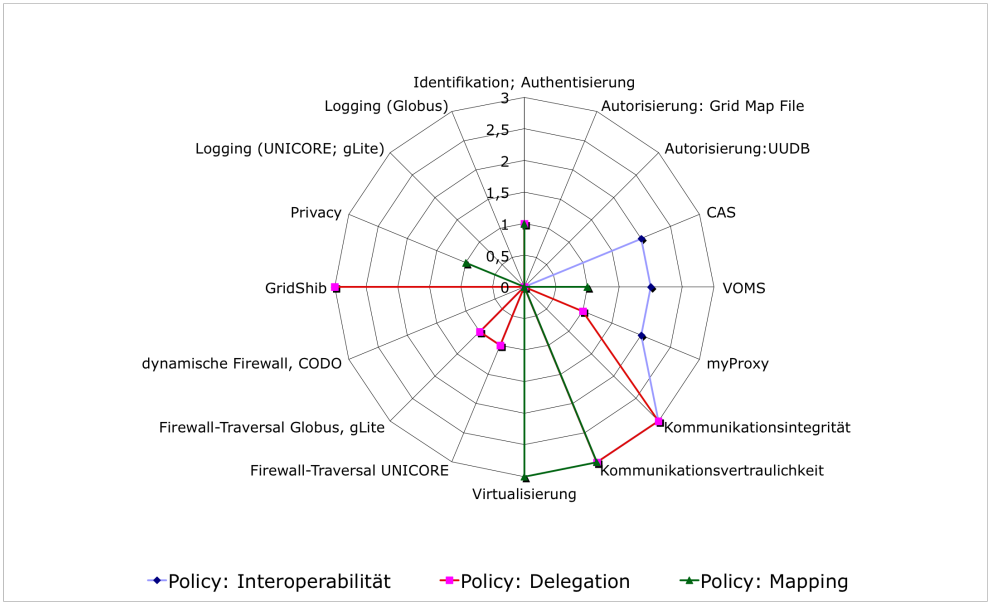

Abbildung 4.50: Vergleich der Berwertung von Blattkriterien zu Policies

Abbildung 4.50 fasst die Bewertungen dieser drei Blattkriterien für die betrachteten Sicherheitsmechanismen nochmal zusammen und hierbei wird ganz deutlich, dass bei vielen Mechanismen erhebliche Defizite beim Policy Management existieren.

4.8.3 Trust Management

individuelle Vertrauenswerte für alle Entitäten

Die Aufgabe des Trust Managements umfasst die Festlegung und Formalisierung von Trust Metriken, um Vertrauensbeziehungen überhaupt darstellen und differenzieren zu können. Für alle Grid Nutzer, Ressource-Provider, VOs und Organisationen muss es möglich sein individuell für alle Entitäten im Grid (d.h. Nutzer, Organisationen, Ressourcen, VOs, Gruppen, Rollen, Code/Implementierungen und ggf. Jobs) Vertrauenswerte festzulegen.

Das Trust Management muss technische Mechanismen bereitstellen, um die Vertrauenswerte validieren zu können. Einem Grid Nutzer muss es bspw. möglich sein den Vertrauenswert, den seine VO einem Ressource-Provider entgegenbringt, bestimmen zu können und mit seinem eigenen zu vergleichen.

dynamische Anpassung und Nutzung durch Sicherheitsmechanismen

Desweiteren sind Verfahren erforderlich, die eine dynamische Anpassung von Vertrauenswerten möglich machen (z.B. durch Reputationssysteme o.ä.). Die anderen Sicherheitsmechanismen, die Vertrauenswerte nutzen, sind so zu erweitern, dass die formalisierten Vertrauenswerte mit in die Entscheidungen beim Sicherheitsmechanismus einbezogen werden können. So sollte bspw. sowohl die Autorisierung als auch die Zugangskontrolle zu Ressourcen in Abhängigkeit vom Vertrauenswert des Nutzers erfolgen.

4.8.4 Bewertung Trust Management

Die Vertrauensmodelle in heutigen Grids erfüllen i.d.R. nicht einmal die grundlegende Forderung nach formalisierten Metriken und individueller Festlegung der Vertrauenswerte. Sowohl in LCG/EGEE, als auch in DEISA und D-Grid, gibt es nur implizite Vertrauensbeziehungen zu Trusted Third Parties (TTPs), CAs und VOs [Neil 04, EGEE b, NSSM 07, DEIS , BDE$^+$ 07]. Das Vertauensverhältnis zu CAs wird über den IGTF Verbund (vgl. Abschnitt 4.2.2 und 4.8.1) gebildet. Die Vertrauenswerte sind binär und nicht weiter abgestuft, d.h. jeder Nutzer, der ein Zertifikat vorweisen kann, das in seiner Zertifikatskette eine Signatur von einer IGTF zertifizierten PMA enthält, wird als Grid Nutzer des weltweiten Grid-Verbundes angesehen. Der CA wird vollstes Vertrauen im Hinblick auf die Zertifizierung von Nutzern entgegengebracht. Das Vertrauensverhältnis wird durch die Menge an gemeinsam anerkannten Policies, die zumindest schriftlich fixiert sind, begründet. Bei den VO-basierten Vertrauensverhältnissen ist es noch extremer. Hier existieren i.d.R. keine schriftlichen Regelwerke, welche die Mitgliedschaft in einer VO regeln. Eine Ausnahme ist hier allenfalls im Betriebskonzept für die D-Grid Infrastruktur zu sehen, das neben Regelungen über den Zugang zum

Heutige Vertrauensmodelle erfüllen Grundanforderungen nicht

Grid auch Anforderungen an virtuelle Organisationen und deren Mitglieder sowie Ressourcen-Provider festschreibt [BDE+ 07]. In dem Dokument werden Richtlinien für VOs festgelegt und die VOs verpflichtet sich eine Satzung zu geben, die diese Richtlinien berücksichtigt. Desweiteren wird festgelegt, welche Entität welche Use Policies zu erlassen hat.

Obwohl im Normalfall keine schriftlichen Regelwerke für die VOs existieren, vertrauen alle VO-Mitglieder bspw. einem Zertifikat, das die Mitgliedschaft in einer VO bestätigt. Auch hier gilt ein binäres Vertrauen, d.h. alle Mitglieder einer VO besitzen erst einmal denselben Vertrauenswert. Ein individueller Vertrauenswert pro Nutzer ist nicht vorgesehen.

Nachdem bereits die Basis für ein Trust Management fehlt, sind auch die anderen Aufgaben des Trust Management nicht umgesetzt. Im Rahmen der Bewertung der Sicherheitsmechanismen nach dem Kriterienkatalog aus Abschnitt 3 wurde auch die Unterstützung und Verwendung von Trust Management bei den einzelnen Sicherheitsmechanismen bewertet. Als Blattkriterien wurden die Formalisierung der Vertrauenswerte, die Berechnung bzw. Ableitung von Vertrauenswerten und die unterstützte Granularität bewertet.

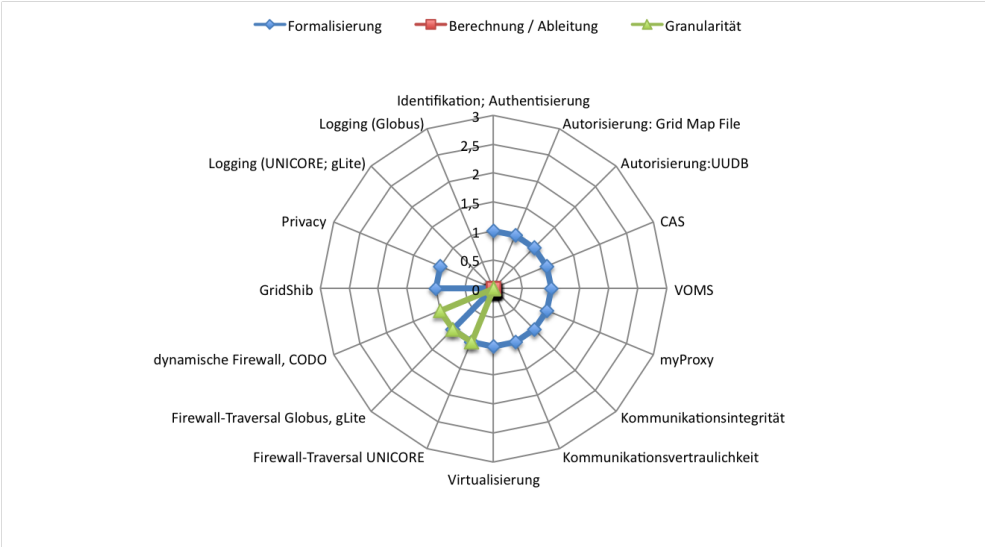

Abbildung 4.51: Vergleich der Berwertung von Blattkriterien zum Trust Management

Abbildung 4.51 fasst die Einzelbewertungen dieser Blattkriterien für die unterschiedlichen Mechanismen zusammen. Hier zeigt sich, dass die Bewertung maximal den Wert 1 erreicht, d.h. bei allen betrachteten Mechanismen liegt ein implizites Vertrauensmodell mit binären Vertrauenswerten zugrunde. Verfahren zur dynamischen Anpassung oder Berechnung von Vertrauenswerten existieren nicht und bei der Granularität werden allenfalls vereinzelt Vertrauensbeziehungen an Gruppen von Entitäten geknüpft. Die Vergabe individueller Trust Level ist nicht vorgesehen.

4.9 Gefahrenabwehr

Im Grid-Umfeld werden im Bereich der Gefahrenabwehr derzeit i.d.R. nur Firewall-Konzepte und -Problematiken betrachtet [NAG+ 06]. Eine Firewall soll jede Domäne so gut wie möglich von externen Netzen abschotten und nur berechtigten Nutzern von außen Zugang auf interne Ressourcen ermöglichen. Außerdem soll die Firewall sicherstellen, dass nur bestimmte Dienste nach außen zur Verfügung gestellt werden.

Konflikt zwischen Grid-Nutzbarkeit und Sicherheit

Eine Kooperation im Grid stellt in gewissen Aspekten ein Problem für die Konfiguration von Firewalls dar. Die Grid-Nutzer sollen einerseits das Grid möglichst nahtlos nutzen können. Am besten soll von den Nutzern der Einsatz einer Firewall überhaupt nicht bemerkt werden. Auf der anderen Seite kann ein Firewall-Administrator nicht alle Netze aller Partner so behandeln wie sein sicheres internes Netz. In diesem Spannungsfeld zwischen Nutzbarkeit und Benutzerfreundlichkeit einerseits und strengen Sicherheitsrichtlinien und starken Schutzmaßnahmen andererseits bewegen sich die Firewall-Mechanismen.

Fragen, die hierbei interessieren, sind die Anordnung und die Architektur der Firewall, die Firewall-Policies (insbesondere die zu öffnenden Ports) und die Platzierung von Grid-Komponenten in einer Demilitarisierten Zone (DMZ).

Die bestehenden und empfohlenen Verfahren bei den Middlewares unterscheiden sich erheblich. Deshalb werden UNICORE (vgl. Abschnitt 4.9.1) auf der einen und Globus sowie gLite (vgl. Abschnitt 4.9.3) auf der anderen Seite betrachtet und bewertet. Die Ansätze, die derzeit in der Praxis Anwendung finden, lassen sich als statische Firewall-Konfigurationen klassifizieren. Daneben gibt es in der aktuellen Forschung auch Überlegungen zu dynamischen Firewalls, die in Abschnitt 4.9.5 vorgestellt werden.

4.9.1 Firewall-Traversal: UNICORE

klare und strukturierte Firewall Konzepte

Bei der Entwicklung von UNICORE wurden systematisch Sicherheitsüberlegungen und auch Firewalls mit berücksichtigt. Die UNICORE-Architektur (vgl. Abbildung 4.9, S. 68) berücksichtigt Firewalls und gibt Empfehlungen für deren Platzierung [Erwi 00, Erwi 03].

Es gibt drei Möglichkeiten die Firewall(s) bzw. die Komponenten von UNICORE anzuordnen :

1. Das Gateway wird hinter einer Firewall platziert und ein SSL Tunnel von außen auf das Gateway wird erlaubt. Das Gateway entscheidet dann anhand des Endorsers, ob der Request zugelassen wird.

2. Das Gateway wird vor der Firewall betrieben und die SSL–Verbindung zwischen Gateway und NJS wird durch die Firewall getunnelt.

3. Das Gateway wird in einer demilitarisierten Zone (DMZ) zwischen zwei

Firewalls betrieben, die dann beide entsprechende SSL Verbindungen zulassen müssen.

Das besondere am UNICORE-Sicherheitskonzept im Hinblick auf Firewalls sind die klaren und sehr strikten Kommunikationsbeziehungen zwischen den UNICORE-Komponenten. Die Kommunikation „von außen“ mit dem Gateway erfolgt über einen vordefinierten Port. Das Gateway kommuniziert mit dem NJS und der NJS mit dem TSI ebenfalls über nur je einen Port. Die VO muss sich auf einen entsprechenden externen Gateway-Port einigen. Alle anderen Festlegungen, d.h. NJS und TSI-Port sowie die Festlegung der Firewall-Architektur können organisations-lokal erfolgen. Bei allen Architekturvarianten ist darauf zu achten, dass der NJS von innen zu anderen Usites außen, d. h. zu deren Gateways, SSL Verbindungen aufbauen darf.

VO muss sich nur auf einen Port einigen

Im D-Grid gibt es bspw. eine Empfehlung für den Betrieb und die Architektur von Firewalls beim Einsatz von UNICORE [VoGr 06a]. Darin wird empfohlen das Gateway in einer DMZ zu betreiben, d.h. vor dem Gateway und zwischen Gateway und NJS je eine Firewall zu platzieren. Diese Architektur entspricht dem Beispiel der Usite B in Abbildung 4.9 auf S. 68. Die freizuschaltenden Ports orientieren sich an den Standard-Vorgaben von UNICORE und die Firewall-Regeln sind in Tabelle 4.6 zusammengefasst.

Dienst	Quelle		Ziel	
	Rechner/Netz	Port	Rechner/Netz	Port
Gateway	Extern	⋆	Gateway	4433
(in DMZ)	Gateway	⋆	NJS	8181
	NJS	8181	Gateway	⋆
NJS	NJS	8181	Gateway	⋆
(im internen Netz)	Gateway	⋆	NJS	8181

Tabelle 4.6: Empfehlung für Firewall-Regeln beim Betrieb von UNICORE im D-Grid nach [VoGr 06a]

4.9.2 Bewertung Firewall-Traversal: UNICORE

Da sich die Firewall-Konzepte der Middlewares deutlich unterscheiden, muss eine getrennte Bewertung erfolgen. Im folgenden werden die Kriterien aus Kapitel 3 auf die Firewall-Konzepte von UNICORE angewendet.

- **Middleware-Integration:**
  Das Firewall Konzept war integraler Bestandteil bei der Entwicklung von UNICORE und ist dementsprechend auch voll in die Middleware integriert (Bewertung UNICORE: 3). Dies gilt jedoch nicht für die anderen Middlewares, die weder das Konzept des Gateways noch Consigner und Endorser unterstützen (Bewertung Globus und gLite: 0). Middleware übergreifend ergibt sich eine Bewertung von 1 (Bewertung: 1).

- **Ressourcen-Integration:**
  Der Mechanismus ist für die Endsystem-Ressourcen vollständig transparent (Bewertung: 3).
- **Erweiterbarkeit:**
  Eine Erweiterung des Konzeptes erfordert eine internationale Abstimmung zwischen den UNICORE Entwicklern (Bewertung 1).
- **Middleware übergreifende Interoperabilität:**
  Das Firewall Konzept wird nur von UNICORE unterstützt und umgesetzt (Bewertung 0).
- **Interoperabilität auf Policy-Ebene:**
  Innerhalb der Spezifikation von UNICORE werden Vorschläge für die Firewall-Regeln und die Firewall-Architektur gemacht. Insbesondere im Hinblick auf die Firewall-Regeln ist eine globale Abstimmung erforderlich. Lokal spezifizierte unterschiedliche Regeln sind nicht möglich. (Bewertung: 1).
- **Interoperabilität beim ID-Management:**
  Die Filterung nach Endorser und Consigner ist möglich. Der Bezug von Identitätsinformationen stützt sich voll auf X.509 Zertifikate ab. Insofern muss hier die Bewertung aus Abschnitt 4.2.4 übernommen werden (Bewertung 1).
- **Trust Management; Formalisierung:**
  Die Vertrauenswerte zwischen den Partnern erfolgen einzig über den impliziten Vertrauenswert, der an das Zertifikat bzw. an die Adressen gebunden ist. Es gibt kein formalisiertes Modell (Bewertung 1).
- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).
- **Trust Management; Granularität:**
  Grundsätzlich besteht ein globales Vertrauensverhältnis innerhalb einer VO. Über lokale Firewall-Regeln und über das Gateway können für einzelne Systeme oder für einzelne Consigner oder Endorser (vgl. Abschnitt 4.2.3 individuelle Regeln und damit gruppenspezifische Trust Level umgesetzt werden. Im Prinzip wären sogar individuelle Regelungen möglich, allerdings wäre der administrative Aufwand in diesem Fall erheblich. Einfach anwendbare Mechanismen sind nicht vorgesehen. (Bewertung: 1).
- **Rechtedelegation; Granularität und Sicherheit:**
  Konzepte zur Delegation von Rechten sind nicht vorgesehen (Bewertung: 0).
- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Eine erweiterte Delegation oder die Beschränkung des Delegationsrechtes sind nicht vorgesehen (Bewertung: 0).

- **Delegation von Policies:**
  Die Regeln zur Konfiguration lokaler Firewalls werden informell VO-weit spezifiziert. Eine Berücksichtigung lokaler Gegebenheiten und Policies ist nur über eine VO-weite Abstimmung möglich. Formale Verfahren zur Beschreibung existieren nicht (Bewertung: 1).

- **Delegation von Aufgaben:**
  Die Delegation von Aufgaben, z.B. zur Konfiguration von Firewalls, sind nicht vorgesehen (Bewertung: 0).

- **Mapping:**
  Für den Fall, dass es zu Policy Konflikten kommt, gibt es keine Mechanismen zur Auflösung dieser Konflikte. Konflikte können z.B. dadurch entstehen, dass lokale Policies die Freischaltung von Ports, die von der VO vorgeschrieben werden, nicht zulassen (Bewertung: 0).

- **Skalierbarkeit:**
  Die zu spezifizierenden Regeln sind relativ einfach und der Umfang ist sehr stark begrenzt. Das Konzept geht davon aus, dass jede Organisation ihre eigene lokale Firewall betreibt. Das Konzept skaliert damit sowohl räumlich als auch organisatorisch. Das Firewall-Konzept von UNICORE stützt sich auf vorhandene lokale Firewalls ab. Hierbei gibt es das Problem, dass insbesondere im Grid-Umfeld oft sehr hohe Datenraten benötigt werden. In DEISA sind beispielsweise alle Partner mit 10 GBit/s an das DEISA-Netz angebunden. Eine Erweiterung einzelner Standorte auf 40 GBit/s ist absehbar. Für 10 GBit/s full wire speed gibt es vereinzelte Firewall-Lösungen, für 40 GBit/s sind solche derzeit nicht in Sicht. Insofern kann nicht von einer Skalierbarkeit von Hard- und Software ausgegangen werden (Bewertung: 2).

- **Flexibilität; Entitätenvielfalt:**
  Die Filterung erfolgt auf Basis der IP-Adresse der jeweiligen Ressource, andere Entitäten werden nur mittelbar über Funktionen im UNICORE Gateway unterstützt (Bewertung: 1).

- **Flexibilität; Organisationsflexibilität:**
  Die Filterung erfolgt ausschließlich auf lokaler Ebene. Zentrale oder föderierte Konzepte werden nicht unterstützt (Bewertung: 0).

- **Administrierbarkeit:**
  Die Freischaltungen sollten pro Ressource bzw. pro Subnetz für den Fall, dass alle Systeme des Subnetzes freigeschaltet werden sollen, erfolgen. Je größer die VO, umso größer wird auch das Regelwerk und damit der Administrationsaufwand. Der Umfang aller Regelsätze beträgt $O(n \cdot m)$ bei n Ressourcen bzw. Subnetzen und m Regeln. Durch das klare Konzept von UNICORE ist der Regelsatz (m, vgl. Tabelle 4.6) pro Ressource sehr überschaubar (Bewertung: 1,5).

- **Sicherheit; Sicherheitsniveau:**
  Das Sicherheitsniveau einer Firewall wird bestimmt durch die fehlerfreie

Implementierung der Firewall und des Regelwerks. Dabei gilt der Grundsatz, je größer das Regelwerk und je mehr geöffnete Ports und Dienste eine Firewall unterstützt, umso größer ist auch das Risiko einer Sicherheitsverletzung. Da für UNICORE nur sehr wenige Ports freizuschalten sind, kann bei einer sauberen Spezifikation der berechtigten Partner von einer niedrigen Eintrittswahrscheinlichkeit ausgegangen werden.

Die Tatsache, dass ein Angreifer in der Lage ist, die Firewall zu umgehen und Verbindungen zu Grid-Systemen aufzubauen, bedeutet noch nicht zwangsläufig, dass er damit auch unberechtigten Zugang zu Grid-Ressourcen erhalten kann. Die Schadenshöhe kann deshalb erst einmal als Mittel angenommen werden (Bewertung: 2).

- **Sicherheit; Zusicherung, QoP:**
  Weder für die lokale Umsetzung der Firewall-Architektur noch für die sichere Freischaltung berechtigter und die zuverlässige Sperrung unberechtigter Ressourcen bzw. Nutzer gibt es Mechanismen zur Überprüfung oder Zusicherung (Bewertung: 0).

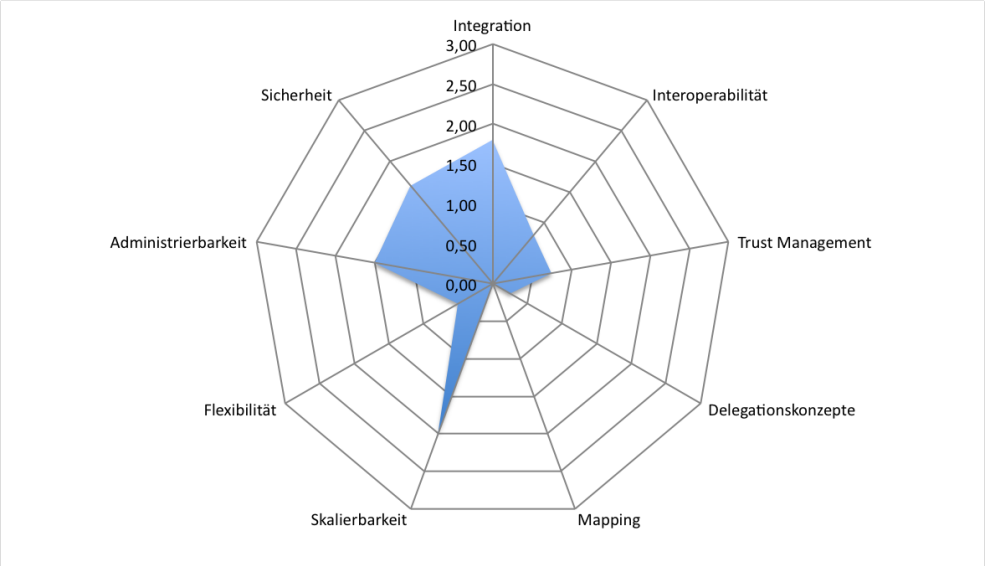

Abbildung 4.52: Bewertung Firewall-Konzept von UNICORE

Abbildung 4.52 fasst die Bewertung der UNICORE Firewall-Konzepte in einem Netzdiagramm zusammen.

4.9.3 Firewall-Traversal: Globus und gLite

keine Gateway Komponente

Bei der Konzeption von Globus findet sich kein solch stringentes und konsequentes Design im Hinblick auf Firewalls. Ein UNICORE vergleichbares Konzept einer Gateway-Komponente gibt es bei Globus nicht. Ein Gateway hat, im Hinblick auf Firewalls, zwei Vorteile:

1. Das Gateway kann als Sicherheitskomponente den Zugriff auf interne Systeme absichern, eine Zugriffskontrolle durchführen und als Relay wirken. Es ist damit ein Mittel, um interne und externe Komponenten mit jeweils unterschiedlichen Sicherheitsmechanismen zu schützen. Insbesondere kann die direkte Erreichbarkeit interner Komponenten an der Firewall bzw. am Gateway verhindert werden.
2. Das Gateway ist bei einer DMZ-Architektur die einzige Komponente, die von außen sichtbar ist, und damit auch die Komponente, die potentiell angreifbar ist. Dementsprechend kann das Gateway besonders geschützt und so realisiert werden, dass es möglichst wenig Angriffspunkte bietet.

Alle Systeme müssen von außen erreichbar sein

Da Globus keine derartigen architekturellen Konzepte unterstützt, sind für alle Systeme, die am Grid teilnehmen, Regeln in der Firewall einzupflegen, welche die Erreichbarkeit von außen sicherstellen, d.h. eine Unterscheidung in interne und externe Systeme ist nicht möglich. Soll eine DMZ realisiert werden, müssen alle am Grid teilnehmenden Systeme in der DMZ platziert werden.

Auch bei den Ports benötigt Globus deutlich mehr Ports, die in der Firewall freigeschaltet werden müssen. Tabelle 4.7 fasst die für D-Grid empfohlenen Firewall-Regeln zusammen [VoGr 06b, VoGr 06a]. Hierbei werden nur die Regeln für GT4 angegeben, denn die Regeln für GT2 und gLite sind strukturell vergleichbar. Aus der Tabelle wird ersichtlich, dass bei Globus sehr viel mehr Dienste von außen erreichbar sein müssen.

Dienst	Quelle		Ziel	
	Rechner/Netz	Port	Rechner/Netz	Port
GRAM	Extern	$\star$	GRAM	8433
(Job Management)	GRAM	CEPR	Extern	$\star$
MDS	Extern	$\star$	MDS	8443
GridFTP Kontrollkanal	Extern	$\star$	GridFTP	2811
GridFTP Daten (einzelner Kanal)	Extern	$\star$	GridFTP	CEPR
GridFTP Daten (mehrere Kanäle)	Extern	$\star$	GridFTP	CEPR
GridFTP Daten (mehrere Kanäle)	GridFTP	CEPR	Extern	$\star$
GSI-SSH	Extern	$\star$	GSI-SSH	2222
MyProxy	Extern	$\star$	MyProxy	7512

Tabelle 4.7: Empfehlung für Firewall Regeln beim Betrieb von GT4 im D-Grid nach [VoGr 06a]

Beim Betrieb von Globus wird davon ausgegangen, dass ein Benutzer Serverseitig durchschnittlich 20 Ports benötigt, um die in der Tabelle angegebenen Dienste zu nutzen. Insbesondere für parallele GridFTP Verbindungen werden die zusätzlichen Ports benötigt. Unter der Annahme, dass geschätzte 250 Nutzer gleichzeitig aktiv sind, müssen $250 \cdot 20$, d.h. 5000 Ports freigeschaltet werden, damit keiner der Nutzer in seiner Arbeit behindert wird. Dieser Bereich von Portnummern wird in der Literatur als Controllable Ephemeral Port Range (CEPR) bezeichnet [Welc 06, Glob b]. Im D-Grid wurde in Anlehnung an

das LCG Projekt eine CEPR von 20000 bis 25000 vereinbart. Dieser Bereich von 5000 Ports muss für jeden Server an der Firewall freigeschaltet werden.

GRAM (Grid Ressource Allocation Manager) ist die Komponente, mit der ein Client Jobs instantiieren, überwachen und verwalten kann. Der Client verbindet sich standardmäßig mit dem GRAM auf Port 8443.

Der MDS (Monitoring and Discovery Service) dient primär dem Ressourcen-Monitoring und ist über denselben Port wie GRAM erreichbar.

GridFTP dient der Übertragung großer Datenmengen und wurde in Abschnitt 4.1.5, Seite 64, beschrieben. Für den Kontrollkanal ist Serverseitig der Port 2811 freizugeben. Für den Datenkanal sind die CEPR freizuschalten.

GSI-SSH realisiert eine auf ssh basierende Terminal-Verbindung zur entsprechenden Ressource. Dafür muss Serverseitig der Port 2222 geöffnet werden.

MyProxy wurde in Abschnitt 4.2.10 beschrieben. Für dessen Verwendung ist Port 7512 erforderlich.

Besondere Herausforderung durch GridFTP

Durch die Verschlüsselung des Kontrollkanals und die Trennung von Front End und Datenknoten entstehen besondere Herausforderungen und Probleme bei der Verwendung von GridFTP (vgl. auch Abschnitt 4.1.5, S. 64). Dies wird im folgenden an zwei Beispielen erläutert.

Protokollablauf: GridFTP, aktive Mode, Stripping

Abbildung 4.53 zeigt ein Interaktionsdiagramm für GridFTP im Active Mode mit aktivierten Stripping. Nachdem der Client einen Kontrollkanal zum Server Front End des GridFTP-Servers aufgebaut hat, aktiviert er mit Hilfe des `MODE E` Kommandos den Stripping Mode. Der Parameter `PORT` beinhaltet die Client-seitigen Ziel-Ports, über die der Client die Daten empfängt. Im Schritt 4 versetzt das Front End einen oder mehrere Daten-Knoten in eine parallele Sendebereitschaft. Danach beginnt die eigentliche Datenübertragung direkt zwischen den Datenknoten und dem Client.

Empfangs-Port für Firewall nicht erkennbar

In diesem Ablauf entstehen Probleme für den Firewall-Betreiber auf Client Seite. Der mit dem `PORT` Parameter übergebene Empfangs-Port (im Beispiel `zzz`) ist für den Firewall-Betreiber nicht bekannt, da die Kommunikation zwischen Client und Server auf dem Kontrollkanal verschlüsselt erfolgt. Der Ansatz einer Paket- und Protokollanalyse durch die Firewall, wie dies bei Standard-FTP üblich ist, ist damit nicht umsetzbar.

Datenknoten unabhängig vom Front End

Ein weiteres Problem stellen die Datenknoten dar. Eigenständige Systeme mit eigenen Adressen, die völlig unabhängig vom Front End Knoten sein können. Aus Sicht der Client Firewall ist erkennbar, dass eine Grid Verbindung zwischen Client und Front End aufgebaut wird. Im Anschluss daran versuchen aber völlig andere Maschinen (die Datenknoten) Verbindungen mit dem Client aufzubauen.

Protokollablauf: Third Party Transfer

Auch der Third Party Transfer stellt ein Problem für Firewall-Administratoren dar (vgl. Abbildung 4.54). Hierbei steuert der Client den Datentransfer zwischen zwei Servern. Dazu baut er einen Kontrollkanal zu beiden Servern auf. Einen der beiden versetzt er in Empfangsbereitschaft (Kommando `SPAS`), den anderen in Sendebereitschaft (Kommando `SPOR`). Die Antwort des `SPAS`

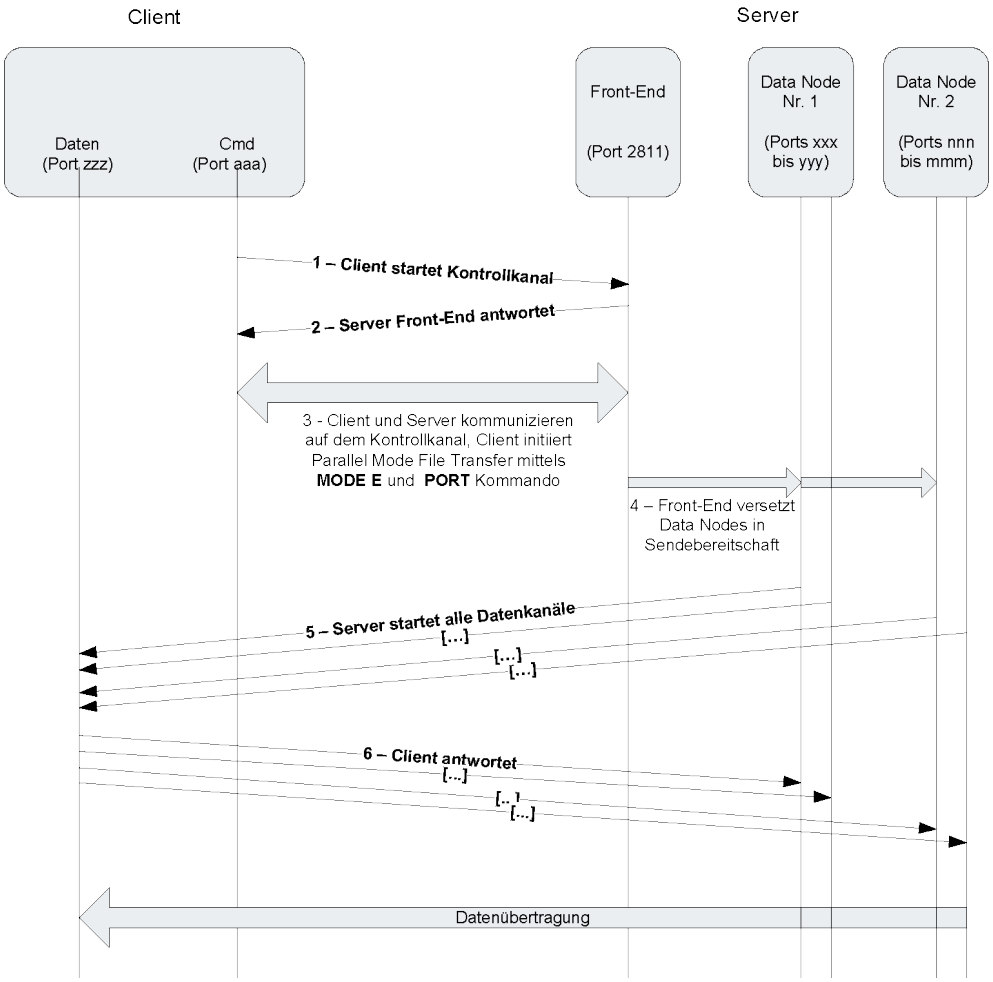

Abbildung 4.53: Protokollablauf bei GridFTP mit Stripping [GPW 06]

Kommandos enthält IP-Adressen und Ports der Empfänger-Seite. Diese Informationen werden im `SPOR` Kommando an die Sendeseite weitergereicht. Der Sender baut mit Hilfe dieser Informationen dann eine Datenverbindung zu den Empfängern auf und beginnt mit der Übertragung der Daten.

Kontrollinformation nicht zugänglich für Firewall

Der Firewall-Administrator — der im Beispiel zwischen den Rechnern A1 und B1, B2 — eine Firewall betreiben muss, hat keine Möglichkeiten an diese Kontrollinformationen zu gelangen (Verschlüsselter Kontrollkanal und Trennung von Front End und Daten Knoten). Dementsprechend kann auch keine automatische und gezielte Freischaltung erfolgen.

Im Rahmen von D-Grid werden auch die Auswirkungen von Firewalls auf Leistungsparameter im Netz (Ausbreitungsverzögerung, Verarbeitungsverzögerung, Serialisierungsverzögerung und Durchsatz) untersucht. Es werden auch Testszenarien spezifiziert, um den durch die Firewall induzierten Leistungsabfall zu messen [GMNE 07].

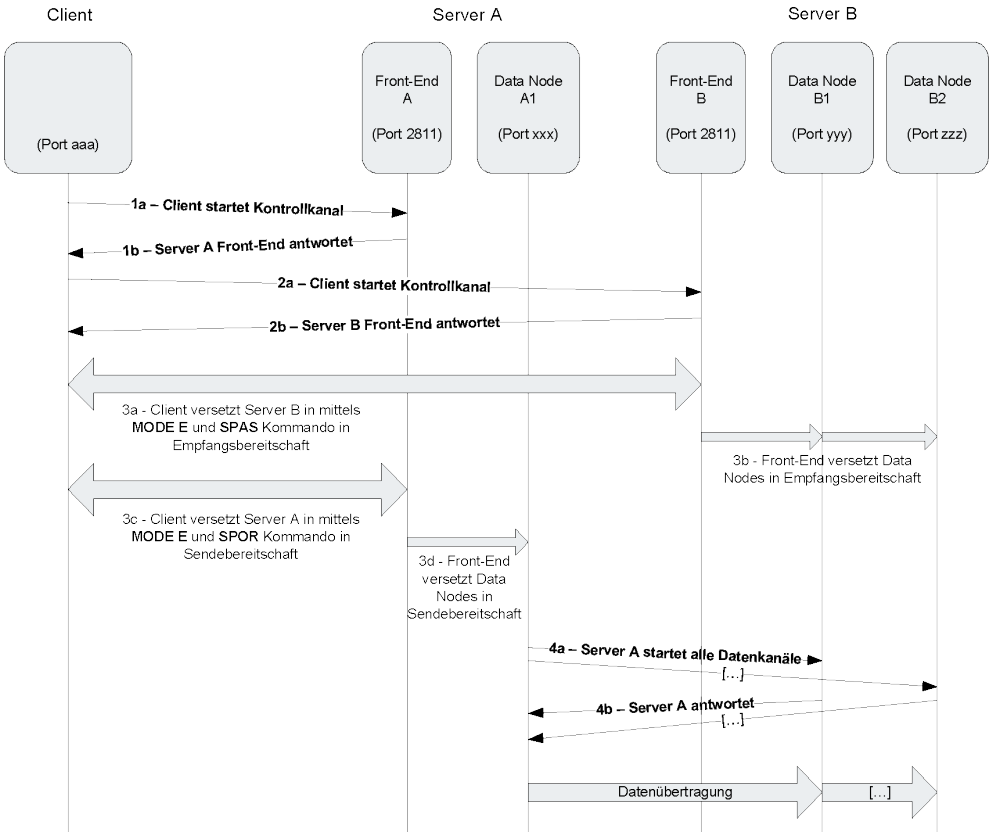

Abbildung 4.54: Protokollablauf bei GridFTP mit Third Party Transfer [GPW 06]

4.9.4 Bewertung Firewall-Traversal: Globus und gLite

Da die Platzierungsempfehlungen für Firewalls und die Empfehlungen für Freischaltungsregeln bei Globus und gLite sehr ähnlich und auch die sonstigen Firewall-Konzepte gut vergleichbar sind, erfolgt eine gemeinsame Bewertung im folgenden Teilabschnitt.

- **Middleware-Integration:**
  Bei der Entwicklung von Globus und gLite wurden Firewall-Aspekte nicht systematisch mit in den Entwicklungsprozess einbezogen. Sondern es wurden Anforderungen an Firewalls veröffentlicht [Welc 06] und Empfehlungen für den Betrieb und die Konfiguration von Firewalls spezifiziert [Glob b]. Dabei werden für sehr viele Dienste dedizierte Port-Freischaltungen gefordert. Firewall-Konzepte sind kein integraler Bestandteil der Middleware (Bewertung: 0).

- **Ressourcen-Integration:**
  Der Mechanismus ist für die Endsystem-Ressourcen vollständig transparent (Bewertung: 3).

- **Erweiterbarkeit:**
  Eine Erweiterung des Konzeptes erfordert eine internationale Abstimmung zwischen den Entwicklern (Bewertung 1).
- **Middleware übergreifende Interoperabilität:**
  Das Firewall-Konzept wird prinzipiell von Globus und gLite unterstützt (Bewertung 1).
- **Interoperabilität auf Policy-Ebene:**
  Es werden Vorschläge für die Firewall-Regeln gemacht. Hier ist jedoch eine globale Abstimmung erforderlich. Lokal spezifizierte unterschiedliche Regeln sind nicht möglich (Bewertung: 1).
- **Interoperabilität beim ID-Management:**
  Die Filterung erfolgt auf Basis der Ressourcen und deren Adressen. Ein ID-Management wird nicht unterstützt (Bewertung 0).
- **Trust Management; Formalisierung:**
  Die Vertrauenswerte zwischen den Partnern erfolgen einzig über den impliziten Vertrauenswert, der an die Adressen gebunden ist. Es gibt kein formalisiertes Modell (Bewertung 1).
- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**
  Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).
- **Trust Management; Granularität:**
  Grundsätzlich besteht ein globales Vertrauensverhältnis innerhalb einer VO. Über lokale Firewall-Regeln können für einzelne Systeme oder Subnetze individuelle Regeln und damit Ressourcen- bzw. Organisationsspezifische Trust Level umgesetzt werden. Eine echte Gruppenbildung, die unabhängig von organisatorischen Einheiten, bzw. Subnetzbereichen, definiert sein kann ist nur mit sehr hohem Aufwand umsetzbar. Einfach anwendbare Mechanismen sind nicht vorgesehen (Bewertung: 1).
- **Rechtedelegation; Granularität und Sicherheit:**
  Konzepte zur Delegation von Rechten, z.B. für die temporäre Freischaltung von Firewall-Regeln, sind nicht vorgesehen (Bewertung: 0).
- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Eine erweiterte Delegation oder die Beschränkung des Delegationsrechtes sind nicht vorgesehen (Bewertung: 0).
- **Delegation von Policies:**
  Die Regeln zur Konfiguration lokaler Firewalls werden informell VO-weit spezifiziert. Eine Berücksichtigung lokaler Gegebenheiten und Policies ist nur über eine VO-weite Abstimmung möglich. Formale Verfahren zur Beschreibung existieren nicht (Bewertung: 1).
- **Delegation von Aufgaben:**
  Die Delegation von Aufgaben, z.B. zur Konfiguration von Firewalls, sind nicht vorgesehen (Bewertung: 0).

- **Mapping:**
  Für den Fall, dass es zu Policy Konflikten kommt, gibt es keine Mechanismen zur Auflösung dieser Konflikte. Konflikte können z.B. dadurch entstehen, dass lokale Policies die Freischaltung von Ports, die von der VO vorgeschrieben werden, nicht zulassen (Bewertung: 0).

- **Skalierbarkeit:**
  Die zu spezifizierenden Regeln sind zwar deutlich umfangreicher als bei UNICORE, aber die einzelnen Regeln sind nicht sehr aufwändig und für eine Firewall einfach umzusetzen. Das Konzept geht davon aus, dass jede Organisation ihre eigene lokale Firewalls betreibt und skaliert damit sowohl räumlich als auch organisatorisch. Auch das Firewall-Konzept von Globus und gLite stützt sich auf vorhandene lokale Firewalls ab, d.h. es gibt die selben Probleme mit hohen Datenraten wie in der Bewertung der UNICORE Firewall-Konzepte (vgl. S. 181) bereits beschrieben (Bewertung: 2).

- **Flexibilität; Entitätenvielfalt:**
  Die Filterung erfolgt auf Basis der IP-Adresse der jeweiligen Ressource, andere Entitäten werden nicht unterstützt (Bewertung: 1).

- **Flexibilität; Organisationsflexibilität:**
  Die Filterung erfolgt ausschließlich auf lokaler Ebene. Zentrale oder föderierte Konzepte werden nicht unterstützt (Bewertung: 0).

- **Administrierbarkeit:**
  Die Freischaltungen sollten pro Ressource bzw. pro Subnetz für den Fall, dass alle Systeme des Subnetzes freigeschaltet werden sollen, erfolgen. Je größer die VO, umso größer wird auch das Regelwerk und damit der Administrationsaufwand. Der Umfang aller Regelsätze beträgt $O(n \cdot m)$ bei n Ressourcen bzw. Subnetzen und m Regeln. Der Regelsatz ist auch deutlich umfangreicher als dies z.B. bei UNICORE der Fall wäre (Bewertung: 1).

- **Sicherheit; Sicherheitsniveau:**
  Das Sicherheitsniveau einer Firewall wird bestimmt durch die fehlerfreie Implementierung der Firewall und des Regelwerks. Dabei gilt der Grundsatz, je größer das Regelwerk und je mehr geöffnete Ports und Dienste eine Firewall unterstützt, umso größer ist auch das Risiko einer Sicherheitsverletzung. Wegen der Controllable Ephemeral Port Range, welche die Freischaltung von 5000 Ports — quasi auf Vorrat — erfordert, egal ob diese gerade gebraucht werden oder nicht, kann nicht mehr von einer niedrigen Eintrittswahrscheinlichkeit ausgegangen werden.

  Bei Globus und gLite müssen grundsätzlich alle Ressourcen, die Dienste der Tabelle 4.7 implementieren, von außen erreichbar sein, d.h. in der Regel werden nahezu alle Ressourcen an der Firewall freigeschaltet. Ein Angreifer, der in der Lage ist, die Firewall zu überwinden und Verbindungen zu Grid-Systemen aufzubauen, kann damit im Zweifelsfall sehr viele Grid-Ressourcen angreifen. Die Schadenshöhe ist deshalb als hoch anzunehmen (Bewertung: 0).

- **Sicherheit; Zusicherung, QoP:**
  Weder für die lokale Umsetzung der Firewall-Architektur noch für die sichere Freischaltung berechtigter und die zuverlässige Sperrung unberechtigter Ressourcen bzw. Nutzer gibt es Mechanismen zur Überprüfung oder Zusicherung (Bewertung: 0).

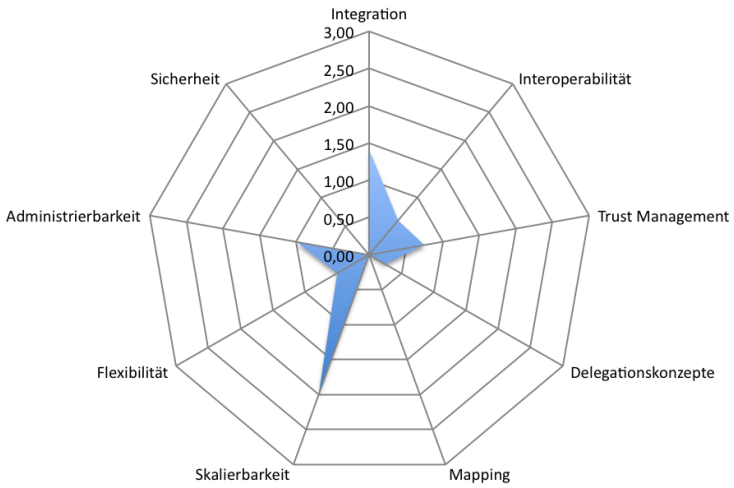

Abbildung 4.55: Bewertung Firewall-Konzept von Globus und gLite

Abbildung 4.55 fasst die Bewertung der Firewall-Konzepte von gLite und Globus zusammen. Ein Vergleich des Netzdiagramms mit dem der Bewertungen der UNICORE Firewall-Konzepte (vgl. Abbildung 4.52) zeigt doch deutliche Unterschiede. Das UNICORE Konzept ist sicherer und besser administrierbar. Insgesamt erscheint das Konzept hier durchdachter und ausgereifter.

4.9.5 Dynamische Firewalls

Die im vorangegangenen Abschnitt dargestellten Probleme, die durch verschlüsselte Kontrollkanäle, den Third Party Transfer u.ä. entstehen und mit klassischen statischen oder dynamischen Packet- oder Application-Level Firewalls nicht adäquat lösbar sind, haben zu Entwicklungen für die dynamische Konfiguration von Firewalls geführt. Die Grundidee hierbei ist, dass ein Endsystem vor der eigentlichen Datenübertragung die lokale sowie die fremde Firewall über die Kommunikationsparameter (Protokoll, Adressen, Ports, u.ä.) informiert und die Firewalls dann dynamisch und temporär diese Kommunikation freischalten. Das bedeutet, dass ein Client System quasi selbständig Regeln in einer Firewall freischalten kann. Im folgenden werden Beispiel für Mechanismen zur dynamischen Konfiguration von Firewalls vorgestellt.

Hole Punching

Hole Punching nutzt Freischaltung für ausgehenden Verkehr

Hole Punching wurde durch Anwendungen wie Voice over IP (VoIP), Skype und P2P Protokolle bekannt. Der Zweck von Hole Punching ist die direkte Kommunikation zwischen zwei Systemen, von denen sich mindestens eines hinter einer NAT-Firewall befindet und deshalb eigentlich von außen nicht erreichbar ist. Hole Punching Techniken nutzen die Tatsache, dass die meisten Firewalls ausgehenden (UDP) Verkehr erlauben.

Relay Server zur Ermittlung von NAT-Art und als Vermittler

Mit Hilfe eines **Relay-Servers** (mit öffentlicher Adresse) kann ein Client ermitteln, ob er sich hinter einem NAT-Gateway befindet. Außerdem kann der Client die öffentliche IP-Adresse, die Art des NAT-Gateways und den entsprechenden externen Port ermitteln, über den seine eigene Verbindung geleitet wird. Diese ermittelten Informationen werden verwendet um eine UDP basierte Kommunikation zu einem anderen System aufzubauen.

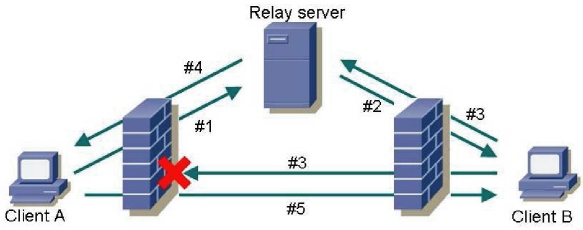

Abbildung 4.56: Prinzip des UDP Hole Punching [GMNP 06]

Funktionsweise UDP Hole Punching

Abbildung 4.56 stellt das Prinzip des UDP Hole Punching dar [GMNP 06]. Der Relay Server stellt die zentrale Komponente des Konzeptes dar. Beide Clients verbinden sich mit dem Relay Server und übermitteln hierbei ihre internen Adressen und Ports. Aus dem IP-Header kann der Relay Server die öffentlichen Adressen und Ports ermitteln. Außerdem kann die Art des NAT-Gateways ermittelt werden. Im folgenden werden die einzelnen Nachrichten aus Abbildung 4.56 näher erläutert:

1. Wenn die Clients eine Verbindung zwischen sich aufbauen wollen, sendet der Initiator (Client A) eine Nachricht an den Relay-Server (Nachricht #1 in der Abbildung) und teilt diesem mit, dass er mit Client B über den Port p kommunizieren möchte.
2. Der Server teilt dem Client B die öffentliche IP-Adresse a von A und den UDP Port p mit.
3. Client B sendet seinen bevorzugten UDP Port q an den Relay Server und gleichzeitig ein UDP Datagramm mit Quell-Port q und Ziel-Port p an a.
4. Die Firewall auf Seiten von B lässt das Datagramm passieren und erzeugt eine dynamische Firewall-Regel, die Antworten auf das Datagramm zulässt. Die Firewall auf Seite von A blockiert das entsprechende Datagramm.

5. Der Relay Server informiert A über IP-Adresse b und Port q von B.
6. A sendet nun seinerseits ein UDP Datagramm mit Quell-Port p zum Ziel-Port q und „öffnet“ damit seine lokale Firewall. Wegen des dynamischen Eintrags in der Firewall von B, die noch aktiv ist, kann die Nachricht die Firewall von B passieren und ab diesem Zeitpunkt besteht ein bidirektionaler Kommunikationskanal zwischen A und B.

Da UDP ein verbindungsloser, ungesicherter Dienst der Transportschicht ist, bietet das Protokoll keine Mechanismen zur Erkennung von Paketverlusten oder zur Reihenfolgesicherung. Dies muss entweder von der Anwendung erkannt werden oder es wird ein Protokoll verwendet, das diese Funktionen nachbildet. Hierzu könnte das **UDP-based Data Transfer Protocol (UDT)** [GuGr 07, GuGr 04] verwendet werden.

Die bekannteste UDP Hole Punching Technik ist **STUN (Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs))** [RWHM 03], die allerdings nur für drei von vier möglichen NAT-Betriebsarten funktioniert.

Im folgenden werden die vier verschiedenen Betriebsarten von NAT-Gateways eingeführt:

1. **Full Cone NAT**: In diesem Fall werden die interne (private) IP-Adresse sowie der interne Port immer auf denselben externen Port und dieselbe externe Adresse abgebildet. Außerdem kann jedes externe System den internen Rechner über die externe Adresse erreichen. — Full Cone NAT
2. **Restricted Cone NAT** arbeitet wie Full Cone, allerdings kann ein externes System mit Adresse x den internen Rechner nur erreichen, wenn dieser vorher eine Nachricht an die Adresse x geschickt hat. — Restricted Cone NAT
3. **Port Restricted Cone NAT** ist vergleichbar dem Restricted Cone NAT; allerdings umfasst die Restriktion auch Ports, d.h. ein externer Rechner mit Adresse x und Port p kann nur Daten an den internen Rechner senden, wenn dieser vorher Pakete an x und Port p geschickt hat. — Port Restricted Cone NAT
4. **Symmetric NAT**: Beim symmetrischen NAT werden alle Verbindungen (definiert durch Quell-Adresse, Quell-Port, Ziel-Adresse, Ziel-Port) auf dieselbe externe Adresse und denselben externen Port abgebildet. Falls das interne System von derselben IP-Adresse und demselben Port Verbindungen zu verschiedenen Zielsystemen aufbaut, erfolgt ein unterschiedliches externes Mapping. Außerdem kann das externe System nur Pakete an das interne System senden, wenn es bereits ein Paket erhalten hat. — Symmetric NAT

STUN funktioniert mit den ersten drei NAT Betriebsarten.

CODO (Cooperative On-Demand Opening)

Cooperative On-Demand Opening (CODO) wurde ursprünglich für die

dynamische Konfiguration von Firewalls entwickelt, die Network Address Translation (NAT) verwenden, sowie für die speziellen Anforderungen von Grid Anwendungen [SAL 05a]. Zwei Systeme, die durch eine oder mehrere Firewalls/NAT-Gateways getrennt sind, können i.A. keine Verbindung zueinander aufbauen. CODO wurde entwickelt, um autorisierten Anwendungen die dynamische Freischaltung von Firewalls, und damit die Kommunikation „durch“ diese Firewalls, zu ermöglichen. Die Entwickler argumentieren, dass dadurch die Sicherheit der Gesamtinfrastruktur steigt, denn anstatt große Port-Bereiche auf Vorrat freizuschalten (vgl. bspw. CEPR in Abschnitt 4.9.3), öffnet die autorisierte Anwendung nur gezielt einzelne Ports, die auch nur so lange geöffnet bleiben, wie sie auch tatsächlich für eine Kommunikation benötigt werden.

CODO erlaubt autorisierten Anwendungen dynamische Freischaltung

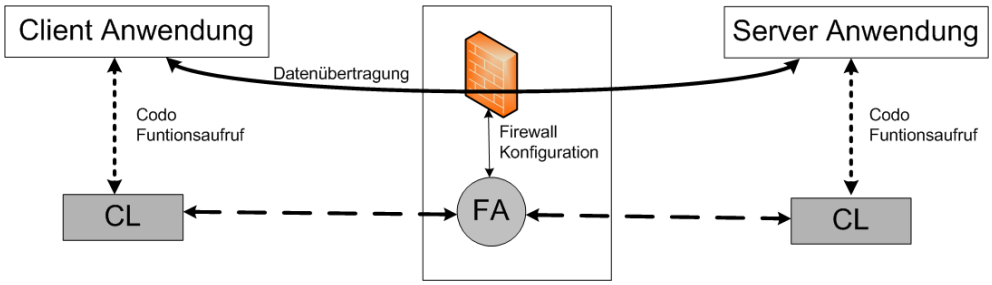

Abbildung 4.57: CODO: architekturelle Komponenten und Grundprinzip; nach [SAL 05a]

Abbildung 4.57 stellt die Komponenten sowie die grundsätzliche Funktionsweise von CODO dar. Die Firewall wird um einen Dienst — den **Firewall Agent (FA)** — erweitert. Der FA ist in der Lage dynamisch neue Regeln in der Firewall zu aktivieren, bestehende Regeln zu ändern oder zu löschen. Die Client- und Server-Anwendungen müssen um eine so genannte **Client Library (CL)** erweitert werden, die das CODO Protokoll implementiert. Vor der eigentlichen Kommunikation werden über CODO Funktionen der CL die notwendigen Ports freigeschaltet. Danach kann der eigentliche Datentransfer zwischen Client und Server erfolgen. Im Normalfall werden sowohl das Netz, in dem sich der Client befindet, als auch das Server-Netz durch eigene Firewalls gesichert (vgl. Abbildung 4.58). Im folgenden wird dargestellt, wie mit Hilfe von CODO eine Verbindung zwischen Client und Server aufgebaut werden kann. Dabei wird davon ausgegangen, dass der Client Adresse und Port des Servers kennt:

Erweiterung der Firewall um Firewall Agent (FA)

Erweiterung von Client u. Server um Client Library (CL)

CODO Protokollablauf

1. Der Server muss seinen Dienst über CODO registrieren. Sobald der Server-Prozess einen lokalen Socket öffnet, sendet die Server CL einen Registration Request an seinen lokalen FA. Dazu wird eine gesicherte TCP Verbindung zwischen der Server-Anwendung und dem FA aufgebaut. Die Anwendung autorisiert sich über ein X.509 Zertifikat und der FA kann diese Identitätsinformationen zur Autorisierung verwenden. Falls der Server berechtigt ist, erfolgt das Official Binding, d.h. die Re-

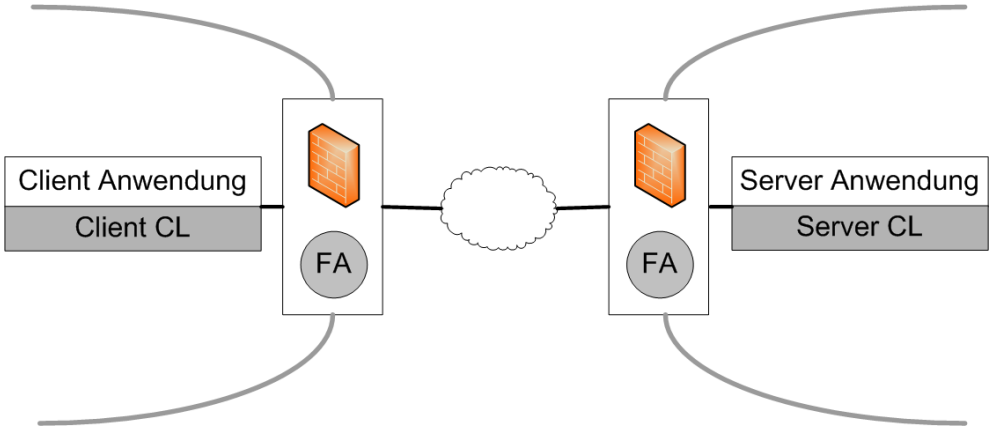

Abbildung 4.58: CODO: Funktion bei mehreren Firewalls [SAL 05a]

gistrierung von $S1$, d.h. Server-Port und Server-Adresse bei der lokalen Firewall.

2. Der Client baut ebenfalls eine CODO Verbindung zu seinem lokalen FA auf und wird wie der Server authentisiert. Der Client fordert eine Freischaltung von Client Port und Client Adresse ($C1$) zu $S1$ sowie ggf. eine Adressumsetzung an. Falls der Client berechtigt ist, die gewünschte Verbindung freizuschalten, verbindet sich der Client FA mit dem Server FA. Auch hier erfolgt wieder eine Authentisierung und der Client FA bittet um Freischaltung der Verbindung von $C1$ zu $S1$.
3. Bei ausreichenden Rechten ergänzt der Server FA die Firewall um eine Regel, die Verbindungen von $C1$ zu $S1$ und Antworten von $S1$ an $C1$ erlauben.
4. Dasselbe geschieht durch den Client FA auf Seiten der Client Firewall.
5. Der eigentliche Datenverkehr zwischen Client und Server kann erfolgen.
6. Nach Abschluss der Kommunikation werden auf beiden Seiten die Freischaltungsregeln wieder gelöscht.

Zusammenfassend lässt sich feststellen, dass CODO starke Authentisierung verwendet. Regeln werden nur hinzugefügt, wenn sie auch wirklich gebraucht werden. Die Port-Bereiche können so eng wie möglich gehalten werden und die Freischaltungen erfolgen auf Ebene der Einzelverbindung und nicht etwa für ganze Netzbereiche. Durch das CODO Protokoll entsteht natürlich zusätzlicher Overhead, insbesondere beim Verbindungsaufbau. Die Autoren haben hier eine Verlangsamung um den Faktor 50 gemessen [SAL 05b, SAL 05a].

Problem: Adressverteilung

Für eine nahtlose Verwendung von CODO müssen die Adressen der FAs für alle potentiellen Kommunikationswege bekannt sein, d.h. Client und Server müssen die Adresse der lokalen FA kennen. Dies lässt sich einfach realisieren. Der Client FA muss jedoch die Adressen aller potentiellen Server FAs kennen. Derzeit wird dies durch eine lokale Tabelle beim jeweiligen FA realisiert.

4.9.6 Bewertung dynamische Firewalls: CODO

Das in Abschnitt 4.9.5 vorgestellte Hole-Punching wird zwar gelegentlich auch als Firewall-Konzept für Grids vorgeschlagen [GMNP 06], hier jedoch nicht weiter untersucht. Die Technik wurde entwickelt, um trotz einschränkender Firewall-Regeln für bestimmte Anwendungen eine Kommunikation durch die Firewall hindurch zu ermöglichen. Hole Punching hebelt dabei häufig existierende Firewall-Policies aus. Aus diesem Grund wird im Folgenden nur noch CODO bewertet.

- **Middleware-Integration:**
  Die Client Library von CODO kann beim Kompilieren der Middleware als Bibliothek dazugebunden werden. Danach kann jeder Funktionsaufruf zum Aufruf einer Netzwerkverbindung durch die CL und über CODO vermittelt werden (Bewertung: 2).
- **Ressourcen-Integration:**
  Der Mechanismus ist für die Endsystem-Ressourcen vollständig transparent (Bewertung: 3).
- **Erweiterbarkeit:**
  Eine Erweiterung des Konzeptes erfordert eine Abstimmung innerhalb der Entwicklergemeinde von CODO (Bewertung 1).
- **Middleware übergreifende Interoperabilität:**
  Durch die Realisierung der Client-Komponente von CODO als Betriebssystem-nahe Bibliothek ist die Verwendung nicht auf bestimmte Middlewares beschränkt (Bewertung 3).
- **Interoperabilität auf Policy-Ebene:**
  Die Prüfung der Berechtigung zur dynamischen Änderung der Firewall erfolgt beim jeweiligen FA. Die Festlegung von Policies zur Spezifikation und zum Austausch entsprechender Autorisierungspolicies ist bisher nicht spezifiziert [VoGr 06c, Grün 07]. Derzeit muss eine Abstimmung über diese Rechte auf individueller Basis erfolgen (Bewertung: 0).
- **Interoperabilität beim ID-Management:**
  Zur Ermittlung der Identität werden ausschließlich X.509 Zertifikate verwendet. Insofern muss hier die Bewertung aus Abschnitt 4.2.4 übernommen werden (Bewertung 1).
- **Trust Management; Formalisierung:**
  Die Administratoren der Firewalls bzw. der Firewall Agents könnten implizite Vertrauenswerte auf individueller Basis des Zertifikates vergeben, allerdings wird kein Mechanismus zur Spezifikation des Vertrauenswertes zur Verfügung gestellt. Damit ist nur eine binäre Unterscheidung in „Vertrauen“ und „kein Vertrauen“ möglich (Bewertung 0).
- **Trust Management; Berechnung oder Ableitung der Vertrauenswerte:**

Verfahren zur dynamischen Berechnung oder Ableitung von Vertrauenswerten existieren nicht (Bewertung: 0).

- **Trust Management; Granularität:**
  Über die lokale Rechtevergabe bzw. Konfiguration des FA können für jedes Zertifikat individuelle Regeln spezifiziert werden und damit implizit Benutzer- und auch Organisations-spezifische Trust Level umgesetzt werden. Eine echte Gruppenbildung, die unabhängig von organisatorischen Einheiten bzw. Subnetzbereichen definiert sein kann, ist nur mit sehr hohem Aufwand umsetzbar. Einfach anwendbare Mechanismen sind nicht vorgesehen (Bewertung: 1).

- **Rechtedelegation; Granularität und Sicherheit:**
  Konzepte zur Delegation von Rechten, z.B. für die temporäre Freischaltung von Firewall-Regeln, sind nicht vorgesehen (Bewertung: 0).

- **Rechtedelegation; Beschränkung des Delegationsrechtes:**
  Eine erweiterte Delegation oder die Beschränkung des Delegationsrechtes sind nicht vorgesehen (Bewertung: 0).

- **Delegation von Policies:**
  Formale Verfahren zur Beschreibung oder zur Delegation von Policies existieren nicht (Bewertung: 0).

- **Delegation von Aufgaben:**
  Die Delegation von Aufgaben für die Freischaltung entfernter Firewalls ist im CODO Protokoll gegeben. Allerdings beschränkt sich dies auf Freischaltungen, die Delegation allgemeiner Aufgaben ist nicht vorgesehen (Bewertung: 1).

- **Mapping:**
  Für den Fall, dass es zu Policy Konflikten kommt, gibt es keine Mechanismen zur Auflösung dieser Konflikte (Bewertung: 0).

- **Skalierbarkeit:**
  Die Autoren selbst weisen bereits auf den Overhead hin, der durch die Anwendung von CODO beim Verbindungsaufbau entsteht [SAL 05b]. Mit Hilfe von CODO können nicht mehrere Systeme oder Anwendungen gruppiert und mit Hilfe einer einzelnen Regel (z.B. für einen gesamten Netzbereich) freigeschaltet werden. Die Freischaltung erfolgt immer auf Basis der einzelnen Anwendung. Für sehr viele Kommunikationsbeziehungen kann es dadurch zu Skalierungsproblemen sowohl beim FA oder den Firewalls kommen, d.h. eine Skalierbarkeit in Hardware und Software ist nicht gegeben. Ein weiteres Skalierungsproblem entsteht dadurch, dass jeder FA die Adressen aller FAs kennen muss, über die eine potentielle Kommunikation eines lokalen Clients vermittelt werden muss. Auch die organisatorische Skalierbarkeit ist daher problematisch. Die räumliche Ausdehnung der Ressourcen hat dagegen keinen großen Einfluss (Bewertung: 1).

- **Flexibilität; Entitätenvielfalt:**
  Die Filterung erfolgt auf Basis der Anwendung bzw. des Nutzers der Anwendung, andere Entitäten sind nicht vorgesehen (Bewertung: 1).

- **Flexibilität; Organisationsflexibilität:**
  Die Filterung erfolgt sowohl auf lokaler Ebene als auch föderiert (Bewertung: 1).

- **Administrierbarkeit:**
  Für die Spezifikation oder Delegation von Autorisierungsinformationen sind keine Mechanismen vorgesehen, d.h. jeder FA-Administrator muss für alle potentiellen Entitäten eigene Freischaltungsregeln spezifizieren. Desweiteren muss die Liste mit den Adressen der anderen FAs eingerichtet und gepflegt werden (Bewertung: 1).

- **Sicherheit; Sicherheitsniveau:**
  Der klare Vorteil von CODO im Vergleich zu anderen Firewall-Konzepten ist die dynamische und kurzfristige Freischaltung von Firewall-Regeln. Jede Anwendung kann genau die benötigten Ports für den Zeitraum der Nutzung freischalten. Dadurch werden keine Ports auf Vorrat freigeschaltet und es sind auch keine Ports offen, wenn der entsprechende Dienst gar nicht genutzt wird. Damit lässt sich mit Hilfe von CODO ein minimaler Regelsatz realisieren und deshalb von einer geringen Eintrittswahrscheinlichkeit ausgehen.

  Ein Angreifer der in der Lage ist, CODO zu umgehen und Verbindungen zu Grid-Systemen aufzubauen, kann damit nicht notwendigerweise sehr viele Grid-Ressourcen angreifen. Die Schadenshöhe ist deshalb als niedrig anzunehmen (Bewertung: 3).

- **Sicherheit; Zusicherung, QoP:**
  Weder für die lokale Umsetzung der Firewall-Architektur noch für die sichere Freischaltung berechtigter und die zuverlässige Sperrung unberechtigter Ressourcen bzw. Nutzer gibt es Mechanismen zur Überprüfung oder Zusicherung (Bewertung: 0).

Abbildung 4.59 fasst die Bewertung von CODO nach den Kriterien des Kriterienkatalogs zusammen.

4.10 Defizit- und Schwachstellenanalyse

In diesem Kapitel wurden existierende Sicherheitsmechanismen für Grids untersucht und bewertet. Dabei wurde der in Abschnitt 3.2 entwickelte Kriterienkatalog zugrunde gelegt. Diese Bewertung kann neben der primär intendierten Nutzung, d.h. der Unterstützung des Sicherheitsverantwortlichen bei der Mechanismenauswahl, auch für eine allgemeinere Bewertung der

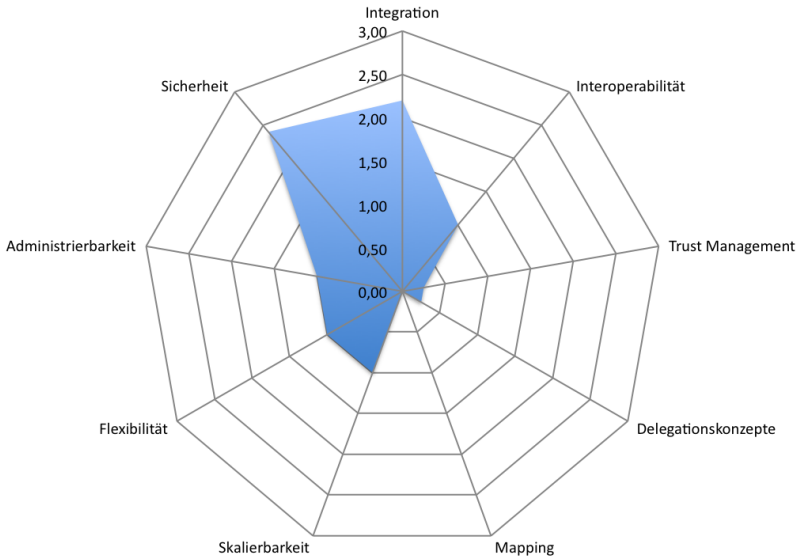

Abbildung 4.59: Bewertung dynamischer Firewall-Konzept

Bemühungen zur Grid-Sicherheit sowie für eine Defizit- und Schwachstellenanalyse verwendet werden. Hierzu fasst Tabelle 4.8 die Gesamtbewertungen der Sicherheitsmechanismen zusammen. Die Berechnungsvorschrift für die Ermittlung der Gesamtbewertung wurde so konzipiert, dass der errechnete Wert innerhalb des Wertebereichs liegt, der auch für die Einzelkriterien gilt. Damit lässt sich eine allgemeine Vergleichbarkeit zwischen den einzelnen Kriterien und abgeleiteten Bewertungszahlen als auch zwischen den verschiedenen Mechanismen herstellen. Die Bewertung der Kriterien erfolgt nach den in Tabelle 3.1 angegebenen Bewertungszahlen von 0 bis 3, wobei der Wert 0 für die Nichterfüllung des Kriteriums und der Wert 3 für die vollständige Erfüllung steht. Aus der Tabelle lässt sich auch ableiten, dass keiner der Mechanismen in der Lage ist alle Kriterien vollständig zu erfüllen. Es ist sogar so, dass der größte Teil der Mechanismen die Kriterien nur teilweise erfüllt. Nur ein Mechanismus hat eine Bewertungszahl die größer als 2 ist. Acht Mechanismen liegen in ihrer Bewertung zwischen 1 und 2 und vier Mechanismen sind schlechter als 1 bewertet. Das heißt, bei der Umsetzung der Mechanismen gibt es zum Teil erhebliche Defizite.

Gesamtbewertung aller Mechanismen

Zur Ermittlung von Mechanismen-übergreifenden generellen Stärken und Schwächen wurde neben der Einzelbewertung eine vergleichende Bewertung aller Mechanismen herangezogen. Abbildung 4.60 fasst alle Netzdiagramme aller Mechanismen in einem Diagramm zusammen.

Netzdiagramm aller Mechanismen

Die drei besten Bewertungen wurden bei der Skalierbarkeit, der Integration sowie der Flexibilität erzielt. Die drei schlechtesten Bewertungen sind beim Trust Management, bei den Delegationskonzepten und beim Mapping ablesbar. Abbildung 4.61 fasst die Mittelwerte der Bewertungen der ersten Ebene des Kriterienkataloges (vgl. Abschnitt 3.2) über alle Mechanismen zusam-

Netzdiagramm der durchschnittlichen Bewertungen

Sicherheitsmechanismus (Abschnitt)	Gesamtbewertung
Identifikation und Authentisierung (4.2.4)	**1,06**
Grid Map File und UUDB (4.2.7)	**0,70**
CAS (4.2.9)	**1,24**
myProxy (4.2.11)	**1,16**
VOMS (4.3.2)	**1,37**
GridShib (4.3.4)	**1,39**
Integrität der Kommunikation (4.4.1)	**2,13**
Vertraulichkeit der Kommunikation (4.4.3)	**1,92**
Privacy-Dienste (4.5.3)	**0,93**
Sandboxing und Virtualisierung (4.6.6)	**0,90**
Logging (UNICORE, gLite / Globus) (4.7.2)	**0,63** / **0,70**
Firewall-Traversal (UNICORE) (4.9.2)	**1,13**
Firewall-Traversal (Globus, gLite) (4.9.4)	**0,74**
Dynamische Firewall (4.9.6)	**1,20**

Tabelle 4.8: Gesamtbewertung der Sicherheitsmechanismen (Übersicht)

men.

In Abbildung 4.62 sind die Mittelwerte der entsprechenden Kriterien aller drei Ebenen im Kriterienkatalog zusammengefasst.

4.10.1 Stärken der Mechanismen

Die Stärken der Verfahren liegen in Grid-typischen Aspekten der Middleware-Implementierung und der Umsetzung Grid-typischer Anforderungen. Für eine effiziente interorganisationale Zusammenarbeit und eine effektive Nutzung von Ressourcen und Middleware spielen Skalierbarkeit, Integration, Flexibilität und Interoperabilität eine entscheidende Rolle.

Skalierbarkeit

Mechanismen skalieren in allen Dimensionen

Eine Grid-Infrastruktur für sehr große Projekte, wie z.B. das LHC Grid, sollte in der Lage sein sowohl räumlich, organisatorisch als auch in Hard- und Software im globalen Maßstab zu wachsen, ohne das dabei der Aufwand überproportional zunimmt. Diese Forderung gilt natürlich nicht nur für die Grid-Infrastruktur und deren Middleware-typischen Komponenten, sondern auch für die Sicherheitsmechanismen. Ein Mechanismus, der nicht in der Lage ist, im selben Maße wie die Middleware selbst zu skalieren, würde das Gesamtsystem bremsen und dementsprechend im Zweifelsfall nicht eingesetzt werden (wie z.B. bei Mechanismen zur Unverkettbarkeit (vgl. Abschnitt 4.5) geschehen). Eine Maßzahl, die im Mittel in Richtung 3 bei der Bewertung der Skalierbarkeit geht, bedeutet, dass die Mechanismen in allen drei Dimensionen (Hard- und Software, organisatorisch, räumlich) skalieren. Über alle bewerteten Mechanismen wird bei der Skalierbarkeit ein Mittelwert von 2,33 erreicht.

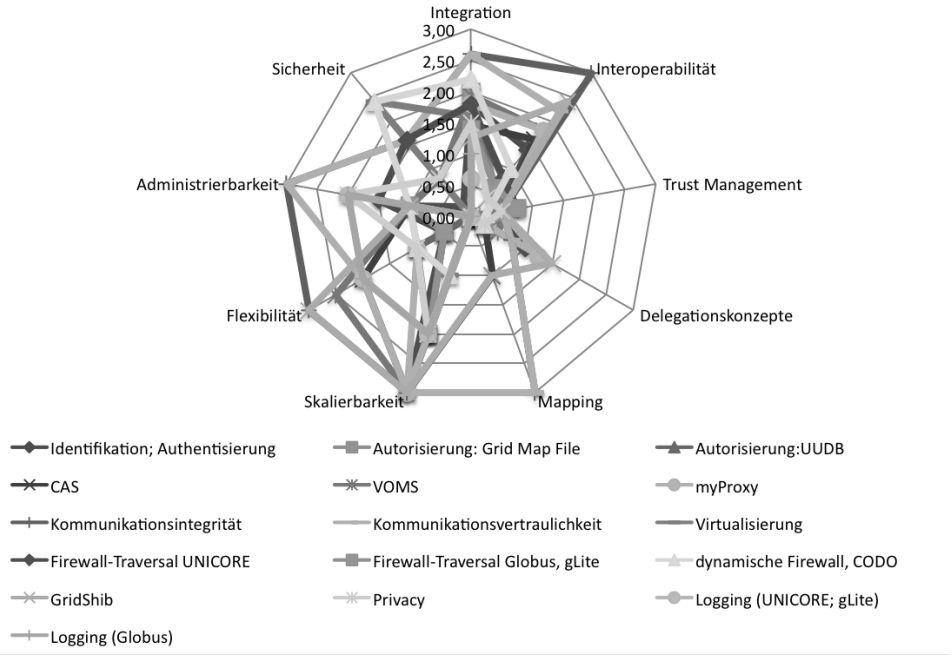

Abbildung 4.60: Vergleichende Bewertung der Mechanismen

Integration

Die Bewertung der Integration setzt sich aus den Bewertungen der Middleware-Integration, der Ressourcen-Integration und der Erweiterbarkeit zusammen (vgl. auch Kriterienkatalog in Abbildung 4.62). Ein Sicherheitsmechanismus ist im Optimalfall vollständig in die Middleware und in die verschiedenen Ressourcen integrierbar bei gleichzeitiger einfacher Erweiterbarkeit.

Kriterium Integration wird erfüllt

Die betrachteten Mechanismen erreichen hier eine Bewertung, die in Richtung von 2 tendiert, d.h. auch diese Kriterien werden im Mittel erfüllt. Die Erweiterbarkeit erhält dabei die besten Bewertungen, die Möglichkeiten zur Ressourcen-Integration schneiden am schlechtesten ab.

Flexibilität

Das Kriterium Flexibilität setzt sich aus den beiden Teilkriterien Entitätenvielfalt und Organisationsflexibilität zusammen. Mit diesem Kriterium wird bewertet, inwiefern Sicherheitsmechanismen in der Lage sind komplexe organisatorische Verbünde mit einer Vielzahl von Entitäten abbilden und unterstützen zu können. Die unterschiedlichen Mechanismen schneiden im Mittel bei der Entitätenvielfalt etwas besser ab als bei der Unterstützung verschiedener Organisationsformen. Auch hier gilt, dass viele Sicherheitsmechanismen entweder für den rein lokalen Einsatz oder allenfalls noch für eine

Entitätenvielfalt etwas besser als Organisationsflexibilität

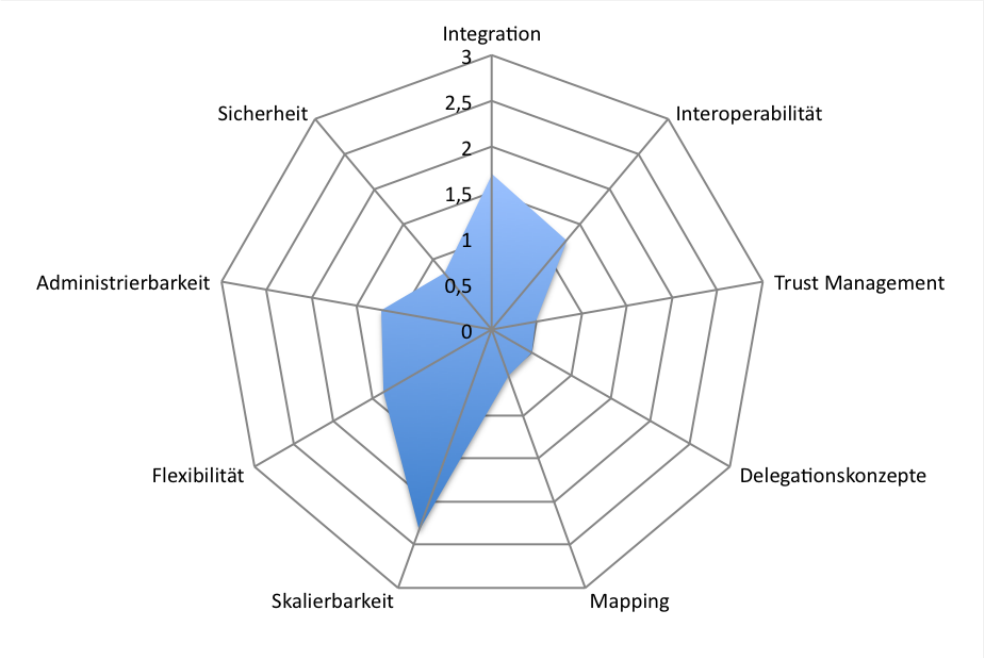

Abbildung 4.61: Durchschnitt der Bewertungen (Ebene 1 des Kriterienkatalogs) über alle Mechanismen

zentrale Verwendung entwickelt wurden. Föderative Prinzipien werden nur von sehr wenigen Mechanismen unterstützt.

Interoperabilität

Die Interoperabilität setzt sich aus drei Teilkriterien zusammen. Die Middleware-übergreifende Interoperabilität berücksichtigt Kooperationsformen wie z.B. das D-Grid, bei dem mehrere verschiedene Middlewares eingesetzt werden. Die eingesetzten und einsetzbaren Sicherheitsmechanismen sollten am besten unter allen Middlewares verwendbar sein und von diesen auch unterstützt werden.

Die Mechanismen werden über Policies parametrisiert und gesteuert. Möglichkeiten zur Spezifikation von Policies und deren formalisierter Austausch bildet die Basis für eine einheitliche und abgestimmte Verwendung der Sicherheitsmechanismen. Das dritte Teilkriterium der Interoperabilität beim Identitätsmanagement berücksichtigt die Einbettung der Sicherheitsmechanismen in, sowie eine möglichst nahtlose Nutzung von, lokalen Identity & Access Management Systeme. Nur bei einer Interoperabilität der Schemata und Verfahren der lokalen I&AM Systeme können Redundanzen und Inkonsistenzen verhindert und der Managementaufwand reduziert werden. Der Mittelwert der Bewertungszahlen der Interoperabilität liegt allerdings näher bei 1 als bei 2. Obwohl der Mittelwert bei der Interoperabilität noch zu den Besseren gehört, ist er absolut betrachtet allerdings suboptimal.

Bewertung Interoperabilität bei 1,27

Bei den Teilkriterien der Interoperabilität schneidet die Interoperabilität beim Identitätsmanagement am besten und die Interoperabilität der Policies am

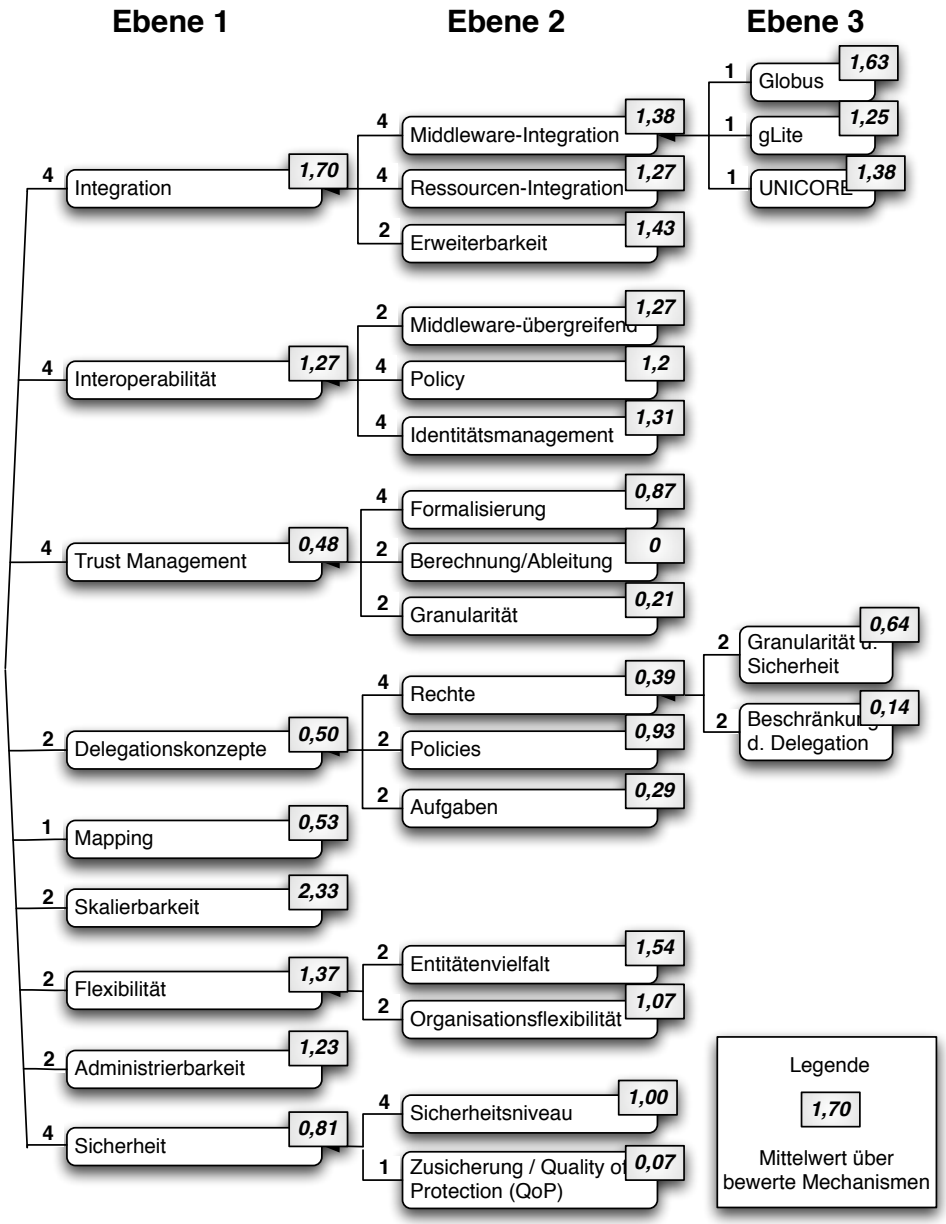

Abbildung 4.62: Mittelwert der Kriterienbewertung im Kriterienkatalog

schlechtesten ab.

4.10.2 Defizite und Schwachstellen

Nichterfüllung bei Trust Management, Delegation und Mapping

Die Sicherheitsmechanismen schneiden bei der Bewertung der Trust Management Konzepte am schlechtesten ab. Hier ergibt sich mit einem Mittelwert von 0,48 eine Bewertung, die einer Nichterfüllung des Kriteriums entspricht. Es folgen die Bewertungen zu den Delegationskonzepten und zum Mapping deren Mittelwerte sehr nah bei denen des Trust Managements liegen. Insofern kann man auch hier von Nichterfüllung der Kriterien ausgehen. Auch die Be-

wertungen zur Sicherheit der Mechanismen selbst liegt mit einem Mittelwert über alle Mechanismen noch unterhalb von 1 (vgl. Abbildung 4.62).

Trust Management

Das Trust Management stellt nach der Gruppierung und Hierarchie der Dienstklassen (vgl. Abschnitt 2.3) einen absoluten Basisdienst dar, der von vielen anderen Mechanismen genutzt oder als gegeben vorausgesetzt wird. Die Sicherheitsmechanismen selbst müssen Konzepte des Trust Managements unterstützen oder umsetzen. Dementsprechend ist das Merkmal Trust Management auch ein Bewertungskriterium für Sicherheitsmechanismen. Das Kriterium setzt sich aus den drei Teilkriterien Formalisierung, Berechnung und Ableitung von Vertrauenswerten sowie der Granularität des Trust Level Managements zusammen. Bei allen untersuchten Mechanismen liegt bestenfalls ein implizites Vertrauensmodell mit binär repräsentierten Vertrauenswerten zugrunde. Diese gelten i.d.R. global für alle Anwendungsfälle und Dienste. Konzepte zur dynamischen Berechnung oder Ableitung von Vertrauenswerten und feingranularere Vertrauenswerte werden von keinem Mechanismus unterstützt. Das Trust Level Management erfolgt global auf Ebene der VO oder bestenfalls auf Ebene der einzelnen Organisationen. Eine individuelle Festlegung von Vertrauenswerten auf Nutzer- oder Dienstebene ist ebensowenig vorgesehen wie die Realisierung von Attribute Release Policies (ARPs) oder Attribute Acceptance Policies (AAPs).

nur implizite Vertrauensmodelle

binäre Vertrauenswerte

keine dynamische Anpassung

Delegationskonzepte

Obwohl die Delegation ein fundamentales Konzept in Grids ist, fällt die Bewertung des Kriteriums trotzdem schlecht aus. Das Kriterium setzt sich auch den drei Teilkriterien Rechtedelegation, sowie Delegation von Policies und Aufgaben zusammen. Bei der Rechtedelegation sind mit Granularität und Sicherheit sowie der Möglichkeit zur Beschränkung der delegierten Rechte zwei weitere Teilkriterien zu bewerten. Hier liegt auch eines der Hauptprobleme heutiger Grids. Die Rechtedelegation erfolgt bei sehr vielen Mechanismen in der Form der Impersonation, d.h. dass bspw. ein Job alle Rechte desjenigen erbt, der ihn erzeugt hat. Der Job tritt quasi als Stellvertreter des Nutzers auf. Auf diese Art lässt sich natürlich kein Need-to-Know Prinzip umsetzen. Bei den Sicherheitsmechanismen gibt es einige, die überhaupt keine Delegation von Rechten zulassen. Alle diese Mechanismen gehen, zumindest implizit, von der Prämisse der lokalen Entscheidung als letzter Instanz aus. Dies ist besonders schmerzlich, wenn versucht werden soll echt föderierte Sicherheitsmechanismen umzusetzen. Noch schlechter schneiden die Mechanismen bei der Beschränkung einmal delegierter Rechte ab. Für nahezu alle bewerteten Sicherheitsmechanismen gibt es keine Möglichkeit ein Recht nach der Delegation zu widerrufen. Die Mechanismen definieren zwar eine Lebensdauer eines Rechts, d.h. ein delegiertes Recht wird nach einer gewissen Zeit ungültig,

Granularität bei Rechtedelegation gering

Widerruf delegierter Rechte kaum möglich

aber es gibt keine Konzepte zum vorzeitigen Widerruf. Noch schlechter fällt die Bewertung der Delegation von Aufgaben bei den Sicherheitsmechanismen aus. Bei den Mechanismen ist in der Regel nicht vorgesehen, dass eine Aufgabe, den Sicherheitsmechanismus betreffend, an eine andere Entität übertragen wird. Etwas besser schneidet das Kriterium der Delegation von Policies ab, dies liegt daran, dass die Sicherheitsmechanismen bei der Kommunikation (vgl. Abschnitte 4.4.1 und 4.4.3) und auch GridShib (vgl. Abschnitt 4.3.4) Verfahren zur formalen Beschreibung von Policies und Konzepte zum institutionellen Austausch beinhalten.

Delegation von Sicherheitsaufgaben kaum vorgesehen

Mapping

Die Bewertung des Mapping berücksichtigt die echte kooperative Zusammenarbeit autonomer, gleichberechtigter Organisationen, die naturgemäß jeweils eigene Policies und eigene Rechtesysteme betreiben bzw. für ihre eigenen Mitarbeiter selbst Rechte und Pflichten festlegen wollen und müssen.

Bei einer kooperativen Zusammenarbeit unter Gleichberechtigten besteht der Bedarf, Policies zwischen Quell- und Ziel-Domäne abzustimmen und ggf. auftretende Konflikte zu lösen. Die hierfür nötigen Verfahren sind in den Sicherheitsmechanismen nicht vorhanden oder es gibt keine Möglichkeit in beiden Domänen entsprechende Rechte und Policies zu spezifizieren. Die meisten Projekte in der Praxis lösen dieses Problem durch den Verzicht auf Gleichberechtigung zwischen den Partnern.
Die VO wird als sehr starke zentrale Entscheidungs- und Regelungsinstanz etabliert, die für alle Beteiligten zentral Policies vorgibt. Diese müssen dann von allen Partnern akzeptiert werden. Bei den Rechten gibt es auch den Fall, dass die VO für alle Beteiligten zentral die Rechte vergibt, oder es wird die implizite Prämisse der „letzten Entscheidung“ durch die Ziel-Domäne zugrunde gelegt. In diesem Fall hat immer die Ziel-Domäne, in der die Ressourcen genutzt werden, das Recht die eigenen Policies durchzusetzen und lokale Rechte zu spezifizieren und auch umzusetzen. Die Quell-Domäne muss sich damit der Ziel-Domäne oder einer „allmächtigen“ VO unterordnen.

Verzicht auf Gleichberechtigung zwischen Partnern

Bei keiner dieser Alternativen kommt es jedoch zu einer interorganisationalen Kooperation mit der Möglichkeit der dynamischen Abstimmung von Policies und Rechten zwischen zwei oder mehereren autonomen Organisationseinheiten.

Keine dynamische Abstimmung von Policies u. Rechten

Sicherheit

Die Bewertung für die Sicherheit setzt sich aus dem Sicherheitsniveau, das der entsprechenden Mechanismus erreicht, und den Möglichkeiten der Zusicherung eines bestimmten Sicherheitsniveaus durch QoP-Werte zusammen.

Das Sicherheitsniveau bestimmt sich in Abhängigkeit von der Schadenshöhe und der Eintrittswahrscheinlichkeit eines Schadens (vgl. Abbildung 3.3 auf Seite 50). Wenn eine große Schadenshöhe angenommen werden muss, kann

selbst bei einer niedrigen Eintrittswahrscheinlichkeit nur eine Bewertung von 1 erreicht werden. Bereits bei mittlerer Eintrittswahrscheinlichkeit ist das Sicherheitsniveau mit 0 zu bewerten. Grid-Systeme sollen möglichst transparent die Verteilung von Ressourcen übernehmen, einen nahtlosen Zugang zu Systemen und Ressourcen ermöglichen, ein Single Sign On realisieren und die einfache Delegation von Aufgaben und Rechten realisieren. Alle diese gewünschten Eigenschaften führen bei der Umsetzung von Sicherheitsmechanismen oft dazu, dass keine abgestuften Sicherheitszonen innerhalb des Grids realisiert werden. Bestenfalls können lokale Ressourcen-Provider eigene verschärfte Sicherheitsmechanismen implementieren. Die fehlenden Sicherheitszonen führen auch dazu, dass einem möglichen Angreifer, der in der Lage ist an einer Stelle unberechtigten Zugang zum Grid zu erlangen, nahezu die gesamte Grid-Infrastruktur offen liegt und er diese ausgehend von der einen Schwachstelle missbrauchen kann.

fehlende Sicherheitszonen führen zu hohen Schäden

In diesen Fällen muss, unter der Annahme von großen Grid-Infrastrukturen und teuren Ressourcen, die in der Praxis eher den Regelfall darstellen, auch eine große Schadenshöhe angenommen werden.

Keine QoP

Keiner der Mechanismen unterstützt formale Verfahren, um einem Nutzer die Möglichkeit zu geben sich über das Sicherheitsniveau und die Umsetzung des Sicherheitsdienstes in einer Ziel-Domäne zu informieren. Assurance Level oder QoP Parameter werden nicht zur Verfügung gestellt.

Allgemeine Defizite

Generell lässt sich noch feststellen, dass eine saubere Trennung der Mechanismen anhand der Sicherheitsanforderungen häufig nicht umgesetzt wird. Es gibt Mechanismen, die versuchen mehrere Sicherheitsdienste gleichzeitig innerhalb eines Sicherheitsmechanismus zu realisieren. Dies führt einerseits dazu, dass Sicherheitsanforderungen nicht mehr einzeln umgesetzt werden können, andererseits kann diese Vorgehensweise einem sauberen Sicherheitsdesign widersprechen.

Mechanismen implementieren mehrere Sicherheitsdienste

Bisher zeichnen sich die Organisationsformen im Grid, die Bildung, Pflege sowie die Administration von VOs durch relativ statische Prozesse und Strukturen aus. Man spricht hier auch von statischen und langlebigen VOs. Die Arbeiten in den Forschergruppen von Dreo, Stiller und Hegering [Schi 07, DHS 06, MW 06] gehen jedoch davon aus, dass statische VOs kein adäquates Mittel zur Bildung aller Arten von Grids sind. Erst die zunehmende Dynamik und deren Unterstützung durch **Dynamische Virtuelle Organisationen (DVO)** ermöglicht eine breitere Anwendung von Grid Technologien. Erst wenn die Grid-Technologien in der Lage sind, DVOs zu unterstützen, werden hochdynamische, mobile oder sich ad-hoc bildende Kooperationsformen überhaupt in die Lage versetzt Grid Technologien für ihre Kooperation zu nutzen.

Dynamik nicht ausreichend unterstützt

Existierende VO-Managementsysteme sind nicht in der Lage mit einer hohen Dynamik umzugehen, laufend und schnell neue Benutzer und Ressourcen in

die VO aufzunehmen oder wieder zur löschen.

Auch bei der Unterstützung dezentraler Konzepte zeigen existierende Grid-Systeme Defizite. Grids gehen häufig von einer logisch zentralen VO-Administration oder einem zentralen Gruppenmanagement mit zentraler Rechtevergabe aus. Dies widerspricht aber häufig der Projektrealität innerhalb der Forscher- und Nutzergruppen, die das Grid als Werkzeug verwenden wollen. Insbesondere beim Gruppenmanagement gilt hier in der Praxis häufig das Lokalitätsprinzip. Das heißt, innerhalb einer VO bilden sich aufgabenbezogene Gruppen und nutzen Grid-Ressourcen zur Erfüllung ihrer Aufgaben und zur Lösung ihrer Ziele. Die Gruppenmitglieder wissen in der Regel am besten, wer in ihre Gruppe aufgenommen werden soll und welche Rechte das neue Gruppenmitglied erhalten soll. Natürlicherweise sind sie auch am besten geeignet, um diese Aufgaben des Gruppenmanagements durchzuführen. Die bestehenden Mechanismen lassen ein dezentrales Konzept für ein dynamisches Gruppenmanagement nicht zu (vgl. auch Abschnitt 4.3.5).

Dezentrale Konzepte zum Gruppenmanagement fehlen

Beim Policy Management und der Unterstützung formalisierter Policies zeigen viele Mechanismen deutliche Schwächen (vgl. Abbildung 4.50). Eine Interoperabilität von Policies oder deren Delegation oder eine Konfliktlösung wird von den Mechanismen kaum unterstützt. Es gibt in derzeitigen Grids strategische (im wesentlichen CA-Policies und AUPs) und technische Policies (für die Konfiguration bestimmter Mechanismen), aber für den Policy-Raum zwischen diesen beiden Extremen existieren i.d.R. keine Policies. Auch bei den strategischen Policies wird von einer häufig sehr zentralistischen Struktur ausgegangen, d.h. eine Organisation oder eine VO gibt eine Policy vor und alle anderen müssen sich an diese Policy halten. Die Berücksichtigung lokaler Gegebenheiten oder anderer Richtlinien ist nicht vorgesehen. Der Abstimmungsprozess für AUPs ist deshalb auch sehr aufwändig und lang. Die Policies sind sehr statisch und damit ist man auch nicht in der Lage schnell auf irgendwelche Änderungen zu reagieren. Die in diesem Bereich vorherrschende technische Unterstützung heißt Papier mit entsprechenden Unterschriften (vgl. auch Abschnitt 4.8.1).

Schwächen beim Policy Management

Fehlende Sicherheitsdienste und -mechanismen

In dieser Arbeit wurde eine Anforderungsanalyse bezüglich der erforderlichen Sicherheitsdienste und -mechanismen im Grid durchgeführt (vgl. Abschnitt 2). Es wurde auch untersucht inwieweit diese erforderlichen Sicherheitsdienste voneinander abhängen (vgl. Abschnitt 2.3). Bei der Anwendung des in Kapitel 3.2 entwickelten Kriterienkataloges zeigte sich jedoch, dass Sicherheitsdienste nicht durch entsprechende Mechanismen unterstützt werden oder das bestehende Mechanismen den Anwendungsfall Grid nicht ausreichend unterstützen. Diese Ergebnisse sind neu und für die Grid Community von hoher theoretischer und praktischer Bedeutung. Im folgenden werden diese Defizite nochmal kurz zusammengefasst:

1. Für den Schutz der Integrität gespeicherter oder auf fremden Systemen

Datenintegrität fehlt

verarbeitete Daten gibt es keine adäquaten Mechanismen (vgl. Abschnitt 4.4.1).

Verbindlichkeit fehlt

2. Derzeit gibt es keine Mechanismen, mit denen sich Verbindlichkeit realisieren ließe (vgl. Abschnitt 4.4.2). Damit kann ein Leugnen einer Aktion oder eines Ereignisses nicht verhindert werden.

Mechanismen zur Unverkettbarkeit werden nicht eingesetzt

3. Mit Hilfe eines Dienstes der Unverkettbarkeit realisiert wäre es möglich Verkehrsflussanalysen zu verhindern. Für diesen Dienst existieren grundsätzlich zwar Mechanismen, diese werden aber im Grid nicht eingesetzt, weil diese Mechanismen zu Verzögerungen und einer Verschlechterung der Antwortzeit führen (vgl. Abschnitt 4.5.2).

Keine Sicherheitsalarme

4. Es gibt kein Alarming bei den Sicherheitsdiensten (vgl. Abschnitt 4.7.3).

Keine Festlegung von Sicherheitsprozessen und Auditverfahren

5. Im Grid-Umfeld existieren derzeit weder Beschreibungen oder Festlegungen, die vergleichbar mit ISO/IEC 270001 oder IT-Grundschutz, Sicherheitsverfahren oder Prozesse festlegen noch interne oder externe Auditierungs- oder Zertifizierungsrichtlinien. Eine zweifelsfreie, normierte Überprüfung der implementierten Sicherheitsprozesse, -dienste oder -mechanismen ist derzeit nicht möglich (vgl. Abschnitt 4.7.4).

Offene Fragestellung: Kooperative Frühwarnsysteme

Die Analyse und Erkenntnisse aus bestehenden Grid-Projekten zeigen aber auch, dass es noch grundlegendere Defizite bei der Sicherheit in Grids gibt. Es zeigt sich, dass für Sicherheitsdienste, die notwendig, sinnvoll und wichtig wären, bisher weder Mechanismen im Grid existieren, noch diese bisher überhaupt tiefer in der Literatur diskutiert wurden.

Viele Organisationen, die an einem Grid teilnehmen, betreiben lokale Sicherheitswerkzeuge wie Firewalls, Sicherheitsmonitoringsysteme, Intrusion Detection- (IDS) und Intrusion Prevention Systeme (IPS) und lokale Überwachungssysteme und -werkzeuge. Damit versucht jede Organisation seine lokale Infrastruktur möglichst optimal zu sichern, Angriffe möglichst frühzeitig zu erkennen und sich möglichst effizient gegen diese zu schützen.

Durch die Autonomie der beteiligten Organisationen liegt der Fokus der Gefahrenabwehr somit bisher üblicherweise auf lokalen Ressourcen. Eine globale Sichtweise fehlt; keine der Organisationen fühlt sich verantwortlich für die Gefahrenabwehr des interorganisationalen Gesamtsystems. Auch hierdurch entstehen wieder Redundanz und zusätzliche Kosten ohne die Vorteile und möglichen Synergieeffekte einer föderativen Gefahrenabwehr zu nutzen.

Föderative Gefahrenabwehr wird derzeit überhaupt nicht betrachtet und es gibt auch keine Mechanismen für die Koppelung lokaler Gefahrenabwehr- und Frühwarnsysteme

4.10.3 Zusammenfassung

Im Kapitel 2 wurde eine Hierarchie der Sicherheitsdienste vorgestellt. Die dort präsentierte Struktur und die Abbildung 2.3 wird an dieser Stelle nochmal aufgegriffen, um die Bewertung der Mechanismen zu visualisieren.

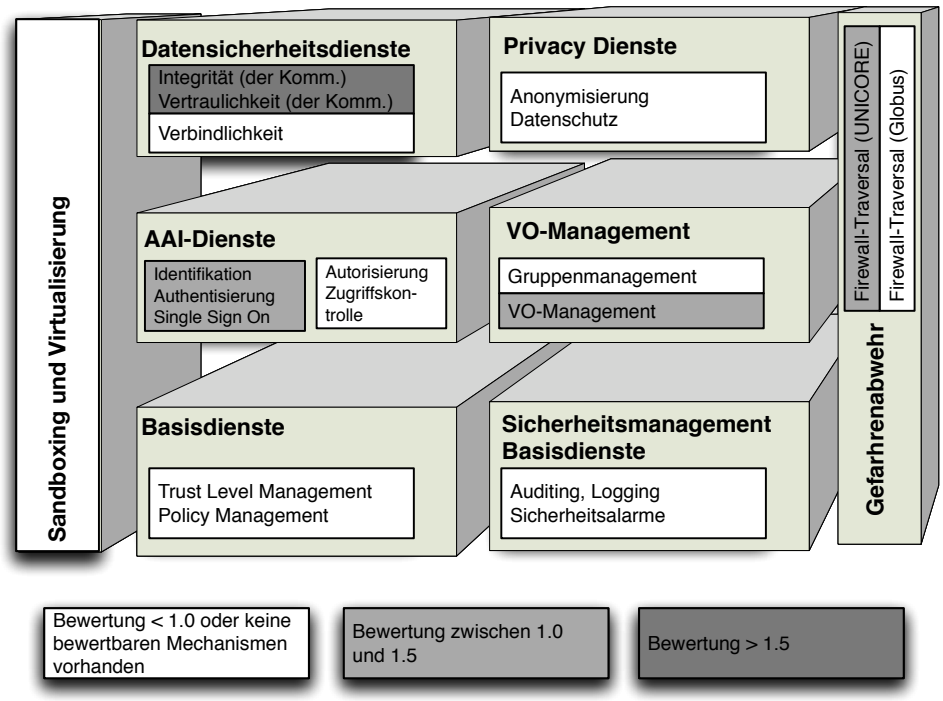

Abbildung 4.63: Hierarchie der Sicherheitsdienste; Defizite und Schwachstellen

Abbildung 4.63 stellt die Bewertungen der Dienste graphisch dar. Die Dienste werden verschieden gefärbt und je dunkler die Farbe, desto besser die Bewertung, bzw. je heller, desto schlechter. Obwohl die Bewertungskala von 0 bis 3 definiert ist, werden aufgrund der Ergebnisse der Bewertung in diesem Kapitel die Intervalle für die (Bewertungs-) Farben nicht äquidistant aufgeteilt, denn insgesamt überwiegen die schlechteren Bewertungen.

Kapitel 5

Spezifikation zusätzlicher Komponenten

Inhaltsverzeichnis

In Abschnitt 4.10.2 wurden Defizite und Schwachstellen existierender Sicherheitsmechanismen identifiziert. Hier hat sich gezeigt, dass die Bewertungen eher schlecht ausfallen und die Sicherheitsdienste, im Hinblick auf den Kriterienkatalog, viele Defizite aufweisen. Alle Sicherheitsdienste, die in der Abbildung 4.63 weiß eingefärbt sind, erfüllen die Kriterien noch nicht einmal teilweise (Bewertungszahl < 1). Das heißt, bei allen diesen Diensten besteht erhebliches Potential für Verbesserungen. In diesem Abschnitt werden für Probleme des Trust Managements, genauer für die dynamische Berechnung von Trust Levels, sowie des dezentralen und dynamischen Gruppenmanagement mit Implanted Chain Certificates neue Lösungen vorgeschlagen (vgl. Abbildung 5.1).

Obwohl das Trust Management ein absolut unverzichtbares Basiskonzept darstellt und von allen anderen Sicherheitsmechanismen verwendet und vorausgesetzt wird, ist die Realisierung in heutigen Grid-Umgebungen unzureichend. Eine Formalisierung und Festlegung abgestufter Vertrauenswerte ist nicht gegeben. Beim Trust Level Management wird von keinem Mechanismus die Ableitung oder dynamische Berechnung von Vertrauenswerten unterstützt (Mittelwert der Bewertungen ist 0, vgl. Abbildung 4.62). Deshalb wird im

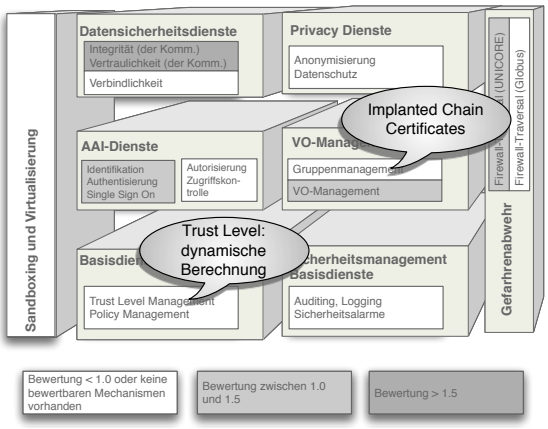

Abbildung 5.1: Einordnung der zusätzlichen Komponenten in die Hierarchie der Sicherheitsdienste

Abschnitt 5.1 ein neuartiges Konzept zum Trust Level Management und zur dynamischen Ableitung von Vertrauenswerten vorgestellt, das ursprünglich für Szenarien des föderierten Identitätsmanagements (Federated Identity Management, FIM) entwickelt wurde [BoRe 07], aber auch sehr gut für Grids anwendbar ist.

Die im vorherigen Kapitel betrachteten Mechanismen zum VO- oder Gruppenmanagement gehen von einer logisch zentralen Sicht auf die VO aus, d.h. sie implizieren einen zentralen „VO-Administrator“, der berechtigt und in der Lage ist, neue Mitglieder in die VO aufzunehmen oder neue Gruppen zu bilden. Im praktischen Betrieb zeigt sich jedoch häufig der Fall, dass eine lokale Gruppe einen ihnen bereits bekannten Mitarbeiter mit in die Gruppe aufnehmen möchte, da dieser im entsprechenden eng begrenzten Projekt mitarbeiten soll. Dies ist aber ohne den zentrale VO-Administrator nicht möglich. Kapitel 5.2.2 stellt ein vom Autor dieser Arbeit neu entwickeltes Konzept vor, das auch eine dezentrale und in gewissem Grad autonome Gruppenverwaltung möglich macht. Außerdem wird mit dem Implanted Chain Certificate (IChC) in Abschnitt 5.2.2 ein Verfahren vorgeschlagen, mit dem sich lange Zertifikatsketten, wie sie bei der Delegation von Aufgaben im Grid entstehen können, deutlich effizienter verifizieren lassen als dies bisher möglich war.

5.1 Trust Level Management; dynamische Berechnung

Das Trust Level Management in Grids erfolgt derzeit allenfalls implizit über Trusted Third Parties oder über bilaterale bzw. multilaterale Verträge. Die Partner vertrauen bspw. alle einer zentralen Komponente wie einer CA oder einem VOMS Server in dem Sinne, dass die Aussagen, die von dieser Kom-

ponente gemacht werden, auch als richtig angenommen werden. Zum Teil geht dieses implizite Vertrauen so weit, dass Unternehmens-kritische Entscheidungen, wie bspw. die Rechtevergabe, an diese zentralen Komponenten übertragen werden. Diese impliziten Vertrauensverhältnisse werden allenfalls über Verträge, die Schadensersatz-Klauseln beinhalten, abgesichert.

Grid-Teilnehmer können Wert des Vertrauens nicht quantifizieren

Dies führt dazu, dass die Partner in einem Grid nicht in der Lage sind den Wert des Vertrauens, den sie einem bestimmten Partner entgegenbringen, zu quantifizieren. Häufig sind sie auch nicht in der Lage bei zwei oder mehr Partnern zu sagen, welchem dieser Partner sie mehr bzw. am meisten und welchem sie weniger bzw. am wenigsten vertrauen. Allen Partner innerhalb einer VO wird üblicherweise das gleiche Vertrauen entgegengebracht. Wer in die VO aufgenommen wird, für den gilt das allgemein innerhalb der VO angenommene Vertrauensverhältnis. Es existieren i.d.R. auch keine abgestuften Konzepte, um Fehlverhalten zu sanktionieren.

gleiches Vertrauen zu allen Partnern

Ein Ziel des Trust Level Managements muss es sein, genau dieses abgestufte Vertrauen explizit zu machen und damit auch als Basis für Entscheidungen, z.B. über eine Dienstnutzung oder den Zugriff auf bestimmte Daten, zu nutzen. Im Abschnitt 5.1.1 wird ein hier entwickeltes Konzept für Trust Level und deren Repräsentation bzw. Verwendung in Grids vorgestellt.

Gibt es formalisierte Werte für Vertrauen, ist ein weiteres Ziel, diese dynamisch anpassbar zu machen und auch für Partner, die der Bewertende selbst nicht kennt, Vertrauenswerte ableiten zu können. Dazu wird im Abschnitt 5.1.2 ein Reputationssystem und ein Algorithmus vorgestellt, mit Hilfe dessen über Vertrauenswerte bekannter Partner auch ein Vertrauenswert für einen nicht unmittelbar bekannten Partner abgeleitet werden kann.

5.1.1 Vertrauen und die Repräsentation im Grid

Eine Vertrauensbeziehung in heutigen Grid-Infrastrukturen beschränkt sich auf die institutionelle Ebene, d.h. eine Organisation hat eine implizite Vertrauensbeziehung zu einer Partnerorganisation. Dieses Vertrauen gilt dann implizit für alle Beziehungen zu diesem Partner und für alle seine Mitarbeiter, Dienste und Ressourcen. In der Praxis zeigt sich jedoch, dass weitere Vertrauensbeziehungen existieren, eine Repräsentation erfordern und sich diese auch inhaltlich von denen auf institutioneller Ebene unterscheiden können.

Ein Trust Management im Grid muss in der Lage sein, alle geforderten Vertrauensbeziehungen auch abbilden zu können, d.h. für jede Entität im Grid muss es möglich sein einen Vertrauenswert festzulegen. Entitäten in diesem Sinn sind:

Vertrauenswerte für alle Entitäten im Grid

- Virtuelle Organisationen (VOs)
- Institutionen und Organisationen
- Personen
- Ressourcen

Diese Entitäten lassen sich im Grid über Authentisierungsmechanismen zweifelsfrei identifizieren und damit kann ihnen ein konkreter Vertrauenswert zugeordnet werden. Eine Vertrauensbeziehung zwischen zwei Entitäten A und B wird durch einen **Vertrauenswert (Trust Level)** t beschrieben. Dabei bezeichnet t_{AB} den Vertrauenswert, den A B entgegenbringt.

Die Festlegung eines einzigen Vertrauenswertes pro Entität ist jedoch häufig nicht ausreichend. Aus diesem Grund werden Dienst- bzw. Szenarioabhängige Vertrauenswerte benötigt. Damit lassen sich unterschiedliche Vertrauensverhältnisse zur selben Entität darstellen. Ein Beispiel verdeutlicht diese notwendige Unterscheidung. Für eine Nutzung von „normalen“ Ressourcen und Diensten wird ein Dienstleister einen niedrigeren Vertrauenswert akzeptieren als für Dienste, bei deren Nutzung externe Kosten für den Dienstleister anfallen (wie z.B. kostenpflichtige Datenbanken, o.ä.).

$t_{AB}(s)$ Vertrauen von A gegenüber B im Hinblick auf Dienst s

Um dies im Trust Management abzubilden, wird ein parametrisierter Vertrauenswert $t_{AB}(s)$ eingeführt, der den Vertrauenswert von A gegenüber B im Hinblick auf die Situation oder den Dienst s beschreibt.

Skala für Vertrauenswerte sollte nicht zu groß sein

Vertrauensskala: 1 = Misstrauen, 4 = ultimatives Vertrauen

Der Vertrauenswert selbst wird auf unterschiedlichen Skalen durch Zahlen oder durch Beschreibungen spezifiziert. In PGP (Pretty Good Privacy) [Zimm 94a, Zimm 94b, AR 97, Sv 02, WWW 06] bspw. gibt es die Vertrauenswerte *unknown*, *untrusted*, *marginally trusted*, *completely trusted* und *ultimately trusted*. Goldberg [GPH 03] klassifiziert Trust mit Werten zwischen 1 und 9, wobei 1 für absolutes Misstrauen und 9 für absolutes Vertrauen steht. Die Größe dieser Skala ist jedoch für Einsatzzwecke, in denen verschiedene Nutzer die Klassifizierung vornehmen, aus unserer Sicht zu breit. Eine echte Vergleichbarkeit zwischen verschiedenen Klassifizierungen ist damit nicht mehr gegeben. Aus diesem Grund wird in dieser Arbeit die Skala auf den Wertebereich von 1 bis 4 beschränkt und die Wertebelegung in Anlehnung an PGP verwendet. Dabei steht 1 für Misstrauen und 4 für ultimatives Vertrauen.

Diese Vertrauenswerte lassen sich global, lokal oder individuell mit Hilfe eines Verzeichnisdienstes an die Identität einer Entität binden. Auf diese Weise lässt sich ein statisches Trust Repository aufbauen, in dem dienstabhängig alle Entitäten berücksichtigt werden können, zu denen eine direkte Vertrauensbeziehung besteht.

5.1.2 Rekursiver Trust Algorithmus

Für die Ableitung von indirekten Vertrauenswerten sind statische Trust Repositories nicht geeignet. Sie sind auch nicht in der Lage mit hoher Dynamik und mittelbarem Vertrauen oder Reputationssystemen umgehen zu können. Aus diesem Grund wurde ein rekursiver Trust Algorithmus zur Ableitung von mittelbaren Vertrauenswerten entwickelt.

Vertrauensbeziehung als Vertrauensgraph

Vertrauensbeziehungen zwischen Entitäten lassen sich neben der Repräsentation in Verzeichnisdiensten auch noch allgemeingültiger in Form von Vertrauensgraphen (Trust Graphs) darstellen. Die Knoten des Trust Graph re-

präsentieren die Entitäten und eine gerichtete und bewertete Kante von A nach B repräsentiert den Vertrauenswert t_{AB}.

Der im folgenden vorgestellte Algorithmus arbeitet auf einem Trust Graph und ist in der Lage den Vertrauenswert zu einem Partner dynamisch zu berechnen. Dabei wird ein dienst- bzw. situationsbezogenes Reputationssystem implementiert, das auf Empfehlungen von Partner-Entitäten basiert. Für die Ableitung eines Vertrauenswertes von A gegenüber dem nicht unmittelbar bekannten B arbeitet der Algorithmus auf einem angepassten Trust Graphen, der Nachbarschaftsinformationen enthält (vgl. Abbildung 5.2). Dieser gerichtete azyklische Graph startet bei A und beinhaltet alle Nachbarn $j_1, \ldots, j_n$, zu denen A eine direkte Vertrauensbeziehung besitzt. Die Abbildung 5.3 stellt den rekursiven Algorithmus `TrustLevel(x,B,s)` graphisch dar, der einen Vertrauenswert von x gegenüber B für die Situation oder den Dienst s rekursiv ableitet. Um den Vertrauenswert von A gegenüber B abzuleiten, wird `TrustLevel(A,B,s)` aufgerufen und es werden folgende Schritte ausgeführt:

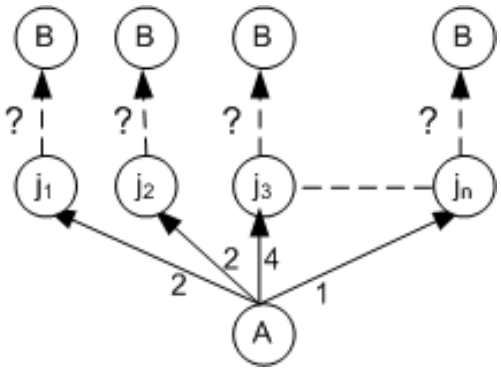

Abbildung 5.2: Trust Graph[BoRe 07]

- Zu Beginn wird der Algorithmus initialisiert und ggf. werden neue Situationen in einer entsprechenden Datenbank gespeichert.
- Danach wird überprüft, ob es zu B nicht bereits eine direkte Beziehung und damit einen Vertrauenswert gibt. In diesem Fall ist B Nachbar von A und der entsprechende Vertrauenswert kann verwendet werden.
- Falls es keine direkte Verbindung zwischen A und B gibt, wird der Vertrauenswert durch das gewichtete Mittel der Vertrauenswerte aller Nachbarn mit Vertrauensbeziehungen zu B berechnet. Der Vertrauenswert von A zu B für die Situation s aus der Menge S von Situationen lässt sich durch folgende Formel angeben:

$$t_{AB}(s) = \frac{\sum_{j=1}^{n} \begin{pmatrix} t_{jB} \cdot t_{Aj} & \text{if } t_{Aj} \geq t_{jB} \\ (t_{Aj}^2) & \text{if } t_{Aj} < t_{jB} \end{pmatrix}}{\sum_{j=1}^{n} t_{Aj}} \tag{5.1}$$

Dabei wird angenommen, A hat n Nachbarn mit Vertrauensbeziehungen zu B. Außerdem kann das mittelbare Vertrauen über einen Nachbarn j nicht größer sein als der eigene Vertrauenswert gegenüber j, d.h. falls $t_{Aj} < t_{jB}$ wird das Quadrat des Vertrauenswertes zum Nachbar als Summand verwendet. Im anderen Fall wird das mittelbare Vertrauen mit dem Vertrauen zum Nachbarn multipliziert. Dadurch und durch die Berechnung des gewichteten Mittels wird sichergestellt, dass keinem entfernten Partner mehr Vertrauen entgegengebracht wird als den direkten Nachbarn. Für den Fall, dass alle Nachbarn vollstes Vertrauen zu B haben, d.h. B den höchsten Vertrauenswert t_{max} gegeben haben, würde sich folgender Vertrauenswert ergeben:

$$t_{AB} = \frac{\sum_{j=1}^{n} t_{Aj}^2}{\sum_{j=1}^{n} t_{Aj}} = \frac{t_{A1}^2 + t_{A2}^2 + ... + t_{An}^2}{t_{A1} + t_{A2} + ... + t_{An}} = Ctn \text{ mit } t_{min} <= Ctn <= t_{max}$$

Das heißt, hier würde das Vertrauen zu B nicht größer sein als der Mittelwert aller Vertrauenswerte zu Nachbarn, die ihrerseits wieder eine Vertrauensbeziehung zu B haben.

Im umgekehrten Fall, d.h. wenn alle Nachbarn B völlig misstrauen würden und B mit dem Vertrauenswert t_{min} bewerten, würde sich t_{AB} wie folgt berechnen:

$$t_{AB} = \frac{\sum_{j=1}^{n} t_{min} \cdot t_{Aj}}{\sum_{j=1}^{n} t_{Aj}} = \frac{\sum_{j=1}^{n} 1 \cdot t_{Aj}}{\sum_{j=1}^{n} t_{Aj}} = 1 = t_{min}$$

In diesem Fall würde das einheitliche Misstrauen der Nachbarn gegenüber B voll auf das mittelbare Vertrauen t_{AB} durchschlagen.

- Um diese Formel rekursiv zu berechnen, werden zu Beginn Hilfsvariablen initialisiert.
- Der Vertrauensgraph wird in einer Tiefensuche durchlaufen, d.h. wenn ein Nachbar j keine direkte Vertrauensbeziehung zu B hat, wird der Algorithmus rekursiv aufgerufen (`TrustLevel(j,B,s)`).
- Falls der Nachbar j eine direkte Vertrauensbeziehung zu B besitzt, wird überprüft, ob diese größer ist als die zum „empfehlenden“ Nachbarn. Abhängig vom Ergebnis des Vergleichs wird der entsprechende zu verwendende Teil der Formel (5.1) ausgewählt und berechnet.
- Solange es weitere Nachbarn gibt, wird auch für diese deren mittel- oder unmittelbarer Vertrauenswert abgefragt bzw. berechnet.
- Die Szenario-abhängigen Vertrauenswerte von A gegenüber B bzw. von x gegenüber B werden in einer Datenbank und im Vertrauensgraph gespeichert.

- Der Algorithmus und die Daten können auch verwendet werden, um ein durchschnittliches Vertrauen zu einem Partner B über alle Dienste bzw. Szenarien zu berechnen. Dazu werden die Vertrauenswerte bezüglich aller Situationen s aus der Menge möglicher Situationen S addiert und durch die Mächtigkeit der Menge S dividiert:

$$t_{\overline{AB}} = \lfloor \frac{\sum_{s=0}^{S} t_{AB}(s)}{|S|} \rfloor$$

Die Vertrauenswerte innerhalb des Graphen bzw. in der Datenbank sind mit Zeitstempeln versehen, die den Zeitpunkt der Berechnung bzw. Festlegung des entsprechenden Vertrauenswertes beinhalten. Damit ließe sich, bei entsprechenden Anforderungen, auch das „Alter“ des Vertrauenswertes mit in die Ableitung eines neuen Vertrauenswertes einbeziehen. Andererseits kann sich eine Vertrauensbeziehung jederzeit ändern. Eine Entität A, die ein mittelbares Vertrauen $t_{AB}(s)$ zu B berechnet hat, kann nach einigen Transaktionen mit B, im Hinblick auf den Dienst s, B auch zum direkten Nachbarn machen und $t_{AB}(s)$ entsprechend anpassen.

5.2 IChC: verteilte Autorisierung, verteiltes Gruppenmanagement, Rechtedelegation

In bestehenden Grid-Systemen ist die Authentisierung und die Autorisierung sehr stark vermischt. Für beide Zwecke werden EECs oder Proxy-Zertifikate verwendet. Auch wenn mit den Proxy–Zertifikaten eine mehrstufige Delegation möglich ist, wird am Ende trotzdem i.d.R der DN des Nutzers, der am Anfang der Proxy-Zertifikatskette steht, zur Autorisierung verwendet. In der Regel werden die Rechte während der Delegation auch nicht beschränkt. Entscheidend ist häufig, dass der Name bzw. die Informationen über die VO-Mitgliedschaft delegiert werden. Die Autorisierung erfolgt abhängig von diesen Informationen immer lokal.

Bereits relativ früh wurde auch gefordert, dass individuelle Nutzer in der Lage sein sollten, Gruppen zu einem bestimmten Zweck zu etablieren [HuTh 01] (vgl. auch Anforderungen an Grids in Abschnitt 2.2.2). Dies ist mit den in interorganisationalen Kooperationen eingesetzten Konzepten kaum umsetzbar.

Auch beim VO-Management und insbesondere beim Gruppen-Management werden im Regelfall logisch zentrale Konzepte eingesetzt. Eine zentrale Instanz („VO-Admin“) muss die Gruppenstruktur in Form eines komplexen azyklischen Graphen aufbauen und muss jedes VO-Mitglied seinen Gruppen und Subgruppen zuteilen. Ein dezentrales Management der Gruppen, ein verteilter oder föderativer Ansatz lässt sich damit nur sehr schwer realisieren. Obwohl

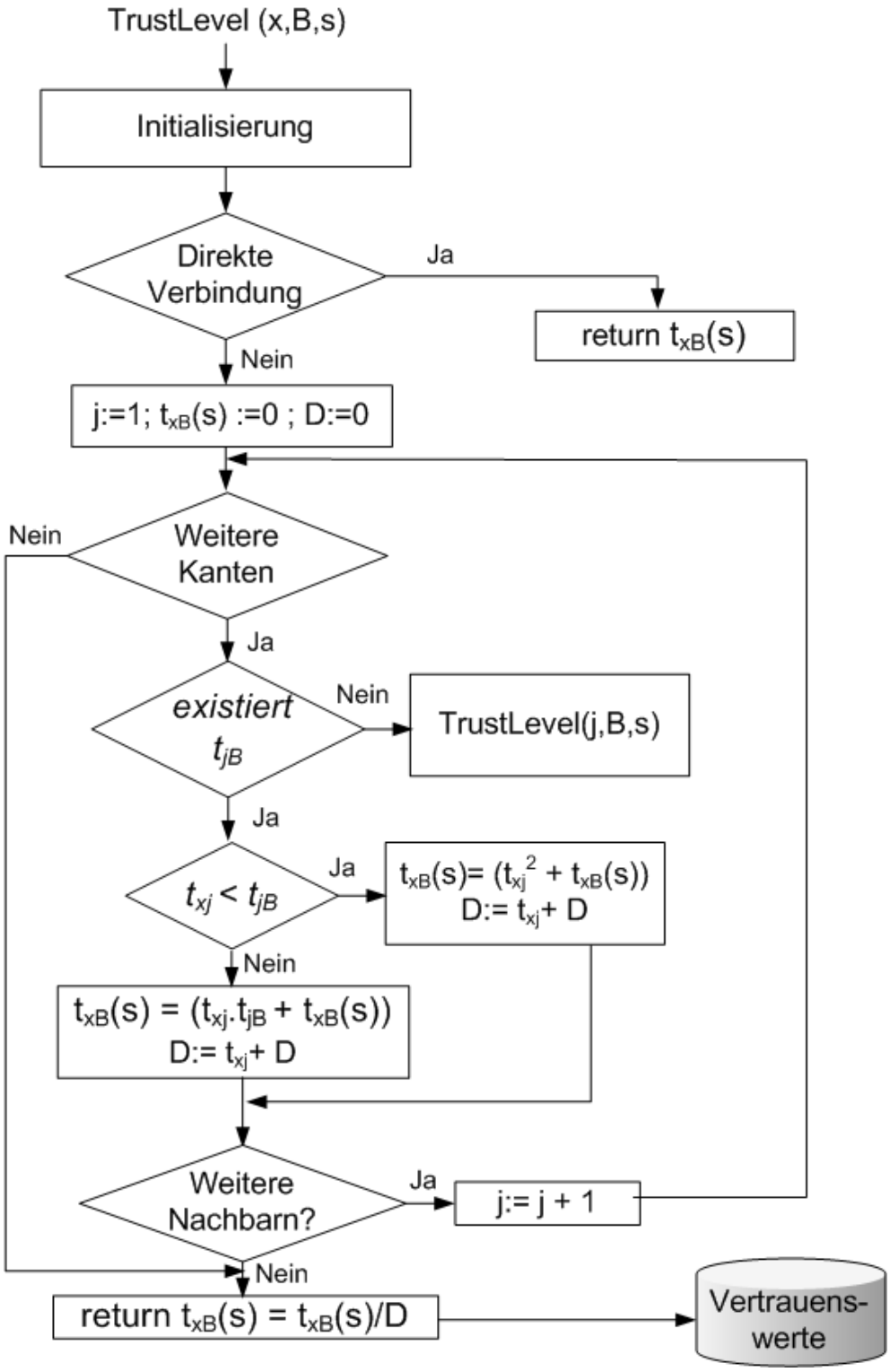

Abbildung 5.3: Rekursiver Trust Algorithmus nach [BoRe 07]

eine Gruppenmitgliedschaft häufig „Gruppen-nahe“ festgelegt werden kann, muss das zentrale VO-Management die Gruppenmitgliedschaft eines Nutzers festlegen. In der Praxis wäre der dezentrale Ansatz oft sicherer, effizienter, ef-

fektiver und vom Workflow der kooperativen Problembearbeitung schlüssiger. Eine lokale Gruppe, die eine bestimmte Aufgabe zu erfüllen hat, kann am besten bestimmen, wer der Gruppe angehören soll.

Gibt man die Prämisse der lokalen Autorisierung und der Vermischung von Authentisierung und Autorisierung auf, können effizientere Verfahren zur verteilten Autorisierung, zur echten Delegation von Rechten und für das verteilte Gruppenmanagement zum Einsatz kommen.

Im folgenden werden hierzu Implanted Chain Certificates (IChC) vorgestellt, die auf der Simple Public Key Infrastructure (SPKI) Theorie aufsetzen.

5.2.1 Verteiltes Gruppenmanagement mit SPKI

SPKI Theorie

Mit klassischen Zertifikaten wird durch die Signatur einer CA ein Name (DN) zweifelsfrei mit einem öffentlichen Schlüssel verknüpft. Dieser Ansatz ist identitätsbasiert und wurde primär zur zweifelsfreien Identifikation von Entitäten entwickelt. Die Theorie der **Simple Public Key Infrastructure (SPKI)** bricht ziemlich radikal mit diesem Ansatz [Elli 99, EFL+ 99]. Die Haupt-Motivation für die Entwicklung war, dass die SPKI-Zertifikate primär für die Autorisierung und nicht für die individuelle Authentisierung verwendet werden sollten. Dementsprechend enthält ein SPKI-Zertifikat keinerlei Identitätsinformationen. Ein SPKI-Zertifikat wird direkt für einen öffentlichen Schlüssel ausgestellt, d.h. ein öffentlicher Schlüssel wird vom Zertifizierenden mit seinem Private Key digital signiert. Zusätzlich dazu können noch Attribute mit in das Zertifikat aufgenommen werden. Auf diese Art kann ein Recht in Form eines SPKI-Zertifikates generiert werden. Derjenige, der durch eine Signatur nachweisen kann, dass er im Besitz des zum zertifizierten öffentlichen Schlüssel gehörenden privaten Schlüssels ist, kann damit das Recht nutzen.

keine Identitätsinformation im SPKI Zertifikat

dezentrales Gruppenmanagement

Ein weiteres Anwendungsfeld für SPKI-Zertifikate ist das dezentrale Gruppenmanagement [AuM 01]. Eine neue Gruppe wird durch ein neues Schlüsselpaar, dem so genannten **Group Key**, etabliert. Ein Nutzer erzeugt sich ein Schlüsselpaar und lässt seinen öffentlichen Schlüssel vom Besitzer des Group Key signieren und wird damit zum Mitglied der Gruppe. D.h., das SPKI-Zertifikat bescheinigt die Gruppenmitgliedschaft. Die Verifikation der Gruppenmitgliedschaft erfolgt mit Hilfe des öffentlichen Group Key, der innerhalb der Gruppe allgemein bekannt ist, da der öffentliche Schlüssel als Identifikator für die Gruppe verwendet wird und als solches Teil jedes Zertifikates ist.

Leader können neue Mitglieder aufnehmen

Um ein dezentrales Verfahren realisieren zu können, gibt es ausgezeichnete Mitglieder in der Gruppe, so genannte **Leader**. Ein Leader kann wie der Group Key Owner selbst neue Mitglieder in die Gruppe aufnehmen oder die Leader–Rolle an andere Gruppenmitglieder weitergeben. Ein Leader muss nicht im Besitz des Group Key sein, sondern erzeugt sich ein eigenes Schlüsselpaar. Der Group Key Owner oder ein anderer Leader signieren den entsprechenden öffentlichen Schlüssel mit der Zusatzinformation „Leader“.

Leader, die direkt vom Besitzer des Group Key zertifiziert sind, werden als Top Level Leader bezeichnet. Ein Leader oder auch ein normales Mitglied, das seine Mitgliedschaft belegen möchte, benötigt alle Zertifikate, die seinen Status innerhalb der Gruppe belegen. Die entsprechenden Zertifikate bilden eine Kette so genannter Delegationszertifikate, an deren Anfang ein Zertifikat stehen muss, das mit dem Group Key signiert wurde (vgl. Abb. 5.5). Der Nutzer kann mit seinem privaten Schlüssel und der **Zertifikatskette** seinen Status in der Gruppe belegen.

Zertifikatskette belegt Status

Um den Status zu verifizieren muss die gesamte Zertifikatskette verifiziert werden. Dies bedeutet:

klassische Verifikation

1. Es muss geprüft werden, ob die Zertifikatskette mit einem Zertifikat beginnt, das mit dem Group Key signiert wurde.
2. Für alle folgenden Zertifikate Z_i mit $i \in \{2, \ldots n\}$ ist zu prüfen, ob Z_i mit dem öffentlichen Schlüssel aus Z_{i-1} verifiziert werden kann.
3. Für jedes Zertifikat aus der Kette, außer dem letzten, muss geprüft werden, ob das entsprechende Mitglied die Rollenzugehörigkeit zur Leader Rolle besitzt und damit andere Leader ernennen bzw. Mitglieder aufnehmen durfte.
4. Die Gültigkeitsdauer der Kette muss als Schnittmenge aller Gültigkeitszeiträume der Zertifikate in der Kette berechnet werden:

   Sei $X^i = (X^i_{min}, X^i_{max})$, $1 \leq i \leq n$, die Gültigkeitszeiträume der Zertifikate, wobei X_{min} die untere Schranke (not before date) und X_{max} die obere Schranke (not-after date) der Gültigkeitsdauer eines Zertifikates Z_i bezeichnet. Die Schnittmenge der Daten berechnet sich dann wie folgt $V = \bigcap_{i=1..n} \{X^i\}$ mit $V_{min} = \max_{i=1..n}(X^i_{min})$, und $V_{max} = \min_{i=1..n}(X^i_{max})$. Falls $V_{min} > V_{max}$ gilt, ist die Schnittmenge leer und die Zertifikatskette ist nicht mehr gültig [EFL+ 99]. Falls ein Zertifikat in der Kette bereits abgelaufen sein sollte, müssen die nachfolgenden Zertifikate nicht weiter betrachtet werden.

Diese Verifikation von Zertifikatsketten werden wir im folgenden als „klassische Verifikation“ bezeichnen.

Bildung von Subgruppen

Innerhalb von Gruppen lassen sich auch Subgruppen bilden. Eine Subgruppe besteht aus ihren Mitgliedern, einem Subgruppen-Leader und einem Sub-Group Key. Die Verknüpfung zwischen Supergruppe und Subgruppe erfolgt durch ein Subgruppen–Zertifikat, das analog zu Leader– oder Mitglieder–Zertifikaten eine Gültigkeitsdauer, den Gruppen- und Subgruppen-Identifikator und eine Signatur beinhaltet. Alle Mitglieder einer Subgruppe sind automatisch Mitglieder der Supergruppe. Ein Supergruppen-Leader ist Leader in alle Subgruppen wohingegen ein Subgruppen-Leader nicht automatisch auch Leader in der Supergruppe wird [SM 00].

Im Gruppenmanagement bilden die Zertifikate mit einer Baumstruktur die Gruppen- und Mitgliederstruktur nach. Für praktische Anwendungsfälle des

Gruppenmanagements können die resultierenden Zertifikatsketten sehr lang werden. Der Aufwand für die klassische Verifikation von Zertifikatsketten ist, wegen der rechenintensiven kryptographischen Funktionen zur Verifikation aller Signaturen innerhalb einer Kette, sehr hoch. Um eine effizientere Verifikation bei gleichzeitigem Erhalt der Delegationsinformation zu erreichen, wurde das Konzept der Implanted Chain Certificates [HuRe 04] eingeführt, das im nächsten Abschnitt vorgestellt wird.

Aufwand für klassische Verifikation sehr hoch

5.2.2 Implanted Chain Certificates (IChC)

Die Grundidee eines **Implanted Chain Certificates (IChC)** [HuRe 04] ist die Zertifizierung der Korrektheit einer kompletten Zertifikatskette und die Sicherstellung ihrer Integrität. Man kann sich ein IChC als ein Zertifikat für eine ganze Kette von Zertifikaten vorstellen (vgl. Abbildung 5.4).

IChC zertifiziert Korrektheit einer ganzen Kette

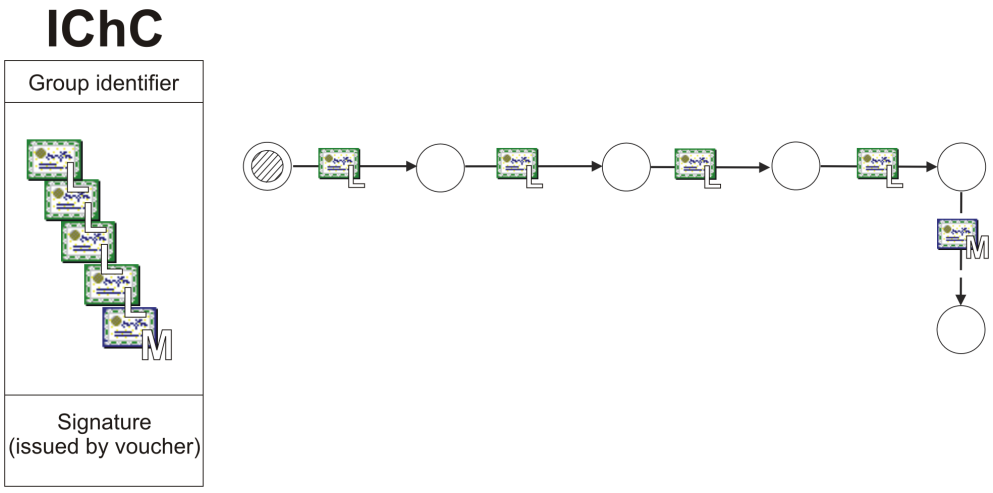

Abbildung 5.4: Struktur eines IChC

Mit der Einführung des IChC gibt es drei verschiedene Arten von Zertifikaten, die im interorganisationalen Kontext eingesetzt werden:

1. Identitätsbasierte Zertifikate, die einen öffentlichen Schlüssel an den DN des Eigentümers des Schlüsselpaares knüpfen.
2. Delegationszertifikate, welche die Delegation von Rechten zertifizieren.
3. IChC die ganze Zertifikatsketten zertifizieren.

Für die Ausstellung eines IChC wird eine neue Rolle eingeführt, die als **Voucher** bezeichnet wird. Zur Ausstellung von Voucher Zertifikaten wird ein eigenes Schlüsselpaar der **Common Voucher Key (CVK)** verwendet. Der CVK wird Initial vom Group Key Owner erzeugt. Ein normales Mitglied oder ein Leader in der Gruppe können Voucher werden. Dazu muss ein Schlüsselpaar generiert werden, das Voucher Zertifikat auf den öffentlichen Schlüssel ($Vouch_p$) ausgestellt und mit dem CVK signiert werden.

Voucher stellt IChC aus

IChC Struktur und Erzeugung

Der Voucher muss, bevor er ein IChC ausstellen kann, die Korrektheit und Integrität der zu zertifizierenden Kette prüfen, d.h. er muss eine klassische Verifikation, wie auf S. 218 beschrieben, durchführen. Danach wird die gesamte Kette als Inhalt dem IChC hinzugefügt und der Voucher kann das IChC mit seinem privaten Schlüssel signieren.

Eine Zertifikatskette wird wie folgt dargestellt:

$$Cert\_Chain = (cert1, sig1, cert2, sig2, ..., certN, sigN)$$

wobei $cert_i$ die Zertifikatsinformation ohne die entsprechende Signatur sig_i darstellt. Um ein IChC auszustellen wird ein Object Certificate ($certOC$) und eine entsprechende Signatur hinzugefügt:

$$IChC = (cert1, sig1, cert2, sig2, \ldots, certN, sigN, certOC, signOC)$$

mit

$$certOC = cert(hash(cert1, sig1, cert2, sig2, \ldots, certN, sigN), Vouch_P, \ldots)$$

$certOC$ beinhaltet einen Hash über die gesamte Zertifikatskette sowie den öffentlichen Schlüssel des Vouchers ($Vouch_P$).

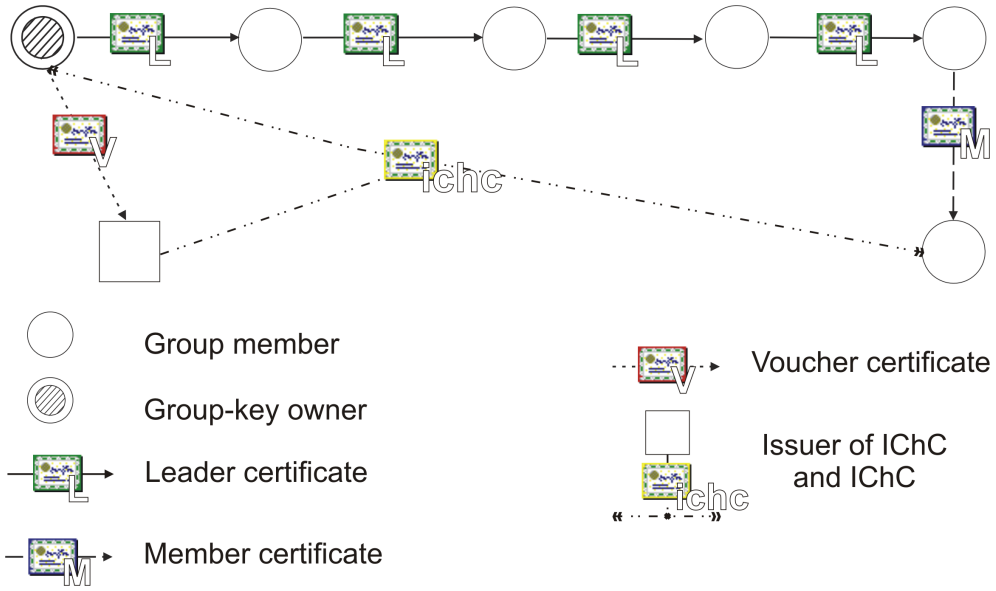

Abbildung 5.5: Beispiel für eine Gruppenstruktur und der entsprechenden SPKI-Zertifikate

5.2.3 Verifikation von IChC

Die Verifikation eines Implanted Chain Certificates ist ein fünfstufiger Prozess:

1. Der Verifikator muss überprüfen, ob der öffentliche Schlüssel desjenigen, der das IChC vorlegt, in der Zertifikatskette innerhalb des IChC

enthalten ist. Das Zertifikat, das diesen Schlüssel enthält, muss nicht am Ende der Kette stehen, sondern kann überall innerhalb der Zertifikatskette platziert sein.

2. Der Hash über die Zertifikatskette muss berechnet und mit dem Hash in $certOC$ verglichen werden. Sind beide gleich, ist die Integrität des IChC gewährleistet.
3. Die Signatur des IChC muss kryptographisch verifiziert werden.
4. Das Voucher Zertifikat muss verifiziert werden, um sicherzustellen, dass es sich um ein rechtmäßig erstelltes IChC handelt.
5. Zuletzt muss die Gültigkeitsdauer des IChC als Schnittmenge aller Gültigkeitszeiträume der Kette berechnet werden (vgl. Abschnitt 5.2.1, Seite 218).

IChC Verifikation deutlich effizienter

Ein Nutzer, der seine Gruppenmitgliedschaft belegen will, kann nun anstelle seiner Zertifikatskette ein IChC verwenden. Für die Verifikation des IChC sind nur noch drei teure kryptographische Operationen notwendig. Die Signatur des IChC ist zu überprüfen sowie das Voucher Zertifikat und ggf. der CVK sind zu verifizieren. Falls der CVK, analog zum Group Key, allgemein bekannt gemacht werden kann, kann auf die Verifikation des CVK verzichtet werden und der Aufwand reduziert sich auf zwei kryptographische Operationen.

IChC mehrfach verwendbar

Ein weiterer Vorteil des IChC ist dessen breite Verwendbarkeit. Ein IChC, ausgestellt für ein Mitglied an einem Blatt des Baumes, der die Gruppe repräsentiert, kann von allen Mitgliedern, die auf dem Pfad vom Group Key zu diesem Blatt liegen, verwendet werden. Es muss also nicht für jedes Mitglied ein eigenes IChC erstellt werden.

5.2.4 Sicherung der Voucher Infrastruktur

Ein Voucher Zertifikat ist ein Spezialfall eines normalen Mitgliedszertifikates, das mit dem CVK signiert wurde. Beim Aufbau einer Gruppe kann der CVK vom Group Key Owner generiert und es können auch gleich die ersten Voucher zertifiziert werden. Um die Flexibilität und Dezentralität des SPKI-Ansatzes beibehalten zu können, muss es Möglichkeiten geben, ohne den Group Key Owner neue Voucher zu ernennen. Ansonsten wäre der Group Key Owner eine zentrale Komponente und damit ein Single Point of Failure.

Voucher und CVK als kritische Komponenten

Grundsätzlich sollte ein Leader in der Lage sein einen Voucher zu ernennen. Das heißt, wenn ein neuer Voucher erzeugt werden soll, wird ein Schlüsselpaar generiert und der Voucher erhält vom Leader ein Zertifikat auf seinen öffentlichen Schlüssel sowie eine Zertifikatskette, welche die Mitgliedschaft des Vouchers in der Gruppe belegt. Zum Voucher wird das Mitglied aber erst durch die Signatur mit Hilfe des CVK.

Die naive Idee ist, den privaten Schlüssel des CVK an alle Voucher zu verteilen. Um seine Rolle als Voucher bestätigen zu lassen, muss der neue Voucher

einen bestehenden kontaktieren. Dieser stellt dann ein neues Voucher Zertifikat aus und übermittelt den privaten Schlüssel des CVK über einen gesicherten Kanal an den neuen Voucher. Danach wäre der neue Voucher in der Lage IChCs auszustellen und weitere neue Voucher zu zertifizieren. In Abbildung 5.6 a ist dieser Vorgang beispielhaft dargestellt. Der Leader erstellt ein neues Zertifikat für den Voucher V_x (1), der dann seinerseits den bereits existierenden Voucher V_e kontaktiert. V_e zertifiziert mit seinem privaten CVK den Voucher V_x (2). Danach kann V_x eigene IChCs ausstellen (3).

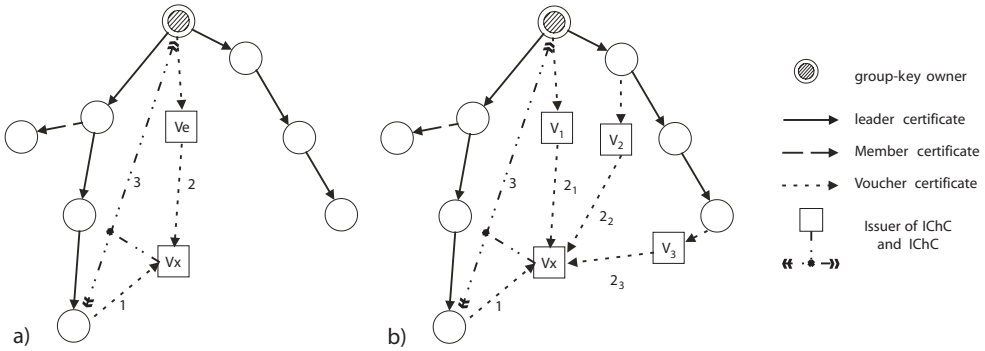

Abbildung 5.6: Ernennung eines neuen Voucher

Dieser naive Ansatz ist allerdings in höchstem Maße verwundbar. Aus sicherheitstechnischen Überlegungen ist der CVK die sensibelste Komponente des gesamten Systems. Ein Angreifer, der in der Lage ist, in den Besitz des privaten CVK zu gelangen, kann damit komplette Gruppenstrukturen fälschen, eigene Mitglieder aufnehmen und eigene Voucher generieren.

Schutz durch (n, k)–Secret Sharing Scheme

Daher ist ein Konzept erforderlich, das den CVK schützt, gleichzeitig aber die dezentrale Struktur nicht zu sehr behindert. Hierzu wird ein (n, k)**–Secret Sharing Scheme** verwendet. Das Prinzip hierbei ist ein Geheimnis, in unserem Fall den privaten CVK, auf n Mitglieder zu verteilen. Jeder besitzt ein Teil des Geheimnisses (ein Share). Der Zweck ist, dass nur mindestens k Mitglieder gemeinsam in der Lage sind, das Geheimnis wieder zu rekonstruieren. Weniger als k Mitglieder sind nicht in der Lage an das Geheimnis zu gelangen. Ferner kann für ein neues Mitglied, das auch ein Share erhalten soll, dieses aus k Anteilen berechnet werden. Dieses Prinzip wird auf bestehende Voucher angewendet. Um beispielsweise ein neues Voucher Zertifikat signieren zu können, müssen k existierende Voucher „zusammenkommen“, um den privaten CVK zu berechnen und damit das Zertifikat zu signieren.

Um dies für die Voucher–Infrastruktur zu realisieren, kann ein Secret Sharing Scheme verwendet werden, das auf Arbeiten von Shamir [Sham 79] basiert und eine Lagrange Interpolation verwendet (bspw. [KZL+ 01]).

Am Beginn etabliert der Group Key Owner k Voucher, signiert deren Zertifikate und verteilt die Shares des CVK an die Voucher. Danach kann der Group Key Owner den CVK löschen, denn außer in der Initialisierungsphase wird der CVK nie mehr von einem einzigen Nutzer genutzt werden. Es müssen

immer k Voucher gemeinsam den privaten Schlüssel wiederherstellen.

Erzeugung eines Vouchers

Abbildung 5.6 b) stellt die Erzeugung eines neuen Vouchers dar:

1. Ein Voucher erstellt sein Schlüsselpaar und erhält vom Leader ein Zertifikat und die Zertifikatskette, welche die Mitgliedschaft in der Gruppe belegt.
2. Der zukünftige Voucher muss k (im Beispiel $k = 3$) existierende Voucher kontaktieren um sein Zertifikat re-signieren zu lassen und von diesen sein CVK Share zu erhalten.
3. Danach kann der Voucher IChC ausstellen.

5.2.5 Komplexität von IChC

IChC reduziert Komplexität auf $O(1)$

Eine klassische Verifikation einer Zertifikatskette hat die Komplexität $O(n)$ für eine Kette der Länge n. Mit Hilfe von IChC lässt sich die Komplexität auf $O(1)$ reduzieren.

IChC stellen aufgrund der zusätzlichen Zertifikate jedoch eine zusätzliche Komplexität dar. Für sehr kurze Ketten kann die klassische Verifikation u.U. effizienter sein als die Verifikation eines IChC. Interessant ist natürlich, wann der Break-Even Point erreicht ist, d.h. ab welcher Länge der Kette eine IChC Verifikation günstiger ist als eine klassische Verifikation. Im Folgenden wird diese Frage untersucht.

Die Verifikation eines Zertifikates besteht aus drei Schritten:

1. Berechnung des Hash-Wertes über den Inhalt des Zertifikate
2. Verifikation der digitalen Signatur, d.h. Entschlüsseln des Hash-Wertes aus der Signatur
3. Vergleich von berechnetem und entschlüsseltem Hash-Wert

$T_{pkc}(h, c, cert)$ sei die Zeit, die für die kryptographische Verifikation eines Zertifikates benötigt wird, welches das Hash-Verfahren *h* und das asymmetrischen Kryptosystem *c* verwendet. Die Zeit für den Vergleich der beiden Hash-Werte kann im Vergleich zu den kryptographischen Operationen vernachlässigt werden.

Zeit für kryptographische Verifikation

Die Zeit für die kryptographische Verifikation lässt sich darstellen als:

$$T_{pkc}(h, c, cert) = t_h + \frac{S(cert)}{\lambda_h} + t_c \tag{5.2}$$

wobei *S(m)* eine Funktion ist, welche die Länge in Bits des Arguments *m* zurückgibt; t_h ist der konstante Zeitanteil (in Millisekunden), der benötigt wird um das Hash-Verfahren *h* zu initialisieren . λ_h ist der Durchsatz des Hash–Verfahrens *h* in Bits/msec; *c* ist das asymmetrische Kryptosystem; t_c ist

die Zeit (in Millisekunden), die für die Verifikation einer Signatur im Kryptosystem *c* benötigt wird. Das zu verifizierende Zertifikat wird als *cert* notiert.

Der klassische Verifikationsprozess (beschrieben in Abs. 5.2.1) verwendet den Hash-Algorithmus *h* und das asymmetrische Kryptosystem *c*. Die Anzahl der Zertifikate sei *n*. Die Zeiten für den Vergleich von öffentlichen Schlüsseln (um den öffentlichen Schlüssel des Mitglieds in der Kette zu finden) sowie für die Berechnung des Gültigkeitszeitraumes können im folgenden wieder vernachlässigt werden.

$S(cert_{avrg})$ ist die durchschnittliche Größe eines Zertifikates aus der Kette $S(cert_{avrg}) = \sum_{i=1..n} \frac{S(cert_i)}{n}$ und die Größen der Einzel–Zertifikate werden im Folgenden durch diesen Mittelwert ersetzt.

Zeit für Verifikation der Kette

Die Zeit für die Verifikation der Zertifikatskette lässt sich wie folgt angeben:

$$T_{chain}(h,c) = \sum_{i=1..n} (t_h + \frac{S(cert_i)}{\lambda_h} + t_c) \tag{5.3}$$

$$T_{chain}(h,c) = nt_h + \frac{n}{\lambda_h} S(cert_{avrg}) + nt_c \tag{5.4}$$

Die Zeit $T_{ichc}(h,c)$ wird für die in Abschnitt 5.2.3 beschriebene Verifikation des IChC benötigt. Dabei gehen wir davon aus, dass dasselbe Hash-Verfahren, dasselbe Kryptosystem und dieselbe Kette der Länge n verifiziert werden. Die Verifikation eines IChC besteht aus zwei grundsätzlichen Operationen: Der Berechnung des Hash-Wertes der Kette und der kryptographischen Verifikation des $certOC$. Daneben müssen zusätzlich zwei weitere Zertifikate, das Voucher-Zertifikat und die CVK Signatur geprüft werden.

Zeit für IChC Verifikation

Die Zeit für die IChC Verifikation ergibt sich:

$$T_{ichc}(h,c) = t_h + \frac{\sum_{i=1..n} S(cert_i)}{\lambda_h} + t_h + \frac{S(certOC)}{\lambda_h} + t_c + 2t_h + 2\frac{S(certVC)}{\lambda_h} + 2t_c \tag{5.5}$$

Durch die Verwendung der durchschnittlichen Zertifikatslänge $S(cert_{avrg})$ anstelle der exakten Länge erhält man:

$$T_{ichc}(h,c) = 4t_h + 3t_c + (n+3)\frac{S(cert_{avrg})}{\lambda_h} \tag{5.6}$$

Um den Speedup-Faktor zu berechnen wird die Zeit der klassischen Verifikation mit der der IChC Verifikation verglichen:

$$T_{chain}(h,c) \; > \; T_{ichc}(h,c) \tag{5.7}$$

$$nt_h + \frac{n}{\lambda_h} S(cert_{avrg}) + nt_c \; > \; 4t_h + 3t_c + (n+3)\frac{S(cert_{avrg})}{\lambda_h}$$

$$(n-3)t_c + (n-4)t_h \; > \; \frac{3}{\lambda_h} S(cert_{avrg}) \tag{5.8}$$

Die Werte von λ_h und $S(cert_{avrg})$ sind alle positiv und konstant. Deshalb ist auch die rechte Seite der Ungleichung 5.7 positiv und konstant. Auch die Werte t_c und t_h sind positiv und konstant, damit hängt der Wert der linken Seite der Ungleichung 5.7 lediglich von *n* ab; je größer *n*, umso größer ist der Wert.

Es gibt also eine positive Zahl n_0, sodass für alle $n \geq n_0$ die linke Seite der Ungleichung 5.7 immer größer ist als die linke. Dies beweist die Korrektheit von 5.6 Daraus folgt, dass der Speedup-Faktor:

$$f = \frac{T_{chain}(h,c)}{T_{ichc}(h,c)} \tag{5.9}$$

für alle $n \geq n_0$ größer als 1 ist, und:

$$n_0 > \frac{\frac{3}{\lambda_h}S(cert_{avrg}) + 3t_c + 4t_h}{t_c + t_h}. \tag{5.10}$$

gilt.

Damit ist gezeigt, dass eine ganz Zahl n_0 mit $n \geq n_0$ existiert, für die die Verifikation eines IChC für eine Zertifikatskette der Länge n effizienter ist als die klassische Verifikation der Kette.

IChC Speedup

Ausgehend von diesen Ergebnissen und basierend auf den in [LeCa 99] gemessenen Zeiten für verschiedene Krypto- und Hash-Systeme wurde der Speedup-Faktor in Abbildung 5.7 dargestellt. Für alle dargestellten Kombinationen aus Krypto- und Hash-Verfahren ist spätestens bei Ketten der Läge 4 das IChC Verfahren effizienter als die klassische Verifikation.

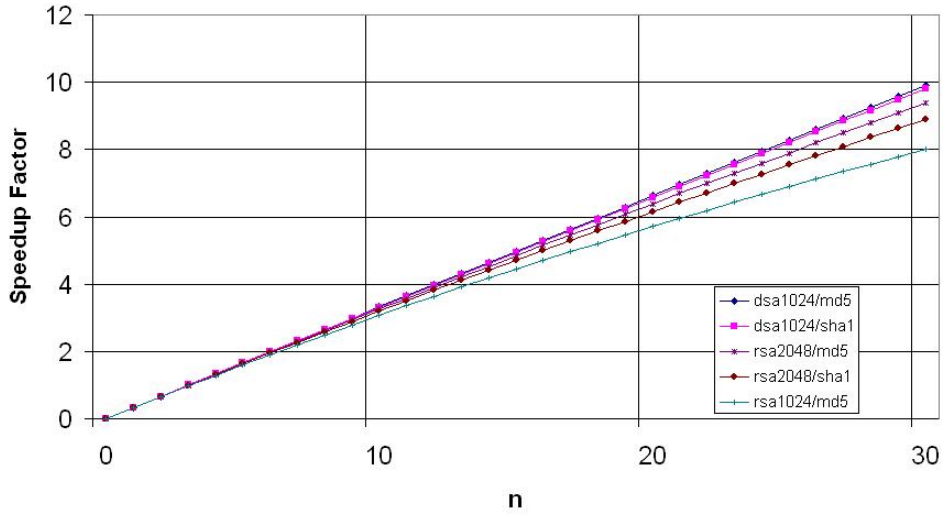

Abbildung 5.7: Vergleich IChC zu klassischer Verifikation: Speedup

Die hier vorgestellten Algorithmen wurden implementiert und die Laufzeiten für verschiedenen Kombinationen aus Signatur- und Hash-Verfahren bestimmt. Tabelle 5.1 stellt die Ergebnisse dieser Analyse dar.

Dabei bezeichnet T_{chain} die klassische Verifikation, T_{ichc^s} die Zeit die allein für die Verifikation der IChC benötigt wird, T_{ichc} die Gesamtzeit einschließlich der Zeit die für die Verifikation der Voucher Zertifikate benötigt wird. TS bezeichnet die Zeitersparnis von IChC im Vergleich zur klassischen Verifikation und f den Speedup-Faktor.

In Kapitel 5 wurden zwei Sicherheitsmechanismen entwickelt, die sich für Sicherheitsdienste in den Dienstklassen VO-Management und bei den Basisdiensten einsetzen lassen (vgl. Abbildung 5.1).

Mit dem neuen Algorithmus zur dynamischen Ableitung von Vertrauenswerten (vgl. Abschnitt 5.1.2) wurde ein wichtiger Beitrag zum Trust Level Management erbracht. Damit lassen sich auch mittelbare Vertrauenswerte ableiten und der Algorithmus ist in der Lage mit einer hohen Dynamik umzugehen.

Mit dem IChC (vgl. Abschnitt 5.2.2) kann das VO-Management, bzw. das Gruppenmanagement innerhalb des VO-Management, dynamisiert und flexibilisiert werden und ein dezentraler und föderierter Einsatz des Konzeptes wird möglich. Außerdem verringert das Verfahren die Komplexität der Verifikation von Zertifikatsketten.

		(msec)				
Algorithmen		T_{chain}	T_{ichc^s}	T_{ichc}	TS	f
RSA1024/MD5	n = 10	20263.60	2254.19	6231.31	14032.29	3.25
	n = 20	40835.23	2333.14	6316.58	34518.65	6.46
	n = 50	107138.39	2741.20	6724.64	100413.75	15.93
	n = 100	211827.09	2912.97	6896.42	204930.67	30.72
RSA1024/SHA-1	n = 10	20339.56	2262.61	6240.58	14098.98	3.26
	n = 20	40928.00	2366.86	6344.83	34583.17	6.45
	n = 50	110441.14	2997.65	6981.10	103460.04	15.82
	n = 100	212454.09	3110.22	7088.18	205365.91	29.97
RSA2048/MD5	n = 10	138314.56	14468.79	41628.76	96685.81	3.32
	n = 20	278764.47	14975.56	42135.53	236628.94	6.62
	n = 50	690833.22	16207.58	43367.55	647465.67	15.93
	n = 100	1403689.29	17223.24	44383.21	1359306.08	31.63
RSA2048/SHA-1	n = 10	145235.59	14644.33	45296.92	99938.67	3.21
	n = 20	292713.37	15157.25	45809.84	246903.53	6.39
	n = 50	725401.36	17012.27	47675.92	677725.44	15.22
	n = 100	1472073.23	18389.47	48894.63	1423178.60	30.11
DSA1024/MD5	n = 10	2638.96	397.96	917.48	1721.48	2.88
	n = 20	5007.07	452.87	972.38	4034.69	5.15
	n = 50	13505.32	619.02	1138.53	12366.79	11.86
	n = 100	27920.15	937.81	1441.46	26478.70	19.37
DSA1024/SHA-1	n = 10	2667.07	406.89	976.53	1690.54	2.73
	n = 20	5406.09	505.64	1075.28	4330.82	5.03
	n = 50	13410.57	620.53	1190.16	12220.41	11.27
	n = 100	27887.04	945.94	1515.58	26371.46	18.40
DSA2048/MD5	n = 10	8272.15	1262.41	2945.27	5326.88	2.81
	n = 20	17312.32	1287.76	2970.62	14341.70	5.83
	n = 50	43231.29	1659.20	3342.06	39889.23	12.94
	n = 100	83649.62	2162.53	3845.40	79804.22	21.75
DSA2048/SHA-1	n = 10	8303.06	1291.47	3027.40	5275.66	2.74
	n = 20	17337.82	1369.45	3105.38	14232.44	5.58
	n = 50	41769.84	1702.38	3438.31	38331.53	12.15
	n = 100	83125.88	2351.79	4087.72	79038.17	20.34

Tabelle 5.1: Messung der IChC Performance

Kapitel 6

Anwendung des Rahmenwerkes

In dieser Arbeit wurde ein Rahmenwerk für föderiertes Sicherheitsmanagement entwickelt. Dieses Rahmenwerk kann sowohl von lokalen Sicherheitsverantwortlichen als auch für ganze Föderationen oder VOs angewendet werden. Der lokale Sicherheitsverantwortliche, der mit Ressourcen an einer Föderation teilnimmt, kann das Rahmenwerk unter seinem lokalen Fokus in seiner Rolle als Föderationsmitglied nutzen. Auf der anderen Seite kann aber auch eine gesamte VO bzw. deren Sicherheitsverantwortliche mit Hilfe des Rahmenwerkes die konkrete Sicherheitsarchitektur für den speziellen Anwendungsfall dieser VO gestalten.

Ableitung einer Sicherheitsarchitektur

Dieses Kapitel beschreibt das Vorgehen für die Anwendung des Rahmenmodells auf einen konkreten Anwendungsfall. Das primäre Ziel, das mit der Anwendung erreicht werden kann, ist die Erhöhung des Sicherheitsniveaus innerhalb einer Föderation oder innerhalb einer lokalen Domäne, die an einer Föderation teilnimmt. Der Sicherheitsadministrator, der das in diesem Abschnitt vorgestellte Vorgehensmodell des Szenario-unabhängigen Rahmenwerkes anwendet, wird als Ergebnis des Vorgehens eine spezifische Sicherheitsarchitektur für sein konkretes Anwendungsszenario erstellen.

Das Vorgehensmodell, das in Abbildung 6.1 dargestellt ist, umfasst zwei grundlegende Phasen: die Analyse- sowie die Synthesephase, die im folgenden näher erläutert werden.

6.1 Analysephase

Requirements Engineering für konkreten Anwendungsfall

Im Rahmen der Analysephase führt der Sicherheitsverantwortliche ein Requirements Engineering für seinen Anwendungsfall durch. Ziel dieses Teilprozesses des Vorgehensmodells ist die Ableitung der Sicherheitsanforderungen für das konkrete Anwendungsszenario und eine Vorauswahl von Sicherheitsdiensten. Der Sicherheitsverantwortliche wird in die Lage versetzt, seine Anforderungen vollständig zu erfassen und keinen wichtigen Sicherheitsdienst zu „vergessen“.

Das Rahmenwerk beinhaltet eine umfassende und systematische, dabei aber

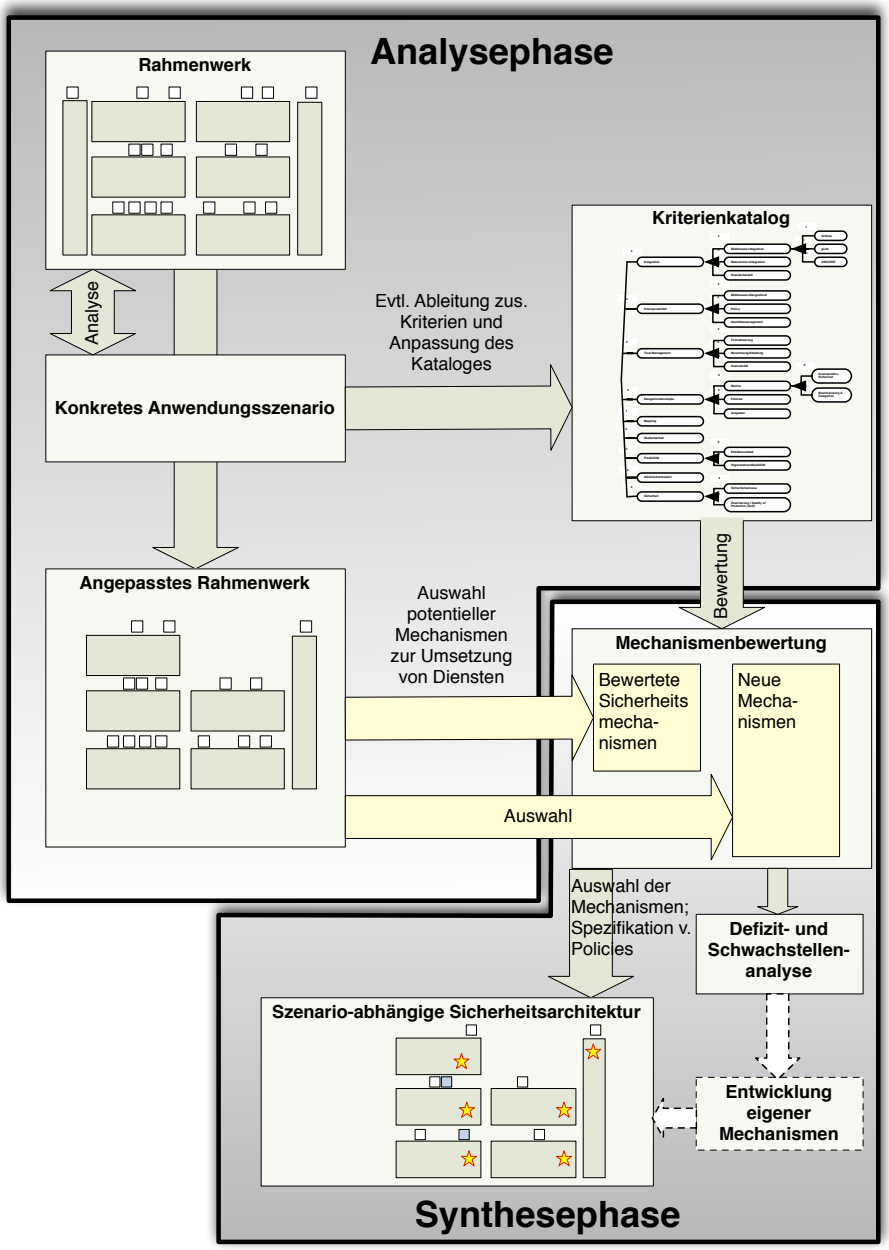

Abbildung 6.1: Anwendung des Frameworks durch den Sicherheitsadministrator

Szenario-unabhängige Anforderungsanalyse (vgl. Abschnitt 2.2) von Sicherheitsdiensten, die in einer Föderation hilfreich und notwendig sein können. Der Sicherheitsverantwortliche kann für sein Szenario die Notwendigkeit jeder einzelner der in Kapitel 2.2.2 aufgeführten Anforderung prüfen. Ein Ressourcen Provider kann bereits bei diesemSchritt die Anforderungen mit den vorgegebenen lokalen Sicherheitsanforderungen abgleichen und ggf. Widersprüche identifizieren. Ein Widerspruch tritt dann auf, wenn eine Anforderung ermittelt wird, die für die Teilnahme an einer Föderation erfüllt sein muss, lokal aber nicht unterstützt wird. Es kann auch lokale Anforderungen geben,

die in der Föderation nicht unterstützt oder umgesetzt werden sollen. Diese Widersprüche können durch Anpassung lokaler oder globaler Policies aufgelöst werden. Sie können aber auch dazu führen, dass eine Teilnahme an der Föderation nicht möglich ist.

Die vom Sicherheitsverantwortlichen abgeleiteten Anforderungen können durch Sicherheitsdienste umgesetzt werden. Das Rahmenwerk definiert starke und schwache Abhängigkeiten zwischen diesen Diensten und gruppiert die Sicherheitsdienste in Dienstklassen (vgl. Abbildung 2.3). Damit kann der Sicherheitsadministrator ggf. Widersprüche in seiner Anforderungsanalyse erkennen. Wenn bspw. die Anforderung nach einem Sicherheitsdienst A vorliegt, dieser in starker Abhängigkeit zum Dienst B steht, muss auch B mit in die Liste der Anforderungen aufgenommen werden.

Auswahlprozess liefert erforderliche Sicherheitsdienste

Die linke Seite im Block „Analysephase“ der Abbildung 6.1 stellt diesen Auswahlprozess dar, an dessen Ende der Sicherheitsadministrator weiß, welche Sicherheitsdienste für seinen Anwendungsfall benötigt werden.

Anpassung des Kriterienkataloges

Das Rahmenwerk definiert in Abschnitt 3 die Struktur und Methodik eines Kriterienkataloges sowie konkrete Kriterien zur Bewertung von Sicherheitsmechanismen. Dieser Katalog unterstützt den Sicherheitsadministrator in der Synthesephase bei der Auswahl konkreter Sicherheitsmechanismen zur Realisierung der erforderlichen Sicherheitsdienste. Um ein für den konkreten Anwendungsfall möglichst optimales Syntheseergebnis zu erhalten, kann der Sicherheitsadministrator den Kriterienkatalog auf zwei Arten an seine spezifischen Gegebenheiten anpassen. Auf der einen Seite kann er sehr einfach lokale Präferenzen in Form eigener Kriterien zum Katalog hinzufügen oder vorgegebene Kriterien aus dem Katalog streichen. Auf der anderen Seite kann er eigene Prioritäten durch eine Veränderung der Gewichte bei den Kriterien im Kriterienkatalog abbilden.

Das Ergebnis der Analysephase ist also auf der einen Seite eine Liste von Sicherheitsdiensten, die umgesetzt werden müssen. Auf der anderen Seite steht ein auf den konkreten Anwendungsfall angepasster Kriterienkatalog zur Bewertung von Sicherheitsmechanismen.

6.2 Synthesephase

Mit Hilfe der beiden Ergebnise der Analysephase wird in der Synthesephase des Vorgehensmodells eine szenario-abhängige Sicherheitsarchitektur erstellt.

Mechanismenauswahl und -bewertung

Das in dieser Arbeit vorgestellte Rahmenwerk liefert in Abschnitt 4 bereits eine Vielzahl von Mechanismen zur Umsetzung einzelner Sicherheitsdienste mit einer entsprechenden Bewertung nach dem in Kapitel 3 vorgestellten Kriterienkatalog. Der Sicherheitsadministrator muss ausgehend von seiner Liste der Sicherheitsdienste potentielle Mechanismen auswählen, die geeignet sind die entsprechenden Dienste umzusetzen. Hierzu kann er auf die

Ergebnisse von Kapitel 4 zurückgreifen und diese ggf. um weitere Mechanismen ergänzen. Auf alle diese Mechanismen muss er dann, wie im Kapitel 4 gezeigt, den Kriterienkatalog anwenden. Wird der Katalog unverändert übernommen, kann er auch die Bewertungen einfach übernehmen. Wurden die Gewichte geändert, müssen für die Hauptkriterien neuen Bewertungszahlen berechnet werden. Wurden dem Katalog in der Analysephase neue Kriterien hinzugefügt, sind die Mechanismen im Hinblick darauf noch zu bewerten. Für neue Mechanismen, die in dieser Arbeit nicht untersucht wurden („Neue Mechanismen“ in Abbildung 6.1) ist eine Gesamtbewertung durchzuführen.

Am Ende der Phase der Mechanismenbewertung hat der Sicherheitsadministrator für jeden Mechanismus eine Gesamtbewertungszahl sowie detaillierte Bewertungszahlen für die Teilkriterien (in Kapitel 4, z.B. graphisch als Netzdiagramme repräsentiert). Diese Bewertung kann er für die Auswahl der für seinen Anwendungsfall besten Sicherheistmechanismen genauso verwenden wie für eine Defizit- und Schwachstellenanalyse, wie sie für den szenariounabhängigen Fall bereits in Abschnitt 4.10.2 exemplarisch durchgeführt wurde.

Sollte die Mechanismenbewertung und die Defizitanalyse ergeben, dass für bestimmte Sicherheitsdienste keine adäquaten Mechanismen zur Umsetzung existieren, so könnte ein Entwicklungsprozess angestoßen werden. Dasselbe gilt auch, falls die bewerteten Mechanismen eklatante Schwächen in einzelnen Kriterien besitzen, die behoben werden müssen. Ein Spezifikationsprozess für fehlende Mechanimsen wurde in Abschnitt 5 gezeigt. Die Entwicklung eigener Mechanismen kann i.d.R. jedoch nur sinnvoll im Kontext einer gesamten Föderation erfolgen. Dieser Prozessschritt ist deshalb in Abbildung 6.1 optional (d.h. gestrichelt) dargestellt.

Kontinuierliche Verbesserung

Am Ende der Synthesephase hat der Sicherheitsverantwortliche im Idealfall für jeden Sicherheitsdienst den oder die besten Sicherheitsmechanismen ausgewählt. Durch die Bewertung ist er auch in der Lage die Schwächen seiner resultierenden Sicherheitsarchitektur zu benennen und eine Prioritätenliste für die kontinuierliche Verbesserung und damit eine Erhöhung des Sicherheitsniveaus zu erstellen.

Vor der Instantiierung der Sicherheitsarchitektur in einer Föderation müssen Policies bezüglich Verwendung, Einsatzort und -art, Konfiguration, Pflege, usw. festgelegt und umgesetzt werden.

Kapitel 7

Zusammenfassung und Ausblick

Die Vision von der „Rechenleistung aus der Steckdose“ war ein Momentum für die Entwicklung von Grid Technologien. Föderationen und insbesondere Grids bieten im wissenschaftlichen Bereich erstmals die Möglichkeit die Daten riesiger Experimente, wie z.B. des LHC (vgl. Abschnitt 4.1.11), von einer sehr großen Zahl von Wissenschaftlern, die weltweit verteilt sind, gemeinsam zu nutzen und in kooperativer Weise zu analysieren. Durch die Virtualisierung und das Sharing von Ressourcen, unabhängig von deren Standort, dem Betreiber oder dem rechtlichen Eigentümer, kann der Wissenschaftler Rechenleistung nutzen. Er kann sich aus dem Ressourcen-Pool auch die für seine Problemstellung optimale Ressource auswählen.

Grid wird zunehmend von der Wirtschaft genutzt

Das Grid stellt damit eine optimale Kooperations- und Koordinationsplattform dar. Die anfängliche Beschränkung auf rein wissenschaftliche Anwendungsfelder beginnt sich auch zu Anwendungen in der Wirtschaft hin zu verändern. Zum Beispiel versucht das BauVOGrid im Rahmen des D-Grid die Struktur, Funktionsweise und Operabilität von virtuellen Unternehmen im Bauwesen durch Grid und VO-Techniken entscheidend zu verbessern [BauVOGrid]. GDI-Grid verwendet Grid-Technologien, um landesweit verteilte Geo-Datenbestände zu integrieren und für Katastrophenfälle oder zur Geosimulation (z.B. für Lärmausbreitung oder Hochwasser) zu verwenden [GDI-Grid].

Sicherheitsfragen werden immer wichtiger

Je mehr aber die Grid Technologien und Föderationen nicht mehr nur eng abgegrenzten wissenschaftlichen Communities dienen, sondern in Projekte von ansonsten konkurrierenden Unternehmen Verwendung finden, umso wichtiger werden und sind Sicherheitsfragen. Sicherheitsverantwortliche, auf Organisations-lokaler Ebene oder auch auf Ebene der Föderation oder VO, benötigen Werkzeuge, um eine Sicherheitsanalyse durchführen und um Sicherheitsdienste bewerten und zu einer Sicherheitsarchitektur verbinden zu können. Oberstes Ziel muss dabei ein einheitlich hohes und durchgehendes Sicherheitsniveau sein.

7.1 Ergebnisse

Im Rahmen dieser Arbeit wurde nicht der Ansatz verfolgt für eine konkrete Middleware bzw. ein konkretes Grid-Projekt eine spezifische Sicherheitsarchitektur zu erstellen, sondern es wurde ein Szenario-unabhängiges und Middleware-übergreifendes Rahmenwerk und ein entsprechendes Vorgehensmodell entwickelt und präsentiert. Dieses umfassende und systematische Rahmenwerk kann dann von den verschiedenen Sicherheitsverantwortlichen für ihre jeweiligen konkreten Anwendungsfälle entsprechend angewendet werden, um eine Szenario- und Middleware-spezifische Sicherheitsarchitektur abzuleiten, das Sicherheitsniveau bewerten und kontinuierlich verbessern zu können.

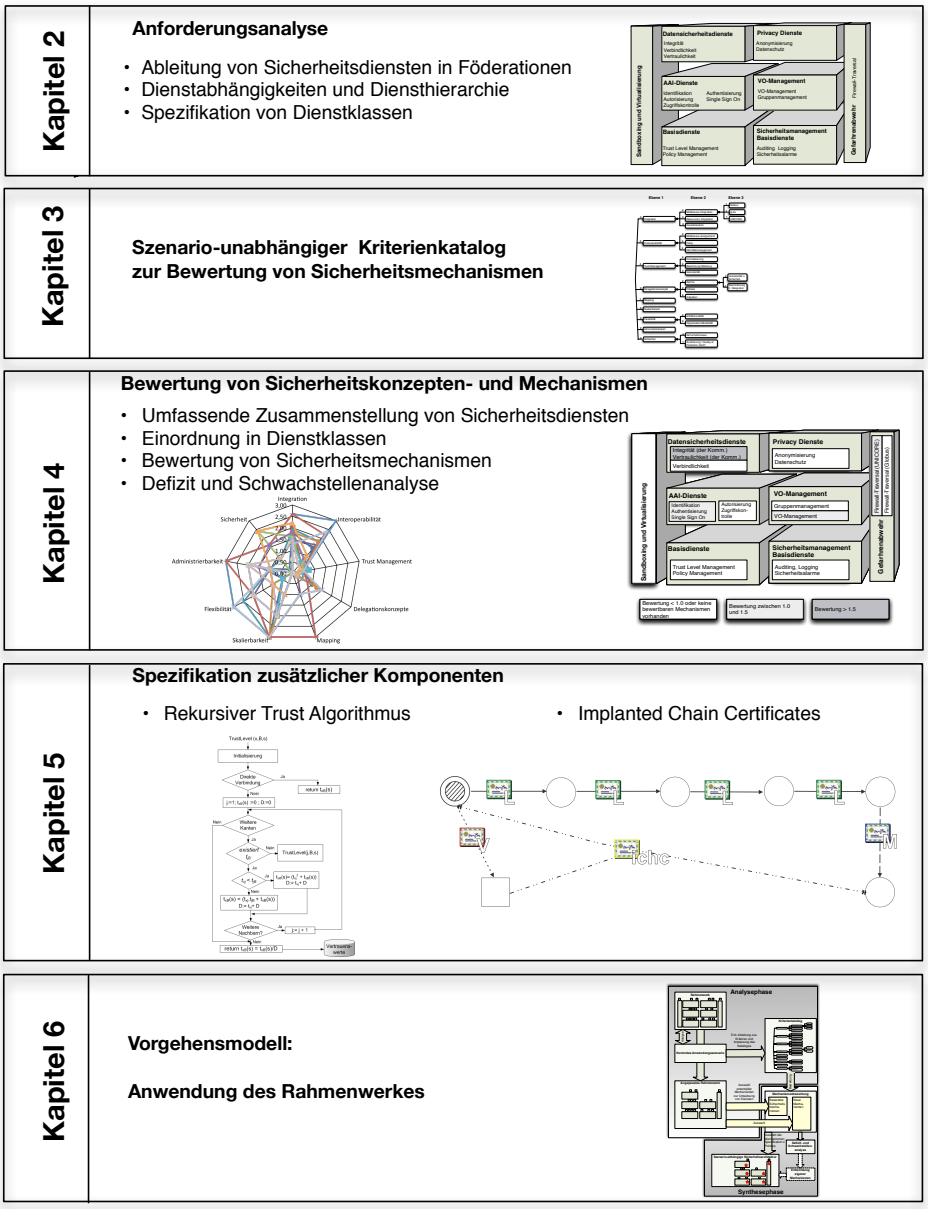

Abbildung 7.1: Überblick über Beiträge dieser Arbeit

Die Hauptbeiträge dieser Arbeit sind in der Abbildung 7.1 skizzenhaft visualisiert und lassen sich wie folgt zusammenfassen:

- In Kapitel 2 wurde eine systematische und umfassende Anforderungsanalyse durchgeführt. Die Hauptbeiträge in diesem Abschitt sind:
  - Umfassende Ableitung von Sicherheitsdiensten für Föderationen
  - Definition eines Abhängigkeitsbegriffs zwischen den Sicherheitsdiensten
  - Ableitung einer Diensthierarchie
  - Gliederung der Sicherheitsdienste in Dienstklassen
- In Kapitel 3 wurde ein Kriterienkatalog zur Bewertung der Sicherheitsmechanismen entwickelt.
- Kapitel 4 befasste sich mit der Bewertung der Sicherheitsmechanismen. Die Hauptbeiträge sind:
  - Umfassende Zusammenstellung von Sicherheitsdiensten und entsprechender Mechanismen für drei Middleware-Technologien
  - Einordnung der Mechanismen in Dienstklassen
  - Bewertung der Mechanismen
  - Allgemeine Defizit- und Schwachstellenanalyse
- Aus der Defizit- und Schwachstellenanalyse ergibt sich ein erhebliches Verbesserungspotential bei den Sicherheitsmechanismen. In Kapitel 5 wurden für die Dienstklasse Basisdienste und VO-Management neue Mechanismen vorgeschlagen. Die Hauptbeiträge hier sind:
  - Trust-Level-Management: Entwicklung eines Algorithmus zur dynamischen Berechnung von Trust Levels
  - Gruppenmanagement: Entwicklung von Implanted Chain Certificates (IChC), Implementierung und Leistungsbewertung des Konzeptes
- Kapitel 6 beschreibt das zweistufige Vorgehensmodell aus Analyse- und Synthesephase. Mit dessen Hilfe kann der Sicherheitsverantwortliche das in dieser Arbeit entwickelte, Szenario-unabhängige und Middleware-übergreifende Rahmenwerk auf sein konkretes Anwendungsszenario anwenden, um die Instanz einer Sicherheitsarchitektur zu entwickeln.

7.2 Offene Forschungsfragestellungen

Aus dieser Arbeit, insbesondere aus der Defizit- und Schwachstellenanalyse (vgl. Abschnitt 4.10.2) zeigt sich, dass ein erhebliches Verbesserungspotential

im Hinblick auf das Sicherheitsmanagement in Grids existiert. Daraus ergeben sich eine Vielzahl weiterführender wissenschaftlicher Fragestellungen. Im Folgenden werden einige sehr wichtige und interessante Projekte vorgestellt.

- **Frühwarnsysteme in Grids:**
  Die am Grid beteiligten Organisationen betreiben eigene Sicherheitseinrichtungen, wie Firewalls, Intrusion-Detection (IDS) sowie Intrusion-Prevention-Systeme (IPS) und erheben mit diesen Systemen eine Vielzahl von Sicherheits- und Angriffs-relevanten Daten mit dem Ziel, Angriffe auf ihre Infrastruktur möglichst früh zu erkennen und möglichst optimal darauf reagieren zu können. Das Problem dabei ist jedoch der ausschließlich lokale auf die jeweilige Organisation bezogene Fokus dieser Aktivitäten. Die Partner im Grid nutzen derzeit die Föderation nicht, um ein globales Grid-übergreifendes Frühwarnsystem zu etablieren.

  In einer aktuellen Arbeit wird daher die Entwicklung eines Frühwarnsystems in Grids untersucht. Dabei wird die Idee der kooperativen Nutzung lokaler Sicherheitssysteme und der Austausch von Angriffsdaten verfolgt. Über eine Datenschutz-konforme Aufbereitung und Bereinigung von Log-Daten, unter Wahrung individuell bestehender Sicherheitsrichtlinien sowie lokalen Richtlinien zum Austausch vertraulicher Daten, können die Informationen in einem zentralen oder verteilten Grid-IDS analysiert werden. Damit besteht die Möglichkeit, die Grid-weite Erkennungsrate zu verbessern, die Reaktionszeiten zu verkürzen und verteilte Angriffe zu detektieren, die aus dem ausschließlich lokalen Blickwinkel der einzelnen Organisation nicht erkennbar wären.

- **Trust Management:**
  Obwohl das Trust Management einen absoluten Basisdienst darstellt, fällt die Bewertung entsprechender Verfahren sehr schlecht aus (vgl. auch Abschnitte 4.8.4 und 4.10.2). Dies liegt entweder daran, dass die Trust Level implizit und global festgelegt werden, oder an der Tatsache, dass überhaupt kein Trust Management vorgesehen ist.

  In Abschnitt 5.1 wurde ein Verfahren vorgeschlagen, um dynamisch Vertrauenswerte abzuleiten. In Fortsetzung dieser Arbeit entsteht derzeit eine Dissertation, die ein allgemeines und umfassendes Konzept für das Trust Management in Föderation entwickelt. Die Arbeit beschäftigt sich neben der Formalisierung und dynamischen Berechnung von Trust und Trust Metriken, deren Repräsentation in Datenstrukturen sowie mit Reputationsverfahren, mit dem effizienten Zugriff, der Aktualisierung, dem Schutz, der Verbindlichkeit und der Integrität der Vertrauenswerte. Daneben werden auch Managementkonzepte für die zu entwickelnde Trust-Infrastruktur betrachtet.

- **Prozessorientiertes Sicherheitsmanagement und Audit-Verfahren im Grid:** In derzeitigen Grids läßt sich die Umsetzung von Sicherheitsdiensten und der effektive und richtige Einsatz von Sicherheitsmechanis-

men der beteiligten Organisationen weder von der VO noch von einem unabhängigen Dritten überpürfen. Es gibt auch keine Mechanismen, um erreichbare Sicherheitsniveaus zu definieren und deren Einhaltung zu bestimmen. Der Mittelwert über alle betrachteten Mechanismen für Zusicherung / Quality of Protection liegt bei 0,7 (vgl. Abbildung 4.62), d.h. das Kriterium wird von keinem Mechanismus erfüllt. Sicherheitsaudits werden derzeit auch nicht durchgeführt.

Im Rahmen einer Forschungsarbeit sollte die Anwendung und Umsetzung einschlägiger internationaler (wie ISO/IEC 270001 [ISO 270001]) wie nationaler Sicherheitsnormen (z.B. BSI Grundschutz [Bund 07]) im Grid untersucht werden. Dabei können Prinzipen des prozessorientierten IT-Management nach ITIL oder ISO/IEC 20000 [ISO 20000-1, ISO 20000-2] auf Sicherheitsarchitekturen im Grid angewandt werden mit dem Ziel, Sicherheitsrichtlinien und Sicherheitsprozesse sowie Messverfahren und Maßzahlen für das Sicherheitsniveau zu definieren. Um die Umsetzung der Prozesse und die Einhaltung der Sicherheitsrichtlinien überprüfen und unabhängig auditierbar zu machen, sollten auch Audit-Richtlinien und Prüfverfahren und -prozesse festgelegt werden.

Eine Fragestellung in diesem Umfeld ist auch die externe Überprüfung der Verwundbarkeit einer Grid Infrastruktur. Bei Betriebs- oder Verteilten Systemen wird dies durch Vulnerability Scans oder Angriffsversuche sogenannter „Tiger Teams“ realisiert. Dazu bedarf es der systematischen Ermittlung von Schwachstellen der verschiedenen Middleware-Technologien oder der darauf aufsetzenden Grid Dienste. Mit diesem Wissen kann dann ein System (Security Scanner) entwickelt werden, das die Verweundbarkeit automatisch oder semi-automatisch ermittelt.

ABKÜRZUNGEN

AAI Authentication, Authorization, Identification

AAP Attribute Acceptance Policy

AA Attribute Authority

ACL Access Control List

ACS Assertion Consumer Service

AES Advanced Encryption Standard

AJO Abstract Job Object

ALICE A Large Ion Collider Experiment

APGrid PMA Asia Pacific Grid PMA

API Application Programming Interface

APPEL A P3P Preference Exchange Language

AP Authentication Profile

ARP Attribute Release Policy

AR Attribute Requestor

ASN.1 Abstract Syntax Notation Nr. 1

ATLAS A Toroidal LHC ApparatuS

AUP Acceptable Use Policies

BDSG Bundesdatenschutzgesetz

BMBF Bundesministerium für Bildung und Forschung

BSC Barcelona Supercomputing Center

BSI Bundesamt für Sicherheit in der Informationstechnik

CAS Community Authorization Service

CA Certification Authority

CEA Computing Element Acceptance

CEMon CE Monitor

CE Computing Element

CINECA Consorzio Interuniversitario del Nord est Italiano Per il Calcolo Automatico

CL Client Library

CMS Compact Muon Solenoid

CNRS Centre National de la Recherche Scientifique

CODO Cooperative On-Demand Opening

CP Certification Policy

CP Community Projekt

CSO Chief Security Officer

CVK Common Voucher Key

DES Data Encryption Standard

DFN Deutsches Forschungsnetz

DGI D-Grid Integrationsprojekt

DICOM Digital Imaging and Communications in Medicine

DMZ Demilitarisierte Zone

DN Distinguished Name

DSS Data Security Standard

DVO Dynamische Virtuelle Organisation

ECC Elliptic Curve Cryptography

ECMWF European Centre for Medium-Range Weather Forecasts

EDG European DataGRID

EEC End Entity Certificates

EGA Enterprise Grid Alliance

EGEE Enabling Grids for E-Science

EPCC Edinburgh Parallel Computing Centre

EUGrid PMA Europäische Grid PMA

FA Firewall Agent

ff. folgende

FIM Federated Identity Management

FQAN Fully Qualified Attribute Name

FZJ Forschungszentrum Jülich

GACG German Astro Community Grid

GGF Global Grid Forum

GRAM Grid Ressource Allocation Manager

GSH Grid Service Handle

GSI Grid Security Infrastructure

GSR Grid Service Reference

GSS Generic Security Services

HLRS Höchstleistungsrechenzentrum Stuttgart

HS Handle Service

IChC Implanted Chain Certificate

IDAT identifizierende Daten

IDB Incarnation Data Base

IDEA International Data Encryption Algorithm

IdM Identity Management

IDRIS Institut du Développement et des Ressources en Informatique Scientifique

IDS Intrusion Detection System

IGTF International Grid Trust Federation

IPS Intrusion Prevention System

IPY International Polar Year

IP Internet Protocol

IVOA International Virtual Observatory Alliance

JMC Job Monitor Controller

JMS Java Messaging Service

JNI Java Native Interface

JPA Job Preparation Agent

JVM Java Virtual Machine

LCAS Local Centre Authorization Service

LCG LHC Computing Grid

LCMAPS Local Credential Mapping Service

LHCb Large Hadron Collider beauty

LHCf Large Hadron Collider forward

LHC Large Hadron Collider

LRMS Local Ressource Management System

LRZ Leibniz Rechenzentrum

MAC Message Authentication Code

MA Mobiler Agent

MD5 Message Digest No. 5

MDAT medizinische Daten

MDS Monitoring & Discovery System

MICS Member Integrated Credential Services

NAT Network Address Translation

NCSA National Center for Supercomputing Applications

NJS Network Job Supervisor

Nspace No UNICORE Space

OASIS Organization for the Advancement of Structured Information Standards

OGA Open Grid Forum

OGSA-DAI Open Grid Service Architecture – Data Access and Integration

OGSA-SEC-WG OGSA Security Working Group

OGSA Open Grid Service Architecture

OGSI Open Grid Service Infrastructure

OPN Optical Private Network

OSI–RM OSI Referenzmodell

OU Organizational Unit

O Organization

P2P Peer to Peer

P3P Platform for Privacy Preference

PBS Portable Batch System

PCI Payment Card Industry

PGP Pretty Good Privacy

PMA Policy Management Authority

QoP Quality of Protection

RA Registration Authority

RC4 Ron's Code 4

RFT Reliable File Transfer

RLS Replica Location Service

RM Referenzmodell

RM Ressource Manager

RPC Remote Procedure Call

RZG Rechenzentrum Garching der Max Panck Gesellschaft

SAML Security Assertion Markup Language

SAP SSL Authentication Protocol

SHA Secure Hash Algorithm

SIM Security Information Management

SLCS Short Lived Credential Service

SLC Short Lived Credential

SOAP Simple Object Access Protocol

SPKI Simple Public Key Infrastructure

SP Service

SSO Single Sign On

SSO Site-Specific Security Object

STUN Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)

TAGPMA The Americas Grid PMA

TCP Transmission Control Protocol

Torque Tera Scale Open-Source Ressource and Queue Manager

TOTEM Total Cross Section, Elastic Scattering and Diffraction Dissociation at the LHC

TSI Target System Interface

UDDI Universal Description, Discovery and Integration

UDT UDP-based Data Transfer Protocol

UPL UNICORE Protocol Layer

URI Uniform Resource Identifier

Usite UNICORE Grid Site

Uspace UNICORE File Space

UUDB UNICORE User Databse

VMM Virtual Machine Monitor

VoIP Voice over IP

VOMS Virtual Organization Membership Service

VO Virtuelle Organisation

VPN Virtual Private Network

Vsite Virtual Site

VT-i Intel Virtualization Technology for the Itanium architecture

VT-x Intel Virtualization Technology for the IA-32 architecture

WAYF Where Are You From

WG Working Group

WMS Workload Management Service

WN Worker Node

WSDL Web Service Description Language

WSRF Web Services Resource Framework

XACML eXtensible Access Control Markup Language

XIO Extensible Input/Output

XML Extensible Markup Language

Xspace Unix File Space

ABBILDUNGEN

TABELLEN

LITERATUR

[ABB+ 05] ANDERSON, S., J. BOHREN, T. BOUBEZ, G. DELLA-LIBERA, B. DIXON, P. GARG, M. GUDGIN, S. HADA, P. HALLAM-BAKER, M. HONDO, C. KALER, H. LOCKHART, R. MARTHERUS, H. MARHYAMA, A. NADALIN, N. NAGARATNAM, A. NASH, R. PHILPOTT, D. PLATT, H. PRAFULLCHANDRA, M. SAHU, J. SHEWCHUK, D. SIMON, D. SRINIVAS, E. WAINGOLD, D. WAITE, D. WALTER und R. ZOLFONOON: *Web Services Secure Conversation Language (WS-SecureConversation)*. Technischer Bericht, Februar 2005, ftp://www6.software.ibm.com/software/developer/library/ws-secureconversation.pdf .

[ABD+ 05] ANJOMSHOAA, A., F. BRISARD, M. DRESCHER, D. FELLOWS, A. LY, S. MCGOUGH, D. PUSHIPHER und A. SAVVA: *Job Submission Description Language (JSDL) Specification, Version 1.0*. Technischer Bericht GFD-R-P.056, GGF, November 2005, http://www.gridforum.org/documents/GFD.56.pdf .

[ACC+ 03] ALFIERI, R., R. CECCHINI, V. CIASCHINI, L. DELL'AGNELLO, A. GIANOLI, F. SPATARO, F. BONNASSIEUX, P. J. BROADFOOT, G. LOWE, L. CORNWALL, J. JENSEN, D. P. KELSEY, A. FROHNER, D. L. GROEP, W. SOM DE CERFF, M. STEENBAKKERS, G. VENEKAMP, D. KOURIL, A. MCNAB, O. MULMO, M. SILANDER, J. HAHKALA und K. LÖRENTEY: *Managing Dynamic User Communities in a Grid of Autonomous Resources*. In: *Computing in High Energy and Nuclear Physics 2003 Conference Proceedings (CHEP03)*, La Jolla, California, USA, März 2003. Stanford Linear Accelerator Center (SLAC); University of California, San Diego (UCSD), http://www.slac.stanford.edu/econf/C0303241/proc/papers/TUBT005.PDF .

[ACC+ 04] ALFIERI, R., R. CECCHINI, V. CIASCHINI, L. DELL'AGNELLO, K. LORENTEY A. FROHNER, A. GIANOLI und F. SPATARO: *VOMS, an Authorization System for Virtual Organizations*. In: RIVERA, F. F., M. BUBAK, A. G. TATO und R. DOALLO (Herausgeber): *Grid Computing — First European Across Grids Conference*, Nummer LNCS 2970 in *Lecture Notes in Computer Science*, Seiten 33–40, Santiago de Compostela, Spain, February 13-14 2004. Springer, http://grid-auth.infn.it/docs/VOMS-Santiago.pdf .

[ACC+ 05] ALFIERI, R., R. CECCHINI, V. CIASCHINI, L. DELL'AGNELLO, A. FROHNER, K. LÖRENTEY und F. SPATARO: *From gridmap-file to VOMS: managing authorization in a Grid environment*. Future Generation Computer Systems (FGCS), 21(4):559–571, April 2005, http://www.fis.unipr.it/lca/grid/doc/from-gridmap.pdf .

[Alta] ALTAIR ENGINEERING: *Altair PBS Professional*, http://www.altair.com/software/pbspro.htm .

[APGrid PMA] *APGrid PMA — Asia Pacific Grid Policy Management Authority*, http://www.apgridpma.org/ .

[AR 97] ABDUHL-RAHMAN, A.: *The PGP Trust Model*. In: *EDI - Forum*, April 1997, http://www.wim.uni-koeln.de/uploads/media/The_PGP_Trust_Model.pdf .

[Asia 05] ASIA PACIFIC GRID POLICY MNAGEMENT AUTHORITY: *Asia Pacific Grid Minimum CA Requirements*. Technischer Bericht Version 1.1, IGTF; APGridPMA, Mai 2005, http://www.apgridpma.org/docs/APGridPMA-Minimum-CA-Requirements-1.1.doc .

[AuM 01] AURA, T. und S. MÄKI: *Towards a survivable security architecture for ad-hoc networks*. In: *Proceedings Security Protocols, 9th International Workshop,*, Band 2467 der Reihe *LNCS*, Seiten 63–79, Cambridge, UK, April 2001. Springer, http://research.microsoft.com/users/tuomaura/Publications/aura-maki-protocols01.pdf .

[Autograph] MAMS: *Autograph — a personal privacy manager*, http://federation.org.au/Autograph .

[AzMa 02] AZZEDIN, F. und M. MAHESWARAN: *Towards trust-aware resource management in Grid computing systems*. In: *Cluster Computing and the Grid 2nd IEEE/ACM International Symposium CCGRID2002*, Seiten 452–457. IEEE ComSoc, 2002.

[BAK+ 02] BUTT, A.R., S. ADABALA, N.H. KAPADIA, R. FIGUEIREDO und J.A.B. FORTES: *Fine-Grain Access Control for Securing Shared Resources in Computational Grids*. In: *Proceedings of the Parallel and Distributed Processing Symposium (IPDPS 2002)*, Seiten 206–213. IEEE ComSoc, 2002.

[BAR+ 05] BÜCHLI, M. (ED.), V. APOSTOLESCU, H. REISER, H. DÖBBELING, R. NIEDERBERGER, O. MEXTORF, K. ULLMANN und D. VANDROMME: *DEISA Backbone Architecture*. Technischer Bericht V 1.0, DANTE, November 2005, http://www.net.deisa.fz-juelich.de/docs/DeisaBackboneArchitectureV1.0.pdf .

[BauVOGrid] *BauVOGrid — Grid-basierte Plattform für die virtuelle Organisation im Bauwesen*, http://www.d-grid.de/index.php?id=403 .

[BBF+ 06] BARTON, T., J. BASNEY, T. FREEMAN, T. SCAVO, F. SIEBENLIST, V. WELCH, R. ANANTHAKRISHNAN, B. BAKER, M. GOODE und K. KEAHEY: *Identity Federation and Attribute-based Authorization through the*

Globus Toolkit, Shibboleth, GridShib, and MyProxy. In: POLK, W. T., N. E. HASTINGS und K. SEAMONS (Herausgeber): *5th Annual PKI R&D Workshop — Making PKI Easy to Use; Proceedings*, Nummer NISTIR 7313, Seiten 54–67. National Institute of Standards and Technology (NIST), 2006, http://csrc.nist.gov/publications/nistir/ir-7313/NIST-IR-7313_Final.pdf .

[BBM+ 01] BALLINGER, K., P BRITTENHAM, A. MALHOTRA, W. A. NAGY und S. PHARIES: *Web Services Inspection Language (WS-Inspection)*. Technischer Bericht, IBM, 2001, http://www-128.ibm.com/developerworks/library/specification/ws-wsilspec/ .

[BDE+ 07] BÜCHNER, O., C. DOHMEN, T. EIFERT, H. ENKE, T. FIESELER, A. FRANK, S. FREITAG, A. GARCIA, C. GRIMM, W. GÜRICH, H. HELLER, T. JEIKAL, H. NITSCHE, O. SCHNEIDER und W. ZIEGLER: *Betriebskonzept für die D-Grid Infrastruktur*. Technischer Bericht Version 1.1c, D-Grid, Oktober 2007, http://www.d-grid.de/fileadmin/user_upload/documents/Kern-D-Grid/Betriebskonzept/D-Grid-Betriebskonzept.pdf .

[BDSG] *Bundesdatenschutzgesetz (BDSG) in der Fassung der Bekanntmachung vom 14. Januar 2003 (BGBl. I S. 66), zuletzt geändert durch Artikel 1 des Gesetzes vom 22. August 2006 (BGBl. I S. 1970)*, August 2006, http://www.gesetze-im-internet.de/bundesrecht/bdsg_1990/gesamt.pdf .

[BEF+ 04] BALLINGER, K., D. EHNEBUSKE, C. FERRIS, M. GUDGIN, C. K. LIU, M. NOTTINGHAM und P. YENDLURI: *WSI — Basic Profile Version 1.1*. Technischer Bericht, Web Services Interoperability Organization (WSI), August 2004, http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html .

[BgFGR 06a] BAUR, T., N. GENTSCHEN FELDE, M. GARSCHHAMMER und H. REISER: *Ergebnisse der Studie und Anforderungsanalyse in den Fachgebieten Monitoring, Accounting, Billing bei den Communities und Ressourcenanbietern im D-Grid*, Kapitel Monitoring — Anwendungsfälle und Anforderungen, Seiten 21–44. D-Grid, Juni 2006, https://www.d-grid.de/index.php?id=90 .

[BgFGR 06b] BAUR, T., N. GENTSCHEN FELDE, M. GARSCHHAMMER und H. REISER: *Ergebnisse der Studie und Anforderungsanalyse in den Fachgebieten Monitoring, Accounting, Billing bei den Communities und Ressourcenanbietern im D-Grid*, Kapitel Anforderungen an das Monitoring, Ergebnisse aus der Erhebung bei den Communities und Ressourcenanbietern im D-Grid, Seiten 45–64. D-Grid, Juni 2006, https://www.d-grid.de/index.php?id=90 .

[BgFGR 06c] BAUR, T., N. GENTSCHEN FELDE, M. GARSCHHAMMER und H. REISER: *Studie und Anforderungsanalyse in den Fachgebieten Monitoring, Accounting, Billing — Vorläufige Ergebnisse*, Kapitel Monitoring, Seiten 13–37.

D-Grid, Februar 2006, https://www.d-grid.de/index.php?id=90 .

[BHR 01] BRANDT, R., C. HÖRTNAGL und H. REISER: *Dynamically Adaptable Mobile Agents for Scaleable Software and Service Management*. Journal of Communications and Networks, 3(4):307–316, Dezember 2001, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=bhr01 .

[BoRe 07] BOURSAS, L. und H. REISER: *Propagating Trust and Privacy Aspects in Federated Identity Management Scenarios*. In: *Proceedings of the 14th Annual Workshop of HP Software University Association*, Band 2007, Munich, Germany, Juli 2007. Leibniz Supercomputing Center, https://www.nm.informatik.uni-muenchen.de/pub/Publikationen/bore07/ .

[Bren 02] BRENNER, M.: *Entwicklung eines Kriterienkatalogs zur Evaluierung des Anwender Supports in der BMW AG*. Diplomarbeit, Ludwig–Maximilians–Universität München, Januar 2002, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=bren02 .

[BrRe 01] BRANDT, R. und H. REISER: *Dynamic Adaptation of Mobile Agents in Heterogenous Environments*. In: PICCO, G. P. (Herausgeber): *Mobile Agents: Fifth International Conference, MA 2001, Proceedings*, Nummer 2230 in *Lecture Notes in Computer Science (LNCS)*, Seiten 70–87, Atlanta, Georgia, USA, Dezember 2001. Springer, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=brre01 .

[BRS 02] BRENNER, M., I. RADISIC und M. SCHOLLMEYER: *A Criteria Catalog based Methodology for Analyzing Service Management Processes*. In: FERIDUN, M., P. KROPF und G. BABIN (Herausgeber): *Proceedings of the 13th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM 2002)*, Lecture Notes in Computer Science (LNCS) 2506, Seiten 145–156, Montreal, Canada, Oktober 2002. IFIP/IEEE, Springer, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=brs02 .

[BuGe 03] BUTLER, R. und T. J. GENOVESE: *Global Grid Forum Certificate Policy Model*. Technischer Bericht GFD-C.16, Global Grid Forum, Juni 2003, www.ggf.org/documents/GFD/GFD-C.16.pdf .

[Bund] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *IT-Grundschutz*, http://www.bsi.bund.de/gshb/index.htm .

[Bund 04] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNOLOGIE (BSI): *Analyse Kritischer Infrastrukturen — Die Methode AKIS*. Technischer Bericht, BSI, 2004, http://www.bsi.bund.de/fachthem/kritis/acis_paper_d.pdf .

[Bund 05a] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *IT–Grundschutz–Vorgehensweise*. BSI-Standard 100-2 V 1.0, BSI, 2005, http://www.bsi.bund.de/literat/bsi_standard/standard_1002.pdf .

[Bund 05b] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI) (Herausgeber): *IT–Grundschutz-Kataloge*. Bundesanzeiger Verlag, 2005, http://www.bsi.de/gshb/deutsch/download/index.htm .

[Bund 05c] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *Managementsysteme für Informationssicherheit (ISMS)*. BSI-Standard 100-1 V 1.0, BSI, 2005, http://www.bsi.bund.de/literat/bsi_standard/standard_1001.pdf .

[Bund 05d] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *Risikoanalyse auf der Basis von IT-Grundschutz*. BSI-Standard 100-3 V 2.0, BSI, 2005, http://www.bsi.bund.de/literat/bsi_standard/standard_1002.pdf .

[Bund 07] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI) (Herausgeber): *IT–Grundschutz-Kataloge; Stand 9. Ergänzungslieferung*. Bundesanzeiger Verlag, 2007, http://www.bsi.bund.de/gshb/deutsch/download/it-grundschutz-kataloge_2007_de.pdf .

[Bund 08a] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *Notfallmanagement*. BSI-Standard 100-4 (Entwurf) V 0.7, BSI, 2008, http://www.bsi.bund.de/literat/bsi_standard/bsi-standard_100-4_v070.pdf .

[Bund 08b] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *Zertifizierung nach ISO 27001 auf der Basis von IT-Grundschutz — Prüfschema für ISO 27001-Audits*. Technischer Bericht, BSI, 2008, http://www.bsi.bund.de/gshb/zert/ISO27001/Pruefschema_V.2.1.pdf .

[BWBG+ 06] BLAKE-WILSON, S., N. BOLYARD, V. GUPTA, C. HAWK und B. MOELLER: *RFC 4492: Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)*. RFC, IETF, Mai 2006, ftp://ftp.isi.edu/in-notes/rfc4492.txt .

[BWE+ 00] BUTLER, R., V. WELCH, D. ENGERT, I. FOSTER, S. TUECKE, J. VOLMER und C. KESSELMAN: *A National–Scale Athentication Infrastructure*. IEEE Computer, Seiten 60–66, 2000, http://www.globus.org/research/papers.html#GSI1 .

[BWNH+ 06] BLAKE-WILSON, S., M. NYSTROM, D. HOPWOOD, J. MIKKELSEN und T. WRIGHT: *RFC 4366: Transport Layer Security (TLS) Extensions*. RFC, IETF, April 2006, ftp://ftp.isi.edu/in-notes/rfc4366.txt .

[C3-Grid] *C3-Grid — Über das Projekt*, http://www.c3grid.de/index.php?id=32 .

[Catl 02] CATLETT, CHARLIE: *TeraGrid: A Primer*. Technischer Bericht, TeraGrid Project, September 2002, http://www.teragrid.org/about/TeraGrid-Primer-Sept-02.pdf .

[CCIT-1] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *Common Criteria for Information Technology Security Evaluation — Part 1: Introduction and general model*. Technischer Bericht Version 3.1, Revision 1, BSI, September 2006, http://www.commoncriteriaportal.org/public/files/CCPART1V3.1R1.pdf .

[CCIT-2] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *Common Criteria for Information Technology Security Evaluation — Part 2: Security functional components*. Technischer Bericht Version 3.1, Revision 1, BSI, September 2006, http://www.commoncriteriaportal.org/public/files/CCPART2V3.1R1.pdf .

[CCIT-3] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI): *Common Criteria for Information Technology Security Evaluation — Part 3: Security assurance components*. Technischer Bericht Version 3.1, Revision 1, BSI, September 2006, http://www.commoncriteriaportal.org/public/files/CCPART3V3.1R1.pdf .

[CCMW 01] CHRISTENSEN, E., F. CURBERA, G. MEREDITH und S. WEERAWARANA: *Web Services Description Language (WSDL) 1.1*. W3C Recommendation, World Wide Web Consortium (W3C), März 2001, http://www.w3.org/TR/2001/NOTE-wsdl-20010315 .

[CCO+ 03] CANNON, S., S. CHAN, D. OLSON, C. TULL, V. WELCH und L. PEARLMAN: *Using CAS to Manage Role-Based VO Sub-Groups*. In: *Proceedings of Computing in High Energy Physics 03 (CHEP '03)*, März 2003, http://www.globus.org/alliance/publications/papers/CAS-group-CHEP03.pdf .

[CDS 08] CHAKRABARTI, A., A. DAMODARAN und S. SENGUPTA: *Grid Computing Security — A Taxonomy*. IEEE Security & Privacy, 6(1):44–51, Februar 2008.

[Ceki 02] CEKI GÜLCÜ: *Short introduction to log4j*. Technischer Bericht, Apache Software Foundation, März 2002, http://logging.apache.org/log4j/1.2/manual.html .

[CFS+ 03] CHOKHANI, S., W. FORD, R. SABETT, C. MERRILL und S. WU: *RFC 3647: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*. RFC, IETF, November 2003, ftp://ftp.isi.edu/in-notes/rfc3647.txt .

[Chau 85] CHAUM, D.: *Security without Identification: Transaction Systems to Make Big Brother Obsolete*. Communications of the ACM, 28(10):84–88, Oktober 1985.

[Chiv 03] CHIVERS, H.: *Grid Security: Problems and Potential Solutions*. Technischer Bericht YCS-2003-354, University of York, 2003, http://www.cs.york.ac.uk/ftpdir/reports/YCS-2003-354.pdf .

[ChLi 06] CHAPPELL, DAVE und LILY LIU: *Web Services Brokered Notification 1.3 (WS-BrokeredNotification)*. OASIS Standard, OASIS, Oktober 2006, http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf .

[ChOt 03] CHADWICK, D.W. und A. OTENKO: *The PERMIS X.509 Role Based Privilege Management Infrastructure*. Future Generation Computer Systems, 19(2):277–289, Februar 2003, http://www.cs.kent.ac.uk/pubs/2003/2109 .

[Chow 02] CHOWN, P.: *RFC 3268: Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)*. RFC, IETF, Juni 2002, ftp://ftp.isi.edu/in-notes/rfc3268.txt .

[chrbreak] NETBSD: *How to break out of a chroot environment*, http://wiki.netbsd.se/How_to_break_out_of_a_chroot_environment .

[chrjail] *How to break out of a* `chroot()` *jail*, http://www.bpfh.net/simes/computing/chroot-break.html .

[CHT 03a] CHANNABASAVAIAH, K., K. HOLLEY und E. M. TUGGLE: *Migrating to a service-oriented architecture, Part 1*. Technischer Bericht, IBM developer Works, Dezember 2003, http://www-128.ibm.com/developerworks/library/ws-migratesoa/ .

[CHT 03b] CHANNABASAVAIAH, K., K. HOLLEY und E. M. TUGGLE: *Migrating to a service-oriented architecture, Part 2*. Technischer Bericht, IBM developer Works, Dezember 2003, http://www-128.ibm.com/developerworks/library/ws-migratesoa2/ .

[CHvRR 04] CLEMENT, L., A. HATELY, C. VON RIEGEN und T. ROGERS: *UDDI Version 3.0.2*. Technischer Bericht, OASIS, Oktober 2004, http://uddi.org/pubs/uddi-v3.0.2-20041019.htm .

[Cias 04] CIASCHINI, V.: *A VOMS Attribute Certificate Profile for Authorization*. Technischer Bericht, DataGrid WP6 Authorization Working Group, Oktober 2004, http://infnforge.cnaf.infn.it/voms/AC-RFC.pdf .

[CIK 00] CHANG, F., A. ITZKOVITZ und V. KARAMCHETI: *User-level Resource-constrained Sandboxing*. In: *Proceedings of the USENIX Windows Systems Symposium*, Seiten 25–36, 2000, http://cs.nyu.edu/vijayk/papers/user-sandbox.pdf .

[CJK+ 04] CORNWALL, L., J. JENSEN, D. P. KELSEY, A. FROHNER, D. KOURIL, F. BONNASSIEUX, S. NICOUD, K. LÖRENTEY, J. HAHKALA, M. SILANDER, R. CECCHINI, V. CIASCHINI, L. DELL'AGNELLO, F. SPATARO, D. O'CALLAGHAN, O. MULMO, G. L. VOLPATO, D. L. GROEP,

M. STEENBAKKERS und A. MCNAB: *Authentication and Authorization Mechanisms for Multi-Domain Grid Environments*. Journal of Grid Computing, (2):301–311, 2004, http://www.nikhef.nl/grid/lcaslcmaps/publications/edg-security_paper.pdf .

[CLM 02] CRANOR, L., M. LANGHEINRICH und M. MARCHIORI: *A P3P Preference Exchange Language 1.0 (APPEL1.0)*. W3C Working Draft, W3C, April 2002, http://www.w3.org/TR/P3P-preferences/ .

[Clus] CLUSTER RESOURCES INC.: *TORQUE Resource Manager*, http://www.clusterresources.com/pages/products/torque-resource-manager.php .

[Comm 05] COMMUNITY MEDIZIN UND BIOMEDIZINISCHE INFORMATIK: *MediGRID: Projektantrag im Rahmen des D-Grid Vorprojektes überarbeiteter Neuantrag, Vers. 1.3*. Projektantrag zur Bekannmachung ëScience und GRID–Middleware zur Unterstützung wissenschaftlichen Arbeitens"des BmBF, Oktober 2005.

[Czyz 06] CZYZEWSKI, M.: *Management von Mitgliedschaften zu Virtuellen Organisationen im Grid*. Fortgeschrittenenpraktikum, Ludwig–Maximilians–Universität München, Mai 2006.

[D-Gr 07] D-GRID: *D-Grid — Einverständniserklärung*. Technischer Bericht Version 1.2, D-Grid, Februar 2007, http://www.fz-juelich.de/dgrid/AUP/D-Grid-User-AUP.pdf .

[D-Grid] *D–Grid Initiative*, http://www.d-grid.de/ .

[Dani 03] DANIELS, T.: *Security and Availability Policy for LCG*. Technischer Bericht Version 4c, LCG Security Group, 17.10. 2003, http://lcg.web.cern.ch/LCG/documents/GDB/Security%20Policy_T.Daniels_30.09.03.doc .

[Data 01] DATAGRID: *DataGrid Requirements for Grid-Aware Biology Applications*. Technischer Bericht DataGrid-10-D10.1, DataGRID, September 2001, https://edms.cern.ch/document/332412/3.8 .

[Data 02] DATAGRID: *Security Requirements and Testbed 1 Security Implementation*. Technischer Bericht DataGrid-07-D7.5, DataGRID, Mai 2002, https://edms.cern.ch/document/340234/4.0 .

[DataGRID] *The DataGrid Project*, http://eu-datagrid.web.cern.ch/eu-datagrid/ .

[DEF+ 06] DUSSA, T., U. EPTING, B. FILIPOVIC, G. FOEST, J. GLOWA, G. GÖTZE, C. GRIMM, M. HILLENBRAND, C. KOHLSCHÜTTER, R. LOHNER, S. MAKEDANZ, P. MÜLLER, M. PATTLOCH, S. PIEGER, T. STRAUB und J. WIEBELITZ: *Analyse der AA-Infrastrukturen im Grid-Middleware*. Technischer Bericht Version 1.1, D-Grid Integrationsprojekt, Fachgebiet 3-4, März 2006, https://www.d-grid.de/fileadmin/user_upload/documents/DGI-FG3-4/Analyse-AAI_v1_1.pdf .

[DEIS] DEISA: *Security*, http://www.deisa.org/organisation/security.php .

[DEISA] *DEISA — Distributed European Infrastrucutre for Supercomputing Applications*, http://www.deisa.org .

[DFN- 06] DFN-VEREIN: *Certification Practice Statement of the Public Key Infrastructure in the Deutsche Forschungsnetz — Grid*. Technischer Bericht Grid-CPS V1.3, DFN-Verein, November 2006, https://www.pki.dfn.de/fileadmin/PKI/DFN-PKI_grid-cps_v13.pdf .

[DFN-PKI] *DFN-PKI Policy Certification Authority (DFN-PKI PCA)*, http://www.pca.dfn.de/ .

[DHS 06] DREO RODOSEK, G., H.-G. HEGERING und B. STILLER: *Dynamic Virtual Organisations as Enablers for Managed Invisible Grids*. In: *Proceedings of the 2006 IEEE/IFIP Network Operations and Management Symposium (NOMS)* , Band 2006, Vancouver, Canada, April 2006. IEEE/IFIP.

[DiAl 99] DIERKS, T. und C. ALLEN: *RFC 2246: The TLS Protocol Version 1.0*. RFC, IETF, Januar 1999, ftp://ftp.isi.edu/in-notes/rfc2246.txt .

[DIC] *DICOM — Digital Imaging and Communications in Medicine*, http://dicom.nema.org/ .

[DICO 07] DICOM – DIGITAL IMAGING AND COMMUNICATIONS IN MEDICINE: *Strategic Document — Version 7.1*. Technischer Bericht, NEMA Diagnostic Imaging and Therapy Systems Division, März 2007, http://medical.nema.org/dicom/geninfo/Strategy.pdf .

[DiRe 06] DIERKS, T. und E. RESCORLA: *RFC 4346: The Transport Layer Security (TLS) Protocol Version 1.1*. RFC, IETF, April 2006, ftp://ftp.isi.edu/in-notes/rfc4346.txt .

[DSMS 06] DREPPER, J., S. C. SEMLER, Y. MOHAMMED und U. SAX: *Akruelle Themen des Datenschutzes und der Datensicherheit in der biomedizinischen Forschung*, Kapitel 2, Seiten 25–36. In: SAX, U. et al. [SMVR 06], 2006.

[EaJo 01] EASTLAKE 3RD, D. und P. JONES: *RFC 3174: US Secure Hash Algorithm 1 (SHA1)*. Technischer Bericht, September 2001, ftp://ftp.isi.edu/in-notes/rfc3174.txt .

[Ecke 06] ECKERT, C.: *IT–Sicherheit: Konzepte — Verfahren — Protokolle*. Oldenbourg, München, 4 Auflage, 2006.

[EFL+ 99] ELLISON, C., B. FRANTZ, B. LAMPSON, R. RIVEST, B. THOMAS und T. YLONEN: *RFC 2693: SPKI Certificate Theory*. RFC, IETF, September 1999, ftp://ftp.isi.edu/in-notes/rfc2693.txt .

[EGA] *Enterprise Grid Alliance*, http://www.gridalliance.org .

[EGEEa] *EGEE (Enabling Grids for E-sciencE)*, http://www.eu-egee.org .

[EGEE b] EGEE: *EGEE results*, http://www.eu-egee.org/introduction/results .

[EGEEc] *Enabling Grids for E–science in Europe (EGEE)*. http://public.eu-egee.org.

[EGEE-JRA3] *EGEE JRA3: Security*, http://egee-jra3.web.cern.ch/egee-jra3/ .

[EGEE-Tec] *EGEE Technical Pages*. http://egee-intranet.web.cern.ch/.

[EGEE 04a] EGEE: *EGEE Middleware Architecture and Planning (Release 1)*. Technischer Bericht, CERN, August 2004, https://edms.cern.ch/document/476451/ .

[EGEE 04b] EGEE MIDDLEWARE ENGINEERING AND INTEGRATION JRA1: *Design of the EGEE Middleware Grid Services*. Technischer Bericht Version 1-0, EGEE, Oktober 2004, https://edms.cern.ch/document/487871 .

[EGEE 04c] EGEE SECURITY JRA3: *EGEE Activity User Requirements*. Technischer Bericht Version 1-0, EGEE, August 2004, https://edms.cern.ch/document/485295 .

[EGEE 04d] EGEE SECURITY JRA3: *EGEE Global Security Architecture for web and legacy services*. Technischer Bericht Version 1-1, EGEE, September 2004, https://edms.cern.ch/file/487004/1.1/EGEE-JRA3-TEC-487004-DJRA3.1-1.1.pdf . Document identifier: EGEE-JRA3-TEC-487004-DJRA3-v-1-1.

[EGEE 05a] EGEE: *EGEE Middleware Architecture and Planning (Release 2)*. Technischer Bericht, CERN, Juli 2005, https://edms.cern.ch/document/594698/ .

[EGEE 05b] EGEE SECURITY JRA3: *EGEE Global Security Architecture for web and legacy services*. Technischer Bericht Version 1-2, EGEE, September 2005, https://edms.cern.ch/file/602183/1.3/EGEE-JRA3-TEC-602183-DJRA3.3-1.2.pdf . Document identifier: EGEE-JRA3-TEC-602183-DJRA3-v-1-2.

[Elli 99] ELLISON, C.: *RFC 2692: SPKI Requirements*. RFC, IETF, September 1999, ftp://ftp.isi.edu/in-notes/rfc2692.txt .

[Enge 07] ENGEL, T.: *Evaluierung und Positionierung biometrischer Authentisierungsverfahren bei der BMW Group*. Diplomarbeit, Ludwig–Maximilians–Universität München, März 2007.

[Erwi 00] ERWIN, D. (Herausgeber): *UNICORE — Uniformes Interface für Computer–Ressourcen*. UNICORE Forum, 2000, http://www.unicore.org/forum/documents.htm .

[Erwi 03] ERWIN, D. (Herausgeber): *UNICORE Plus Final Report — Uniform Interface to Computing Ressources*. Unicore Forum e.V., 2003, http://www.unicore.org/forum/documents.htm .

[EU P 04a] EU POLICY MANAGEMENT AUTHORITY FOR GRID AUTHENTICATION IN E-SCIENCE: *Accreditation Procedures*. Technischer Bericht Version 1.0, EUGridPMA, April 2004, http://eugridpma.org/guidelines/EUGridPMA-accreditation-20040402-1-0.pdf .

[EU P 04b] EU POLICY MANAGEMENT AUTHORITY FOR GRID AUTHENTICATION IN E-SCIENCE: *EU Grid PMA Charter*. Technischer Bericht Version 1.0, EUGridPMA, April 2004, http://eugridpma.org/charter/EUGridPMA-charter-20040401-1-0.pdf .

[EUGrid PMA] *The EU Grid PMA*, http://www.eugridpma.org/ .

[FaHo 02] FARRELL, S. und R. HOUSLEY: *RFC 3281: An Internet Attribute Certificate Profile for Authorization*. RFC, IETF, April 2002, ftp://ftp.isi.edu/in-notes/rfc3281.txt .

[FBSR 06] FLIEGL, D., T. BAUR, B. SCHMIDT und H. REISER: *Ein generisches Intrusion Prevention System mit dynamischer Bandbreitenbeschränkung*. In: MÜLLER, P., G. PETER und E. JESSEN (Herausgeber): *20. DFN–Arbeitstagung über Kommunikationsnetze*, Seiten 219–230, Heilbronn, Juni 2006. .

[FIPS 197] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST): *Advanced Encryption Standard (AES)*. Federal Information Processing Standards 197, U.S. Department of Commerce, Gaithersburg, November 2001, http://csrc.nist.gov/encryption/aes/ .

[FIPS 46-2] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST): *Data Encryption Standard (DES)*. Federal Information Processing Standards 46-2, U.S. Department of Commerce, Gaithersburg, Dezember 1993, http://www.itl.nist.gov/fipspubs/fip46-2.htm .

[FKK+ 97] FOSTER, I., N. T. KARONIS, C. KESSELMAN, G. KOENIG und S. TUECKE: *A Secure Communications Infrastructure for High–Performance Distributed Computing*. In: *Proceedings of the 6th International Symposium on High Performance Distributed Computing (HPDC '97)*, Seiten 125–136. IEEE Computer Society, 1997, http://www.globus.org/research/papers.html#Zipper .

[FKKT 98] FOSTER, I., N. T. KARONIS, C. KESSELMANN und S. TUECKE: *Managing Security in High-Performance Distributed Computations*. Cluster Computing, 1(1):95–107, 1998, http://www.globus.org/research/papers.html#cc-security .

[FKS+ 05] FOSTER, I., H. KISHIMOTO, A SAVVA, D. BERRY, A. DAJAOUI, A. GRIMSHAW, B. HORN, F. MACIEL, F. SIEBENLIST, R. SUBRAMMANIAM, J. TREADWELL und J. VON REICH: *The Open Grid Services Architecture, Version 1.0*. Technischer Bericht GFD-I.030, Global Grid Forum, Januar 2005, http://www.ggf.org/documents/GFD.30.pdf .

[FKT 01] FOSTER, I., C. KESSELMAN und S. TUECKE: *The Anatomy of the Grid — Enabling Scalalable Virtual Organizations*. Intl. Journal of High Performance Computing Applications, Seiten 200–222, 2001, http://www.globus.org/research/papers.html#anatomy .

[FKTT 98] FOSTER, I., C. KESSELMANN, G. TSUDIK und S. TUECKE: *A Security Architecture for Computational Grids*. In: *Proc. 5th ACM Conference on Computer and Communications Security Conference*, Seiten 83–92, 1998, http://www.globus.org/research/papers.html#security-arch .

[FoCh 05] FOSTER, IAN und LISA CHILDERS: *Introduction to GT4*. Tutorial at the APAC Conference and Exhibition on Advanced Computing, Grid Applications and eResearch (APAC'05), Royal Pines Resort, Gold Coast, Australien, September 2005, http://www.globus.org/toolkit/tutorials/BAS/APAC/APACGlobusIntro.pdf .

[FoKe 99] FOSTER, I. und C. KESSELMANN (Herausgeber): *The Grid: Bluprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, 1999.

[Fost 05] FOSTER, I.: *A Globus Primer*. Draft 0.6, Globus Alliance, August 2005, http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf .

[FoWa 03a] FOX, G. und D. WALKER: *Appendix: UK Grid Services and Activities — e-Science Gap Analysis*. Technischer Bericht, National e-Science Centre, June 2003, http://www.nesc.ac.uk/technical_papers/UKeS-2003-01/Appendix30June03.pdf .

[FoWa 03b] FOX, G. und D. WALKER: *e-Science Gap Analysis*. Technischer Bericht, National e-Science Centre, June 2003, http://www.nesc.ac.uk/technical_papers/UKeS-2003-01/GapAnalysis30June03.pdf .

[FrCi 04] FROHNER, A. und V. CIASCHINI: *VOMS Credential Format*. Technischer Bericht, DataGRID, Februar 2004, http://edg-wp2.web.cern.ch/edg-wp2/security/voms/edg-voms-credential.pdf .

[Froh 04a] FROHNER, A.: *EDG-VOMS-Admin Install Guide*. Technischer Bericht, DataGRID, Februar 2004, http://edg-wp2.web.cern.ch/edg-wp2/security/voms/edg-voms-admin-install-guide.pdf .

[Froh 04b] FROHNER, A.: *EGD-VOMS-Admin User's Guide*. Technischer Bericht, DataGRID, Februar 2004, http://edg-wp2.web.cern.ch/edg-wp2/security/voms/edg-voms-admin-user-guide.pdf .

[GDI-Grid] *GDI-Grid - Geodateninfrastruktur-Grid — Effiziente Erschließung und Prozessierung von Geodaten für die Geosimulation von Lärmausbreitung und Katastrophenfällen*, http://www.gdi-grid.de/ .

[Geno 05] GENOVESE, T. J.: *Profile for Short Lived Credential Services X.509 Public Key Certification Authorities with secured infrastructure*. Technischer Bericht Version 1.1, IGTF; TAGPMA, November 2005, http://eugridpma.org/guidelines/SLCS/IGTF-AP-SLCS-20051115-1-1.pdf .

[Geno 06] GENOVESE, T. J.: *Profile for Member Integrated X.509 Credential Services with Secured Infrastructure*. Technischer Bericht Version 1.0c, TAGPMA, Oktober 2006, http://eugridpma.org/igtf/IGTF-AP-MICS-draft-1.1c.pdf .

[GGFa] *Global Grid Forum*. http://www.gridforum.org/.

[GGFb] *Global Grid Forum*, http://www.ggf.org .

[GGG+ 07] GIETZ, PETER, CHRISTIAN GRIMM, RALF GRÖPER, MARTIN HAASE, SIEGFRIED MAKEDANZ, HANS PFEIFFENBERGER und MICHAEL SCHIFFERS: *IVOM: Interoperability and Integration of VO Management Technologies in D-Grid — Work Package 3: A Concept for Authorization on D-Grid Resources*. Technischer Bericht V 1.0, D-Grid, September 2007, http://www.d-grid.de/fileadmin/user_upload/documents/DGI-FG1-IVOM/AP3-Report-v1.0.pdf .

[GHM+ 03a] GUDGIN, M., M. HADLEY, N. MENDELSOHN, J.-J. MOREAU und H. F. NIELSEN: *SOAP Version 1.2 Part 1: Messaging Framework*. W3C Recommendation, World Wide Web Consortium (W3C), Juni 2003, http://www.w3.org/TR/2003/REC-soap12-part1-20030624/ .

[GHM+ 03b] GUDGIN, M., M. HADLEY, N. MENDELSOHN, J.-J. MOREAU und H. F. NIELSEN: *SOAP Version 1.2 Part 2: Adjuncts*. W3C Recommendation, World Wide Web Consortium (W3C), Juni 2003, http://www.w3.org/TR/2003/REC-soap12-part2-20030624/ .

[GHM 06] GRAHAM, STEVE, DAVID HULL und BRYAN MURRAY: *Web Services Base Notification 1.3 (WS-BaseNotification)*. OASIS Standard, OASIS, Oktober 2006, http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf .

[GHR 99] GRUSCHKE, B., S. HEILBRONNER und H. REISER: *Mobile Agent System Architecture — Eine Plattform für flexibles IT–Management*. Technischer Bericht 9902, IFI, August 1999, http://wwwmnmteam.informatik.uni-muenchen.de/php-bin/pub/show_pub.php?key=ghr99 .

[Giem 00] GIEMSA, F.: *Evaluation von Outsourcing–Beziehungen für die IT–Hotline der BMW AG*. Diplomarbeit, Ludwig–Maximilians–Universität München, November 2000, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=giem00 .

[Glob a] GLOBUS ALLIANCE: *About the Globus Toolkit*, http://www.globus.org/toolkit/about.html .

[Glob b] GLOBUS ALLIANCE: *Firewall HowTo*, http://dev.globus.org/wiki/FirewallHowTo .

[Glob c] GLOBUS ALLIANCE: *GT 4.0 Java WS Core : Developer's Guide*, http://www.globus.org/toolkit/docs/4.0/common/javawscore/developer-index.html .

[Glob d] GLOBUS ALLIANCE: *GT 4.0: OGSA-DAI*, http://www.globus.org/toolkit/docs/4.0/techpreview/ogsadai/ .

[Glob e] GLOBUS ALLIANCE: *Large-scale Data Replication for LIGO*, http://www.globus.org/solutions/data_replication/ .

[Glob f] GLOBUS ALLIANCE: *WS-GRAM Audit Logging*, http://www.globus.org/toolkit/docs/4.0/execution/wsgram/WS_GRAM_Audit_Logging.html .

[Glob 02] GLOBUS ALLIANCE: *Globus Toolkit Developer Tutorial - Grid Security Infrastructure*, Februar 2002, http://www-fp.globus.org/about/events/US_tutorial/slides/index.html .

[Glob 03] GLOBUS ALLIANCE: *Globus Toolkit History*, 2003, http://www.globus.org/toolkit/presentations/GThistory2.ppt .

[globus] *The globus alliance*. http://www.globus.org/.

[Glow 06] GLOWKA, J.: *Die Sicherheitsstrukturen bei LCG 2*. Technischer Bericht, Forschungszentrum Karlsruhe – IWR, 26.01. 2006, http://www.d-grid.de/fileadmin/user_upload/documents/DGI-FG1.4/documentation/LCG-Security_DE.pdf#search=%22Glowka%20Sicherheitsstrukturen%22 .

[GMNE 07] GRÜNTER, E., M. MEIER, R. NIEDERBERGER und T. EICKERMANN: *Firewall-Testszenarien — Methoden, Ausführungen und Auswertungen*. Technischer Bericht; DGI Fachgebiet 3-5, DGI, Januar 2007, http://www.d-grid.de/fileadmin/user_upload/documents/DGI-FG3-5/Firewall-Testszenarien.pdf .

[GMNP 06] GRÜNTER, E., M. MEIER, R. NIEDERBERGER und F. PETRI: *Dynamic Configuration of Firewalls Using UDP Hole Punching*. Technical Report IB-2006-13, Forschungszentrum Jülich; Zentralinsitut für angewandte Mathematik (ZAM), Jülich, 2006, http://www.fz-juelich.de/zam/files/docs/ib/ib-06/ib-2006-13.pdf .

[googleWS] *Google Web APIs — Develop Your Own Application Using Google*, www.google.com/apis/ .

[GORP 04] GERLONI, H., B. OBERHAITZINGER, H. REISER und J. PLATE: *Praxisbuch Sicherheit für Linux–Server und -Netze*. Hanser, Mai 2004, http://www.nm.ifi.lmu.de/~sicherheitsbuch . ISBN 3–446–22626–5, 430 p.

[GPH 03] GOLBECK, JENNIFER, BIJAN PARSIA und JAMES HENDLER: *Trust Networks on the Semantic Web*. In: *Proceedings of Cooperative Intelligent Agents 2003*, Nummer 2782 in *LNCS*, Seiten 238–249, Helsinki, Finland, August 2003. , http://www.mindswap.org/papers/CIA03.pdf .

[GPR 06] GRIMM, C., M. PATTLOCH und H. REISER: *Sicherheit in Grids*. PIK — Praxis der Informatikonsverarbeitung und Kommunikation, 2006(29/3):159–164, Juli 2006.

[GPW 06] GRIMM, C., S. PIGER und J. WIEBELITZ: *Evaluation von Security-Mechanismen in Grid-Umgebungen*. In: *20. DFN-Arbeitstagung*. DFN, Juni 2006, http://dfn2006.uni-kl.de/programm/mittwoch.shtml .

[Gree 06] GREENE, S. STERN: *Security Policies and Procedures — Principles and Practices*. ISBN: 0-13-186691-5. Pearson Prentice Hall, 2006.

[GrGr 02] GRÖPNER, R. und C. GRIMM: *Aktuelle Entwicklungen zu GridShib*, März 2002, http://medigrid.de/u_veranst/070327security-ws/v15_ivom_gridshib_presentation_1.00.pdf . 2. D-Grid Security Workshop; Göttingen.

[GridKA-CA] *Zertifizierungsstelle GridKA-CA im Forschungszentrum Karlsruhe (GermanGrid)*, http://grid.fzk.de/cgi-bin/welcome_ca.pl .

[grid.org] *GRID.ORG – Grid Computing Projects*. http://www.grid.org/.

[GridPMA] *International Grid Trust Federation — Grid Policy Management Authority*, http://www.gridpma.org/ .

[GridShib] UNIVERSITY OF CHICAGO: *GridShib – A NSF-funded project between NCSA and the University of Chicaco*, http://gridshib.globus.org/ .

[GRIP] *GRid Interoperability Project*, http://www.grid-interoperability.org/index.html .

[Groe] GROEP, D. L.: *International Grid Trust Federation – towards worldwild interoperability in identity management*. UK Presidency 2005 e-IRG Meeting, www.e-irg.org/meetings/2005-UK/eugridpma-eirg-20051213.ppt .

[Groe 05] GROEP, D.: *International Grid Trust Federation*. Technischer Bericht, IGTF, Oktober 2005, http://www.gridpma.org/IGTF-Federation-Constitution.pdf .

[Groe 06a] GROEP, D.: *Authentication Profile for Classic X.509 Public Key Certification Authorities with secured infrastructure*. Technischer Bericht Version 4.1, IGTF; EUGridPMA, Dezember 2006, http://eugridpma.org/guidelines/IGTF-AP-classic-4-1.pdf .

[Groe 06b] GROEP, D.: *Authentication Profile for Classic X.509 Public Key Certification Authorities with secured infrastructure*. Technischer Bericht Version 4.0, IGTF; EUGridPMA, Oktober 2006, http://www.eugridpma.org/guidelines/IGTF-AP-classic-20050930-4-0.pdf .

[GRSW 06] GEMMILL, JILL, JOHN-PAUL ROBINSON, TOM SCAVO und VON WELCH. Internet2 Member Meeting, April 2006, http://grid.ncsa.uiuc.edu/presentations/i2mm-myvocs-gridshib-april06.ppt .

[Grün 07] GRÜNTER, E.: *Firewalls und Grids – Herausforderungen oder "Quadratur des Kreises"?* In: PAULSEN, C. (Herausgeber): *14. Workshop Sicherheit in vernetzten Systemen*, Seiten F1 – F13, Hamburg, Februar 2007. DFN CERT, DFN CERT Services GmbH.

[GuGr 04] GU, Y. und R. L. GROSSMAN: *UDT: A Transport Protocol for Data Intensive Applications*. Internet Draft draft-gg-udt-01.txt, IETF, Chicago, August 2004, http://udt.sourceforge.net/doc/draft-gg-udt-01.txt .

[GuGr 07] GU, Y. und R. L. GROSSMAN: *UDT: UDP-based Data Transfer for High-Speed Wide Area Networks* . Computer Networks: The International Journal of Computer and Telecommunications Networking, 51(3):1777–1799, Mai 2007, http://www.ncdm.uic.edu/publications/files/journal-035.pdf .

[HAN 99] HEGERING, H.-G., S. ABECK und B. NEUMAIR: *Integrated Management of Networked Systems — Concepts, Architectures and their Operational Application*. Morgan Kaufmann Publishers, ISBN 1–55860–571–1, Januar 1999. 651 p.

[HaRe 00] HAUCK, R. und H. REISER: *Monitoring Quality of Service across Organizational Boundaries*. In: *Trends in Distributed Systems: Towards a Universal Service Market. Proceedings of the third International IFIP/GI Working Conference, USM 2000*, September 2000, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=hare00 .

[HaRe 99] HAUCK, R. und H. REISER: *Monitoring of Service Level Agreements with flexible and extensible Agents*. In: *Workshop of the OpenView University Association (OVUA 99)*, Bologna, Italy, Juni 1999. , http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/6_HPOVUAWS/Papers/hauck_reiser.pdf .

[Hege 04] HEGERING, H.-G.: *D-Grid: Schritte zu einer nationalen e-Science-Initiative*. In: KNOP, J. VON, W. HAVERKAMP und E. JESSEN (Herausgeber): *E-Science und Grid, Ad-hoc-Netze, Medienintegration*, Band P-55 der Reihe *Lecture Notes in Informatics (LNI)*, Seiten 267–284, Bonn, 2004. GI-Edition.

[Heth 07] HETHMON, P.: *RFC 3659: Extensions to FTP*. RFC, IETF, März 2007, ftp://ftp.isi.edu/in-notes/rfc3659.txt .

[HKLR 03] HEGERING, H.-G., A. KÜPPER, C. LINNHOFF-POPIEN und H. REISER: *Management Challenges of Context–Aware Services in Ubiquitous Environments*. In: *Self–Managing Distributed Systems; 14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2003, Heidelberg, Germany, October 2003, Proceedings*, Nummer LNCS 2867, Seiten 246–259, Heidelberg, Germany, Oktober 2003. Springer, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=hklr03 .

[Holl 04] HOLLENBECK, S.: *RFC 3749: Transport Layer Security Protocol Compression Methods*. RFC, IETF, Mai 2004, ftp://ftp.isi.edu/in-notes/rfc3749.txt .

[HoLu 97] HOROWITZ, M. und S. LUNT: *RFC 2228: FTP Security Extensions*. RFC, IETF, Oktober 1997, ftp://ftp.isi.edu/in-notes/rfc2228.txt .

[Homm 05a] HOMMEL, W.: *Using XACML for Privacy Control in SAML–based Identity Federations*. In: *9th IFIP TC–6 TC–11 Conference on Communications and Multimedia Security (CMS 2005)*. Springer, September 2005.

[Homm 07] HOMMEL, W.: *Architektur- und Werkzeugkonzepte für föderiertes Identitäts–Management*. Dissertation, Ludwig–Maximilians–Universität München, Juli 2007.

[HoRe 05] HOMMEL, W. und H. REISER (Herausgeber): *Federated Identity Management in B2B Outsourcing*, Band 2005 der Reihe *Proceedings of the 12th Annual HPOVUA Workshop*, Porto, Portugal, Juli 2005. .

[HoRe 05a] HOMMEL, W. und H. REISER: *Federated Identity Management: Shortcomings of existing standards*. In: CLEMM, A., O. FESTOR und A. PRAS (Herausgeber): *9th IFIP/IEEE International Symposium on Integrated Network Managment (IM 2005) — Managing New Networked Worlds*, Nice, France, Mai 2005. IEEE, IEEE, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=hore05a .

[HoRe 05b] HOMMEL, W. und H. REISER: *Federated Identity Management: Die Notwendigkeit zentraler Koordinationsdienste*. In: *Kommunikation in Verteilten Systemen (KiVS)*, P–61, Seiten 65–72, Kaiserslautern, Germany, März 2005. Gesellschaft für Informatik, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=hore05 .

[HPFS 02] HOUSLEY, R., W. POLK, W. FORD und D. SOLO: *RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC, IETF, April 2002, ftp://ftp.isi.edu/in-notes/rfc3280.txt .

[HuRe 04] HURAJ, L. und H. REISER: *Efficient Verification of Delegation in Distributed Group Membership Management*. In: *Research Directi-*

ons in Data and Applications Security XVIII; IFIP TC11/WG11.3 Eighteenth Annual Conference on Data and Applications Security;, Seiten 265–280, Sitges, Catalonia, Spain, Juli 2004. Kluwer Academic Publishers, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=hure04 .

[HuTh 01] HUMPHREY, M. und M. THOMPSON: *Security Implications of Typical Grid Computing Usage Scenarios*. In: *10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10 01)*, Seiten 95–103. IEEE ComSoc, 2001.

[IBM 05] IBM RESEARCH (Herausgeber): *Service Oriented Architecture*, Band 44 der Reihe *IBM Systems Journal*, Yorktown Heights, NY, USA, 2005. , http://www.research.ibm.com/journal/sj44-4.html .

[IBMi 02] IBM COOPERATION und MICROSOFT: *Security in a Web Services World: A Proposed Architecture and Roadmap*. Whitepaper Version 1.0, IBM, April 2002, http://www-128.ibm.com/developerworks/library/specification/ws-secmap/ .

[IETF] IETF: *IETF Working Group: Transport Layer Security (tls)*, http://www.ietf.org/html.charters/tls-charter.html .

[IPY] *International Polar Year 2007 – 2008*, http://www.ipy.org/ .

[ISO 10181-1] *Information Technology – Open Systems Interconnection – Security Frameworks in Open Systems – Overview*. IS 10181-1, International Organization for Standardization and International Electrotechnical Committee, November 1995.

[ISO 20000-1] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Information technology — Service management — Part 1: Specification*. ISO/IEC 20000-1:2005, International Organization for Standardization and International Electrotechnical Committee, 2005.

[ISO 20000-2] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Information technology — Service management — Part 2: Code of pracitce*. ISO/IEC 20000-2:2005, International Organization for Standardization and International Electrotechnical Committee, 2005.

[ISO 270001] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Information technology — Security techniques — Information security management systems — Requirements*. ISO/IEC 270001:2005, International Organization for Standardization and International Electrotechnical Committee, 2005.

[ISO 7498] *Information Processing Systems – Open Systems Interconnection – Basic Reference Model*. IS 7498, International Organization for Standardization and International Electrotechnical Committee, 1984.

[ITSEC] DEPARTMENT OF TRADE AND INDUSTRY: *Information Technology Security Evaluation Criteria (ITSEC)*. Technischer Bericht Version 1.2, Department of Trade and Industry, 1991, http://www.bsi.de/zertifiz/itkrit/itsec-en.pdf .

[ITU- 93] ITU–T: *X.500 – Information technology – Open systems Interconnection – The Directory: Overview of concepts, models and services*. Technischer Bericht, International Telecommunication Union, November 1993.

[IVOM] *D-Grid Projekte — IVOM — Interoperabilität und Integration der VO-Management Technologien im D-Grid*, http://www.d-grid.de/index.php?id=314 .

[jails] KAMP, P.-H. und R. N. M. WATSON: *Jails: Confining the omnipotent root*. Technischer Bericht, The FreeBSD Project, http://docs.freebsd.org/44doc/papers/jail/jail.html .

[JAP] *JAP Anonymity & Privacy — Projekt AN.ON — Anonymität.Online*, http://anon.inf.tu-dresden.de/ .

[Jaqu 07] JAQUITH, A.: *Security Metrics — Replacing Fear, Uncertainty and Doubt*. ISBN-13: 978-0-321-34998-9. Addison-Wesley; Pearson Education, 2007.

[JeGr 06] JENSEN, J. und D. GROEP: *IGTF Policy for High-Level Certification Authorities — Non-Entity Issuing CAs*. Technischer Bericht Version 0.2, IGTF; EUGridPMA, August 2006, http://www.eugridpma.org/guidelines/igtf-policy-hlca-0.2.pdf .

[Join 06a] JOINT SECURITY POLICY GROUP: *Grid Acceptable Use Policy; V 3.1*. Technischer Bericht, LCG, EGEE, September 2006, https://edms.cern.ch/document/428036 .

[Join 06b] JOINT SECURITY POLICY GROUP: *Grid Security Policy; V 5.4*. Technischer Bericht, LCG, EGEE, Dezember 2006, https://edms.cern.ch/document/428008 .

[KBC 97] KRAWCZYK, H., M. BELLARE und R. CANETTI: *RFC 2104: HMAC: Keyed-Hashing for Message Authentication*. RFC, IETF, Februar 1997, ftp://ftp.isi.edu/in-notes/rfc2104.txt .

[KeRe 99] KELLER, A. und H. REISER: *Dynamic Management of Internet Telephony Servers: A Case Study based on JavaBeans and JDMK*. In: *Proceedings Third International Enterprise Distributed Object Computing Conference (EDOC 99)*, Seiten 135–146, Mannheim, Germany, September 1999. IEEE Publishing, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=kere99 .

[KeWe 02] KEAHEY, K. und V. WELCH: *Fine-Grain Authorization for Resource Management in the Grid Environment*. In: *Grid Computing - GRID 2002 : Third International Workshop*, Band 2536 der Reihe *LNCS*, Baltimore, USA, November 2002. Springer, http://www.globus.org/alliance/publications/papers/gauth02.pdf .

[Kim 02] KIM, S.: *Security in Computational Grid*. Informal Security Seminar, University of Minnesota, Oktober 2002, http://www.cs.umn.edu/research/sclab/Slides/grid.ppt .

[KoRe 07] KORNBERGER, R. und H. REISER: *"Die Suche nach der Nadel im Heuhaufen" — Nyx — Ein System zur Lokalisierung von Rechnern in großen Netzwerken anhand IP- oder MAC–Adressen*. In: *21. DFN Arbeitstagung über Kommunikationsnetze*, Kaiserslautern, Juni 2007. .

[Kren 05] KRENEK, A.: *EGEE User's Guide — Service Logging and Bookkeeping (L&B)*. Technischer Bericht EGEE-JRA1-TEC-571273, EGEE, Dezember 2005, https://edms.cern.ch/file/571273/2/LB-guide.pdf .

[KRRV 01] KEMPTER, B., H. REISER, H. ROELLE und G. VOGT: *Implementierung eines MASIF konformen Agentensystems — Die Mobile Agent System Architecture (MASA) — .* PIK — Praxis der Informationsverarbeitung und Kommunikation, 24(3):141–148, September 2001, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=krrv01 .

[KRS 04] KÜPPER, A., H. REISER und M. SCHIFFERS: *Mobilitätsmanagement im Überblick: Von 2G zu 3,5G*. PIK — Praxis der Informationsverarbeitung und Kommunikation, 04(2):68–73, Juni 2004, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=krs04 .

[KRSS 06] KARSCH, S., H. REISER, H. L. STAHL und R. SKERKA: *Vorstudie Grid Sicherheits-Infrastruktur (GSI)*. Technischer Bericht, Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn, August 2006.

[KWL+ 04] KEAHEY, K., V. WELCH, S. LANG, B. LIU und S. MEDER: *Fine-Grained Authorization for Job Execution in the Grid: Design and Implementation*. Concurrency and Computation: Practice & Experience, 16(5):477–488, 2004.

[KZL+ 01] KONG, J., P. ZERFOS, H. LUO, S. LU und L. ZHANG: *Providing robust and ubiquitous security support for mobile ad-hoc networks*. In: *Proc. 9th IEEE International Conference on Network Protocols (ICNP)*, Seiten 251–260, 2001.

[LeCa 99] LEVI, A. und M. U. CAGLAYAN: *Analytical Performance Evaluation of Nested Certificates*. In: *1999 Conference of IFIP Working Group 7.3 on Computer System Modeling and Performance Evaluation , INCONNU (1999)*, Band 36-37, Seiten 213–232. Elsevier, 1999.

[LHC] *LHC Computing Grid*. http://lcg.web.cern.ch.

[log4j] APACHE SOFTWARE FOUNDATION: *Apache log4j*, http://logging.apache.org/log4j/ .

[LRg 08] LINDINGER, T., H. REISER und N. GENTSCHEN FELDE: *Virtualizing an IT–Lab for Higher Education Teaching*. In: *Tagungsband zum 1. GI/ITG KuVS Fachgespräch Virtualisierung*, Band 2008, Seiten 97–104, Paderborn, Deutschland, Februar 2008. Gesellschaft für Informatik e.V..

[LRZ-RA] LEIBNIZ-RECHENZENTRUM: *LRZ; akkreditierte Registration Authority (RA) der DFN-PKI*, http://www.grid.lrz.de/de/gacc/certreq.html .

[MaMa 07] MAKEDANZ, S. und J. MATTHES: *IVOM: Interoperability and Integration of VO Management Technologies in D-Grid Work Package 4.1: User Interface for the Selection of Attributes*. Technischer Bericht, D-Grid, Juli 2007, http://dgi.d-grid.de/fileadmin/user_upload/documents/DGI-FG1-IVOM/AP4_1-Report-v1.pdf .

[MAP 05] MANDRICHENKO, I., W. ALLCOCK und T. PERELMUTOV: *GridFTP v2 Protocol Description*. Technischer Bericht, Open Grid Forum, April 2005, http://forge.ogf.org/sf/docman/do/downloadDocument/projects.gridftp-wg/docman.root.working_drafts/doc6216 .

[MaPf 06] MAKEDANZ, S. und H. PFEIFFENBERGER: *Shibboleth – Infrastruktur für das Grid*, März 2006, http://www.ingrid-info.de/modules.php?name=UpDownload&req=getit&lid=41 . D-Grid Security Workshop; Göttingen.

[MCLS 03] MULLEN, S., M. CRAWFORD, M. LORCH und D. SKOW: *Grid Authentication Authorization and Accounting Requirements Research Document*. Technischer Bericht, GGF Site Authentication, Authorization and Accounting Research Group (SEC S3A-RG), Juni 2003, http://www.ggf.org/documents/GFD.32.pdf .

[MediGRID] *MediGRID — GRID-Computing für die Medizin und Lebenswissenschaften*, http://wwww.medigrid.de .

[MeHu 99] MEDVINSKY, A. und M. HUR: *RFC 2712: Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)*. RFC, IETF, Oktober 1999, ftp://ftp.isi.edu/in-notes/rfc2712.txt .

[Metz 04] METZGER, S.: *Föderiertes Identity Management bei der BMW Group*. Diplomarbeit, Ludwig–Maximilians–Universität München, August 2004, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=metz04 .

[Mitr 03] MITRA, N.: *SOAP Version 1.2 Part 0: Primer*. W3C Recommendation, World Wide Web Consortium (W3C), Juni 2003, http://www.w3.org/TR/2003/REC-soap12-part0-20030624/ .

[Mixminion] *Mixminion: A Type III Anonymous Remailer*, http://mixminion.net/ .

[MKK 05] MORIAI, S., A. KATO und M. KANDA: *RFC 4132: Addition of Camellia Cipher Suites to Transport Layer Security (TLS)*. RFC, IETF, Juli 2005, ftp://ftp.isi.edu/in-notes/rfc4132.txt .

[Moha 06] MOHAMMED, Y.: *Juristische, organisatorische und Management Fragestellungen — Datenschutz*, März 2006, http://www.ingrid-info.de/

modules.php?name=UpDownload&req=getit&lid=46 . D-Grid Security Workshop; Göttingen.

[Mose 05] MOSES, T.: *eXtensible Access Control Markup Language (XACML) Version 2.0* . Technischer Bericht, OASIS, Februar 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf .

[MSVR 07] MOHAMMED, Y., U. SAX, F. VIEZENS und O. RIENHOFF: *Shortcomings of Current Grid Middlewares Regarding Privacy in HealthGrids*. In: JACQ, N., Y. LEGRE, H. MULLER, I. BLANQUER, V. BRETON, D. HAUSSER, V. HERNÁNDEZ, T. SOLOMONIDES und M. HOFMAN-APITIUS (Herausgeber): *From Genes to Personalized HealthCare: Grid Solutions for the Life Sciences — Proceedings of HealthGrid 2007*, Seiten 322 – 329, Geneva, April 2007. IOS Press, http://geneva2007.healthgrid.org/proceedings/proceedings/pdf/34.pdf .

[MSWS 03] MEDER, S., F. SIEBENLIST, V. WELCH und T. SANDHOLM: *A GSSAPI profile for security context establishment and message protection using WS-SecureConversation and WS-Trust*. Technical Report, Open Grid Service Architecture Security Working Group (OGSA-SEC-WG), Februar 2003, http://www.cs.virginia.edu/~humphrey/ogsa-sec-wg/GSS-secure-conversation-Feb15.pdf .

[MSZ 07] MILKE, J.-M., M. SCHIFFERS und W. ZIEGLER: *Rahmenkonzept für das Management Virtueller Organisationen im D-Grid (Version 1.0)*. Technischer Bericht, D-Grid, 2007, http://dgi.d-grid.de/fileadmin/user_upload/documents/DGI-FG1-10/VO_Rahmenkonzept_0.9.3.3.doc .

[Murr 07] MURRAY, M.: *Profile for Member Integrated X.509 Credential Services (MICS) with Secured Infrastructure*. Technischer Bericht Version 1.0, IGTF; TAGPMA, Juli 2007, http://www.tagpma.org/files/IGTF-AP-MICS-1.0.pdf .

[MW 06] M. WALDBURGER, B. STILLER: *Toward the Mobile Grid: Service Provisioning in a Mobile Dynamic Virtual Organization*. In: *Toward the Mobile Grid: Service Provisioning in a Mobile Dynamic Virtual Organization*, Seiten 579–583. IEEE, März 2006, http://www.csg.uzh.ch/staff/waldburger/extern/publications/AICCSA-06_camera-ready.pdf .

[MyVOCS] INTERNET2: *GridShib — MyVocs*, 2007.

[NAG+ 06] NIEDERBERGER, R., W. ALLCOCK, L. GOMMANS, E. GRÜNTER, T. METSCH, I. MONGA, G. L. VOLPATO und C. GRIMM: *Firewall Issues overview*. Technischer Bericht GFD-I.083, Global Grid Forum (GGF), August 2006, http://www.ggf.org/documents/GFD.83.pdf .

[Neil 04] NEILSON, I.: *Grid authentication and authorization practice and experiences in LCG/EGEE*. Technischer Bericht, EGEE, September

2004, http://www.jisc.ac.uk/uploaded_documents/CERN_%20PositionPaper.doc .

[Newp 06] NEWPORT NETWORKS LTD.: *NAT Traversal for Multimedia over IP*. White Paper, Newport Networks Ltd., 2006, http://www.newport-networks.com/cust-docs/33-NAT-Traversal.pdf .

[NiMc] NICKULL, D. und F. MCCABE: *OASIS SOA Reference Model TC*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm .

[NJD+ 02] NAGARATNAM, N., P. JANSON, J. DAYKA, A. NADALIN, F. SIEBENLIST, V. WELCH, I. FOSTER und S. TUECKE: *The Security Architecture for Open Grid Services*. Technical Report Vers. 1, Open Grid Service Architecture Security Working Group (OGSA-SEC-WG); Global Grid Forum, Juli 2002.

[NKG 07] NEUROTH, H., M. KERZEL und W. GENTZSCH (Herausgeber): *Die D-Grid Initative*. ISBN 978-3-940344-01. Universitätsverlag Göttingen, 2007, http://webdoc.sub.gwdg.de/univerlag/2007/D-Grid_de.pdf .

[NSL+ 06] NEIGER, G., A. SANTONI, F. LEUNG, D. RODGERS und R. UHLIG: *Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization*. Intel Technology Journal, Seiten 167–177, August 2006, http://download.intel.com/technology/itj/2006/v10i3/v10-i3-art01.pdf .

[NSSM 07] NIEDERBERGER, R., A. STREIT, A. SCHOTT und P. MALFETTI: *DEISA - Cooperative Extreme Computing Across Europe*. Technischer Bericht, DEISA, März 2007, http://event.twgrid.org/isgc2007/presentation/OperationandManagementI-RalphNIEDERBERGER-03292007.ppt .

[Oaks 01] OAKS, S.: *Java Security*. O'Reilly, 2. Auflage, 2001.

[OASI a] OASIS: *eXtensible Access Control Markup Language (XACML) Technical Committee*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml .

[OASI b] OASIS: *OASIS Web Services Notification (WSN) Technical Committee*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn .

[OASI c] OASIS: *Web Services Distributed Management (WSDM) Technical Commitee (TC)*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm .

[OASI d] OASIS: *Web Services Security (WSS) Technical Commitee (TC)*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss .

[OASI 05] OASIS: *Web Services Resource Framework (WSRF) — Primer*. Committee Draft 01, OASIS, Dezember 2005, http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-01.pdf .

[OASIS] *OASIS – Organization for the Advancement of Structured Information Standards*, http://www.oasis-open.org .

[OGF] *Open Grid Forum*, http://www.ogf.org/ .

[Parallels] SWSOFT: *Parallels*, http://www.parallels.com .

[Paym 06a] PAYMENT CARD INDUSTRY (PCI): *Payment Card Industry Data Security Standard (PCI DSS)*. Version 1.1, PCI, September 2006, https://www.pcisecuritystandards.org/pdfs/german_pci_dss_v1-1.pdf .

[Paym 06b] PAYMENT CARD INDUSTRY (PCI): *Payment Card Industry Data Security Standard (PCI DSS) — Sicherheitsprüfverfahren*. Version 1.1, PCI, September 2006, https://www.pcisecuritystandards.org/pdfs/german_pci_dss_audit_procedures_v1-1.pdf .

[Paym 06c] PAYMENT CARD INDUSTRY (PCI): *Payment Card Industry Data Security Standard (PCI DSS) — Verfahren für Sicherheitsscans*. Version 1.1, PCI, September 2006, https://www.pcisecuritystandards.org/pdfs/german_pci_scanning_procedures_v1-1.pdf .

[PfHa 07] PFITZMANN, A. und M. HANSEN: *Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management — A Consolidated Proposal for Terminology*. Technischer Bericht v0.30, TU Dresden, November 2007, http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.30.pdf .

[PfWa 87] PFITZMANN, A. und M. WAIDNER: *Networks without user observability*. Computers & Security, 6(2):158–166, April 1987, http://www.semper.org/sirene/publ/PfWa_86anonyNetze.html .

[PKW+ 03] PEARLMAN, L., C. KESSELMAN, V. WELCH, I. FOSTER und S. TUECKE: *The Community Authorization Service: Status and Future*. In: *Proceedings of Computing in High Energy Physics 03 (CHEP '03)*, März 2003, http://www.globus.org/alliance/publications/papers/CAS_update_CHEP_03-final.pdf .

[PoRe 85] POSTEL, J. und J. REYNOLDS: *RFC 959: File Transfer Protocol*. RFC, IETF, Oktober 1985, ftp://ftp.isi.edu/in-notes/rfc959.txt .

[PWF+ 02] PEARLMAN, L., V. WELCH, I. FOSTER, C.KESSELMANN und S. TUECKE: *A Community Authorization Service for Group Collaboration*. In: *Proceedings of the IEEE 3rd International Workshop on Policies fpr Distributed Systems and Networks*. IEEE, 2002, http://www.globus.org/alliance/publications/papers/gauth02.pdf .

[RaZh 04] RAY, S. und Z. ZHANG: *An Efficient Anonymity Protocol for Grid Computing*. In: *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing*, Seiten 200 – 207. IEEE Computer Society, 2004.

[Reis 01] REISER, H.: *Sicherheitsarchitektur für ein Managementsystem auf der Basis Mobiler Agenten*. Dissertation, Ludwig–Maximilians–Universität München, Dezember 2001, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=reis01 .

[ReVo 00] REISER, H. und G. VOGT: *Security Requirements for Management Systems using Mobile Agents*. In: TOHME, S. und M. ULEMA (Herausgeber): *Proceedings of the Fifth IEEE Symposium on Computers & Communications*, Seiten 160–165, Antibes - Juan Les Pins, France, Juli 2000. IEEE, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=revo00a .

[ReVo 00a] REISER, H. und G. VOGT: *Threat Analysis and Security Architecture of Mobile Agent based Management Systems*. In: HONG, J. W. und R. WEIHMAYER (Herausgeber): *NOMS 2000 IEEE/IFIP Network Operations and Managment Symposium — The Networked Planet: Management Beyond 2000*, Seite 979, Honolulu, Hawaii, USA, April 2000. IEEE, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=revo00 .

[ReVo 04] REISER, H. und G. VOLKER: *Honeynet Operation within the German Research Network — A Case Study*. PIK (Praxis der Informationsverarbeitung und Kommunikation), 04(4):188–194, Dezember 2004, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=revo04a .

[ReVo 04a] REISER, H. und G. VOLKER: *A Honeynet within the German Research Network — Experiences and Results*. In: *Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2004)*, Band 2004 der Reihe *Lecture Notes in Informatics (LNI)*, Seiten 113 — 128, Dortmund, Germany, Juli 2004. GI SIG SIDAR Workshop, Springer, http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=revo04 .

[Rive 92] RIVEST, R.: *RFC 1321: The MD5 Message-Digest Algorithm*. RFC, IETF, April 1992, ftp://ftp.isi.edu/in-notes/rfc1321.txt .

[RSW 07] REETZ, J., A. SCHOTT und J. WOLFRAT: *DEISA Security Activity — Grid Acceptable Use Policy*. Technischer Bericht V 2.0, DEISA, Juni 2007, http://www.deisa.org/userscorner/DEISA_Grid_AUP-v2.pdf .

[RWHM 03] ROSENBERG, J., J. WEINBERGER, C. HUITEMA und R. MAHY: *RFC 3489: STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*. RFC, IETF, März 2003, ftp://ftp.isi.edu/in-notes/rfc3489.txt .

[SAL 05a] SON, S., B. ALLCOCK und M. LIVNY: *CODO: Firewall Traversal by Cooperative On-Demand Opening*. In: *Proceedings of the 14th IEEE International Symposium on High Performance Distributed Computing 2005 (HPDC)*, Seiten 233–242. IEEE, IEEE, Juli 2005, ftp://info.mcs.anl.gov/pub/tech_reports/reports/P1200.pdf .

[SAL 05b] SON, S., B. ALLCOCK und M. LIVNY: *CODO: Firewall Traversal by Cooperative On-Demand Opening*, Juli 2005, http://www.caip.rutgers.edu/hpdc2005/presentations/session7-sson.pdf .

[Sax 06] SAX, U.: *Stand der generischen Datenschutz-Konzepte sowie deren technischen Realisierung in biomedizinischen Grids*, Kapitel 3.1, Seiten 38–43. In: SAX, U. et al. [SMVR 06], 2006.

[Sche 99] SCHEITER, C.: *Erstellung eines Kriterienkatalogs zum Vergleich verschiedener Netzkonzepte für BMW und Rover*. Diplomarbeit, Technische Universität München, August 1999.

[Schi 07] SCHIFFERS, M.: *Management dynamischer Virtueller Organisationen in Grids*. Dissertation, Ludwig–Maximilians–Universität München, Juli 2007.

[Schn 96] SCHNEIER, BRUCE: *Applied Cryptography*. Wiley & Sons, Second Auflage, 1996.

[Scho 05] SCHOPF, JENNIFER: *Introduction to Grid Computing*. Hauptseminar, Ludwig-Maximilians Universität München, Munich, Germany, April 2005, http://www.mnm-team.org/teaching/LMU/Seminare/2005ss/grid/schopf.pdf .

[Scho 06] SCHOPF, JENNIFER: *Globus Toolkit 4 — D-Grid Workshop at LRZ*, Januar 2006, http://www-unix.mcs.anl.gov/~schopf/Talks/gt4-munich-jan06.ppt .

[Sduk 05] SDUKHIN, IGOR: *Web Services Distributed Management: Management of Web Services (WSDM-MOWS) 1.0*. OASIS Standard, OASIS, März 2005, http://docs.oasis-open.org/wsdm/2004/12/wsdm-mows-1.0.pdf .

[SEADL 05] SANTHANAM, S., P. ELANGO, A. ARPACI-DUSSEAU und M. LIVNY: *Deploying Virtual Machines as Sandboxes for the Grid*. In: *Proceedings of the 2nd conference on Real, Large Distributed Systems*, Seite 2. ACM, 2005, http://www.cs.wisc.edu/condor/doc/SandboxingWorlds053.pdf .

[SecArch1.2] GONG, LI: *Java 2 Platform Security Architecture*. Technischer Bericht Version 1.2, Sun Microsystems, Inc., Palo Alto, CA, 2001, http://java.sun.com/j2se/1.4/docs/guide/security/spec/security-spec.doc.html .

[SFE+ 06] SMITH, M., T. FRIESE, M. ENGEL, B. FREISLEBEN, G. KOENIG und W. YURCIK: *Security Issues in On-Demand Grid and*

Cluster Computing. In: *Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops (CCGRIDW'06)*, Seite 24. IEEE, 2006, http://ds.informatik.uni-marburg.de/de/publications/pdf/CCGrid2006-Smith.pdf .

[SFEF 06] SMITH, M., T. FRIESE, M. ENGEL und B. FREISLEBEN: *Countering Security Threats in Service-Oriented On-Demand Grid Computing Using Sandboxing and Trusted Computing Techniques*. Journal of Parallel and Distributed Computing, 66(9):1189–1204, 2006, http://www.uni-marburg.de/fb12/verteilte_systeme/forschung/adhoc_gridcomp/publications .

[Sham 79] SHAMIR, A.: *How to Share a Secret*. Communications of the ACM (CACM), 22(11):612–613, November 1979.

[SM 00] S. MÄKI, T. AURA, M. HIETALAHTI: *Robust Membership Management for Ad-hoc Groups*. In: *Proceedings of the 5th Nordic Workshop on Secure IT Systems (NORDSEC 2000)*, Reykjavik, Iceland, Oktober 2000. , http://research.microsoft.com/users/tuomaura/Publications/maki-aura-hietalahti-nordsec00.pdf .

[SMVR 06] SAX, U., Y. MOHAMMED, F. VIEZENS und O. RIENHOFF (Herausgeber): *Grid-Computing in der biomedizinschen Forschung: Datenschutz und Datensicherheit*. Nummer 90 in *Medizinische Informatik, Biometrie und Epidemiologie*. Urban & Vogel, München, 2006.

[Soto 05] SOTOMAYOR, BORJA: *The Globus Toolkit 4 Programmer's Tutorial*. Technischer Bericht, University of Chicago, 2005, http://gdp.globus.org/gt4-tutorial/multiplehtml/index.html .

[Stei 06] STEINKE, T.: *Zugriffsrechte und Zugriffskontrolle für feingranulare Daten*. D-Grid Security Workshop, März 2006, http://www.ingrid-info.de/modules.php?name=UpDownload&req=getit&lid=48 .

[Stel 05] STELL, ANTHONY: *How to Write GT4 Services — Logging*. Technischer Bericht, University of Glasgow, 2005, http://labserv.nesc.gla.ac.uk/projects/etf/gt4howto/logging.html .

[Sv 02] SRDJAN ČAPKUN, LEVENTE BUTTYÁN, JEAN-PIERRE HUBAUX: *Small worlds in security systems: an analysis of the PGP certificate graph*. In: *Proceedings of the ACM New Security Paradigms Workshop (NSPW)*, Seiten 28–35, Virginia Beach, Virginia, September 2002. ACM Press, http://www.terminodes.org/micsPublicationsDetail.php?pubno=195 .

[SWT+ 02] SIEBENLIST, F., V. WELCH, S. TUECKE, I. FOSTER, N. NAGARATNAM, P. JANSON, J. DAYKA und A. NADALIN: *OGSA Security Roadmap – Global Grid Forum Specification Roadmap towards a Secure OGSA*. Technical Report, Open Grid Service Architecture Security Working Group (OGSA-SEC-WG), Juli 2002, http://www.cs.virginia.edu/~humphrey/ogsa-sec-wg/ogsa-sec-roadmap-v13.pdf .

[TAGPMA] *TAGPMA — The Americas Grid Policy Management Authority*, http://www.tagpma.org/ .

[TCF+ 03] TUECKE, S., K. CZAJKOWSKI, I. FOSTER, J. FREY, S. GRAHAM, C. KESSELMANN, T. MARQUIRE, T. SANDHOLM, D. SNELLING und P. VANDERBILT: *Open Grid Services Infrastructure (OGSI) — Version 1.0*. Proposed Recommendation GFD-R-P.15, Global Grid Forum, Juni 2003, www.ggf.org/documents/GWD-R/GFD-R.015.pdf .

[TCSEC] DEPARTMENT OF DEFENSE: *DEPARTMENT OF DEFENSE STANDARD — Trustet Computer System Evaluation Criteria*. Technischer Bericht DoD 5200.28-STD, Department of Defense, Dezember 1985, http://www.dynamoo.com/orange/fulltext.htm .

[Tor] *Tor: Anonymität online*, http://www.torproject.org/ .

[Trea 06] TREADWELL, J.: *Open Grid Services Architecture — Glossary of Terms*. Technischer Bericht Version 1.5, Open Grid Forum, März 2006, ttp://www.ggf.org/Public_Comment_Docs/Documents/Apr-2006/draft-ggf-ogsa-glossary-1.5-006.pdf .

[Tuec 01] TUECKE, S.: *Security and CAS Futures*. Globus Retreat, August 2001, www-fp.mcs.anl.gov/dsl/scidac/security/www-fp.mcs.anl.gov/dsl/scidac/security/ .

[TWE+ 04] TUECKE, S., V. WELCH, D. ENGERT, L. PEARLMAN und M. THOMPSON: *RFC 3820: Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*. RFC, IETF, Juni 2004, ftp://ftp.isi.edu/in-notes/rfc3820.txt .

[Uni] *UniGrids — Uniform Interface to Grid Services*, http://www.unigrids.org/index.html .

[UniG] UNIGRIDS: *Workpackage Objectives*, http://www.unigrids.org/wpobjectives.html .

[Univ 07] UNIVERSITY OF CAMBRIDGE: *The Xen virtual machine monitor*, 2007, http://www.cl.cam.ac.uk/research/srg/netos/xen/ .

[Vamb 05a] VAMBENEPE, WILLIAN: *Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 1*. OASIS Standard, OASIS, März 2005, http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part1-1.0.pdf .

[Vamb 05b] VAMBENEPE, WILLIAN: *Web Services Distributed Management: Management Using Web Services (MUWS 1.0) Part 2*. OASIS Standard, OASIS, März 2005, http://docs.oasis-open.org/wsdm/2004/12/wsdm-muws-part2-1.0.pdf .

[vdB 06] BERGHE, SVEN VAN DEN: *Using the NJS and TSI*. Technischer Bericht, Fujitsu Laboratories of Europe, März 2006, http://www.unicore.eu/documentation/manuals/unicore5/files/NJS_TSI_Manual.pdf .

[VGN 06] VAMBENEPE, WILLIAM, STEVE GRAHAM und PETER NIBLETT: *Web Services Topics 1.3 (WS-Topics)*. OASIS Standard, OASIS, Oktober 2006, http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf .

[VMware] VMWARE, INC.: *Virtualization Overview*, http://www.vmware.com/overview/ .

[VoGr 06a] VOLPATO, G. L. und C. GRIMM: *Empfehlungen zur statischen Konfiguration von Firewalls im D-Grid*. Technischer Bericht Version 1.2, DGI FG 3-5, Juni 2006, https://www.d-grid.de/fileadmin/dgrid_document/Dokumente/FG3-5_Empfehlungen_Statischer_Firewalls_v121.pdf .

[VoGr 06b] VOLPATO, G. L. und C. GRIMM: *Recommendations for Static Firewall Configuration in D-Grid*. Technischer Bericht Version 1.4, DGI FG 3-5, Januar 2006, https://www.d-grid.de/fileadmin/user_upload/documents/DGI-FG3-5/FG3-5_Recommendations_Static_Firewall.pdf .

[VoGr 06c] VOLPATO, G.L. und C. GRIMM: *Dynamic Firewalls and Service Deployment Models for Grid Environments*. In: BUBAK, M., M. TURALA und K. WIATR (Herausgeber): *Proceedings of the Cracow Grid Workshop 06 (CGW 06)*, Seiten 441–449, Oktober 2006, http://www.l3s.de/web/upload/documents/1/Cracow_paper.pdf .

[WBKS 05] WELCH, V., T. BARTON, K. KEAHEY und F. SIEBENLIST: *Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration*. In: *4th Annual PKI R&D Workshop*. NIST — National Institute of Standards and Technology, April 2005, http://csrc.nist.gov/publications/PubsNISTIRs.html .

[Welc 05] WELCH, V.: *Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective*. Technischer Bericht Version 4, The Globus Security Team, September 2005, http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf .

[Welc 06] WELCH, V.: *Globus Toolkit Firewall Requirements*. Technischer Bericht Version 9, Globus Alliance, Oktober 2006, http://www.globus.org/toolkit/security/firewalls/Globus-Firewall-Requirements-9.pdf .

[WFK+ 04] WELCH, V., I. FOSTER, C. KESSELMANN, O. MULMO, L. PEARLMAN, S. TUECKE, J. GAWOR, S. MEDER und F. SIEBENLIST: *X.509 Proxy Certificates for Dynamic Delegation*. In: *Proceedings of the 3rd Annual PKI R&D Workshop*, 2004, http://www.globus.org/Security/papers/pki04-welch-proxy-cert-final.pdf .

[Whit 07] WHITNEY, J.: *Virtualization and You: What AMD-V Means for the Developer*. Technischer Bericht, AMD, März 2007, http://developer.amd.com/articles.jsp?id=157&num=1 .

[WiEr 03] WIEDER, P. und D. ERWIN: *GRIP — Creating Interoperability between Grids*. Presentation at GGF8 — PGM RG, Forschungszentrum Jülich, Juni 2003, http://www.grid-interoperability.org/presentations03/ggf-pgm-workshop-grip.pdf .

[Wine 99] WINER, D.: *XML-RPC Specification*. Technischer Bericht, XML-RPC.Com, Juni 1999, http://www.xmlrpc.com/spec .

[WWW 06] WARREN, ROBERT H., DANA WILKINSON und MIKE WARNECKE: *Analysis of a dynamic social network built from PGP keyrings*. In: *Statistical Network Analysis: Models, Issues, and New Directions Workshop at the 23rd International Conference on Machine Learning (ICML 2006)*, Pittsburgh PA, USA, June 2006. , http://junobeach.cs.uwaterloo.ca/~warren/publications/warren:icml:2006/warren_paper.pdf .

[X.509] ITU–T: *X.509 — Information technology — Open Systems Interconnection — The Directory: Authentication framework*. ITU–T Recommendation, International Telecommunication Union, August 1997.

[X.800] ITU: *X.800 — Data Communication Networks; Open Systems Interconnection (OSI); Security Structure and Application — Security Architecture for Open Systems Interconnection for CCITT Applications*. Recommendation, International Telecommunication Union, Geneva, 1991.

[X.810] ITU: *X.810 — Data Networks and Open System Communications Security — Information Technology — Open Systems Interconnection — Security Frameworks for Opens Systems: Overview*. ITU–T Recommendation, International Telecommunication Union, Geneva, 1996. also published as ISO/IEC International Standard 10181-1.

[X.811] ITU: *X.811 — Data Networks and Open System Communications Security — Information Technology — Open Systems Interconnection — Security Frameworks for Opens Systems: Authentication Framework*. ITU–T Recommendation, International Telecommunication Union, Geneva, 1995. also published as ISO/IEC International Standard 10181-2.

[X.812] ITU: *X.812 — Data Networks and Open System Communications Security — Information Technology — Open Systems Interconnection — Security Frameworks for Opens Systems: Access Control Framework*. ITU–T Recommendation, International Telecommunication Union, Geneva, 1995. also published as ISO/IEC International Standard 10181-3.

[X.814] ITU: *X.814 — Data Networks and Open System Communications Security — Information Technology — Open Systems Interconnection — Security Frameworks for Opens Systems: Confidentiality Framework*. ITU–T Recommendation, International Telecommunication Union, Geneva, 1995. also published as ISO/IEC International Standard 10181-5.

[X.815] ITU: *X.815 — Data Networks and Open System Communications Security — Information Technology — Open Systems Interconnection — Security*

Frameworks for Opens Systems: Integrity Frameworks. ITU–T Recommendation, International Telecommunication Union, Geneva, 1995. also published as ISO/IEC International Standard 10181-6.

[X.816] ITU: *X.816 — Data Networks and Open System Communications Security — Information Technology — Open Systems Interconnection — Security Frameworks for Opens Systems: Security Audit and Alarms Framework*. ITU–T Recommendation, International Telecommunication Union, Geneva, 1995. also published as ISO/IEC International Standard 10181-7.

[Zeic 06a] ZEICHICK, A.: *Processor-Based Virtualization, AMD64 Style, Part I*. Technischer Bericht, AMD, Juni 2006, http://developer.amd.com/articles.jsp?id=14&num=1 .

[Zeic 06b] ZEICHICK, A.: *Processor-Based Virtualization, AMD64 Style, Part II*. Technischer Bericht, AMD, Juni 2006, http://developer.amd.com/articles.jsp?id=15&num=1 .

[ZiGr 06] ZIEGLER, W. und C. GRIMM: *Interoperabilität und Integration der VO-Management Technologien im D-Grid*. Projektantrag, D-Grid, Juni 2006, http://www.d-grid.de/fileadmin/user_upload/documents/DGI-FG1-IVOM/ShibVoms_v1.3.pdf .

[Zimm 94a] ZIMMERMAN, PHILIP: *PGP User's Guide - Volume I*, Oktober 1994, ftp://ftp.pgpi.org/pub/pgp/2.x/doc/pgpdoc1.txt .

[Zimm 94b] ZIMMERMAN, PHILIP: *PGP User's Guide - Volume II*, Oktober 1994, ftp://ftp.pgpi.org/pub/pgp/2.x/doc/pgpdoc2.txt .

INDEX

D

E

F

J

K

L

M

N

O

W

X

Z