Universität Duisburg-Essen

Virtueller Weiterbildungsstudiengang Wirtschaftsinformatik (VAWi)

Masterarbeit

Ein patternbasiertes Referenzmodell für Identity Management

[A Pattern System for Identity Management]

Vorgelegt dem Fachbereich Wirtschaftswissenschaften der Universität Duisburg-Essen

Verfasser: **Nahrgang, Holger**

Marschner Straße 7
12203 Berlin
Matrikelnummer 2220066

Gutachter: Prof. Dr. Günther Pernul (Universität Regensburg)

Abgabe: 12.06.2007 / Sommersemester 2007

INHALT

TABELLEN

ABBILDUNGEN

ABKÜRZUNGEN

CERT	Computer Emergency Response Team
et al.	et alteri
GOM	Grundsätze ordnungsmäßiger Modellierung
IT	Informationstechnik
IT-Sicherheit	Sicherheit in der Informationstechnik
LDAP	Lightweight Directory Access Protocol
OCL	Object Constraint Language
RBAC	Role based Access Control
SSO	Single Sign On
SOA	serviceorientierte Architekturen
UML	Unified Modeling Language

1 Einleitung

Auf Webseiten zur Sicherheit in der Informationstechnik kann man es verfolgen: Das Computer Emergency Response Team CERT an der Carnegie Mellon Universität listet seit 1995 Sicherheitslücken. Die Anzahl stieg seit 1999 teilweise exponentiell auf über 8000 im Jahr 2006 [Schu03, S. 2]; [CERT07]. Laut MITRE gehen 75% aller Sicherheitslücken auf fehlerhafte Applikationssoftware zurück [Ysk+06, S. 4]; [MITR07]. Neben Mängeln menschlicher Problemlösefähigkeiten ist eine der Ursachen, dass bis heute Sicherheit bei Software erst im Nachhinein berücksichtigt oder „hineingetestet“ wird. In den frühen Phasen der Softwareentwicklung wie Analyse und Entwurf erhalten Fragen der Sicherheit zu oft nicht den angemessen Raum [Pri+04, S. 1]. Die Dimension der geschilderten Probleme legt außerdem nahe, dass Akzeptanz und Anwendung durchaus vorhandener Techniken zur Erstellung sicherer Software deutlich mit Mängeln behaftet sind, auch aus Sicht der Forschung [Ysk+06, S. 1].

Unternehmen sehen sich dieser Entwicklung in zunehmendem Maß ausgesetzt. Hinzu kommt, dass auch eine immer dynamischere Umwelt verstärkt fordert, dass Geschäftsprozesse auch über Organisationsgrenzen hinweg integriert werden, z.B. über webbasierte Portale [Mehl05b, S. 453]. Neben der Absicherung betrieblicher Informationssysteme gegen unzulässige Nutzung gilt es, geschäftliche Vorgaben einer effizienten Administration der Benutzer bei gleichzeitig hoher Sicherheit und Nutzerfreundlichkeit einzuhalten [Rott03, S. 1]; [Mehl 05b].

Sowohl in der Wissenschaft als auch im betrieblichen Alltag wird dieses Thema unter dem Begriff Identity Management subsumiert, siehe z.B. [Baie05, S. 18]; [Wind05, S. 5]; [Wild06, S. 82]. Hierzu gehören bspw. Authentisierungskonzepte, Autorisierungskonzepte, Sign-On-Konzepte, Rollen- und Rechtekonzepte etc. Identitätsmanagement, ursprünglich ein Bereich der IT-Sicherheit, entwickelt sich mehr und mehr zu einer eigenständigen Richtung [FuSa05, S. 259]. Begrifflich wird Identitätsmanagement in dieser Arbeit mit dem englischen Ausdruck Identity Management gleichgesetzt.

Aus Sicht der Wirtschaftsinformatik stellt sich die Frage, mit welchen methodischen Hilfsmitteln die Entwicklung sicherer Software sowie Einführung und Betrieb komplexer Systeme zur Benutzerverwaltung und Zugriffskontrolle unterstützt werden können. Die Basis dafür bilden zwei Ansätze der Wirtschaftsinformatik, Referenzmodelle und Sicherheitsmuster (Security Pattern) [Beck04, S. 325] [Sch+06]. Beide stellen Vorschläge dar, wie

KnowHow für die Anwendungsentwicklung, die Einführung und den Betrieb großer Informationssysteme nachvollziehbar und wiederholbar wieder verwendet werden kann.

Bei Design Pattern oder Entwurfsmustern handelt es sich um ein Konzept zur Verwendung bewährter Lösungen im Softwareentwurf bei wiederkehrenden Problemstellungen [FeLa06, S. 1]. Vor einem Jahrzehnt wurde dieser Ansatz auch auf das Gebiet der Sicherheit in der Informationstechnik (IT) übertragen. Auf Kongressen und in Arbeitsgruppen entstanden Muster zu verschiedenen relevanten Gebieten wie Authentisierung, Autorisierung und Zugriffskontrolle, Firewalls und andere. Diese Patterns können kombiniert und zu komplexen Architekturen zusammengesetzt werden, wie z.B. Single-Sign-On oder Web Services Autorisierungen und andere [FeLa06, S. 1]. Die Begriffe Muster und Pattern sollen im weiteren Verlauf dieser Arbeit synonym verwendet werden. Das gleiche gilt für Sicherheitsmuster und Security Pattern.

Security Pattern sind kein rein akademisches Konzept, sie kommen bereits in kommerziellen Produkten großer Software-Hersteller wie IBM Tivoli, dem Sun One Identity Server und Netegrity Siteminder zum Einsatz [Del+07, S. 3]. Sie können leicht im Software Entwicklungszyklus eingesetzt werden, um so die Erstellung sicherer Software zu unterstützen [Del+07, S. 37].

Bei der Beschäftigung mit Mustern stellt sich dem Autor einer Abschlussarbeit wie auch einem Softwareentwickler letztlich die Frage, welches das richtige (Security) Pattern für ein spezifisches Problem ist. Dadurch stößt man unweigerlich auf Probleme der Praxis, das der Musterklassifikation und das der Orientierung als Neuling in einem komplexen Gebiet.

Um die hohe Komplexität einer Anwendungsdomäne wie dem Identity Management zu bewältigen, können bspw. Referenzmodelle als ein Mittel herangezogen werden. Ein Referenzmodell stellt eine allgemeingültige Repräsentation eines Fachkonzepts für eine Klasse von Informationssystemen dar [Loos06, S. V]. Referenzmodelle beanspruchen Geltung für eine Klasse von Anwendungsfällen. Inhalte können z.B. Datenstrukturen, Prozesse oder Funktionen sein. Durch ihren Einsatz verspricht man sich Vorteile wie Risikoreduktion und Kostendämpfung [Beck04, S. 325]. Die wissenschaftliche Auseinandersetzung mit Referenzmodellen begann bereits in den 60’er Jahren und hat sich zu einer eigenen Richtung mit zahlreichen Veröffentlichungen entwickelt. Referenzmodelle spielen heute insbesondere im Bereich des Enterprise Resource Planing bei der Einführung betriebswirtschaftlicher Standardsoftware eine wichtige Rolle.

Diese Masterarbeit hat das Ziel, die Gebiete der Security Pattern und der Referenzmodelle zu betrachten, Gemeinsamkeiten zu identifizieren und Möglichkeiten zu prüfen, wie Security Pattern in Referenzmodelle integriert werden können.

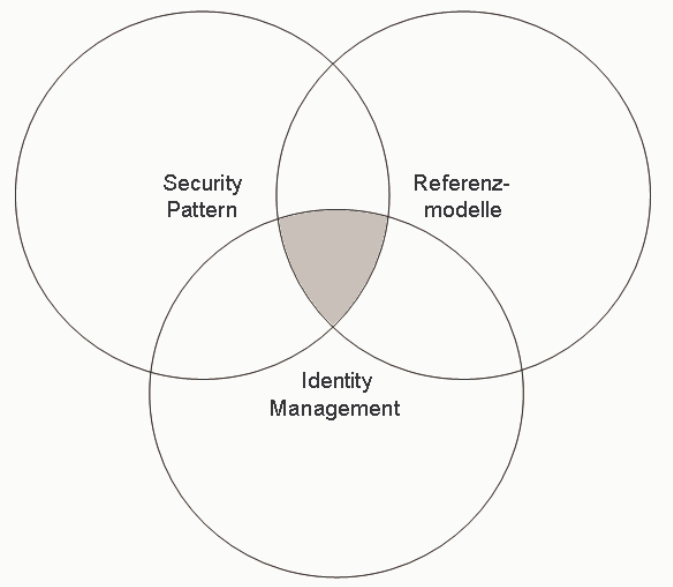

Abbildung 1: Die Themenfelder dieser Arbeit

Dabei soll die Schnittmenge der Felder untersuchen werden, die die beiden Konzepte Security Pattern und Referenzmodelle, angewandt auf die Domäne des Identitätsmananagements bilden (siehe Abbildung 1).

Hierzu werden die genannten Themenkreise in den sich anschließenden drei Kapiteln in ihren Grundlagen dargestellt. Im Kapitel 5 folgt dann eine Betrachtung des Verhältnisses bzw. der Integrationsmöglichkeiten von Security Pattern in Referenzmodelle. Darauf aufbauend wird im Kapitel 6 ein Ordnungsrahmen für die Domäne des Identity Managements entwickelt, die Teilbereiche werden im Kapitel 7 detailliert. Im Kapitel 8. wird als Ergebnis ein eigenes Referenzmodell bzw. Patternsystem für Identity Management vorgestellt. Die Arbeit endet mit einem Ausblick in Kapitel 9.

2 Die Anwendungsdomäne des Identity Managements

Identity Management ist eine vergleichsweise junge Teildisziplin der heutigen Informationstechnologielandschaft in mittleren und großen Unternehmen [FuSa05, S. 259]. Ihre Ursprünge liegen in den Anfängen moderner Datenverarbeitung, als einzelne Benutzer sich die Ressourcen eines Hosts teilten. Die Zuordnung von Aktionen zu einzelnen Personen bei Time-sharing-Computerssystemen können als Anfang der modernen Benutzerverwaltung angesehen werden [Baie05, S. 18]. Das mit der Nutzung von zentralisierten Großrechnern aufkommende Identitätsmanagement wird auch als „lokal“ charakterisiert. Beispiel

für eine Benutzerverwaltung eines Host Systems ist Ressource Access Control Facility (RACF) der Firma IBM auf der z/OS-Betriebssystemplattform [Bena06, S 40].

Die gegenseitige Abschirmung von Benutzern und Prozessen in Host-Umgebungen durch Zugriffskontrollmechanismen entwickelte sich zu policy-basierte Autorisierungssubsystemen [Bena06, S. 1]. Aus der Datenhaltung für die Authentifizierung, zunächst mit Kombinationen aus Name und Passwort, entstanden später Verzeichnisse, mit denen Informationen der Benutzer verwaltet werden [Baie05, S. 19]. Durch den eng begrenzten Einsatz von Informationstechnologien hielten sich die damit verbundenen Aufwände zunächst in Grenzen, heute werden Zugriffskontrolle und Benutzerverwaltung verstärkt zur Absicherung von Transaktionen über große Netzwerke eingesetzt und haben sich damit vom hostzentrierten Paradigma entfernt. Welche Anforderungen die heutigen Geschäftsprozesse und Rahmenbedigungen stellen, wird im nächsten Abschnitt kurz betrachtet.

2.1 Anlässe für Identity Management

Zu den Herausforderungen und Problemen, die mit Identity Management gelöst werden sollen, gehören komplexe Geschäftsbeziehungen und Kooperationen von Organisationen, die immer dynamischer werden und höhere Anforderungen an die Flexibilität der Infrastruktur stellen. Gleichzeitig steigt der Kostendruck bei der parallelen Verwaltung von Identitäten für mehrere Systeme [Wild06, S. 70].

Bei neu hinzukommenden Geschäftsprozessen sind die beteiligten Personen und Systeme sicher zu identifizieren sowie zeitnah mit Zugriffsrechten auszustatten. Die zuverlässige Identifikation der Teilnehmer stellt eine grundlegende Voraussetzung für Geschwindigkeit und Sicherheit bei der Abwicklung von Vorgängen dar [Wind05, S. 3].

Während das Handeln einer Person in der Domäne eines Unternehmens noch zuzuordnen sein mag, ist es bedeutend schwieriger, auch in einem Szenario mit Systemen *mehrerer* Organisationen Identitätsdaten über Unternehmensgrenzen hinweg zu verwalten. Dabei können die Benutzer durchaus mobil sein und der Zugriff kann von verschiedenen, angreifbaren Endgeräten aus erfolgen kann [Del+07, S.1] [Bena06, S. 2].

Selbst im Kontext eines einzelnen Unternehmens ist mit der vorhandenen IT-Infrastruktur bereits genügend Komplexität verbunden. Eine durchschnittliche IT-Landschaft umfasst eine große Zahl an Datenbanken, Dateien und verschiedene Ressourcen, für die eine Vielzahl von Usern administriert werden müssen [FuSa05, S. 260]. Sie hat sich in der Regel über die Zeit entwickelt und ist zu einem Konglomerat heterogener Plattformen, Betriebs-

systeme und einer Vielzahl von Anwendungen gewachsen [FuSa05, S. 260] [Bena06, S. 67]. Die Applikationen sind in aller Regel verteilt und aus verschiedenen Komponenten zusammengesetzt. Sie werden ad hoc entwickelt, sind gekauft oder outgesourced [Del+07, S.31]. Die Nutzung dieser Systeme über Netzwerke hat die Zahl und Verbreitung von Identitäten vergrößert, die durch verschiedene Name-Passwort-Kombinationen ihren Nutzern einiges abverlangen. Dieser Mangel an Benutzerfreundlichkeit erschwert sicheres Verhalten, ausserdem erhöht die Inflation von Passworten die Supportaufwände und hat Produktivitätsverluste zur Folge.

Neben der zunehmenden inner- und überbetrieblichen Integration führen auch die gewachsenen Anforderungen an regulatorische Compliance dazu, dass dezentrales Monitoring und Auditing manuell eigentlich nicht zu bewältigen sind. Eine Organisation wäre dann im Prüfungsfall nicht auskunftsfähig [Wild06, S. 71 / 82] [Wind05, S. 5] [Ste+05, S. 27]. Die Pflege von Benutzerberechtigungen auf Einzelsystemebene begrenzt die Skalierbarkeit, erhöht die Fehlerwahrscheinlichkeit und erzeugt direkte und indirekte Kosten [Bena06, S. 67]. Ohne Identity Management würde die inhärente Dezentralisierung nur mit einer Vervielfachung des Aufwands und einer Gefahr von Inkonsistenzen einhergehen.

Insgesamt resultieren also Probleme wie hohe manuelle Aufwände, Sicherheitsrisiken durch inaktuelle Berechtigungen, zunehmende administrative Kosten sowie Datenqualitätsprobleme [FuSa05, S. 260]. Als Konsequenz hat sich der Bereich des Identity Management als eigene Disziplin in den letzten Jahren entwickelt [Bena06, S. 2]. Sichere Identifizierung und Identitätsverwaltung sind zu einem der Eckpfeiler moderner und sicherer Informationstechnik (IT) geworden und versprechen für diese Herausforderungen verschiedene Lösungsansätze [FuSa05, S. 261].

Zu den Zielsetzungen gehört es, die Kosten der Haltung von Identitäten in einer Vielzahl von Systemen unter Wahrung von Konsistenz und Eindeutigkeit zu begrenzen oder zu senken. Durch Identity Management sollen Identitätsdaten standardisiert und konsolidiert verwaltet und Nutzer konsistent und nach einheitlichen Policies über Applikationen hinweg autorisiert werden. Ein weiterer Vorteil ist die applikationsspezifische Administration durch Provisionierung [FuSa05, S. 261]. Identity und Access Management bedeutet in diesem Sinne, den richtigen Personen den richtigen Zugriff auf die richtigen Ressourcen zur richtigen Zeit zu gewähren [Wild06, S. 70].

2.2 Szenarien für Identity Management

Baier unterscheidet drei Bereiche des Identitätsmanagements [Baie05, S. 18]: Persönliches Identitätsmanagement, Identity Management in Organisationen und föderiertes Identity Management. Persönliches Identitätsmanagement ist demnach eine Unterstützung durch Prozesse und Software auf mobilen Geräten und lokalen Clients, die es Privatanwendern ermöglicht und erleichtert selbst zu entscheiden, welche ihrer vertraulichen Daten sie einem Kommunikationspartner, wie bspw. einer Webseite oder einem Serviceanbieter zur Verfügung stellen. Fokus sind die Daten und Schutzinteressen einzelner Personen. Diese Form des Identitätsmanagement soll nachfolgend nicht weiter berücksichtigt werden.

Die anderen Bereiche des Identitätsmanagements nach Baier betreffen mittlere und große Unternehmen, bzw. Verbünde (Föderationen) von Unternehmen und Organisationen. Mit zunehmender Nutzung verteilter Systeme in Unternehmen hat sich das Netzwerk-Identitätsmodell entwickelt [Bena06, S. 46]. Identitäten authentifizieren sich an einem Netzwerk aus Rechnerknoten und können so ohne erneute Anmeldung auf Ressourcen zugreifen. Der Scope liegt in diesem Modell nicht auf dem Einzelsystem sondern wird durch die Grenzen des Netzwerks definiert.

Im Szenario „Business to Business“ (B2B), das dem Federated Identity Management entspricht, ermöglichen Organisationen den Mitgliedern anderer Organisationen den domänenübergreifenden Zugriff auf Ressourcen [Ste+05, S. 356]. Über Organisationsgrenzen hinausreichende Netzwerke bilden eine Föderation lose gekoppelter Sets von Entitäten und damit Beziehung zwischen zwei oder mehreren Unternehmen [Bena06, S. 46]. Föderierte Organisationen stellen die benötigten Services den Entitäten anderer Organisationen zur Verfügung, tragen jedoch nur eine Teilverantwortung beim Management der Identitäten [Bena06, S. 47].

In dieser Arbeit soll im folgenden das Szenario des Enterprise Identity Management im Mittelpunkt der Betrachtung stehen. Aspekte von Föderationen werden nur am Rande behandelt.

Nach einer Definition von Fumy und Sauerbrey beschreibt der Begriff des Identity Managements Konzepte, Methoden und IT-Systeme für die (teil-) automatisierte Erzeugung und Verwaltung von Identitäten (d.h. Benutzern von Applikationen und die automatisierte Steuerung von der Verwendung von Unternehmensressourcen (z.B. Applikationen, Datenquellen, etc.) [FuSa05, S. 259].

Windley betont demgegenüber in seiner Definition den Aspekt des Lebenszyklus' digitaler Identitäten. Demnach handelt es sich bei Identity Management um den Prozess der Erzeugung, Verwendung, Verwaltung und Beendigung digitaler Identitäten. Digitale Identitäten umfassen Daten, die eine Entität oder ein Objekt auf einzigartige Weise beschreiben und enthalten darüber hinaus Informationen über Beziehungen zu anderen Entitäten [Wind05, S. 8].

2.3 Enterprise Identity Management

Unter technischer Perspektive ist Identity Management die systemübergreifende, zentralisierte und automatisierte Verwaltung und Provisionierung d.h. Versorgung von Teilsystemen mit Identitätsdaten. Durch einheitliche oder angebundene Zugangskontrollsysteme wird der Zugang zu IT-Ressourcen kontrolliert. Technische besteht die Infrastruktur aus den Kernbestandteilen Authentisierung, Provisionierung, Access Management, Identitätsverwaltung (Administration, Policy) und Verzeichnisdiensten [Wind05, S. 217].

Identity Management für heterogene Umgebungen umfasst nach dem Verständnis von Wildgruber Authentisierung, Autorisierung und eine eigene Auditierungskomponente. Unter Identity Management im engeren Sinne kann dann die Bündelung der Funktionen von Provisionierungskomponenten, Self Service-Modulen, Verzeichnissen und einem Speicher für Identity Policy Konfigurationen verstanden werden [Wild06, S. 81].

Zu den Konzepten und Kernfunktionen des Identity Managements gehören die Verwaltung der Identitätsdaten auf der Basis von Verzeichnisdiensten [Wild06, S. 71] [Wind05, S. 73], die Authentisierung von Benutzern [Wind05, S. 50] und das Access- oder Zugriffsmanagement [FuSa05, S. 263] [Wind05, S. 63].

Typische Use Cases sind Neuzugänge einer Organisation („New Hires"), bei denen eine Datenübernahme aus dem Human-Resources-System und eine Rechtezuweisung erfolgt, sowie z.B. interne Wechsel von Personal (Entziehung nicht mehr benötigter und Zuweisung neuer Rechte), außerdem noch Passwortwechsel und Provisionierung [Wild06, S. 71].

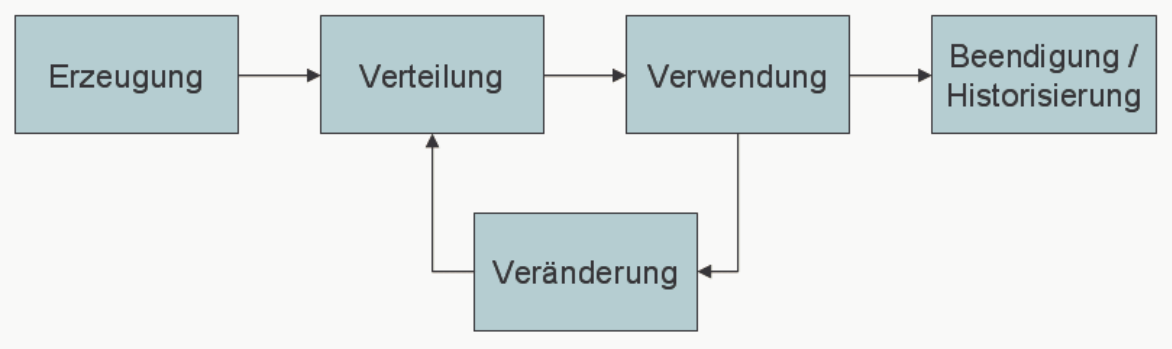

Abbildung 2: Der Lebenzyklus digitaler Identitäten [Wind05, S. 29]

Privilegien und Rechte, im Detail Lese- und Schreibrechte oder abstrakter unterteilt nach Profilen und Rollen mit bestimmten Attributen - werden mit Hilfe von Identity Profiling verwaltet und aktuell gehalten. Die Verwaltung und Veränderung von Berechtigungen in Systemen auf Basis eines Identity Managements ist die Voraussetzung für eine wirksame Zugriffskontrolle.

Zum Identity Establishment gehören Methoden, mit denen ein Benutzer sicher mit einer berechtigten Entität verknüpft wird [Bena06, S. 3]. Hierdurch wird sichergestellt, dass eine verwendete Identität zu der Entität gehört, die den Besitz dieser Identität beansprucht. Daher kann sie authentisch genannt werden.

Identifikation und Authentisierung setzen voraus, dass jede Identität mit einem Authentication credential verknüpft ist, das nur dem Benutzer bekannt ist und das durch das System verifiziert werden kann [Bena06, S. 10]. Als Methoden kommen „Wissen“ (Passwort, PIN-Nummer), „Besitz“ (z.B. Schlüssel oder Karte) oder unveränderliche körperliche Merkmale (biometrische Merkmale wie Fingerabdrücke) in Frage.

Identity Management setzt Mechanismen zur Verwaltung von Massendaten über Individuen, Ressourcen und Räumen in Verzeichnisform voraus [Ste+05, S. 73]. Aufgrund der Heterogenität heutiger IT-Infrastrukturen spielen offene Standards für die Interoperabilität eine entscheidende Rolle [FuSa05, S. 263]. Im Umfeld der Verzeichnisdienste haben bspw. Standards und Protokolle wie X 500 oder LDAP weite Verbreitung gefunden [Baie05, S. 19]. LDAP (Lightweight Directory Access Protocol) oder der mächtigere Verzeichnisstandard x.500 dienen der Strukturierung und Abfrage großer Organisationsverzeichnisse [Ste+06, S. 74]. Mit LDAP-konformen Verzeichnissen werden Entitäten in einer hierarchischen Struktur verwaltet, die für schnelle Queries optimiert und weniger für Analysen wie eine relationale Datenbank.

LDAP-basierende Verzeichnisse eignen sich auch für die Speicherung von Richtlinien (Policies) und Zugriffskontrolllisten. Auch aufgrund dieser Universalität sind auf diesem Standard basierende Verzeichnisdienste zum de-facto-Standard in Organisationen geworden [Ste+05, S. 76].

Authentisierung ist der Prozess der sicheren Identifikation von Entitäten durch Verifizierung eines Identitätsnachweises [Bena06, S. 4]. Damit die Zahl der Authentisierungsvorgänge begrenzt wird, können Mechanismen des Single Sign On zum Einsatz kommen. Dieser Ansatz erlaubt in seinen verschiedenen Umsetzungsformen eine einmalige Anmeldung an verschiedenen Systemen oder zumindest die Anmeldung mit einem einzigen Passwort an verschiedenen Systemen [Ste+05, S. 83f] [FuSa05, S. 270].

Der Zugriffskontrollprozess (oder auch Autorisierung) setzt eine vorher definierte Zugriffsrichtlinie um. Handlungen von Subjekten werden auf das beschränkt, wozu das Subjekt berechtigt ist. Zugriffskontrolle bei Computersystemen dient dem Zweck, ausschließlich berechtigten Nutzern Zugriff auf Informationen, Ressourcen und Services zu gewähren [Bena06, S. 1].

Die Zugriffskontrolle erfolgt häufig nach dem Muster plattformübergreifender, rollenbasierter Berechtigungen. Diese Berechtigungen bauen auf dem Ansatz des Role Based Access Control (RBAC) nach Sandhu, Coyne, Feinstein und Youman auf [San+96] [Sand98]. Es werden Rollen definiert und ihnen 1.) Nutzer und 2.) Rechte zugewiesen, so dass keine oder nur im Ausnahmefall direkte Beziehungen zwischen Personen und Berechtigungen mehr bestehen [Bena06, S. 190 ff].

Workflowfunktionalitäten dienen für die Abbildung von Genehmigungsprozessen bei der Vergabe von Rechten [Wild06, S. 71]. Durch zentralisiertes Identity Management können Veränderungen bei Identitätsdaten per Push-Mechanismen (Provisioning) an alle oder ein Subset von Accounts propagiert werden, auch in der Bottom-Up-Richtung [Bena06, S. 68].

Zur Compliance, der Übereinstimmung mit bzw. Einhaltung gesetzlicher Regelungen gehören bspw. Auditing-Fähigkeiten, damit dies im Fall einer Prüfung problemlos nachgewiesen werden kann [Ste+05, S. 359].

Bei der Verknüpfung von Identity Management Systemen mit ihren Zielapplikationen innerhalb eines Unternehmens bzw. über Unternehmensgrenzen hinweg (Föderationen) spielen Standards wie Auszeichnungssprachen für Zusicherungen (Security Assertions Markup Language, SAML); Provisionierung (Service Provisioning ML) oder bspw. Zugriffskontrolle (Extensible Access Control ML) eine wichtige Rolle [Wind05, S. 98, 118].

Auch die Föderierung von kooperierenden Organisationen setzt Standards voraus. Einige wichtige Standards sind durch die Entwicklung von Sicherheitsspezifikationen für Web Services entstanden, hier sind insbesondere OASIS und die Liberty Alliance zu nennen [Wind05, S. 122].

Mit dieser ersten Übersicht in diesem Abschnitt wurden die wichtigsten Kernbestandteile des Identity Managements eingeführt, die im weiteren Verlauf der Arbeit vertieft werden:

- Authentisierung
- Autorisierung
- rollenbasierte Zugriffskontrolle
- Provisionierung
- Accounting und Auditierung
- Föderierungsdienste

Weitere Funktionen und Bestandteile wie die Administration von Policies, Rollen und Workflows, Verzeichnisdienste und Directories, Public Key Infrastrukturen (PKI) werden nur kursorisch dargestellt.

3 Referenzmodelle

Gegenstand dieses Kapitels sind Referenzmodelle, ihre Einordnung in grundlegende Konzepte und die Methoden ihrer Erstellung. Der Begriff des Referenzmodells wird aufbauend auf grundlegenderen Konzepten eingeführt. Hierzu erfolgt als erstes die Klärung des allgemeinen Modellbegriffs.

3.1 Modelle und Informationsmodellierung

Stachowiak hat in seiner allgemeinen Theorie für Modelle drei wesentliche Merkmale herausgearbeitet [Stac73, S. 129 ff.]. Dies ist erstens der Abbildungscharakter. Das Modell steht zum Original in einer abbildenden Beziehung und gibt dieses wieder. Dabei wird das Original zweitens nur verkürzt oder auch abstrahiert abgebildet. Die Entscheidung, welche Teile dabei hervorgehoben und welche vernachlässigt werden, stellt das dritte Merkmal dar. Diese Entscheidung wird nach pragmatischen Kriterien getroffen.

Brocke charakterisiert den Modellbegriff im Sinne Stachowiaks als abbildungsorientiert und leitet daraus in Anlehnung an Arbeiten von Schütte eine eigene, konstruktionsprozess-

orientierte Definition ab [Broc03, S. 15/16; Schü98a, S. 41f und S. 59 ff]:
„Ein Modell ist die Verdichtung von Wahrnehmungen zu Inhalten eines Gegenstands, um auf diese Weise einem spezifischen Zweck zu dienen. Die Gestaltung von Modellen erfolgt in Konstruktionsprozessen“.

Modelle werden durch den Prozess der Modellierung erstellt [Broc03, S. 24]. In seiner Definition des Begriffs Modellierung betont Brocke dabei besonders den Prozessaspekt: „Mit dem Begriff der Modellierung wird ein Arbeitsgebiet bezeichnet, das die Gestaltung und Ausführung von Prozessen im Zusammenhang mit der Konstruktion von Modellen zum Gegenstand hat [Broc03, S. 25].“

Informationsmodelle stellen Spezialfälle des allgemeinen Begriffs Modell dar. Betrachtet man das Gebiet der Wirtschaftsinformatik, dienen Informationsmodelle der Beschreibung von Informationen betrieblicher Systeme. Informationsmodelle werden in der Regel für die Organisations- und Anwendungssystemgestaltung eingesetzt [Schü98a, S. 63]; [Bec+00, S. 88]. Beispiele für die Anwendung im Kontext einer Organisation sind etwa die Abbildung und Gestaltung betrieblicher Abläufe. Wird eine Standardsoftware für einen konkreten Einsatzzweck in einem Unternehmen angepasst, handelt es sich um eine anwendungssystemorientierte Verwendung.

Die Erstellung von Informationsmodellen, die Informationsmodellierung, stellt ebenfalls ein Teilgebiet des Bereichs der Modellierung dar, das sich durch die Betrachtung von Informationsmodellen auszeichnet [Broc03, S. 30]. Die Informationsmodellierung kann in ihrer Bedeutung für die Untersuchung, Entwicklung und Veränderung von Informationssystemen nicht hoch genug bewertet werden [Fet+05, S. 1].

Um die Qualität von Modellen und des Modellierungsprozesses beurteilen und steuern zu können, wurden verschiedene Ansätze erarbeitet [Schü98a, S. 156ff; Schu05, S. 204]. Stellvertretend für diese Ansätze werden in Anlehnung an Brocke nachfolgend kurz die Grundsätze ordnungsmäßiger Modellierung (GOM) vorgestellt [Broc03, S. 146]. Brocke begründet seine Bevorzugung der GOM gegenüber anderen Beurteilungskriterien mit ihrer Eignung zur Bewertung von Informationsmodellen im Allgemeinen und Referenzmodellen im Speziellen [Broc03, S. 146].

Bei den Grundsätzen ordnungsmäßiger Modellierung handelt es sich um Gestaltungsempfehlungen, die einen Ordnungsrahmen mit konkreten Hinweisen bilden. Die Regeln des Ordnungsrahmens sind zueinander nicht vollständig widerspruchsfrei [Schü98a, S. 119ff].

Durch den Grundsatz der Richtigkeit wird die Forderung nach syntaktischer und semantischer Korrektheit eines Modells aufgestellt. Der Grundsatz der Relevanz verlangt, dass durch ein Modell nur diejenigen Objekte abgebildet werden, die für die Zielerreichung erforderlich sind. Fällt die Kosten-Nutzen-Relation eines Modells angemessen aus, erfüllt es den Grundsatz der Wirtschaftlichkeit. Durch den Grundsatz der Klarheit wird die Anforderung ausreichender Strukturiertheit und Übersichtlichkeit von Modellen erhoben.

Modelle sollten zudem mit anderen Modellen desselben Sachverhalts kompatibel sein, dann entsprechen sie dem Kriterium des Grundsatzes der Vergleichbarkeit. Verfügen Modelle darüber hinaus über ein sichtenübergreifendes Metamodell, dann erfüllen sie auch den Grundsatz des systematischen Aufbaus.

Diese Grundsätze wurden in späteren Entwicklungsschritten des Ordnungsrahmens noch um die Kriterien der Konstruktions- sowie der Sprachadäquanz erweitert. Während unter Konstruktionsadäquanz eine problemangemessene Nachvollziehbarkeit der Modellkonstruktion zu verstehen ist, meint Sprachadäquanz, dass die ausgewählte Sprache den Anforderungen hinsichtlich der Sprachrichtigkeit und Eignung entspricht.

3.2 Referenzmodelle - Definition und Eigenschaften

Referenzmodelle können als Spezialisierung von Informationsmodellen angesehen werden. Im Gegensatz zu Informationsmodellen abstrahieren sie von einem konkreten Einzelfall und werden mit dem Anspruch erstellt, dass sie für mehrere vergleichbar gelagerte Anwendungszusammenhänge Gültigkeit zu besitzen [Beck04, S. 325]. Referenzmodelle stehen für eine Klasse vergleichbarer Fälle, ein Applikationsmodell repräsentiert demgegenüber ein bestimmtes, spezifisches System. Ein Referenzmodell eignet sich daher als Vorlage für mehrere gleichartige Anwendungsfälle, es muss jedoch angepaßt werden, um den konkreten Erfordernissen gerecht zu werden [Fet+05, S. 1].

Ein Referenzmodell ist ein Bezugspunkt für die Entwicklung spezifischer Modelle, da es eine Kategorie von Anwendungen repräsentiert [Thom05, S. 16]. Die Software SAP R/3 im Bereich des Enterprise Resource Managements ist eines der bekanntesten Beispiele und kann als implementiertes Referenzmodell aufgefasst werden [FeLo04, S. 336]. Gerade dieses Beispiel verdeutlicht die Einsatzmöglichkeiten für Referenzmodelle in der Anwendungssystementwicklung und der Organisationsgestaltung [Schl00, S. 54] [Simo98, S. 100] [Thom05, S. 17].

In der Literatur wird der Begriff des Referenzmodells nicht einheitlich gebraucht [Fett06, S. 19] [Thom05, S. 16]. Dies hängt unter anderem damit zusammen, dass viele Autoren ihre Arbeiten nicht in die Forschungslandschaft einordnen und ihre Entwicklungen nicht positionieren [MeHo92, S. 21].

Nach Hars handelt es sich bei jedem Referenzmodell um ein Modell, das für den Entwurf anderer Modelle herangezogen werden kann [Hars94, S. 15 – 18].

Gemäß einer Definition nach Becker spiegeln Referenzmodelle nicht die Gegebenheiten eines spezifischen Objekts (Unternehmung) wider, sondern gelten für eine Klasse von Objekten. Sie zeichnen sich durch einen höheren Abstraktionsgrad als (unternehmens-) spezifische Modelle aus und beinhalten häufig eine größere Anzahl von Teilmodell-Alternativen, die unterschiedliche (Unternehmens-) Szenarios wiedergeben. Referenzmodelle können auf einer technischen Ebene angesiedelt sein oder einen betriebswirtschaftlichen Domänenbezug aufweisen [Beck01, S. 399].

Das in Referenzmodellen enthaltene funktionsbereichs- oder branchenbezogene Domänenwissen bietet die Möglichkeit, die Risiken des Modellierungsprozesses durch hohe Aufwände und inhaltliche Fehler zu begrenzen [Knac01]; [Prob03, S. 44ff]; [Schü98a, S.69]; [FeLo02, S. 3ff] [Fet+05, S. 1].

Damit betonen diese Definitionen insbesondere die Nutzenpotentiale von Referenzmodellen, die sich aus der Wiederverwendung ergeben, wie Kostensenkung, Zeitersparnis, Qualitätssicherung und Risikoreduktion [Prob03, S. 48]; siehe auch [Schü98a, S. 76]. Der Einsatz eines Referenzmodell garantiert jedoch nicht immer die beste Lösung, ggf. erreicht man nur eine geringere Anpassung an die Anforderungen, als eine Eigenentwicklung sie geboten hätte. Schließlich sind auch die Beschaffung und Anpassung eines Referenzmodells mit Aufwänden verbunden [Beck04, S. 325].

In einer großen Zahl an Veröffentlichungen im Bereich der Referenzmodellierung werden Referenzmodelle dahingehen charakterisiert, dass sie den Anspruch erheben , die gegenwärtige Best Practice zu repräsentieren und damit einen Empfehlungscharakter zu besitzen [Schl00, S. 54]; [Schü98a, S. 69]; [Beck04, S. 325]; [FeLo04, S. 332]; [Broc03, S. 31ff]. Hochstein, Zarnekow und Brenner weisen in diesem Zusammenhang darauf hin, dass nicht jedes Referenzmodell diesem Anspruch gerecht wird [Hoc+04, S. 383]. Sie gehen davon aus, dass das wieder verwendbare Wissen in Referenzmodellen eher als Common Practice bezeichnet werden kann, also nicht so sehr eine optimale als eine zur Zeit übliche Lösung darstellt.

Thomas listet in einer Zusammenschau zum Begriff des Referenzmodells allein 30 Definitionen auf, aus denen er die folgenden beiden Hauptpunkte Universalität und Empfehlungscharakter herausarbeitet [Thom05, S. 20]. Beide haben seiner Auffassung nach jedoch keinen Bestand. Die Allgemeingültigkeit eines Referenzmodells kann seines Erachtens nicht absolut oder universell verstanden werden. Vielmehr hält er eine beschränkte Gültigkeit für eine Klasse von Anwendungsfällen für vertretbar [Thom05, S. 20].

In Bezug auf den Empfehlungscharakter stellt Thomas die intersubjektive Überprüfbarkeit der Qualität eines Referenzmodells in Frage auf und verwirft im Ergebnis auch dieses Charakteristikum als konstituierendes Merkmal des Begriffs.

In seiner eigenen Definition des Begriffs verfolgt Thomas einen anderen Ansatz, bei dem er sich auf die Gruppen der Ersteller und Nutzer von Referenzmodellen bezieht [Thom05, S. 23]. Die zumindest einmalige Anwendung eines Referenzmodells durch einen Nutzer, die benutzerseitige Akzeptanz, ist demnach das entscheidende, charakterisierende Merkmal für den Begriff des Referenzmodells aus der Sicht von Thomas. Referenzmodelle sind Informationsmodelle, die für die Konstruktion anderer Modelle verwendet werden.

In Bezug auf das Charakteristikum des hohen Abstraktionsgrads bezweifelt Fettke, dass sich dieser objektiv bestimmen lässt. Modellierung kann unterschiedlich abstrakt sein, und es ist nicht klar, ab welchem Grad ein Modell daher ein Referenzmodell zu nennen ist [Fett06, S. 24].

Der Terminus „Referenzmodell“ wird also nicht einheitlich definiert und zudem unterschiedlich gebraucht. Die vorherrschenden Begriffsexplikationen führen zu formalen Irritationen. Die genannten Merkmale, die ein Modell erfüllen soll, sind zum Teil nicht intersubjektiv zugänglich und führen zu tautologischen und daher inhaltsleeren Bestimmungen oder basieren auf weiteren und unklaren Begriffen, für die keine Definition geliefert werden [Fett06, S. 27 / 28].

Gemäß Fettke stellen eigentlich erst dann eine relevante Forschungsleistung dar, wenn u.a. zahlreiche Anwendungen in verschiedenen Unternehmen nachgewiesen werden [Fett06, S. 253 / 254].

Die Definition von Thomas und in Teilen Fettkes erscheinen im Hinblick auf die Zielsetzung dieser Arbeit als zu streng. Die Entwicklung eines Referenzmodells ohne den Beleg seiner Anwendung als Ganzes und nicht nur einzelner Bestandteile würde dem Begriff nach Thomas nicht gerecht werden.

Demgegenüber wird eine Definition nach vom Brocke als Orientierung in dieser Arbeit verwendet werden. Die Definition bezieht zwar die Modellnutzung durchaus mit ein, ist aber durch die Betonung des Wiederverwendungszusammenhangs etwas pragmatischer ausgerichtet.

„Ein Referenzmodell (aus-führlich: Referenz-Informationsmodell) ist ein Informationsmodell, das Menschen zur Un-terstützung der Konstruktion von Anwendungsmodellen entwickeln oder nutzen. Dabei ist die Beziehung zwischen Referenz- und Anwendungsmodell dadurch gekennzeichnet, dass Gegenstand oder Inhalt des Referenzmodells bei der Konstruktion des Gegenstands oder Inhalts des Anwendungsmodells wieder verwendet werden“ [Broc03, S. 34].

3.3 Referenzmodelle - Erstellung und Anwendung

Für die Entwicklung und Anwendung von Referenzmodellen existieren Referenzmodellierungsmethoden, die Handlungsempfehlungen für Modellierungsträger und –anwender für den Konstruktions- und Anwendungsprozess umfassen [FeLo04, S. 334] [Schl00, S. 60].

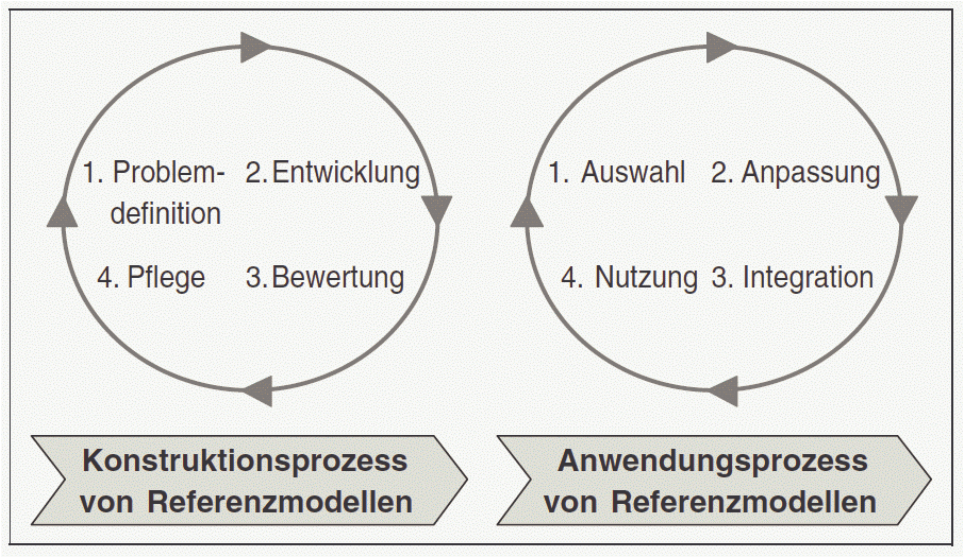

Abbildung 3: Prozesse der Referenzmodellierung [FeLo05, S. 22]

Thomas kritisiert aus methodologischem Blickwinkel, dass es aufgrund der unüberschaubaren Vielzahl von Vorschlägen Referenzmodellerstellern und –nutzern schwerfällt, zu entscheiden, welche Methoden, Techniken und Sprachen nun für den eigenen Fall adäquat sind [Thom06, S. 156].

Bei den Methoden der Referenzmodellierung lassen sich Methoden der Konstruktion von Referenzmodellen und Methoden der Anwendung unterscheiden. Die Erstellung von Referenzmodellen durchläuft Phasen der Problemdefinition, der Entwicklung und der Bewer-

tung. Im Hinblick auf eine mögliche Anwendung stehen Schritte der Auswahl, der Anpassung und der Nutzung an [Schl00, S. 77ff].

Eine durchgehend hohe Qualität des Entwicklungs- und Anwendungsprozesses kann durch Vorgehensmodelle unterstützt werden [Schl00, S. 64]. Sie erlauben ein planmäßiges und ingenieurmäßiges Vorgehen bei der Entwicklung von Referenzmodellen [Schl00, S. 60]. Die Teilphasen der beiden Gesamtprozesse der Konstruktion und Anwendung von Referenzmodellen greifen dabei zyklisch ineinander [Broc03, S.131]. Nach Auffassung von Brocke hat dabei das von Schütte entwickelte Vorgehensmodell den größten Einfluß auf den heutigen State of the Art genommen [Broc03, S. 133] [Schü98a].

Neben empirischen (induktiven) finden sich auch theoriegeleitete (deduktive) Methoden der Konstruktion [Fet+05, S. 9ff; Schu05, S. 198]. Fettke, Loos und Zwicker merken hierzu an, dass induktiv gewonnene Referenzmodelle noch nicht realisierte Möglichkeiten vernachlässigen, während auf Schlussfolgerungen und theoretischen Annahmen beruhende Modelle teilweise an der Realität vorbeientwickelt werden [Fet+05, S. 9]. Der darin zu erkennende Zielkonflikt aus präzisen und konsistenten Modellen einerseits und verständlichen Arbeitshilfen andererseits lässt sich nur schwer auflösen [Fet+05, S. 9].

Die erste Phase der Erstellung eines Referenzmodells dient der Definition des zu lösenden Problems bzw. der zu erreichenden Ziele [Bec+02, S. 34ff; Broc03, S 134; Schl00, S. 65; Prob 03, S. 51]. Als Anwendungsdomäne kommen bspw. betriebswirtschaftliche Funktionen wie etwa das Controlling oder Branchen wie bspw. der Handel in Frage [FeLo04, S. 335]. Ist bereits zu diesem Zeitpunkt die zu bewältigende Komplexität vergleichsweise hoch, stehen Hilfsmittel wie Ordnungsrahmen zur Verfügung, die die Problemdomäne grob vorstrukturieren [Schü98, S. 184]; [Broc03, S. 134ff; Meis01, S. 61 – 64]. Eines der bekanntesten Beispiele ist das Y-CIM-Modell [Sche97].

Desweiteren sind Konventionen wie die Beschreibungssprache festzulegen, mit deren Hilfe z.B. dynamische und statische Eigenschaften abgebildet werden können [Schl00, S. 159]; [Prob03, S. 53; Broc03, S. 131]. In der Regel kommen verbreitete und gängige Modellierungssprachen wie das Entity-Relationship-Model (ERM), ereignisgesteuerte Prozessketten (EPK) oder die objektorientierte Unified Modeling Language (UML) zum Einsatz [Rüff99, S. 86] [Schl00, S. 66] [Schl00, S. 68].

Für gewöhnlich werden zur Deklarierung von Referenzmodellen nicht-proprietäre Modellierungssprachen verwendet, die über breite Akzeptanz im Kreis der potentiellen Anwender verfügen [Schl00, S.54; Broc03, S. 107]. Eine Modellierungssprache für Referenzmo-

delle umfasst im einfachsten Falle Konzepte für die Repräsentation von Entitäten und Systemen sowie einen Satz von Regeln, die festlegen, wie die fraglichen Entitäten miteinander in Beziehung gesetzt werden können. Es können verschiedene Sprachfamilien unterschieden werden, z.B. daten-, prozess- oder objektorientierte Modellierungssprachen [FeLo04, S. 333]. Die verwendeten Modellierungssprachen sind meist auf der Fachkonzeptebene angesiedelt und in der Regel semi-formal definiert. Auf diese Weise wird eine zu Enge Verknüpfung mit einer konkreten Technologie vermieden und der Einsatz in wechselnden Umgebungen erleichtert.

Entscheidend für die Auswahl einer geeigneten Modellierungssprache ist, dass die Sprache einerseits individuelle Anforderungen erfüllt und andererseits dem Hauptziel eines Referenzmodells dient, seiner Wiederverwendung [Broc03, S. 107]. Semi-formale Darstellungen erleichtern bspw. auch Laien den Umgang mit der teilweise nicht unerheblichen Komplexität von Modellen [FeLo04, S. 334].

Im sich anschließenden Schritt wird die Modellierung des Gegenstands vorgenommen, meist in mehreren Iterationen [Prob03, S. 53]; [Broc03, S. 134 / 136]; [Schl00, S. 65]. Der Konstruktionsprozess erfährt im Optimalfall neue Anstöße aus der Anwendung eines Referenzmodells [Schl00, S. 85]; [Broc03, S.131]; [Prob03, S. 55]. Aufgrund des damit verbundenen Aufwands werden Referenzmodelle in der Praxis jedoch nur selten systematisch entwickelt [Thom06, S. 155].

Die Anwendung von Referenzmodellen erfolgt ebenfalls in mehreren Schritten, die sich auf Aspekte der Anpassung und der Spezialisierung der Modelle konzentrieren [Schl00, S. 88]; [Broc03, S. 134ff]; [Schü98a, S. 309]. Nachdem zunächst eine Klärung des Zusammenhangs der Anwendung vorgenommen wurde, beginnt die Suche nach geeigneten Modellen [Schl00, S. 86f]. In Frage kommende Modelle werden im Hinblick auf mehrere Kriterien beurteilt, wie bspw. die Verfügbarkeit, die Anwendungsdomäne, die Potentiale und die Beschränkungen [Fet+05, S. 2]. Das Merkmal Größe, für das verschiedene Metriken verwendet werden (etwa die Zahl der repräsentierten Diagramme und Sichten oder die Zahl der Prozessschritte), ist ebenfalls von Bedeutung [Fet+05, S. 4]. Auch das Vorhandensein eines Evaluationsansatz und Evaluationsergebnisse sind von Interesse [Fet+05, S. 5].

Wurde eine Auswahlentscheidung getroffen, gilt es, ein spezifisches Modell zu erzeugen [Schl00, S. 87]. Hierfür stehen verschiedene Ansätze zur Verfügung, etwa durch Konfigu-

ration, Aggregation, Spezialisierung oder Instanziierung [FeLo04, S. 336]; [Fet+05, S. 7]; im einfachsten Fall manuelles Kopieren.

Aufgrund eines Mangels an akzeptablen, standardisierten Evaluationsansätzen, -kriterien und –methoden, werden Anwendungsergebnisse nur selten einer systematischen Kontrolle unterzogen [Fet+05, S. 9]. Verbesserungspotentiale, die sich bei der unternehmensspezifischen Adaption von Referenzmodellen ergeben, fließen damit insgesamt nicht konsequent in die Referenzmodelle zurück [Thom06, S. 155].

Mit diesem Kapitel wurde gezeigt, welchen Stellenwert Referenzmodelle in Anwendung und Forschung einnehmen. Mit dem Zweck der Wiederverwendung und dem Aspekt des Know How Transfers wurden Eigenschaften von Referenzmodellen benannt und der Stand der Diskussion wurden umrissen. Inwieweit diese Kriterien und Charakteristika bei der Klärung des Verhältnisses zu Pattern Relevanz besitzen, wird im Kapitel 5 herausgearbeitet.

4 Pattern, Security Pattern und Patternsysteme

In diesem Abschnitt wird gemäß dem Ziel dieser Arbeit das Gebiet der Entwurfsmuster und Security Pattern betrachtet.

4.1 Design Pattern

Die Musteridee stammt ursprünglich von dem Architekten Christopher Alexander, der diese in Bezug auf Haus- und Städtebau entwickelt hat [Ale+77]. Muster beschreiben nach seiner Darstellung wiederkehrende Probleme und den Kern einer geeigneten Lösung derart, dass ausreichend Freiheitsgrade für die Anwendung der Lösung bestehen. Die in Mustern enthaltenen bewährten Lösungen sind adaptierbar und abstrahieren von Details [Fowl 97, S. 8]. Grundgedanken sind die Sammlung von Erfahrungswissen und seiner Wiederverwendung [Sch+06, S. 2]. Darüber hinaus sollen Muster die Kommunikation erleichtern [Bus+98, S. 5].

Das Konzept des Musters wurde bereits vor zwei Jahrzehnten in die Domäne der Softwareentwicklung übertragen [Mehl05a, S. 31]. Eine der ersten Arbeiten hierzu stammt von Kent Beck und Ward Cunningham, die 1987 die Ideen Alexanders aus der Architektur aufgriffen und Entwurfsmuster für die Programmiersprache Smalltalk entwickelten [BeCu87]. Unabhängig davon beschäftigte sich Erich Gamma an der Übertragung der Methode Alexanders auf das Gebiet der Softwareentwicklung [Gamm92, S. XI]. In den Jahren 1989 bis

1991 erarbeitete James Coplien musterähnliche Idiome für die Programmiersprache C++ und veröffentlichte diese 1992 [Copl92].

Während Muster im Bereich der Softwareentwicklung ursprünglich den objektorientierten Entwurf als Fokus hatten, entstanden mit der Zeit auch Anwendungen des Mustergedankens für alle Phasen des Entwicklungsprozesses [Mehl05a, S. 32]. So finden sich neben Analysemustern auch Muster für die Architektur von Software, Muster zur Projektgestaltung, aber auch sogenannte Anti-Muster [Bro+98].

Die auch als “Gang of four Book“ bekannte Veröffentlichung von Gamma et al. definiert Design Pattern als “Beschreibungen kommunizierender Objekte und Klassen, die so angepasst sind, dass sie ein allgemeines Designproblem in einem bestimmten Kontext lösen” [Gam+94, S. 3 ff]. Oder noch komprimierter in einer einführenden Begriffsabgrenzung durch Schumacher et al.: „Ein Muster ist die Lösung für ein Problem in einem spezifischen Kontext“ [Sch+06, S. 2].

Die Zahl und die Vielfalt der zwischenzeitlich entwickelten Muster legen eine Einteilung nach Klassifikationskriterien nahe [Mehl05a, S. 33]. Ein Ansatz ist die Klassifikation nach verschiedenen Abstraktionsstufen [Corf98]. Buschmann et al. klassifizieren neben dem Abstraktionsniveau noch nach dem Gegenstands- oder Problembereich eines Musters. Bsp. hierfür sind etwa „Verteilte Systeme“ oder „Zugriffskontrolle“ [BuMe95, S.329ff].

Für die Dokumentation von Mustern hat sich in der Pattern-Community ein natürlichsprachlicher Schreibstil etabliert [Mehl05a, S. 41] [Sch+06, S. 9]. Diese Aussagen können durch stärker formalisierte Modelle, etwa aus dem objektorientierten Umfeld, ergänzt werden.

Die Konventionen und Notationen zur Beschreibung von Mustern werden als Musterschema bezeichnet und können anhand ihres Strukturierungsgrad unterschieden werden. Die Beschreibungen variieren zwischen einem narrativen Stil und einer strukturierten, formularartigen Darstellung [Mehl05a, S. 41]. Bei letzteren werden die drei wesentlichen Gliederungspunkte Problem, Kontext und Lösung durch Spezifika der jeweiligen Domäne abgewandelt.

Eine Zusammenstellung der bekanntesten Musterschema der Autoren Gamma et al. und Buschmann et al. stammt von Mehlau [Mehl05a, S. 41]. In der Zusammenstellung lassen sich Gruppen von Abschnitten unterscheiden, bspw. für die Identifizierung eines Musters, die Verallgemeinerung der Lösung und konkrete Implementierungshinweise. Hinzu kom-

men noch bibliographische Verweise und Referenzinformationen. Mehlau konstatiert insgesamt einen Trend zur strukturierten Beschreibung [Mehl05a, S. 42].

Neben der Beschreibung einzelner Muster sind auch Zusammenfassungen von Mustern entstanden, die in einem Verwendungszusammenhang stehen und untereinander Abhängigkeiten aufweisen [Mehl05a, S. 43]. Mehlau unterscheidet Mustergesamtheiten anhand der Zahl der Muster, der Zahl der beitragenden Autoren, der Vollständigkeit und der Spezifität.

Mustersammlungen weisen verschiedene Einzelmuster ohne thematischen Fokus und ohne näher betrachtete Interdependenzen auf. Ist die Mustersammlung nach einheitlichen Kriterien klassifiziert, spricht man von Musterkatalogen [Schu03, S. 15].

In einem Mustersystem werden Muster nach aufeinander abgestimmten Beschreibungen zusammengestellt und speziell auch in ihren gegenseitigen Abhängigkeiten beschrieben. Erfüllt das Mustersystem darüber hinaus einen Anspruch auf Vollständigkeit in Bezug auf eine eng umgrenzte Problemdomäne, handelt es sich um eine Mustersprache [Bus+98, S. 358].

Der Begriff einer Sprache im Kontext von Mustern stammt als Konzept aus den Arbeiten von Christopher Alexander und darf nicht mit dem Begriff einer Sprache aus anderen Zusammenhängen der Wirtschaftsinformatik (z.B. Auszeichnungssprachen, Programmiersprachen) gleichgesetzt werden [Mehl05a, S. 44].

Ein Muster wird nicht am grünen Tisch entworfen, es ist das Ergebnis der Auseinandersetzung eines erfahrenen Musterautors mit bewährten Lösungen [Mehl05a, S. 45]. Dabei wird versucht, die abstrakte Kernidee herauszuarbeiten, die mehreren verwandten Lösungen gemein ist. Ist das Muster erst einmal verständlich beschrieben, beginnt der iterativ angelegte Community-Prozess, der dem Peer-Review, der Qualitätssicherung und Kommunikation im kleineren Expertenkreis dient [Schu03, S. 20]. Hat das Muster den Prozess erfolgreich durchlaufen, wird es in den allgemeinen Fundus der Pattern-Community aufgenommen und bei ausreichender Relevanz auch in Print-Publikationen veröffentlicht.

Muster machen die Erfahrungen und das Wissen aus erfolgreichen Softwareentwürfen für die Wiederverwendung verfügbar [Mehl05a, S. 47]. Mehlau führt zahlreiche Beispiele für positive Erfahrungen mit Mustern aus der Praxis auf: Muster unterstützen die Kommunikation in einem Team, indem sie Hilfsmittel zur Reduktion von Komplexität liefern. Die prägnante Schilderung der Kernidee erleichtert die Beibehaltung eines Ansatzes über den reinen Entwurf hinaus. Gute Erfahrungen mit einer Lösung lassen sich rascher übernehmen.

Pattern unterstützen außerdem den Prozess der menschlichen Problemlösung, indem sie Schemata und Heurismen liefern [Schu03, S. 24/25].

Zu den offenen Fragen im Zusammenhang mit Mustern gehört z.B., in welchem Umfang man sie einsetzen sollte, in welchem Kontext und wo besser nicht [Mehl05a, S. 52]. Die zunehmende Menge an Mustern erschwert für potenzielle Anwender den Prozess des Findens eines geeigneten Musters. Von den existierenden Klassifikationsschemata hat sich bislang keines als Standard durchsetzen können.

Offen ist auch, wie aus einem Muster eine konkrete Implementierung abzuleiten ist. Bei programmiersprachennahen Mustern erscheint dies in Form von Entwicklungsumgebungen machbar, bei abstrakteren Architekturmustern kommen vermutlich nur Werkzeuge für IT-Bebauungspläne in Frage. In der Zusammenschau rät Mehlau zu einem gewissen Maß an Skepsis in der Bewertung [Mehl05a, S. 53].

4.2 Sicherheit im Softwareentwicklungsprozess

Die Erzeugung und Aufrechterhaltung eines sicheren Betriebsstatus von IT-Systemen ist kein einfaches Unterfangen. Ansatzpunkte für Sicherheitsmaßnahmen bestehen auf verschiedenen Schichten wie Hard- und Software über die gesamte Breite des IT-Stacks [FeLa06, S. 8]. Ein sicherer Systemzustand ist dabei von mehreren Faktoren abhängig, sowohl von der Umsetzung und Planung als auch von den dazugehörigen Maßnahmen. Hierzu ist eine ingenieurmäßige Planung und Durchführung mit einem hohen Maß an Professionalität gefordert [Mehl05a, S. 55].

Die Kombination mehrerer Maßnahmen hat dabei Vor- und Nachteile. Zum einen können bestehende Lücken im Konzept durch die Wirkung andere Maßnahmen aufgefangen werden, andererseits können einzelne Maßnahmen auch kontraproduktiv sein und zu massiven Sicherheitsgefährdungen führen. Die ingenieurmäßige Umsetzung prinzipiell bekannter Maßnahmen weist jedoch oft genug Mängel auf, da es am entsprechenden Fachwissen fehlt [Bish02, S. 21].

Rosado et al. bemängeln, dass der Aspekt Sicherheit im Entwicklungsprozess von Software viel zu oft erst im Nachhinein Berücksichtigung findet, dies ist in der Regel dann jedoch nicht mehr möglich [Ros+06, S. 139]. Sie betonen, dass Sicherheit von vorneherein als Gestaltungsprinzip berücksichtigt werden muss. Schumacher führt als weitere Ursache für diesen Missstand eine unzureichende Gefährdungseinschätzung an [Schu03, S. 52 ff].

Analysiert man den Prozess der Softwareentwicklung unter Sicherheitsgesichtpunkten, besteht in Anbetracht der auftretenden Fehler ein Nachholbedarf bei Methoden und Werkzeugunterstützung zur Erhöhung des Niveaus der IT-Sicherheit [Mehl05a, S. 56]. Bisherige Ansätze zur Erstellung sicherer Software haben sich als mangelhaft herausgestellt, nicht zuletzt aufgrund der Komplexität durch Interdependenzen, die eine ganzheitliche Planung erfordern [Mehl05a, S. 57].

Neben Ansätzen zur Erhöhung der Applikationssicherheit wie „Model Driven Security“ [Bas+05] oder „Business-driven Application Security“ [Nag+05, S. 847] gewinnt das Konzept der Security Patterns zunehmend an Bedeutung. Muster stellen einen vielversprechenden Ansatz zur Verbesserung der Sicherheit von Software dar [Hal+06a, S. 379/ 380; FeLa06, S. 2; Ste+06, S. 440]. Durch Security Patterns wird der Gedanke der Design Patterns aufgegriffen und auf die Domäne der Sicherheit übertragen.

Pattern erleichtern die Weitergabe und Wiederverwendung von sicherheitsrelevantem Know How. Außerdem können Abhängigkeiten auf verschiedenen Abstraktionsstufen durch Muster anschaulich modelliert werden [Mehl05a, S. 57].

Damit Software nicht erst durch Testen sicherer wird, plädieren Steel et al. dafür, dass Software nur in einem strukturierten Prozess entwickelt werden sollte, dessen Schritte Best Practices berücksichtigen und Wiederverwendung unterstützen [Ste+05, S. 439]. Außer in der Anforderungsanalyse, der Architektur- und der Entwurfsphase sollten auch während des Betriebes pro- und reaktive Maßnahmen ergriffen werden. Das setzt eine Methodologie mit einem systematischen und wohldefinierten Ansatz voraus, aus Sicht von Steel et al. sollten dabei Security Pattern zum Einsatz kommen [Ste+05, S. 440].

Schumacher et al beschreiben einen Ansatz für das Sicherheits- und Risikomanagement in Unternehmen, bei dem Security Pattern die Basis bilden [Sch+06, S. 85]. Auf eine Vertiefung dieses umfassenden Sicherheitsansatzes wird jedoch zugunsten wichtigerer Gesichtspunkte dieser Arbeit verzichtet.

4.3 Sicherheitsmuster und der Stand der Forschung

Eine der ersten Publikationen zur Anwendung von Mustern für den Bereich der IT-Sicherheit stammt von Yoder und Barcalow [YoBa97]. In ihrer Arbeit kritisieren Yoder und Barcalow, dass Sicherheitsaspekte in der Entwicklung von Anwendungssystemen oft erst viel zu spät berücksichtigt werden und nicht von Beginn an. Die beiden Autoren stellen hierzu unter anderem Pattern zur Identifikation und Zugriffskontrolle vor [YoBa97].

In den letzten zehn Jahren hat sich die Arbeit an Sicherheitsmustern kontinuierlich weiter entwickelt [Schu03, S. 99]. Muster zu Authentifizierung werden von Smith beschrieben [Smit02]. Auch zur Autorisierung finden sich mittlerweile mehrere Muster, z.B. [Mehl05a, S. 60]. Für das rollenbasierte Zugriffskontrollmodell RBAC (Role Based Access Control) existieren ebenfalls Sicherheitsmuster [z.B. Sch+06, S. 249]. Romanosky et al. haben Pattern zur Absicherung der Vertraulichkeit entwickelt [Rom+06]. Von Fernandez et al. stammt eine Mustersprache für Firewalls [Fer+05]. Braga et al. veröffentlichten ebenfalls eine Mustersprache, und zwar für kryptographische Funktionen [Bra+98]. Einen guten Überblick über die Security Patterns Landschaft geben Schumacher et al. in ihrem aktuellen Buch [Sch+06, S. 59].

Security Pattern kapseln das Wissen von Sicherheitsexperten für wiederkehrende Probleme und bieten so Lernmöglichkeiten und Kommunikationsmittel [Ros+06, S. 139]. Wissen wird durch Security Patterns in vertrauten Repräsentationsformen in einer für alle Abstraktionslevel (z.B. Implementierungs-, Architektur- und Unternehmensebene) einheitlichen Weise dokumentiert [Sch+06, S. 34].

Die Definition des Sicherheitsmusters stellt einen Spezialfall der allgemeineren Definition des Entwurfsmusters dar [Mehl05a, S. 58]. Ein Security Pattern beschreibt ein bestimmtes wiederkehrendes Sicherheitsproblem, das in spezifischen Sicherheitskontexten auftritt und ein bewährtes Schema für eine Lösung liefert [Schu03, S. 10]. Die Lösung besteht aus einem Set interagierender Bestandteile, die zu mehreren Strukturen arrangiert werden können sowie eine Beschreibung, wie diese Strukturen erzeugt werden können [Sch+06, S. 31].

Aus Sicht von Heyman et al. ist diese Definition nicht unproblematisch. Nach Heyman et al . fehlt es dieser Bestimmung an Klarheit und sie führt dazu, dass Muster ganz unterschiedlicher Qualität in dieselbe begriffliche Klasse fallen [Hey+07, S. 2]. Existierende Security Pattern decken verschiedene Abstraktionslevel ab und sind teilweise so abstrakt, dass die Autoren Mühe haben, sie Pattern zu nennen [Hey+07, S. 3].

Heyman et al. plädieren dafür, allgemeine Richtlinien, abstrakte Aktivitäten und Sicherheitsprinzipien nicht als Pattern anzusehen. Dasselbe gilt ihres Erachtens für sehr detaillierte Programmieranweisungen. Als Alternative schlagen Heyman et al. eine Orientierung an der Definition von Entwurfsmustern von Coplien vor. Nach ihrer engeren Definition sollten Security Pattern nicht nur beschreiben, was das Ergebnis sein sollte (Das „Was“), sondern auch das „Wie“, also die Strategie, die Lösung zu erreichen [Hey+07, S. 3]. Legt man diese strenge Security Pattern Definition an (sinnvoll eingegrenztes Problem und mehrfach

bewährte Konstruktionshinweise für eine Lösung), dann können nur 55% der von Heyman et al. gesichteten Security Pattern als „Core Patterns“ bezeichnet werden, während die verbleibenden 35% Richtlinien oder Prozessaktivitäten darstellen (10%) [Hey+07, S. 3].

Als Beleg für die zunehmende Verbreitung und Akzeptanz von Security Patterns verweisen Fernández und Larrondo Petrie auf Unternehmen wie Microsoft, Sun und IBM, die Bücher, Papers und Webseiten zu diesem Thema veröffentlicht haben [FeLa06, S. 11]. Inzwischen existiert auch eine allgemeine Webseite für Security Patterns, die im wesentlichen von der Pattern Community getragen wird [Secu07].

Das Interesse an Security Pattern spiegelt sich in den Ergebnissen einer Untersuchung von Heyman et al. zur Akzeptanz von Security Pattern [Hey+07, S. 1]. Heyman et al. untersuchten 220 Pattern des Zeitraums 1996 bis 2006 und verwenden zur Illustration eine Grafik und die Terminologie des sogenannten „Hype Cycle“ der Beratungsfirma Gartner [Hey+07, S. 1]. In einer Verlaufskurve der Aufmerksamkeit erreicht demnach eine neue Technologie rasch einen Gipfel der übertriebenen Erwartungen, dem ebenso schnell ein Tal der Desillusionierung folgt. Daran schließt sich eine langsam ansteigende Kurve (slope of enlightenment) an, die mit vertieftem Verständnis der Anwendbarkeit, der Vorzüge sowie der Nachteile der Neuigkeit einhergeht.

Die wachsende Zahl der veröffentlichten Pattern macht deutlich, dass es schwerer wird, für ein gegebenes Problem eine geeignete Lösung zu finden. Die Situation wird auch nicht unbedingt einfacher, da gleichartige Security Patterns oft mit unterschiedlichen Namen publiziert wurden [Hal+06a, S. 380]. Ein Hilfsmittel können Klassifikationsvorschläge sein.

Zu diesem Zweck diskutieren Hafiz und Johnson verschiedene Klassifikationsschemata. Ein Vorschlag zieht typische Systemstrukturen in Betracht und teilt Pattern daraufhin ein, welcher Systemteil abgesichert werden soll [HaJo06, S. 17]. Dabei kann weiter zwischen core security, perimeter security und externer security unterschieden werden. Während die Kernsicherheit interne Sicherheitsmechanismen eines Systems zum Gegenstand hat, betrachtet die perimeter security z.B. die Authentisierung und die Autorisierung an Eingangspunkten des Systems. Zur externen security gehören z.B. Aspekte der Datenübertragung und sicherer Kommunikationsprotokolle.

Hafiz und Johnson berücksichtigen in ihrem eigenen Klassifikationsansatz für Security Pattern den Applikationskontext und den STRIDE-Ansatz [HaJo06, S. 21]. STRIDE ist ein Akronym der nachfolgenden Konzepte [HaJo06, S. 19]: *Spoofing* (Systemzugriffsversuch mit manipulierter Identität [HaJo06, S. 20]), *Tampering* (Veränderung von Kommunikati-

onsdaten (Integritätsverlust)), *Repudiation* (Abstreiten einer Transaktionsteilnahme), *Information disclosure* (Vertraulichkeitsverlust von Daten), *Denial of service* (Angriff auf die Systemverfügbarkeit), *Elevation of privilege* (Erschleichung umfassender Zugriffsrechte, z.B. mittels gefälschter Identität).

Avgeriou und Zdun klassifizieren Architekturpattern nach ihrem Verwendungszweck, bspw. Datenfluss, Benutzerinteraktion oder Verteilung. Security Patterns können dem entsprechend nach ihrem Hauptmechanismus klassifiziert werden, etwa Zugriffskontrolle oder Authentisierung [AvZd05].

Fernandez und Larrondo Petrie beziehen sich bei ihrer Klassifikation auf den IT-Stack von Computersystemen als hierarchisch angeordnete Schichten ausgehend von der Dienst- und Applikationsschicht, über die Datenbank- und Betriebssystemschicht, die wiederum auf der Hardwareschicht aufsetzt. Vorteil dieses Vorschlags ist es, daß Sicherheitsrichtlinien auf der obersten Schicht mit leicht nachvollziehbarer Semantik definiert werden, die Umsetzung erfolgt auf den darunter liegenden Ebenen. Da alle Schichten der Architektur den gleichen Sicherheitsstandards genügen müssen, sollten auf allen Ebenen Pattern eingesetzt werden, da Standards so leichter umgesetzt werden können [FeLa06].

Mowbray und Malveau unterscheiden Pattern anhand ihres Abstraktionslevels, z.B. „Application“, „Host“, „Network“ [MoMa97]. Das Abstraktionsniveau wird in der Musterbewegung häufig verwendet und hat sich dabei als nützlich herausgestellt. [Mehl05a, S. 73]

Die Dreiteilung in „Enterprise“, „System“ und „Application“ wird durch Mehlau noch durch eine programmiersprachennah und eine unternehmensübergreifende Kategorie ergänzt [Mehl05a, S. 74], so dass in Anlehnung an Mowbray und Malveau [MoMa97, S. 60], dann fünf Abstraktionsebenen unterschieden werden können.

Auch bei Sicherheitsmustern existieren verschiedene Abstraktionsniveaus wie der Entwurf, die Architektur oder unternehmensübergreifend [Mehl05a, S. 64]. Security patterns addressieren Sicherheitsaspekte nicht nur auf der Unternehmensebene, auch Ebenen der Architektur und der Benutzer werden abgedeckt [Rom+06, S. 2].

Mehlau geht bei seiner Klassifikation davon aus, dass sich ein gewisses Grundverständnis von Sicherheitsgrundfunktionen sowohl in wissenschaftlichen Arbeiten als auch bei Sicherheitsstandards herausgebildet hat [Mehl05a, S. 71]. Er schließt daraus, ausgewählte Sicherheitsgrundfunktionen sind hilfreich für die Strukturierung von Sicherheitsmustern [Mehl05a, S. 72]. Zu den Kategorien gehören z.B. Identifikation und Authentifikation, Rechteverwaltung und Prüfung sowei die Kategorie Protokollierung [Mehl05a, S. 75-77].

Bei weitem am besten durch Pattern abgedeckt ist das Gebiet der Zugriffskontrolle und seine Unterziele, Identifikation, Authentisierung und Autorisierung [Hey+07, S. 6].

4.4 Musterschema und Sicherheitsmustersysteme

Für Security Pattern liegen – wie auch für Design Pattern - Beschreibungsschemata vor, schwerpunktmäßig in strukturierter Textform. Die Beschreibungen bestehen aus fünf grundlegenden Elementen (siehe weiter oben in diesem Kapitel): Der Name, der Kontext, und der Problemabschnitt [Mehl05a, S. 80]. Darauf folgen der Lösungsabschnitt mit den Teilen Generelle Form, Lösungstyp, bekannte Einsatzgebiete sowie Konsequenzabschnitt und Abhängigkeiten [Mehl05a, S. 80].

Im Informationssicherheitsbereich müssen Teillösungen sinnvoll koordiniert werden. Analog zum Bereich der Softwareentwicklung geschieht die Modellierung, insbesondere der Wechselbeziehungen von Sicherheitsmustern in einem Sicherheitsmustersystem [Mehl05a, S. 80].

Der Mehrwert eines Mustersystems sind die erkannten und beschriebenen Interdependenzen. Ein Sicherheitsmustersystem ist ein konsistenter Rahmen zur Beschreibung dieser Beziehungen [Mehl05a, S. 81]. Ein Sicherheitsmustersystem reduziert Komplexität durch eine einheitliche Strukturbeschreibung der zugehörigen Muster nach einer einheitlichen Klassifikation und bietet so eine andere Art der Hilfe beim Auffinden von Mustern.

5 Das konzeptionelle Verhältnis von Pattern und Referenzmodellen

Nachdem die beiden Gebiete Referenzmodelle und Security Pattern zunächst jedes für sich dargestellt wurden, sollen in diesem Abschnitt Gemeinsamkeiten und Unterschiede herausgearbeitet werden, um anschließend die Möglichkeiten zu prüfen, wie Security Patterns mit Referenzmodellen kombiniert oder in diese integriert werden können. In diesem Kapitel werden im Schwerpunkt Pattern diskutiert, die Implikationen haben jedoch auch speziell für Security Pattern Geltung.

Referenzmodelle spiegeln allgemein gültige Strukturen (Vorgehensmodelle, Daten, Geschäftslogik, etc.) wider, die für eine Gruppe von Unternehmen oder Anwendungen Gültigkeit besitzen [Beck04, S. 325]. Demgegenüber versteht man unter einem Design oder auch spezieller Security Pattern bewährte Lösungsschemata für ein (Sicherheits-) Problem in einem spezifischen Kontext [Schu03, S. 10].

Vergleicht man beispielsweise diese Definitionen oder andere der in den Kapiteln 3 und 4 genannten Begriffsbestimmungen miteinander, würde man auf den ersten Blick keine große Übereinstimmung der beiden Ansätze vermuten. Das Gegenteil dieser ersten Annahme ist jedoch der Fall, wenn man den Blickwinkel der Katalogisierung von Referenzmodellen einnimmt.

In der bisherigen wissenschaftlichen Diskussion wurde das Verhältnis von Referenzmodellen und Pattern mehrfach aufgearbeitet. Die dabei eingenommenen Positionen können vereinfachend zu zwei Gruppen zusammengefasst werden:

Vertreter der ersten Gruppe betrachten Pattern aus der Perspektive der Sammlung und Katalogisierung von Referenzmodellen. Beide Konzepte sind demnach einer einzigen Klasse oder Kategorie zugehörig, da die Anforderungskriterien zum Zweck der Katalogisierung von Referenzmodellen eingangs bewusst herabgesetzt sind. Auch nach Auffassung dieser ersten Gruppe bestehen im Detail jedoch durchaus Differenzen.

Vertreter der zweiten Gruppe unterscheiden die Konzepte und betrachten eher die Möglichkeiten einer nutzbringenden Kombination. Die Argumentationen und ihre Relevanz für die Ausgangsfragestellung dieser Arbeit sollen nachfolgend umrissen werden.

5.1 Ergebnisse der Klassifikationsansätze

Die Auffassung, dass Pattern nichts anderes sind als ein Form von Referenzmodellen wird in den Ausführungen von Brocke recht gut wiedergegeben. Untersuchungen des Bestands an Referenzmodellen treffen auf das Problem, wie Modelle als Referenzmodelle zu identifizieren sind, ohne dass die zwar akzeptierten, jedoch nicht als Referenzmodell deklarierten Modelle unberücksichtigt bleiben [Broc03, S. 97].

Brocke schlägt daher folgendes vor: „Wird hingegen untersucht, welche Modelle der Bedeutung eines Referenzmodells entsprechen, werden auch Konstruktionsergebnisse gefunden, die sich hinsichtlich ihres Profils zum Teil stark von den üblicherweise als Referenzmodell bezeichneten Modellen unterscheiden“ [Broc03, S. 97]. So sind seines Erachtens „.... etwa sehr implementierungsnahe Artefakte der Softwareentwicklung (z. B. Patterns, Business Objects) ebenso zu Referenzmodellen zu zählen wie eher gering formalisierte Konzepte des Wissensmanagements (z. B. Lessons Learned)“ [Broc03, S. 97].

Als Ergebnis der Sammlungsbemühungen verschiedener Autoren liegt, z.B. in Form von Referenzmodellkatalogen, ein umfangreicher und durchaus heterogener Bestand an Refe-

renzmodellen mit einem entsprechend breiten Spektrum vor, z.B. [Brock03]; [FeLo02a, S.12]; [Fett06]; [Thom06].

In der Literatur finden sich verschiedene Kriterienkataloge zur Unterscheidung der Informationsmodellierung [Brocke 03, S. 30]. Durch den von Rosemann zusammengestellten morphologischen Kasten wird ein systematischer Rahmen zur Verfügung gestellt [Rose96, S. 22 ff]. Der Vorschlag von Rosemann wurde unter Verwendung von Arbeiten von Brocke, Schütte, Schwegmann und Thomas weiter angepaßt [Thom06, S. 90]; [Schü98a, S. 71]; [Broc03, S. 98] [Schw99, S. 9]. Anhand relevanter Unterscheidungsmerkmale können Referenzmodelle in der nachfolgenden Typologie systematisiert werden (vgl. Tabelle 1).

Tabelle 1: Referenzmodellklassifikation

Merkmal	***Merkmalsausprägung***			
Wiederverwendungsart	Nichtgenerisches Referenzmodell		Generisches Referenzmodell	
Erkenntisweg	Induktives Referenzmodell		Deduktives Referenzmodell	
Aussagenstufe	Referenzobjektmodell		Referenzmetamodell	
Aspekt	Aspektspezifisch			Aspektübergeifend
	Eigenschaftsmodell	Verhaltensmodell	Erweitertes Modell	
Formalität	Unformal	Semi-Formal	Formal	
Fachbezug	Fachkonzept	DV-Konzept	Implementierung	
Inhaltliche Individualität	Unternehmens-spezifisches Modell	Referenzmodell	Mastermodell	
Abstraktions-grad	Ausprägungsebene	Typebene	Metaebene	Meta-Metaebene
Konkretisie-rungsgrad	Ausformuliert		Unkonkret	
Modellbreite	Eingegrenzt		Umfassend	
	Legende: Grau hinterlegt - Typische Merkmalsausprägung in Beiträgen zur Referenzmodellierung im State of the Art nach Auffassung von Brocke [Broc03, S. 98]			

In den Veröffentlichungen zur Sammlung von Referenzmodellen finden sich mehrfach Belege für eine Gleichsetzung von Referenzmodellen und Pattern, etwa bei Fettke und

Loos: „Arbeiten der Pattern-Community können durchaus ebenso als Referenzmodelle verstanden werden“ [FeLo02a, S.18]. Fettke und Loos verweisen in diesem Zusammenhang auch auf die Untersuchung von Schwegmann [Schw99, S. 96].

Ein weiteres Beispiel: „Während ein Applikationsmodell ein spezifisches System in einem Unternehmen repräsentiert, steht ein Referenzmodell für eine Klasse vergleichbarer Unternehmenssysteme. Referenzmodelle werden auch Universalmodelle, generische Modelle oder Modell Patterns genannt …“ [Fet+05, S. 1].

Fettke versteht die in der Literatur verwendeten Begriffe Analysemuster, Baustein, Entwurfsmuster, generische Struktur, Modellierungsbaustein, Referenzmodell und wiederverwendbares Modell als (teil-)synonym [Fett01, S. 2]. Sowohl Referenzmodelle als auch Pattern werden von Fettke als Beispiele ein und desselben Modelltyps angesehen. Diesem Modelltyp können Geltungsansprüche wie Allgemeingültigkeit, Flexibilität und Konsistenz zugeschrieben werden [BeSc96, S. 26]. Modelle dieses Typs zeichnen sich im Idealfall auch durch drei weitere Merkmale aus: „Erstens sind die Vor- und Nachteile sowie die Konsequenzen der Verwendung des Modells wohlbekannt. Zweitens ist die Verwendung des Modells bei der Systemgestaltung erprobt und entsprechende Erfahrungen sind dokumentiert. Drittens sind die Konstruktionsprinzpien und Entwurfsentscheidungen des Modells wohlbegründet und explizit formuliert“ [Fett01, S. 2].

Pattern haben einen DV-technischen Fokus, mit Analysemustern gibt es aber auch Arbeiten, die eine fachkonzeptionelle Prägung aufweisen [FeLo02a, S.13]. Von Beedle stammt etwa ein Ansatz zur Unterstützung des Business Process Reengineering, der die Erstellung objektorientierter Unternehmensmodelle auf hohem Abstraktionsniveau umfasst (vgl. [Beed97a], [Beed97b]).

Dies weist auf eine weitere Analogie zwischen Mustern und Referenzmodellen hin: Beide können auf unterschiedlichen Abstraktionsebenen angesiedelt sein, wenn man z.B. Architektur- und einfache Entwurfsmuster heranzieht [Urba04, S. 37].

Betrachtet man die Typologie in Tabelle 1, so ist eine Gleichsetzung der Konzepte von Referenzmodell und Pattern in all ihren Erscheinungsformen wohl unzutreffend. Größere Übereinstimmungen ergeben sich, wenn man bestimmte Referenzmodelle mit einer spezifischen Patternunterart wie bspw. Analysepattern vergleicht. Dann hat die Gleichsetzung auch eine gewisse Berechtigung. Anhand der Typologie lassen sich aber ebenso gut Aspekte benennen, in denen sich die Ansätze unterscheiden.

Während Referenzmodelle eher der Ebene ihrer Anwendungsdomäne zuzuordnen sind, kommen Pattern schwerpunktmäßig im Software Engineering zum Einsatz [Schw99, S. 95]. Die Beschreibungsbene eines Referenzmodells (entsprechend den Phasen des Softwareentwicklungsprozesses oder auch „Informationstechniknähe“) unterscheidet in die Ebene des Fachkonzepts, des DV-Konzepts und der Implementierungsebene [Thom06, S. 91]. Nach Auffassung von Autoren wie Thomas oder auch Schwegmann sind Design Patterns Beispiele für Referenzmodelle auf DV-Konzeptebene [Schw99, S. 93]. Analysis Patterns sind Muster, die auf der Fachkonzeptebene angesiedelt sind [Schw99, S. 94]. Mit Analysis Pattern können vergleichbare Ziele wie mit der Referenzmodellierung verfolgt werden: Auf Fachkonzeptebene werden Modellierungswissen und fachliche Aspekte dokumentiert [Schw99, S. 96].

Eine ähnliche Haltung nimmt Brocke ein: „Zwar ist in der Referenzmodellierung eine Konzentration auf Fragen fachkonzeptioneller Modellierung zu beobachten, doch bieten z.B. auch Konstruktionsergebnisse implementierungsnäherer Ebenen der Systementwicklung Potenziale zur Wiederverwendung. Nicht zuletzt die erfolgreiche Nutzung der in der Softwareentwicklung als Patterns bezeichneten Referenzmodelle zeigt dies“ [Broc04, S. 390].

In der Beschreibung und Dokumentation der beiden Konzepte finden sich hinsichtlich der Modellierungsnotationen Übereinstimmungen, wenn man an die strukturierte Textform der Musterschemata denkt, jedoch auch ebenso Abweichungen. Viele Entwurfsmuster sind in strukturiertem Prosatext verfasst, wohingegen Referenzmodelle häufig semi-formal modelliert werden.

Becker et al. nehmen ebenfalls eine Gleichsetzung von Referenzmodellen und Entwurfsmustern vor, unterscheiden jedoch auch im Detail. Entwurfsmuster sind ihres Erachtens Referenzmodelle, die im Vergleich zu anderen Formen von Referenzmodellen einer besonders geringen Beschränkung der Modellanpassung unterworfen sind. Die Anwendung des Referenzmodelles stützt sich allein auf Analogieschlüsse durch den Anwender [Bec+06, S. 5]

Beide Konzepte unterscheiden sich in ihrem Scope. Während Referenzmodelle teilweise sehr umfassend sind und große Bereiche einer Domäne integrieren, bilden Analysis-Patterns eher eng umgrenzte Sachverhalt ab. Umfangreichere Modellierungsgegenstände wären demnach durch eine Analysemustersprache oder ein Analysemustersystem abzubilden. Ein weiterer Unterschied besteht in der Abbildung von Varianten. [Schw99, S. 96].

Dem Entspricht in der Tabelle 1 das Kriterium der Modellbreite, mit dem der Umfang der durch das Modell abgebildete Bereiche einer Domäne bestimmt wird. Bezüglich der Modellbreite kann zwischen projekt-, bereichs und organisationsweiten Objektmodellen unterschieden werden [Schm95; Maie96:321]).

Im Gegensatz zur Interpretation anderer Autoren, z.B. Ferstl et al., vertritt Schwegmann die Auffassung, dass Design Pattern primär Strukturaspekte abbilden und nur in geringem Umfang Verhalten [Schw99, S. 96]. Gegen die Interpretation von Schwegman bzgl. Verhalten sprechen die Aussagen von Ferstl et al. und Delessy, Fernandez et al., die teilweise inhaltlich bzw. unter Hinweis auf die Verwendung von UML-Verhaltensmodellen argumentieren [Fer+98]; [Del+07, S. 32].

5.2 Ergebnisse der Kombinationsansätze

Die im vorigen Abschnitt vorgestellten Aussagen zum Verhältnis von Pattern und Referenzmodellen stammen aus Arbeiten, die mit der Zielsetzung der Referenzmodellsammlung erstellt wurden. Diese einfache Kategorisierung von Pattern und Referenzmodellen bleibt unter konzeptionellen Gesichtpunkten jedoch unbefriedigend und wenig fruchtbar. Interessanter ist hingegen die Annahme, dass Referenzmodelle und Pattern sich von ihrem Ansatz her zwar unterscheiden, jedoch sinnvoll miteinander kombiniert werden können.

Im Rahmen der Forschung zur Referenzmodellierung existieren Überlegungen, wie Referenzmodelle weiterentickelt werden können und wie eine größere Implementierungsnähe erreicht werden kann. Hierzu schlägt Brocke in Abgrenzung zu monolithischen Referenzmodellen und in Analogie zur komponentenorientierten Anwendungssystementwicklung die Entwicklung modularer Referenzmodelle vor [Broc03, S. 351]. In diesem Typ von Referenzmodell können abgeschlossene Teilbereiche eigenständig verwendet werden. Brocke bezeichnet sie als Referenzmodellkomponenten. Dabei stellt das bei Softwarekomponenten unterstellte „Plug-and-Play-Prinzip“ in Aussicht, Referenzmodelle zu konstruieren, indem vorgefertigte Referenzmodellkomponenten bedarfsspezifisch miteinander kombiniert werden [Broc03, S. 210].

Zur Gewinnung von Referenzmodellkomponenten ist methodisch aus Sicht von Brocke auch eine induktive Vorgehensweise zulässig. Hierzu könnten existierende Anwendungssysteme daraufhin untersucht werden, ob sich wieder verwendbare Teileinheiten isolieren und zu Referenzmodellierungskomponenten aufbereiten lassen.

Da aus Sicht von Brocke ohnehin kein objektives Abstraktionsniveau für Referenzmodelle angegeben werden kann, muß bei dieser Vorgehensweise seines Erachtens nicht zwingend eine Abstraktion der identifizierten Einheiten vorgenommen werden [Broc03, S. 351].

In Anlehnung an den Aufbau komponentenorientierter Anwendungssysteme hält Brocke die Verwendung schichtenorientierter Ordnungsrahmen für sinnvoll. Außerdem hält er es für naheliegend, zu untersuchen, inwiefern die Modellstruktur der Referenzmodellkomponenten einen Beitrag zur Integration von Referenzmodellen gegenüber implementierungsnäheren Ebenen leisten kann [Broc03, S. 351]. Eine Ausarbeitung dieses Ansatzes im Sinne der Fragestellung dieser Arbeit zur Verknüpfung mit dem Patterngedanken steht jedoch noch aus.

Wie sich Pattern darüber hinaus sinnvoll in Referenzmodelle integrieren lassen, wird in einer Überblicksarbeit von Fettke und Loos deutlich, in der sie verschiedene Methoden zur Wiederverwendung von Referenzmodellen einander gegenüberstellen [FeLo02b, S. 24]. Dabei unterscheiden sie monolithische, bausteinbasierte und universelle Methoden.

Als Vertreter der bausteinbasierten Methoden diskutieren sie die Arbeit von Wolf, der einen mustersprachenbasierten Ansatz wählt, um die Wiederverwendung von Modellen zu unterstützen [Wolf01, S. 1]. Ausgehend von einem initialen Modell werden durch sukzessive Anwendungen der Muster dieser Sprache entsprechende unternehmensspezifische Modelle konstruiert [FeLo02b, S. 25].

Zur Arbeit von Schulze merken Fettke und Loos an: „In Schulzes Ansatz nehmen *Referenzmodelle in Form von Entwurfsmustern* (Pattern) eine herausragende Rolle ein“ ([FeLo 02b, S. 24] verweisen auf [Schu01, S. 1f.], im Original ohne Hervorhebungen). Der Ansatz von Schulze wird trotz der Verwendung von Patterns von Fettke und Loos als Vertreter der Klasse Wissensbasierte Methoden angesehen. In dieser Klasse kommen Methoden und Konzepte der Wissensverarbeitung zur Anwendung kommen und werden in Bezug auf die Wiederverwendung von Referenzmodellen offensichtlich höher gewichtet als Pattern [Fe Lo02b, S. 29].

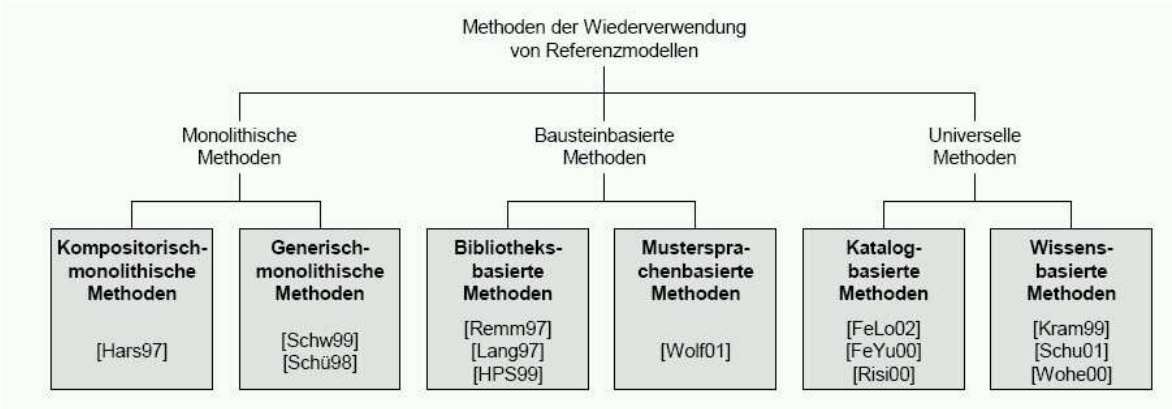

Abbildung 4: Taxonomie von Wiederverwendungsmethoden von Referenzmodellen [FeLo02b, S. 28]

Sowohl die Arbeit von Wolf als auch die von Schulze stützen sich auf Erkenntnisse des Projekts WEGA, in dem von Ferstl et al. wiederverwendbare und erweiterbare Geschäftsprozeß- und Anwendungssystem-Architekturen erarbeitet wurden. Kontext in den Arbeiten von Ferstl et al. ist die Zielsetzung, die Wiederverwendung von fachlichen Modellinhalten zu unterstützen. Einer der Leitgedanken ist, dass in den Geschäftsprozessmodellen korrespondierende Patternsysteme betriebswirtschaftliches Wissen formalisieren und dabei als Grundlage der methodischen Wiederverwendung von Entwurfswissen dienen [Fer+98, S. 11]; [Schl03, S. 142].

Nach Schlitt kann das Konzept der Pattern ganz unterschiedlich verstanden werden [Schl 03, S. 153]: Als Interpretationsmöglichkeiten nennt er Strukurmuster, Entwurfsmuster und generische Entwurfsmuster, die sich auf den Dimensionen “Grad der Kontextberücksichtigung und „Unterstützung des Entwurfsprozesses“ unterscheiden [Schl03, S. 153].

Ein Strukturmuster beschreibt lediglich in ergebnisorientierter Weise Systemzustände, ohne eine Aussage darüber zu treffen, WIE ein Ziel mit einem Lösungsverfahren zu erreichen ist. Durch die Beschränkung auf die Dokumentation fertiger Lösungen und das Fehlen prozeduraler Handlungsempfehlungen erfährt der Entwurfsprozeß nur eine geringe Unterstützung. Strukturmuster schenken dem Entwurfsprozess keine Beachtung, ihr Beispielcharakter dient nur der Orientierung in einem komplexen Entwurf [Fer+98, S. 12].

Bei Entwurfsmustern ist die Gewichtung anders, hier steht die prozedurale Verfahrensbeschreibung zur Herstellung von Lösungen im Vordergrund, wodurch der Entwurfsprozeß deutlich stärker unterstützt wird [Schl03, S. 156]. Mit Hilfe von Entwurfsmustern wird eine spezifische Entwurfsentscheidung für ein gegebenes Problem dokumentiert. Das Entwurfsmuster liefert dabei eine Konstruktionsanweisung für die Problemlösung. Neben Entwurfszielen und dem Objekt stellt das Muster ein Entwurfsverfahren zur Lösung des Problems bereit [Fer+98, S. 13]

Ist das Entwurfsmuster generisch, dann enthält es auch noch eine kontextsensitive Beschreibung des Entwurfsobjekts. Das generische Entwurfsverfahren kann als eine abstrakte, anpassbare Prozessbeschreibung zur Lösung eines Problems aufgefasst werden. Generische Entwurfsmuster zeichnen sich durch eine weitere Differenzierung des Kontextes, z.B. durch Aufzeigen von Constraints aus, die im parametrisierbaren Entwurfsverfahren berücksichtigt werden können [Schl03, S. 156].

Pattern sind damit nichts anderes als gekapselte Entwurfsverfahren und wesentliche Designentscheidungen [Fer+98, S. 15 - 17]. Sie sind oft informale Beschreibungen mit hoher Ausdrucksmächtigkeit, durch ihre nicht-formale Spezifizierung sind sie jedoch auch mehrdeutig [Schl03, S. 151]. Schlitt stellt dabei in Frage, ob eine Formalisierung zum besseren Verständnis des Konzepts beitragen würde [Schl03, S. 153]. Nach Ansicht anderer Autoren wie z. B. Delessy, Fernandez et al. ist das jedoch mit der Notation UML bei Pattern durchaus der Fall [Del+07, S. 32].

Schlitt plädiert für eine Interpretation des Patternbegriffs als generisches Entwurfsmuster, das über hohe Ausprägung auf den Dimensionen „Kontextberücksichtigung“ und „Unterstützung des Entwurfsprozesses“ verfügt [Schl03, S. 157]. Für Schlitt stellen Pattern in der Interpretation als generische Entwurfsmuster gekapseltes, vorgehensorientiertes Konstruktionswissen dar [Schl03, S. 159].

Innerhalb eines komplexen (Meta-)Modells können Pattern als Lösungen für Teil-Probleme angesehen werden. Dabei bestehen zwischen den Pattern Interdependenzen aufgrund von Beziehungen der Ziele [Schl03, S. 196].

Entwurfsmuster stellen Lösungsverfahren für Teilprobleme der Systemkonstruktion dar und weisen auch Beziehungen zueinander auf [Wolf01, S. 177]. Diese Beziehungen werden in einem Patternsystem abgebildet, das bspw. die Reihenfolge von sequentiellen Operationen im Entwurfsprozeß vorgibt [Schl03, S. 147].

Der Patternbegriff dient für Schlitt als Metapher für ein wiederverwendungsorientiertes Verfahren der Modellkonstruktion. Dieses Verfahren kann gleichzeitig als generisches Referenzmodell verstanden werden. Nach Sinz kann ein Referenzmodell dann als generisch bezeichnet werden, wenn ein daraus abgeleitetes konkretes Modell auf das Referenzmodell zurückgeführt werden kann [Sinz97].

Ein (generisches) Refererenzmodell für die Geschäftsprozessmodellierung einer (Anwendungs-) Domäne besteht aus einem Initialmodell und einem Patternsystem [Fer+98, S. 14]. Das Initialmodell repräsentiert das Entwurfsobjekt und den Kontext [Fer+98, S. 15]

Der Nutzer eines nicht-generischen Referenzmodells erfährt bei der Konstruktion eines konkreten Modells nur eine Unterstützung in dem Maß, das die gewählte Modellierungssprache bietet. Nichtgenerische Referenzmodelle sind ausschließlich ergebnisorientiert angelegt und bieten in der Regel keine Ansatzpunkte für spezifische Veränderungen und Weiterentwicklungen [Schl03, S. 198]. Außerdem ist eine Dokumentation der Ziele, die dem nicht-generischen Referenzmodell zugrundeliegen, in der Regel nicht vorgesehen.

Generische Referenzmodelle bieten durch ihre Vorgehensorientierung ein höheres Maß an Unterstützung für den Modellkonstrukteur [Schl03, S. 198]. Die von Schlitt vorgeschlagene Form eines generischen Konstruktionsrahmens enthält neben einem ergebnisorientierten Anteil ein parametrisierbares Entwurfsverfahren, das die Form eines Pattern-Systems aufweist.

In der Arbeit von Schulze ist der Ausgangspunkt für ein generisches Modell ein initiales Modellschema (auch als Initialmodell bezeichnet), das das Modellobjekt und den Kontext beschreibt und ein zugehöriges Patternsystem als generisches Entwurfsverfahren enthält [Schu01, S. 49].

Im Ansatz von Wolf ist das Modellsystem auf jeder Modellebene in mehrere Bestandteile strukturiert. Ein Bestandteil kann eine Menge von Pattern umfassen, die im Sinne einer Mikroarchitektur sinnvolle Lösungen beschreiben [Wolf01, S. 182]. Das Referenzmodell hält neben einer Ergebnissicht auch konstruktives Wissen in Form von Elementen zur kontextsensitiven Problem- und Lösungsbeschreibung bereit [Wolf01, S. 183].

Auf der Ebene der Teilprobleme leisten Patternsysteme Strukturierungshilfe, indem sie mögliche Problemlösungskombinationen und ihre Beziehungen angeben. Teillösungen werden durch Auswahl und Kombination einzelner Patterns entlang der Beziehungen erzeugt [Wolf01, S. 183]. Wolf demonstriert seinen Ansatz anhand eines generischen Referenzmodells für Leasingunternehmen, das Pattern als seine Bestandteile aufweist [Wolf01, S. 221].

Auch im Bereich konkreter, betrieblicher Modelle finden sich in der Literatur Ansätze, Pattern als Bestandteile umfassender Modelle zu verwenden. Um Problemen im Rahmen der Erstellung von Informationsmodellen für das Enterprise Architecture Management zu begegnen, schlagen Buckl et al. einen auf Pattern basierenden Ansatz vor [Buc+07, S.152]. Die Autoren sehen die von ihnen entwickelten Enterprise Architecture Management Patterns als Bausteine für organisationsspezifische Informationsmodelle an (im Original „conceptual information models“).

Im Anwendungsfall werden nach einem Auswahlprozess geeignete Pattern in einem kohärenten organisationsspezifischen Informationsmodell integriert [Buc+07, S. 152]. Anzumerken ist, dass der von Buckl et al. teilweise verwendete Begriff des Conceptual Models in anderen Publikationen mißverständlicherweise als Synonym für Referenzmodelle verwendet wird, siehe etwa [Fet+05, S. 1; PeAa05, S. 30].

In der Lesart von Referenzmodellen der Universität Münster ist ebenfalls Platz für eine Integration von Pattern. In ihrem Vorgehensmodell zur Erstellung von Referenzmodellen verweisen Becker et al. auf das Hilfsmittel eines Ordnungsrahmens [Bec+02, S. 47]. Dieser Ansatz hat sich bei der Untergliederung komplexer Referenzmodelle bewährt. Im Verfeinerungsmodellaspekt des Vorgehensmodells werden Elemente im Ordnungsrahmen angeordnet. Über jedes Teilelement kann ein Modellierungsbereich mit seiner speziellen Semantik in einem höheren Detaillierungsgrad abgebildet werden [Bec+02, S. 48]. Die Verfeinerung kann über mehrere Stufen erfolgen, wobei die Modellierungstechnik nicht auf allen Abstraktionsebenen gleich sein muss. Pattern könnten unter Hinzuziehung der bereits vorgestellten Arbeiten von Ferstl et al. für die Detaillierung eingesetzt werden [Fer+98].

5.3 Schlussfolgerungen für die Ausgangsfragestellung

Es gibt verschiedene Deutungen der Konzepte Pattern und Referenzmodell und ihrer Beziehung zueinander. Die in den beiden vorigen Abschnitten vorgestellten Positionen haben für ihre jeweiligen Zwecke auch eine gewisse Berechtigung. Für die Zielsetzung dieser Arbeit liegt es jedoch näher, dem Kombinationsansatz zu folgen. Die Gründe dafür sollen nachfolgend erläutert werden.

Die großzügige Auslegung der Definition von Referenzmodellen, wie bspw. Fettke und Loos sie vorschlagen, verbreitert auf den ersten Blick die Fallbasis für Referenzmodellbeispiele. Entitäten auf einem Abstraktionslevel wie der Unternehmensebene sind dann konzeptionell auch durchaus vergleichbar. Je konkreter der Scope jedoch wird, desto mehr macht es Sinn, die Konzepte zu unterscheiden. Hier stellen Pattern eher Problemlösungsanweisungen dar, mit denen durch Referenzmodelle vorgegebene Teilziele erreicht werden können.

Folgt man dem Interpretationsvorschlag von Ferstl, Wolf et al. gewinnt man nicht nur eine klarere Abgrenzung der Konzepte sondern auch eine fruchtbare Quelle für weitere Fragestellungen. Die Auffassung, wonach Referenzmodelle als umfassendere Konzepte verstan-

den werden können, die Patternsysteme als Elemente enthalten können, ist konzeptionell interessanter.

Die Betonung des Verfahrenscharakters von Pattern hat auch innerhalb der Patterncommunity Relevanz. Heyman et al. stellen bei einer Untersuchung der Qualität einer großen Zahl von Security Pattern fest, dass viele Pattern oft nur Zielzustände beschreiben, aber keine Konstruktionshinweise liefern [Hey+07, S. 3]. Eben diese „verkürzten" Pattern schnitten bei ihrer Bewertung entsprechend schlechter ab. Empirisch ließe sich die Aussage von Ferstl et al. damit zwar nur zum Teil belegen, aber sie beschreibt doch einen sinnvollen und wünschenswerten Qualitätsanspruch an (Security) Pattern.

Die Ferstl-Patterninterpretation entspricht außerdem einer Forderung von Delessy, Fernandez et al, wonach der Erstellungsprozess für sichere Software auf groben wie auf detaillierteren Betrachtungsebenen ingenieurmäßig, d. h. strukturiert und systematisch ablaufen soll.

Schließlich spielt auch eine Rolle, dass sowohl in der Forschung zu Pattern als auch in der zu Referenzmodellen Ansätze existieren, komplexere Einheiten durch Teilelemente hierarchisch zu gliedern. Diese Idee wurde bereits durch Alexander selbst formuliert: „Each pattern then, depends both on the smaller patterns it contains, and on the larger patterns within which it is contained" [Alex79, S.312], siehe auch [Schl03, S. 144].

Auch bei Fettke und Loos finden sich bei ihrer Darstellung mustersprachenbasierter Methoden entsprechende Hinweise [FeLo02b]: „Die Wiederauffindung bei diesen bausteinbasierten Methoden wird dadurch gelöst, das zwischen den einzelnen Modellierungsbausteinen zahlreiche Beziehungen definiert werden. Im Idealfall sind hierbei die verschiedenen Referenzmodelle soweit miteinander verwoben, dass im Kontext einer konkreten Modellierungssituation leicht ersichtlich wird, welche Referenzmodelle im nächsten Modellierungsschritt zur Anwendung kommen können" [FeLo02b, S. 29].

Becker et al. schlagen vor, dass Referenzmodelle aus Detaillierungselementen zusammengesetzt sein können [Bec+02, S. 48]. Diese Elemente können wiederum durch Teil-Referenzmodelle oder eben Patternsysteme sein. Das methodische Hilfsmittel eines Ordnungsrahmens kann dabei übrigens ähnliche Strukturierungsaufgaben wie ein initiales Modellschema übernehmen.

Dass die Anordnung in der Hierarchie ein geeignetes Unterscheidungsmerkmal für (Referenz-)Modelle und Pattern sein kann, zeigt sich schließlich auch im Ansatz von Buckl et al. [Buc+07]. Pattern kapseln vorgehensorientiertes Konstruktionswissen für eingegrenzte

Probleme und eignen sich gut als Bausteine für Referenzmodelle, um so umfassendere Informations- und Referenzmodelle zu erzeugen.

6 Entwicklung eines patternbasierten Referenzmodells für Identity Management

Nachdem im vorigen Kapitel verschiedene Standpunkte zum Verhältnis von (Security) Pattern und Referenzmodellen dargelegt und diskutiert wurden, soll nun auf Basis der Ergebnisse ein patternbasiertes Referenzmodell entworfen werden.

Zur Entwicklung eines Referenzmodells bedarf es einer Methodik, die ein systematisches Vorgehen im Rahmen des Modellierungsprozesses sicherstellt [Bec+02, S. 34]. Die sachlogische Abfolge der Aufgaben beschreibt das Vorgehensmodell der Methodik [Bec+02, S. 34]. Die Literatur zur Referenzmodellierung stellt inzwischen mehrere Ansätze zur Erstellung bereit, siehe z.B. [Bec+02, S. 34ff]. Aus den zur Verfügung stehenden Ansätzen wird die Vorgehensweise zur Erstellung eines Referenzmodells nach Becker et al. ausgewählt, da sie an einem Lehrstuhl entwickelt wurde, der mittlerweile eine langjährige Tradition in Bezug auf Referenzmodellen vorweisen kann. Die Vorgehensweise wird zunächst kurz vorgestellt und anschließend schrittweise umgesetzt.

In der ersten Phase des Vorgehensmodells nach Becker et al. werden zu Beginn die Ziele des Projekts definiert. Hierzu werden eingangs der Konstruktionsauftrag sowie der Problembereich eingegrenzt und die zu betrachtenden Funktionsbereiche festgelegt. Anschließend werden die Anforderungen analysiert und entschieden, welche Zwecke mit dem Modell verfolgt werden sollen. In dieser Phase wird auch eine Aussage getroffen, für welche Unternehmen mit welchen Merkmalen das Referenzmodell vorgesehen ist. Zu klären ist darüber hinaus, welche Perspektiven unterstützt werden sollen. Aus den Perspektiven werden die passenden Modellierungstechniken abgeleitet [Bec+02, S. 38-40].

In der sich anschließenden zweiten Phase wird die Referenzmodellierungstechnik, ggf. mehrere methodisch und inhaltlich definiert. Die Methode wird durch Auswahl einer Modellierungssprache und einer Vorgehensweise bestimmt [Bec+02, S. 46ff]. Die Modelltypen sind dem Projektziel entsprechend auszuwählen bzw. zu entwickeln. Inhaltlich kann in der zweiten Phase eine Ausarbeitung der Konventionen für wichtige Komponenten der Phase erfolgen. An dieser Stelle im Ablauf wird bestimmt, welche Anordnungsvorschriften für den Modell- bzw. Ordnungsrahmen des Referenzmodells gelten sollen. Modellrahmen geben der Ausgangsdomäne eine grobe Struktur [Meis01, S. 61-64].

Den Elementen im Ordnungsrahmen werden Verfeinerungmodelle zugeordnet, die deren Semantik in einem höheren Detaillierungsgrad zeigen. Die Verfeinerung kann über mehrere Stufen erfolgen [Bec+02, S. 48]. Die Modellierungstechnik muß dabei nicht auf allen Abstraktionsebenen gleich sein.

Schließlich wird in der dritten Phase das eigentliche Referenzmodell unter Verwendung der individuellen Referenzmodellierungstechnik schrittweise konstruiert [Bec+02, S. 36]. Hierzu werden Quellen und Gestaltungsempfehlungen der Literatur ausgewertet und spezifische Informationsmodelle verallgemeinert [Bec+02, S. 49-52].

Nach der Festlegung des Empfehlungscharakters werden der Ordnungsrahmen und anschließend die Verfeinerungsmodelle erstellt. Enthält des Referenzmodell Varianten, sind Konfigurationsregeln zu entwickeln. Die Detaillierung der Funktionsbereiche gemäß den Schritten des Vorgehensmodells von Becker et al. wird schließlich im nachfolgenden Kapitel 7 weiter ausgeführt.

6.1 Definition der Projektziele

Die erste Phase des Vorgehensmodells dient dazu, ein erstes grobes Modell des Problembereichs zu erstellen, für den ein Referenzmodell entwickelt werden soll [Bec+02, S. 37]. Es sind diejenigen Funktionsbereiche zu definieren, die Gegenstand des Referenzmodellierungsprojektes sein sollen. Als Modellanwendungszwecke sind bspw. die Gestaltung von Anwendungssystemen oder Organisationsgestaltung denkbar. Der Zweck der Anwendungssystemgestaltung lässt sich etwa in die Auswahl von Standard-Software, Workflowmanagement und Softwareentwicklung verfeinern [Bec+02, S. 39].

Schließlich sind auch potenziell zu adressierende Modellnutzer zu identifizieren. Die Modellnutzer und ihre unterschiedlichen Subjektivierungen werden über Perspektiven repräsentiert. Es sollte ebenfalls geklärt werden, für welche Klasse von Unternehmen das geplante Referenzmodell geeignet ist.

Neben Sprachanforderungen müssen bei der Anforderungsanalyse weitere Aspekte berücksichtig werden, z.B. in welchem Umfang Wissen über die abgegrenzten Funktionsbereiche vorhanden ist und in welcher Form es vorliegt [Bec+02, S. 41]. Hierzu sollte in einer Marktanalyse untersucht werden, welche Referenzmodelle bereits vorhanden sind und inwieweit diese als Wissensinput dienen können.

Zielsetzung dieser Arbeit ist die Entwicklung eines Referenzmodell für Identity Management, dass geeignete Security Pattern für die Domäne integriert und ggf. Abhängigkeiten aufzeigt.

Das Referenzmodell für Identity Management bzw. Patternsystem kann bspw. als Ausgangspunkt oder als Hilfsmittel bei Design- oder Auswahlprozessen von Enterprise Identity Management Systemen oder bei Sicherheitsanalysen genutzt werden. Zumindest der Ordnungsrahmen kann auch die Einführung und den Betrieb von Identity Management Systemen in Organisationen als erste Orientierung in der Thematik unterstützen.

Für die Hersteller von Identity Management Standardsoftware könnte eine Nutzung zur Überprüfung des Sicherheitsstandards von Produkten anhand bewährter „common practice“ nützlich sein. Anders als andere Referenzmodelle dient das zu entwickelnde Referenzmodell weniger für die Ableitung unternehmensspezifischer Modelle, da davon auszugehen ist, dass der Anteil von Individualentwicklung in diesem Gebiet zumindest auf lange Sicht gering sein wird.

Angesichts des überschaubaren Adressatenkreises dieses Referenzmodells soll auch der wissenschaftliche Verwendungszweck angesprochen werden, der die Klärung des konzeptionellen Verhältnisses von Referenzmodellen und Security Pattern, sowie eine Bestandsaufnahme und Systematisierung zum Gegenstand hat.

Es besteht keine Einschränkung hinsichtlich der Unternehmensfunktion oder Branche, höchstens die Größe wird einen begrenzenden Faktor darstellen. Das Modell besitzt Relevanz für Unternehmen aus dem Mittelstand und aufwärts.

Aus den unterstützten Perspektiven Softwareanforderungsanalyse und Modellierung auf Fachkonzeptebene lässt sich hinsichtlich der Modellierungstechniken die Anforderungen ableiten, dass dem Referenzmodell detaillierte Informationen über Klassen, Funktionsbereiche und den grundlegenden Abhängigkeiten zwischen Modellelementen entnehmbar sein sollten.

6.2 Verwandte und ähnlich gelagerte Arbeiten

In diesem Abschnitt werden ähnlich gelagerte Arbeiten zu den Themengebieten Identity Management, Referenzmodelle und Security Pattern in ihren Grundzügen vorgestellt. In Bezug auf Referenzmodelle wird dabei weniger dem strengen, nutzungsorientierte Referenzmodellbegriff nach Thomas gefolgt [Thom05, S. 23], sondern dem etwas weiter gefassten Verständnis des Begriffs nach Fettke et al. [Fet+05, S. 1]. Dadurch können auch

Referenzmodelle berücksichtigt werden, für die sich keine Belege einer Anwendung finden lassen.

Die Ergebnisse der Literaturrecherche werden, soweit sinnvoll und passend, dargestellt und hinsichtlich ihrer Ergebnisse bewertet. Auf der Bewertung der existierenden Referenzmodelle und Security Pattern für Identity Management aufbauend, wird dann nach den eingeführten Methoden ein eigener Ansatz entwickelt.

6.2.1 Identity Management Referenzmodell „hEAM“ nach Rottleb

Auch wenn man Identity Management erst seit wenigen Jahren als integriertes Themengebiet ansieht, wurden bereits Referenzmodelle entwickelt, die unterschiedliche Aspekte der Domäne betrachten. Hier ist zum einen das Referenzmodell des homogenen Enterprise Access Managements nach Rottleb zu nennen, zum anderen das kommerzielle Emprise Identity Management Referenzmodell [Rott03, S. 1]; [Empr06, S. 1]. Das Referenzmodell von Emprise legt seinen Fokus auf die administrativen Prozesse des Identity Managements, aufgrund des kommerziellen Hintergrunds ist es jedoch nicht öffentlich verfügbar.

Nachfolgend soll kurz Rottlebs Referenzmodell für Identity Management vorgestellt werden [Rott03]. Zur Beschreibung des Modells wird ein Bezugsrahmen für Referenzmodelle von Fettke und Loos verwendet, der sich in die Aspekte Anwendungsdomäne, eingesetzte Modellierungssprachen, Modellgröße, bekannte Evaluationen und Anwendungen gliedert [Fet+05, S. 1].

Den Kontext des Referenzmodells des homogenen Access Managements bilden nach Rottleb jüngere Entwicklungen wie das Supply Chain Management SCM oder Customer Relationship Management CRM [Rott03, S. I Deckblatt]. Diese und andere Veränderungen in der Umwelt von Unternehmen führen dazu, dass neben internen Mitarbeitern verstärkt auch externe Benutzer Zugriff auf Anwendungssysteme eines Unternehmens benötigen. Die hieraus abgeleiteten Anforderungen bezeichnet Rottleb als das Paradigma des homogenen Enterprise Access Managements mit der Abkürzung „heam“.

Umgesetzt werden diese Anforderungen durch ein Referenzmodell zur anwendungssystemübergreifenden konsistenten Zugriffssteuerung (MAKS), mit dem die Realisierung eines zentralen Rollen- und Rechtemanagementsystems (ZR sMS) auf Basis einer Referenzarchitektur unterstützt werden soll.

Das homogene Enterprise Access Management (hEAM) sieht vor, dass in- und externen Aufgabenträgern ein Account und eine Rolle zugewiesen werden [Rott03, S. 50]. Den Rol-

len sind wiederum Rechte für den Ressourcenzugang zugeordnet. Homogen bedeutet in diesem Zusammenhang, dass bei der Konfiguration der rollenbasierten Zugriffskontrolle und Zuordnung von Aufgabenträgern und Rollen über verschiedene Systeme hinweg Konsistenz gewährleistet ist [Rott03, S. 50].

Die fachlichen Anforderungen des homogenen Enterprise Access Managements „heams“ decken u.a. die anwendungssystemübergreifende Modellierung von Rollen und die Handhabung dynamischer Aspekte wie Veränderungen von Rechten, die Löschung oder Sperrung von Benutzerkonten ab.

Im Referenzmodell werden strukturelle Aspekte durch Organisationsmodellierung abgebildet [Rott03, S. 57]. Dynamische Modellbestandteile finden sich im Zugriffssteuerungsmodell, das u.a. die Administrationsprozesse inkl. der Regeln der Konsistenzerhaltung umfaßt. Eine Verknüpfung der strukturellen und der dynamischen Sicht findet sich im Referenzmodell zur anwendungssystemübergreifenden konsistenten Zugriffssteuerung [Rott03, S. 57].

Als wesentlichen Modellierungsobjekte des Organisationsmetamodells sind Rollen, Aufbauorganisation, Stellenbildung, funktionale Gestaltung, Bereichsabgrenzung, Aufgabenträger, Kerngeschäftsprozesse, Geschäftsobjekt, Methode (Verfahren), Vertretung von Aufgabenträgern, Berechtigungsmetamodell sowie Handlungsbeschränkungen und –vollmachten zu nennen [Rott03, S. 63 - 65].

Bei der Auswahl der Modellierungstechnik berücksichtigt Rottleb sowohl strukturelle Aspekte wie die Aufbauorganisation, die Rollenkonfiguration oder die Rollenzuordnung [Rott03, S. 13], aber auch dynamische Aspekte wie die Änderung von Rollenkonfigurationen und Rollenzuordnungen von Organisationen. Das Referenzmodell von Rottleb umfasst insgesamt fünfzehn UML Klassendiagramme und vier UML Aktivitätsdiagramme, die insgesamt zwei Sichten auf das Modellsystem erlauben. Die Aktivitätsdiagramme bestehen aus bis zu acht Prozessschritten.

Eine Evaluation oder Anwendung des Modells, das einem Promotionsvorhaben entstammt, ist zum gegenwärtigen Zeitpunkt (Frühjahr 2007) nicht bekannt. Eine Internet-Recherche nach Veröffentlichungen mit Bezug auf Rottlebs Modell blieb ohne Ergebnisse.

Aus wissenschaftlicher Sicht besteht ein Defizit der Arbeit Rottlebs in der Tatsache, dass das zugrundeliegende Verständnis des Begriffs Referenzmodell nicht expliziert wird [Rott03, S. 1]. Eine Auseinandersetzung mit der umfangreichen Literatur zu Referenzmodellen findet nicht statt; der verwendete Begriff des Referenzmodells wird nicht in die For-

schungslandschaft eingebettet, so wie von Mertens und Holzner empfohlen [MeHo92, S. 21]. Insofern erfolgt bei Rottleb auch keine Strukturierung des Modells im Sinne einer Informationsarchitektur oder der Einsatz von Hilfsmitteln wie einem Ordnungsrahmen. Die fehlende Bezugnahme auf bewährte und akzeptierte Qualitätskriterien für Referenzmodelle erschwert zusätzlich die Bewertung. Positiv ist allerdings, dass das Referenzmodell im Gegensatz zum Modell von Emprise frei, weil online, verfügbar ist.

Das Referenzmodell von Rottleb legt einen Schwerpunkt auf die statischen bzw. eingeschränkt dynamischen Aspekte des Identity Managements wie bspw. die Aufgabenträger, die Aufbauorganisation und die funktionalen Rollen. Dynamische Aspekte, wie etwa Rollenkonstrukte, finden ebenfalls Berücksichtigung, administrative Prozesse, wie Genehmigungs- oder Provisioning-Prozesse stehen jedoch nicht im Fokus. Sie werden zwar betrachtet, jedoch nicht im Detail und lediglich in Textform umschrieben [Rott03, S. 105].

Für die Anwendung des Referenzmodells entwickelt Rottleb kein eigenes Vorgehensmodell, sondern verweist auf ein bereits existierendes Vorgehensmodell von Greiffenberg und Esswein [GrEs01] [Rott03, S. 124].

Ausgangsfragestellung für Rottleb ist zwar die Absicherung von Informationssystemen, trotzdem betont er bereits in der Einleitung der Arbeit: „Technische Schutzziele, Sicherheitstechniken und -verfahren etc. ... liegen hingegen außerhalb des Fokus dieser Arbeit“ [Rott03, S. 1]. Insoweit spielen auch Security Pattern in Rottlebs Ansatz keine Rolle, die größte Nähe erreicht er noch durch eine kurze Betrachtung von Softwarekomponenten.

Aufgrund der beschriebenen Einschränkungen besitzt Rottlebs Referenzmodell nicht für alle Aspekte der Fragestellung dieser Arbeit Relevanz. Für die Funktionen des Identity Managements leistet es jedoch einen Beitrag in Form der Abgrenzung maßgeblicher Teilbereiche.

6.2.2 Security Pattern für Identity Management

Wie bereits erwähnt spielen die Themen Identifikation, Authentisierung und Autorisierung rein quantitativ eine wichtige Rolle bei veröffentlichten Security Pattern [Hey+07, S. 6]. Dies spiegelt sich auch in den nachfolgenden Veröffentlichungen wieder, die für die Detaillierung der einzelnen Funktionsbereiche herangezogen werden sollen.

Eine Gruppe um Steel et al. bei Sun Microsystems hat ein Buch veröffentlicht, das Security Patterns für J2EE basierte Applikationen, Web Services und Identity Management auf Ar-

chitekturebene abdeckt [Ste+05]. Das Buch bietet eine umfassende Sicht auf Java Applikationssicherheit und führt 23 Security Pattern auf [HaJo06, S. 4].

Markus Schumacher leitet eine Arbeitsgruppe für Security Pattern, die eine Webseite betreibt und neben anderen ein umfassendes Buch zum Thema publiziert hat. Diese Veröffentlichung umfasst 46 Security Patterns aus der Domäne des Enterprise Security und Risk Management, Identifikation und Authentisierung, Zugriffskontrolle, Accounting und weitere Themenbereiche wie Firewall-Architekturen und sichere Internetapplikationen [HaJo06, S. 4].

Mehlau stellt zwei Ansätze vor, die übergreifende Sicherheitsmustersysteme zum Gegenstand haben, zum einen frühere Arbeiten von Schumacher et al. und Ergebnisse einer Einheit bei der Open Group [Mehl05a, S. 63]. Das Sicherheitsmustersystem von Schumacher et al. enthält für diese Arbeit vier relevant Bereiche. Dies sind Benutzerauthentisierung, Identifikation und Authentisierung, Audit und Identitätsbasierte Zugriffskontrolle [Rens 03]; [Sch+03]. Aus dem Sicherheitsmustersystem der Open Group erscheinen die Muster Authenticator und Policy relevant. Zwischen den Mustern bestehen Beziehungen, allerdings nur in Form einer einzigen Beziehungsart, der Referenz [BlHe04]; [Mehl05a, S. 65].

Aus der Arbeitsgruppe um Eduardo Fernandez und Delessy stammt eine Mustersprache für Standards des Federated Identity Managements (z.B. Liberty Alliance) [Del+07, S. 31]. Das im Zuge dieser Arbeit entwickelte Identity Provider Pattern zentralisiert die Administration der Subjekte einer Security-Domäne. Weitere Pattern ermöglichen die Bildung von Trust-Beziehungen unter Service Providern (Circle of Trust Pattern) bzw. die Föderierung multipler Identitäten über Organisationsgrenzen hinweg (Identity Federation Pattern). Das SAML Assertion Pattern stellt schließlich ein einheitliches Format zur Kommunikation von Identitätsinformationen zwischen Securitydomänen zur Verfügung.

Delessy et al. schlagen zwar von der Bezeichnung her eine Pattern Language für Identity Management vor [Del+07, S. 37]. Meines Erachtens liegt der Fokus dieser Pattern jedoch spezieller, nämlich auf Federated Identity Management und weniger auf dem von mir betrachteten Szenario des Enterprise Identity Management.

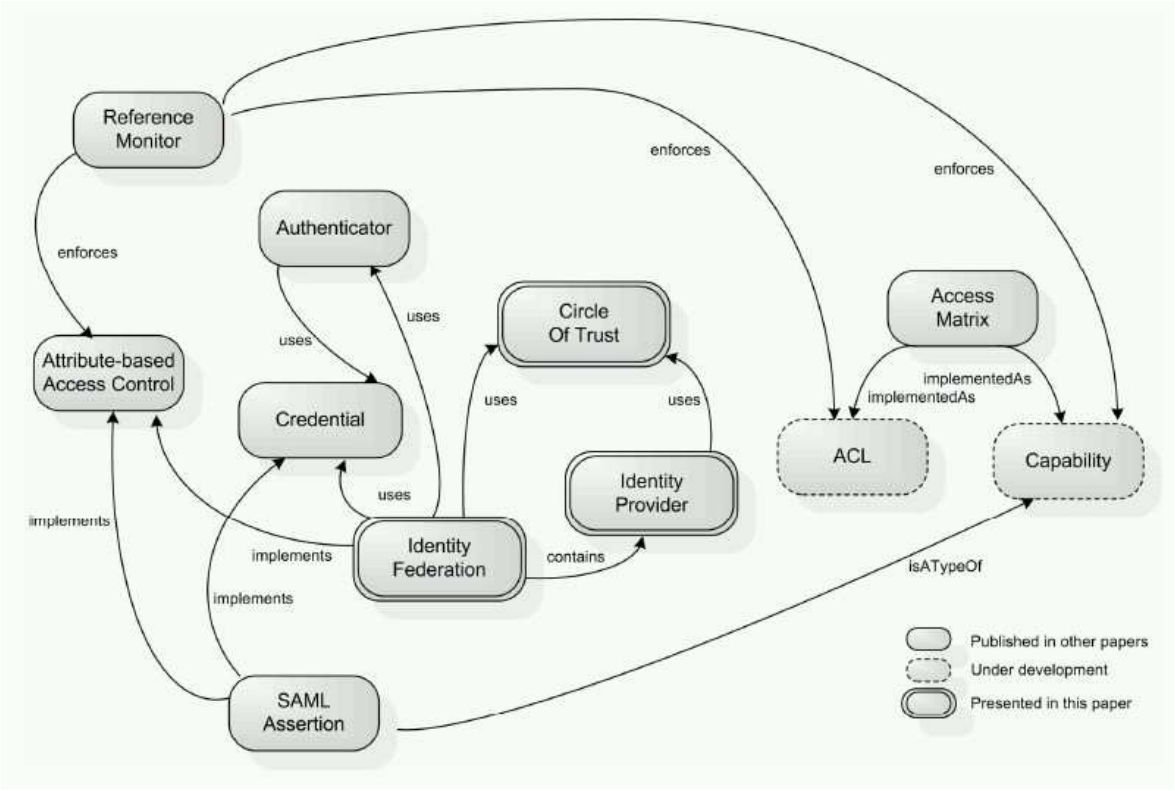

Abbildung 5: Pattern für Federated Identity Management - Delessy, Fernandez et al. [Del+07, S. 32]

Von Emig et al. stammt ein Ansatz für patternbasierte Autorisierungskontrolle im Rahmen serviceorientierter Architekturen (SOA) [Emi+06, S. 62]. Mit Hilfe von Architektur Design Pattern, teilweise aus der Veröffentlichung von Steel et al., werden Legacy Systeme in eine SOA-Umgebung integriert, die um eine Identity Management (IdM) Infrastruktur erweitert wurde [Ste+05]. Die Identity Management Infrastruktur wurde dabei ebenfalls in service-orientierter Art und Weise gestaltet.

Gaedke, Meinecke et al. entwickelten einen Lösungskatalog zur Modellierung und Implementierung für Federated Identity Management, das sowohl auf Analysemuster als auch Bausteine (= Entwurfsmuster) aufsetzt [Gae+05, S. 1156; Mei+05, S. 203]. Das Modellierungs-Framework wird durch einen Katalog von implementierungsnahen Bausteinen ergänzt, deren Richtlinien wie klassische Design Patterns strukturiert sind.

Von Mehlau stammt eine Sammlung sicherheitsrelevanter Pattern, darunter Muster zu Single SignOn, die gut in den Kontext des Enterprise Identity Managements passen und hier daher auch berücksichtigt werden [Mehl05a].

Aus der Darstellung ähnlich gelagerter Arbeiten ergibt sich, dass sich durchaus Quellen und Ansätze in der Literatur finden lassen, die in Teilen der Zielsetzung dieser Arbeit entsprechen. Ein großer Teil der Pattern aus den hier vorgestellten Arbeiten kann der Domäne

des Identity Managements zugeordnet werden. Teilweise werden auch Aussagen über Dependenzen zwischen den Pattern getroffen.

Für die Ausgangsfragstellung, die Klärung des konzeptionellen Verhältnisses von (Security) Pattern zu Referenzmodellen vor dem Hintergrund der Domäne des Identity Managements fanden sich jedoch noch keine Hinweise. Soweit die erwähnten Referenzmodelle und Pattern verfügbar und sinnvoll einsetzbar sind, werden sie für diese Arbeit herangezogen.

6.3 Festlegung der Referenzmodellierungstechnik

In der Phase 2 des Vorgehensmodells von Becker et al. zur Referenzmodellkonstruktion werden die Referenzmodellierungstechniken definiert [Bec+02, S. 43].

Es gibt kaum eigene Sprachen für die Erstellung und Nutzung von Referenzmodellen [Thom06, S. 121]. Eine Ausnahme bilden Referenzprozessbausteine und Referenzmodellkomponenten [Lang97; Broc03, S. 235ff.]. Die meisten Referenzmodelle werden mit einer der etablierten Sprachen zur Informationsmodellierung, ggf. auch einer ihrer Erweiterungen erstellt [Thom06, S. 121]. Rottleb nennt als in Frage kommende Modellierungssprachen ARIS, das Semantische Objektmodell SOM und die Unified Modelling Language UML. ARIS ist verbreitet, UML ist jedoch nach Rottlebs Einschätzung führend [Rott03, S. 14].

Die Modellierungssprachen der ereignisgesteuerten Prozessketten sind in der Literatur zur Referenzmodellierung dominant vertreten. Erst in letzter Zeit wurden stärker objektorientierte Sprachen, speziell die Unified Modeling Language für die Erstellung und Anwendung von Referenzmodellen verwendet [Thom06, S. 134].

In Bezug auf Referenzmodelle benennt Rottleb Anforderungen an eine Modellierungssprache: Die Modellierungstechnik sollte nicht zu abstrakt sein, über eine umfangreiche semantische Ausdruckskraft sowie ein eindeutig interpretierbares Metamodell verfügen [Rott03, S. 14]. Für welche Technik man sich entscheidet, sollte auch davon abhängig gemacht werden, ob eine einfache Überführbarkeit in die Praxis gewährleistet ist [Rott03, S. 14].

Auch bei der Musterbeschreibung ist eine über Text hinausgehende Visualisierung sinnvoll. Eine rein textlich beschreibende Darstellung innerhalb eines Musterschemas ist für einen Musternutzer nur schwer erfassbar [Mehl05a, S. 81]. Zur Unterstützung bei der Ori-

entierung wird daher in der Regel zusätzlich eine grafische, meist semi-formale Notation zur Modellierung von Mustern und Mustersystemen eingesetzt.

Im Bereich der Modellierung von Pattern vertreten Delessy, Fernandez et al. die Ansicht, dass durch UML-Diagramme veranschaulichte Pattern im Gegensatz zu formalen Modellen einen guten Kompromiss bei der Entwicklung sicherer Systeme darstellen. Sie sind hinreichend präzise und gleichzeitig leicht zu verstehen. Dabei betonen die Autoren die Bedeutung von high-level Patterns [Del+07, S. 32]. Nicht unerwähnt soll bleiben, dass auch eine große Zahl der in dieser Arbeit verwendeten Quellen zu Security Pattern UML zu Notationszwecken einsetzen, z.B. Gaedke et al. [Gae+05, S. 1156; Mei+05, S. 203]; Mehlau [Mehl05a, S. 82] und Delessy, Fernandez et al. [Del+07, S. 32] und Steel et al. [Ste+05].

Der Formalisierungsgrad von UML kann durch Verwendung von Beschränkungen (Constraints) der Object Constraint Language (OCL) noch erhöht werden [Schu03, S. 74]. Und ganz generell können Sicherheitsaspekte, unabhängig von Pattern, durch Erweiterungen auch in UML dargestellt werden [Jürj04].

Für die Verwendung von UML auch in der vorliegenden Arbeit spricht eine Reihe von Gründen: Softwareentwickler sind mit UML vertraut und die semi-formalen Abbildungen sind insgesamt auch für Laien gut lesbar [Schu03, S. 74]. UML eignet sich zur Modellierung struktureller Aspekte wie Kardinalitäten und bietet darüber hinaus viele Möglichkeiten hinsichtlich der Modellierung dynamischer Eigenschaften zur Verhaltensmodellierung [Rott03, S. 14].

Aufgrund der Darstellungsmächtigkeit (Struktur- und Verhaltensaspekte) und der weiten Verbreitung der Sprache wird UML für beide Gebiete, die Referenzmodellierung und Security Pattern, verwendet, insbesondere durch Abbildung struktureller Aspekte in UML-Klassenmodellen.

Über die Semantik sicherheitsspezifischer Aspekte besteht bei UML zwar noch keine Einigkeit [Schu03, S. 75], vor dem Hintergrund der genannten Vorteile soll UML dennoch auch in dieser Arbeit als Modellierungssprache eingesetzt werden.

Die Standard UML-Modellierung von Beziehungen sieht lediglich einfache Kanten vor. Das erschwert es, verschiedene Beziehungsarten voneinander zu unterscheiden. Gerade das Erkennen von Wechselwirkungen ist im Zusammenhang von Security Pattern jedoch wichtig. Daher nimmt Mehlau eine Erweiterung der UML für die Modellierung von Security Pattern vor [Mehl05a, S. 82]. Als Erweiterungsmöglichkeit verwendet er UML Profile, die

aus Stereotypen, Eigenschaftswerten und Einschränkungen bestehen [Mehl05a, S. 86]. Beschränkungen nach der Object Constraint Language OCL sind als eigene Klassen definiert und erlauben die Verknüpfung eines Modellelements mit beliebig vielen Constraints [Mehl05a, S. 87].

Im Hinblick auf eine Auswahl der Beziehungsarten soll hier einem Vorschlag Mehlaus gefolgt werden, der Kombination, Spezialisierung, Konflikt und Beeinflussung für ausreichend hält [Mehl05a, S. 90]. Die Beziehungen zwischen Pattern werden ausführlicher in Kapitel diskutiert. Anforderungen an Security Pattern hinsichtlich ihrer Qualität und des Beschreibungsschemas werden zu Beginn des Kapitels 7 erläutert.

Aus inhaltlicher Sicht kann die Definition der konfigurativen Referenzmodellierungstechnik die drei Komponenten Modellrahmen, Verfeinerungsmodelle und Konfigurationsregeln berücksichtigen [Bec+02, S. 47]. Regeln für die Anordnung von Elementen im Modell- oder auch Ordnungsrahmen werden gemeinsam mit der Beschreibung der Konstruktion im nachfolgenden Abschnitt erläutert.

6.4 Entwicklung eines Ordnungsrahmens

Die dritte Phase des Vorgehensmodells von Becker et al. sieht die Entwicklung eines Ordnungsrahmens vor. Für die Konstruktion des Ordnungsrahmens werden verfügbare Quellen analysiert und wesentliche Funktionsbereiche ermittelt, denen als Elementen des Ordnungsrahmens Verfeinerungsmodelle zuzuordnen sind. Für das hier zu entwickelnde Referenzmodell des Identity Managements werden die Teilbereiche durch Pattern bzw. Patternsysteme ausgestaltet.

Die Beziehungen zwischen Elementen des Ordnungsrahmens werden zunächst nicht oder nur rudimentär symbolisiert. Der Ordnungsrahmen dient zu Beginn als Einstieg in ein komplexes Feld und zur Abgrenzung des Betrachtungsgegenstands. Die Details der Zusammenhänge der Bestandteile ist Gegenstand von Kapitel 8.

Diese Phase sieht auch vor, dass das Referenzmodell zum Teil bereits während der Erstellung evaluiert wird [Bec+02, S. 53]. Eine umfangreiche Evaluation ist im Rahmen dieser Arbeit nicht vorgesehen, es soll jedoch versucht werden, diesen Aspekt in der abschließenden Darstellung im Kapitel 8 mit abzudecken.

Zunächst wird das Konzept des Enterprise Identity Managements durch einen Abgleich mit empirischen Quellen herausgearbeitet und präzisiert. Dazu werden die Kern- und Unter-

stützungsfunktionen des Enterprise Identity Managements anhand von Literaturquellen identitifiziert und weitere typische und relevante Strukturmerkmalen betrachtet.

Bei der Ableitung des Referenzmodells wurde eine induktive Vorgehensweise gewählt. Dem Ordnungsrahmen und dem detaillierten Modell liegen empirische, vom Autor als repräsentativ angesehenen Veröffentlichungen und Quellen zum Enterprise Identity Management zugrunde. Die empirische Vorgehensweise bei der Erstellung des Referenzmodells entspricht gängigen und akzeptierten Methoden [Fet+05, S. 9ff]. Die induktive Variante geht darüber hinaus konform mit der Methode des Pattern Mining. Sowohl Design Pattern als auch Security Pattern werden entdeckt und nicht zu allererst am grünen Tisch anhand theoretischer Erkenntnisse geplant und entworfen [Mehl05a, S. 45].

Die empirische Basis soll bewusst auf eine überschaubare Anzahl von Arbeiten beschränkt werden. Die Arbeiten werden auf Übereinstimmungen und Überschneidungen untersucht und einzelne Funktionsbereiche des Identity Managements herausgearbeitet. Weniger relevante Themen werden nur kurz angesprochen, jedoch nicht im Detail erläutert.

Die Auswahl der empirischen Basis für den Ordnungsrahmen des Referenzmodells beruht auf der Einschätzung, dass es sich um Quellen mit ausgeprägter Expertise im Umfeld des Identity Management oder der Security Pattern handelt. Die Arbeiten wurden überwiegend erst in den letzten drei bis vier Jahren veröffentlicht und sind daher als vergleichsweise aktuelles Material anzusehen.

Die Autoren um Casassa-Mont et al. sind für Hewlet Packard im Identity Management Umfeld tätig. Ihre Arbeit soll aber auch deswegen berücksichtigt werden, weil sie den Anspruch erhebt, den State of the Art des Identity Managements im Hinblick auf seine Funktionen zu analysieren [Cas+03]. Ebenfalls thematisch passend ist die Arbeit von Rene Rottleb über ein Referenzmodell aus dem Umfeld des Enterprise Access Management [Rott03].

Für das der Firma Sun zuzuordnende Autorenteam Steel et al. spricht, dass sie durch ihre Mitarbeit bei einem Hersteller eines Identity Management Systems mit Expertenwissen aufwarten können. Außerdem besteht ein eindeutiger Schwerpunkt in einer ihrer jüngsten Veröffentlichungen in der Verknüpfung von Security Pattern mit Fragen des Identity Managements [Ste+05].

Phil Windley kann ebenfalls auf langjährige Erfahrung im Umfeld des Identity Managements zurückblicken. In seine Buchveröffentlichung zum Thema Identity Management fließen Erkenntnisse des in der gleichen Domäne spezialisierten Beratungs- und Analy-

sedienstleisters Burton Group ein, die die Expertensicht auf den Gegenstand noch weiter fundieren [Wind05].

6.5 Typische Enterprise Identity Management Funktionen

Nachfolgend werden die Vorschläge der bereits angesprochenen Autoren geschildert, welche Kernfunktionen und Dienste Enterprise Identity Management Systeme derzeit bereitstellen.

Vor dem Hintergrund wachsender Anforderungen an Flexibilität und organisationsübergreifende Zusammenarbeit stellen Casassa-Mont, Bramhall und Pato eine Zusammenschau derzeitiger Enterprise Identity Management Lösungen vor [Cas+03, S. 7]. Sie arbeiten in ihrer Analyse folgende Schwerpunkte des gegenwärtigen Identity Management Paradigmas heraus: Mehrere, gering integrierte Identity Management Produkte und Lösungen agieren in vergleichsweise geschlossenen und wenig veränderlichen, stark kontrollierten Umgebungen [Cas+03, S. 7].

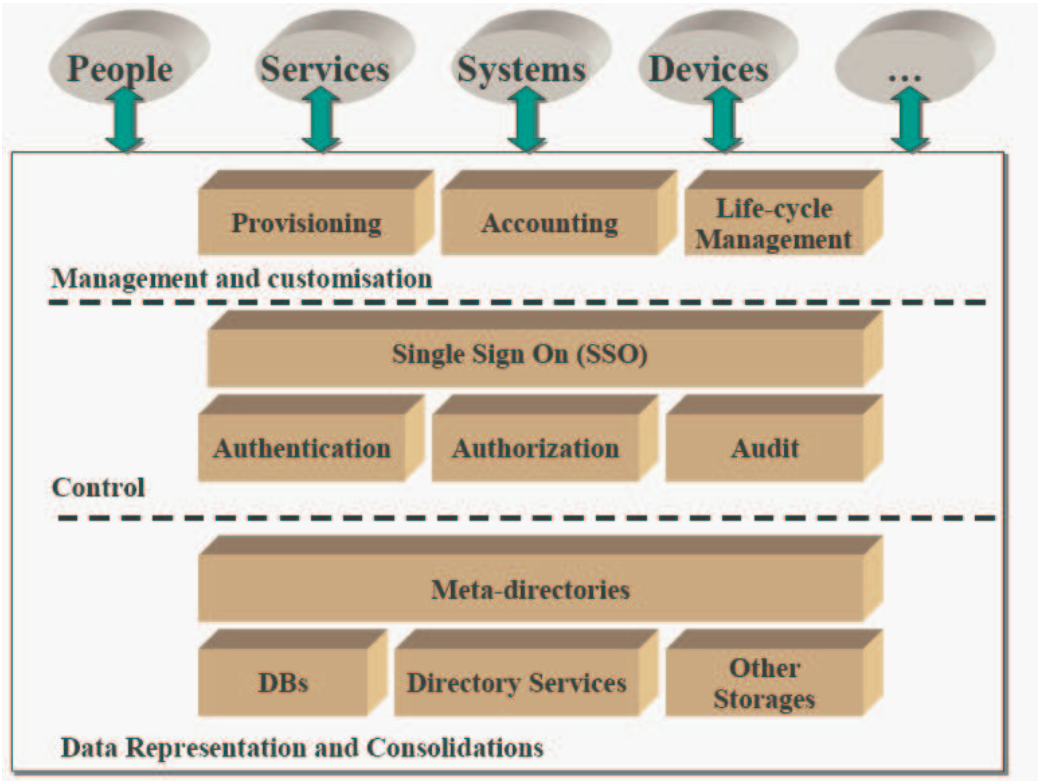

Abbildung 6: Gegenwärtiger Identity Management Lösungs-Stack [Cas+03, S. 7]

Verfügbare Enterprise Identity Management Produkte und Lösungen bieten Funktionen wie die Kernfunktionen Directory services, Authentication, Authorization and Auditing. Hierzu sind Single-Sign-On (SSO) und Role-based Access Control (RBAC) zu rechnen.

Weitere Funktionen sind Provisioning und Accounting Lösungen, Identity Life Cycle Management Lösungen, Web Services Standards und Federated Identity Management. Die vorangehende Abbildung zeigt die Hauptkomponenten und –funktionen heutiger Enterprise Identity Management Produkte und Lösungen aus Sicht von Casassa-Mont et al.

Nach Steel et al. stellt eine typische Identity Management Infrastrukturlösung die folgenden Dienste zur Verfügung [Ste+05, S. 27]:

- *Identity Provisioning Services* dienen der zeitnahen Einrichtung, Veränderung von Accounts und Zugriffsrechten. Workflow Komponenten automatisieren diesen Prozess.
- *Identity Data Synchronization Services* stellen über Systemgrenzen hinweg die Veränderung von Daten sicher und ersparen so manuelle redundante Änderungen
- *Access Management* ermöglicht SingleSignOn zu Anwendungen und Diensten, unabhängig davon, wo diese Dienste residieren [Ste+05, S. 28]
- *Federation Services* stellen ein Framework und geeignete Mechanismen zum Austausch von Authentisierungsinformationen über die Grenzen von Organisationen hinweg zur Verfügung
- Durch *Directory Services* werden gemanagte Identitätsdaten mit hoher Performance, sicher und mit der benötigten Interoperabilität bereitgestellt
- *Auditing und Logging* ermöglichen die Entdeckung von Sicherheitsrisiken und einen Nachweis der Compliance zu vertretbaren Kosten in vernünftiger Zeit

Neben einzelnen „Bausteinen“, wie sie Security Pattern teilweise darstellen, kann die Erstellung sicherer Applikationen auch durch ganze Sicherheits-Frameworks unterstützt werden. Steel et al. stellen in ihrem Buch das Sicherheitsframework der J2EE-Plattform mit seinen logischen Komponenten vor [Ste+05, S. 525]:

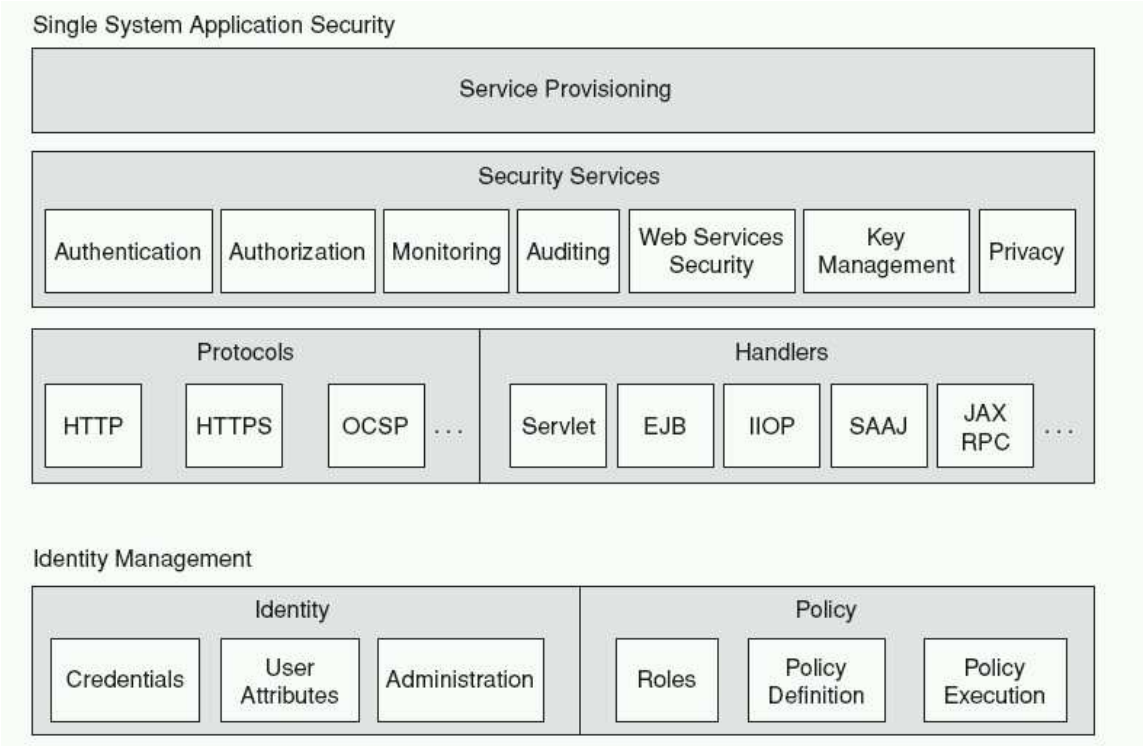

Abbildung 7: Logical Security Framework nach Steel et al. [Ste+05, S. 526]

Ein weiterer Vorschlag zur logischen Strukturierung einer generischen Identity Management Architektur stammt von Phil Windley, der dabei Arbeiten des Beratungs- und Marktforschungsunternehmens Burton Group zitiert [Wind 04, S. 206].

Die Identity Management Architektur der Burton Group nimmt Stellung in Bezug auf die Kernbestandteile einer entsprechenden Infrastruktur. Das sind Benutzerauthentisierung, Zugriffsmanagement, Provisionierungs-Dienste, Identity und Policy-Administration, Verzeichnissysteme, Föderierungsdienste, sowie Werkzeuge zur Verwaltung von Identitäten, Rollen und Gruppen [Wind05, S. 219].

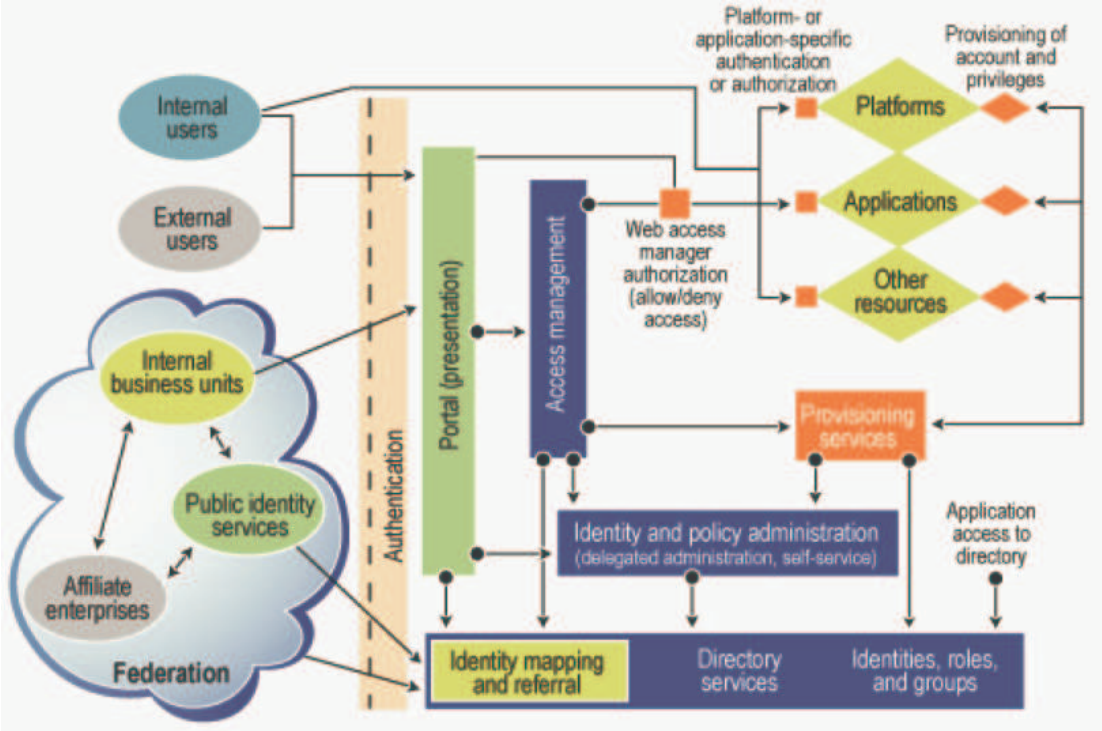

Abbildung 8: Generische Identity Management Architektur (Burton Group, nach [Wind04, S. 217])

Zur besseren Übersicht und zum Vergleich werden die verschiedenen Positionen über die wesentlichen Bestandteile von Identity Management Systemen noch einmal in Tabellenform dargestellt (siehe nachfolgende Tabelle 2):

Tabelle 2: Wesentliche Bestandteile von Identity Management Systemen

	Casassa et al.	**Rottleb**	**Steel et al.**	**Windley / Burton**	***In Form von Pattern verfügbare Funktionen***
Provisioning	Provisioning		Identity Provisioning	Provisioning	*Provisioning pattern*
Directories	Directories	Verzeichnissysteme	Directories	Directories	*Pattern für Datenbanken in Planung*
Authentication	Authentication und SSO	Authentisierung	Access Management / SSO	Authentication	*Authentication /Single Sign-On Sicherheitsmuster*
Authorization	Authorization	Zugriffskontrolle		Zugriffsmanagement	*Authorization Pattern* *RBAC Pattern*
Federation			Federation Services	Federation	*Security Assertion Coordination pattern(Federation)*

	Casassa et al.	**Rottleb**	**Steel et al.**	**Windley / Burton**	***In Form von Pattern verfügbare Funktionen***
Verwaltung Identities, Roles, Groups	(Identity) Life Cycle Management	Berechtigungs / Identitätsverwaltung		Verwaltung von Identitäten, Rollen und Gruppen	
Administration von Policies und Workflows		Policy-Verwaltung		Policy-Administration	
Accounting und Auditierung	Accounting Audit		Auditing und Reporting		*Logging*

Betrachtet man die Tabelle, zeigt sich eine Reihe von Übereinstimmungen, die auf unabhängig voneinander gewonnenen Einschätzungen beruhen. Dies sind die Funktionsbereiche Provisioning, Directory Services, Authentisierung, Autorisierung, Federation Services und Auditierung.

Nimmt man die Übersicht zum Maßstab, werden Themen wie die Verwaltung von Rollen und Gruppen sowie die Administration von Policies und Workflows zum Zeitpunkt der Veröffentlichung der Quellen als weniger zentral angesehen.

Im weiteren Fortgang werden die erstgenannten Funktionen, die in den Veröffentlichungen häufiger genannt wurden, schwerpunktmäßig betrachtet. Natürlich spielt hierbei auch eine Rolle, für welche Funktionen zum heutigen Zeitpunkt eine nennenswerte Zahl von Pattern vorliegt, siehe hierzu auch die Studie zur Security Pattern Landschaft von Heyman et al. [Hey+07]. Dieser Gedanke wird in der Tabelle durch die ganz rechte Spalte verdeutlicht.

Funktionen des Identity Managements, die bisher weder in der Identity Management Literatur noch in der Pattern-Community besondere Aufmerksamkeit auf sich gezogen haben, werden daher im weiteren Verlauf der Arbeit nur überblicksartig behandelt. Konzeptionell sollen sie aber zumindest durch entsprechende Abhängigkeiten oder Schnittstellen Berücksichtigung finden.

6.6 Ordnungsrahmen für das Enterprise Identity Management

Um das hohe Maß an Komplexität von Informationssystemen zu handhaben, präsentieren Modelle ihre Gegenstände in Formen, bei der von der Menge der Details abstrahiert wird [Thom06, S. 111].

Bei umfangreichen Referenzmodellen übernehmen Ordnungsrahmen die überblicksartige Darstellung [Meis01, S. 62]. Ein Ordnungsrahmen strukturiert abgebildetes Gebiet in Hauptbereiche und bildet damit ein Verzeichnis, das in verschiedene Bereiche unterteilt ist. Den Bereichen sind wiederum Modelle und Teilmodelle zugeordnet. Letztlich geht es bei Ordnungsrahmen um die Versinnbildlichung der betrachteten Zusammenhänge [Thom06, S. 115]. Für den Ordnungsrahmen der vorliegenden Arbeit wird das Referenzdesign „Haus“ von Meise gewählt, siehe Abbildung 9 [Meis01, S. 217].

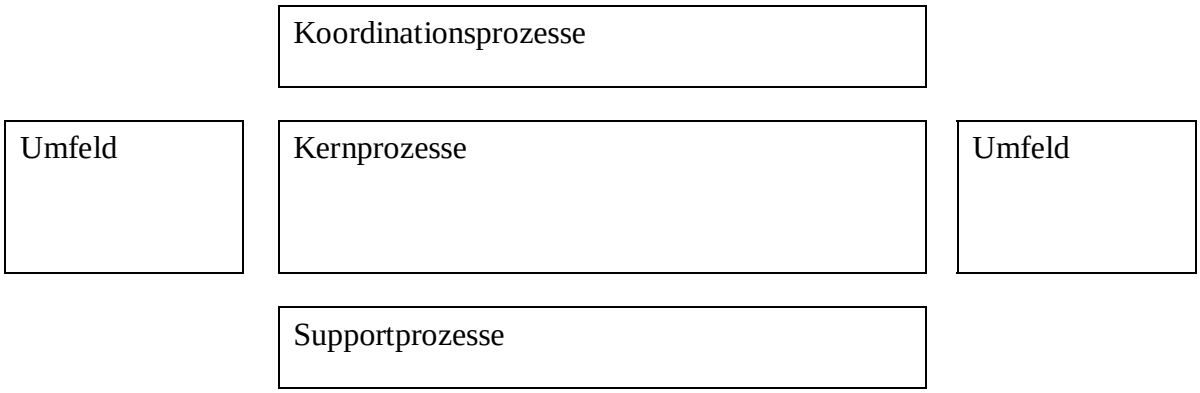

Abbildung 9: Referenzdesign „Haus“ für Ordnungsrahmen nach Meise [Meis01, S. 217]

Für das Referenzdesign existieren Gestaltungsanforderungen, die die Anordnung von Elementen zusätzlich systematisieren. Bspw. sollen zeitlich-sachlogische Abfolgebeziehungen gemäß der üblichen Leserichtung von links nach rechts abgebildet werden [Meis01, S. 210]. Übertragen auf die Domäne des Identity Managements wären hier etwa Zustände wie „nicht authentifiziert“ und „authentifiziert“ oder „assoziiertes Partnerunternehmen“ und „eigenes Unternehmensnetzwerk“ zu nennen. Das entspricht übrigens auch der Darstellung einer Identity Management Referenzarchitektur nach Windley / Burton in der letzten Abbildung im vorigen Abschnitt, siehe oben. Das gleich gilt für die Abbildung von Umweltbeziehungen, diese sollen erkennbar sein [Meis01, S. 211].

Hierarchische Zusammenhänge werden von oben nach unten abgebildet [Meis01, S. 211]. Dem könnten im Themenfeld dieser Arbeit Steuerungs-, Kern- und Unterstützungsaufgaben zugeordnet werden. Zu den domänenspezifischen Anforderungen gehört, dass alle relevanten Elemente der Domäne, d.h., die im vorigen Abschnitt erarbeiteten Funktionen und Aufgaben sich im Ordnungsrahmen wiederfinden müssen, (siehe oben) [Meis01, S. 212 / 213].

In den hier entwickelten Ordnungsrahmen fließen Inhalte des genannten Referenzmodells und der als repräsentativ ausgewählten Veröffentlichungen aus der Identity Management

Literatur ein. Dabei wurde versucht, spezielle Beziehungen zwischen Akteuren etc. angemessen zu berücksichtigen.

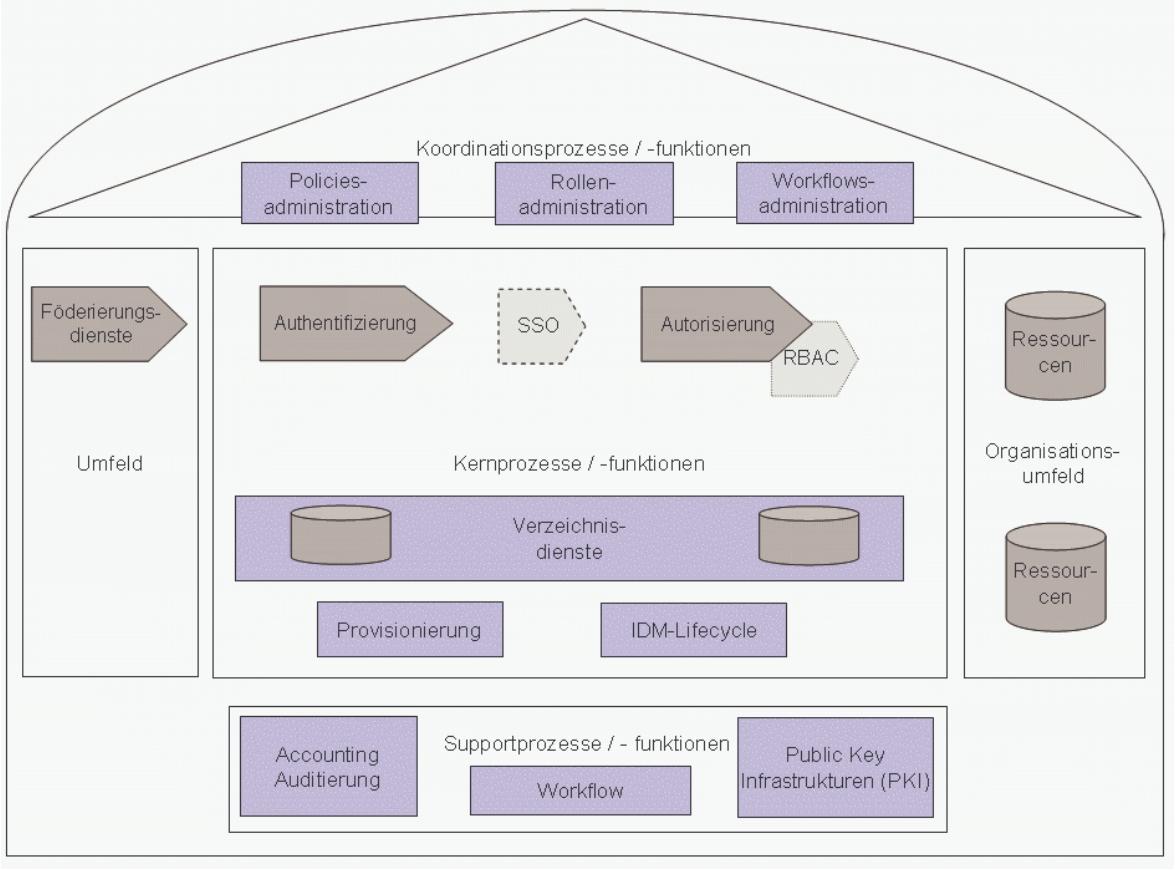

Abbildung 10: Ordnungsrahmen für ein Identity Management Referenzmodell

Die obige Abbildung zeigt den Ordnungsrahmen mit den wichtigsten Teilfunktionen eines generischen Enterprise Identity Managements Systems. Der Ordnungsrahmen gliedert sich grob in die fünf Bereiche des Referenzdesigns „Haus“.

Der Leserichtung folgend von links nach rechts entspricht die (symbolisierte) Abgrenzung von „drinnen“ und „draussen“, also dem nicht authentifizierten Zustand außerhalb einer Organisation und dem identifizierten und berechtigten Zustand innerhalb einer Organisation. Kernfunktionen wie Authentisierung und Autorisierung bilden den zentralen Mittelteil. Dieser wird vertikal eingefasst von den Koordinations- und den Unterstützungsprozessen.

Damit entspricht die Darstellung einem Ansatz von Delessy et al., die ebenfalls abstrakte Konzepte von konkreten Implementierungen trennen [Del+07, S. 37]. Dasselbe Verhältnis besteht zwischen dem hier entwickelten Ordnungsrahmen für ein Referenzmodells für Identity Management zu den Pattern bzw. Patternsystemen, auf denen es basiert.

Die Hauptaspekte des Ordnungsrahmens werden nun jeweils durch Pattern oder Patternsysteme detailliert. Diese bilden zusammen ein Referenzmodell oder auch umfassendes Patternsystem für die betrachtete Domäne Identity Management. Die Darstellung der Pattern erfolgt im nachfolgenden Kapitel 7. Nach Becker et al. kann diese Verfeinerung wiederum über mehrere Stufen erfolgen, wobei die verwendeten Modellierungstechniken wechseln können [Bec+02, S. 48]. Letzteres ist nur insoweit der Fall, als neben der Modellierung mit Hilfe der Unified Modeling Language UML auch gängige Patternschema für die Dokumentation verwendet werden.

Nach der Einzeldarstellung der Bestandteile des Ordnungsrahmens in Kapitel 7 in Form von Security Patterns folgt in Kapitel 8 die Zusammenfassung als Referenzmodell bzw. Patternsystem, inklusive einer detaillierten Darstellung der Beziehung zwischen den Pattern und Patternsystemen.

7 Security Pattern für Funktionen des Identity Managements

In diesem Kapitel erfolgt die Detaillierung der bereits vorgestellten Funktionsbereiche des Ordnungsrahmen für das Referenzmodell des Identity Managements. Die Muster entsprechen in ihrer Herkunft und anderen Kriterien Anforderungen, die ebenfalls im vorigen Kapitel angesprochen wurden. Dies ist zum einen die Erfahrung von Autoren bzw. einer Arbeitsgruppe im Umgang mit Pattern, die thematische Nähe zur Domäne des Identity Managements, aber z.B. auch die Erwähnung und das Abschneiden in vergleichenden Studien wie denen von Heymann et al. oder Yskout et al. [Hey+07]; [Ysk+06]. Bspw. haben die Autoren um Markus Schumacher et al. einen guten Ruf in der Pattern Community und haben sich intensiv bei der Weiterentwicklung von Security Pattern eingebracht [Sch+06].

Nach Heyman et al. definiert sich die Qualität eines Pattern nicht zuletzt durch das Maß, in dem es die Vorgehensweise bei der Konstruktion durch Hinweise unterstützt und nicht lediglich Zielzustände beschreibt [Hey+07]. Damit sind also Pattern auf DV-Konzeptebene gemeint, weniger High-Level-Hinweise, Hilfsmittel oder Sicherheitsrichtlinien.

Aus Sicht von Yskout et al. ist die in der Veröffentlichung von Steel et al. verwendete Granularität durchaus angemessen. Dies spiegelt sich auch darin wieder, dass viele Pattern aus dem Buch von Steel et al. sich in einer abschließenden Auswahl von Pattern von Yskout et al. wiederfinden [Ysk+06, S. 7]; [Ste+05].

Problematisch ist allerdings, dass die Detaillierung in den Sammelarbeiten von Steel et al. und Schumacher et al. nach Auffassung von Yskout et al. nicht immer konsistent ist und

sich auch noch kein Standardformat in der Beschreibung durchgesetzt hat [Ysk+06, S. 8]. Diese Aspekte treffen in Teilen auch auf das vorliegende Kapitel dieser Arbeit zu.

Inhaltlich konzentriert sich die Darstellung der Funktionen im wesentlichen auf die Kernprozesse des Ordnungsrahmens, nicht zuletzt auch, weil zu diesen Themen viele Pattern veröffentlicht wurden. Gerade bei Pattern, für die Spezialisierungen existieren, macht dies z.T. auch Sinn, da so eine größere Auswahl zur Verfügung steht.

7.1 Musterschema

Yskout et al. weisen darauf hin, daß qualitativ hochwertige Security Pattern den Entwicklungsprozess durch konkrete Konstruktionshinweise unterstützen [Ysk+06, S. 8]. Betrachtet man Musterschemata im Hinblick auf diese Aussage, so ist ihres Erachtens eine ausführliche Form der Dokumentation zu bevorzugen. Diese sollte demnach mehr als nur den Minimalstandard erfüllen. Das wäre Name, Aliases, die Motivation, das Problemstatement, die Lösung ein Beispiel, Konsequenzen, Beziehungen zu anderen Pattern und bekannte Anwendungsfälle.

Eine verbesserte Fassung liefert außerdem eine Struktur, Beteiligte und Verantwortlichkeiten, Implementierungsstrategien mit Codebeispielen und UML Sequenzdiagrammen, Sicherheitsfaktoren und Risiken sowie einem abschließenden Realitätscheck.

Auch wenn Yskout e al. für die ausführliche Fassung plädieren, lässt sich das in dieser Arbeit aus Platzgründen und aufgrund der Vielzahl der Quellen nicht immer durchhalten. Es werden jedoch die Verweise in den verwendeten Quellen auf andere Quellen zu Vertiefungszwecken wiedergegeben. Verwendet wird im Schwerpunkt das vergleichsweise kurze Sicherheitsmusterschema von Mehlau [Mehl05a, S. 80].

7.2 Authentisierung

Authentifizierung ist eine der sicherheitsrelevanten Funktionen, zu der in den letzten Jahren vergleichsweise viele Security Pattern veröffentlicht wurden. In der Erhebung von Heyman et al. liegt Authentisierung in Bezug auf die Zahl der Pattern mit fast dreissig Veröffentlichungen an dritter Stelle von insgesamt 15 betrachteten Funktionen [Hey+07, S. 6].

Die große Zahl an Veröffentlichungen bringt für Anwender jedoch nicht nur Vorteile. Teilweise werden Pattern mit gleichen Funktionen unter unterschiedlichem Namen veröffentlicht. Auch Hafis und Johnson weisen anhand der Authentisierungsfunktion auf das Pro-

blem der Überlappung von Pattern in verschiedenen Security Pattern Katalogen hin [HaJo06, S. 5].

So wurde etwa das Authenticator Pattern von Fernandez und Sinibaldi später auch im Security Pattern Buch von Schumacher et al. veröffentlicht [FeSi03]; [Sch+06, S. 321].

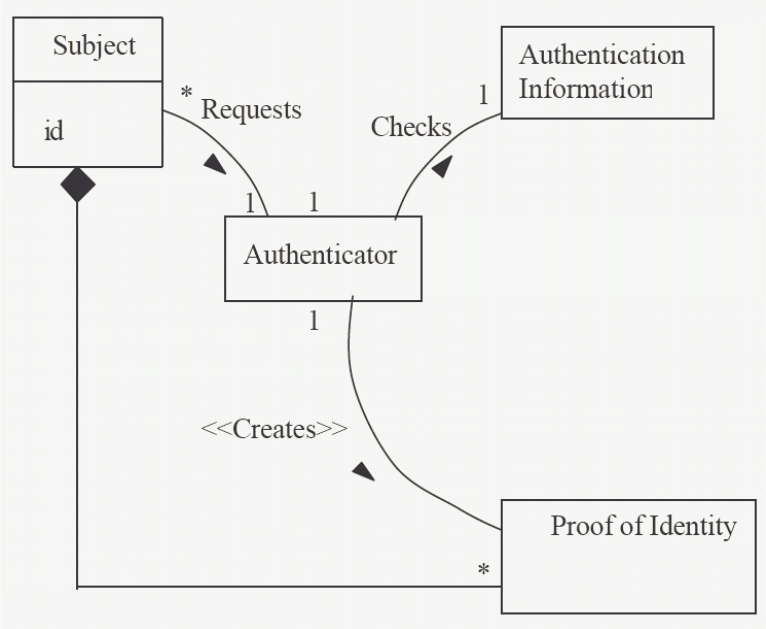

Abbildung 11: Authenticator Pattern nach Fernandez und Sinibaldi [FeSi03, S. 6]

Das Sicherheitsmuster "Authenticator" entspricht im wesentlichen einem Policy Enforcement Point PEP, wie von Yoder und Barcalow beschrieben [YoBa97]. Durch das PEP-Muster werden alle Interaktionen zwischen einem Subjekt und einem System abgefangen und unter Anwendung eines Protokolls wird die Identität des Subjekts geprüft. Welches Protokoll verwendet wird, ist vom gewählten Authentifizierungsverfahren (Passwort, Smartcard, biometrische Merkmale, Bestätigung eines Trust-Servers) abhängig.

Durch Sasha Romanosky wurde das gleiche Pattern unter dem Namen Security Provider beschrieben [Roma01, S. 11]. Darüber hinaus ist das Pattern Teil eines Security Pattern Repositories von Kienzle et al., die es als Authenticated Session führen [Kie+02].

Im Rahmen des Core Security Pattern Buchs wurde das Authentication Enforcer Pattern aus einer Java-orientierten Perspektive beschrieben [Ste+05, S. 535]. Das Muster taucht ebenso in einer Veröffentlichung von Microsoft zum Thema Security Patterns auf [Hog+06].

Als eine Variante von mehreren soll das Authenticator Enforcer Patern nach Steel et al. herausgegriffen werden [Ste+05, S. 535].

7.2.1 Authentication Enforcer

Name:

Authentication Enforcer [Ste+05, S. 535]

Problem:

Es ist erforderlich, dass nur Requests von authentisierten Subjekten weitergeleitet werden [Ste+05, S. 535]. Außerdem sollte der Programmcode möglichst änderbar und wartbar sein. Die Mechanismen sollten möglichst abstrakt und gekapselt sein, um auch in heterogenen Umgebungen eingesetzt werden zu können. Die transportierten Berechtigungsnachweise müssen vertraulich behandelt werden [Ste+05, S. 536].

Lösung:

Als Lösung kommt eine zentralisierte Authentisierung in Frage, die die Identität der Benutzer prüft und Details der Implementierung kapselt. Das Pattern von Steel et al. bildet den Authentisierungsmechanismus auf der logischen Ebene „Web“ ab, die der Einteilung von Steel et al. in Ebenen folgt. Unterstützt werden passwortbasierende Verfahren, aber auch zertifikatsbasierte und Mechanismen wie Kerberos, so dass Veränderung in den Anforderungen aufgefangen werden können [Ste+05, S. 536]. Die generische Schnittstelle ist unabhängig von Protokollen und auf verschiedenen Ebenen (Tiers) einsetzbar. Die nachfolgende Abbildung illustriert mit einem Klassendiagramm die Struktur des Pattern. Die Berechtigungsnachweise werden im RequestContext übergeben, authentifiziert Benutzer werden durch eine Subject-Instanz repräsentiert.

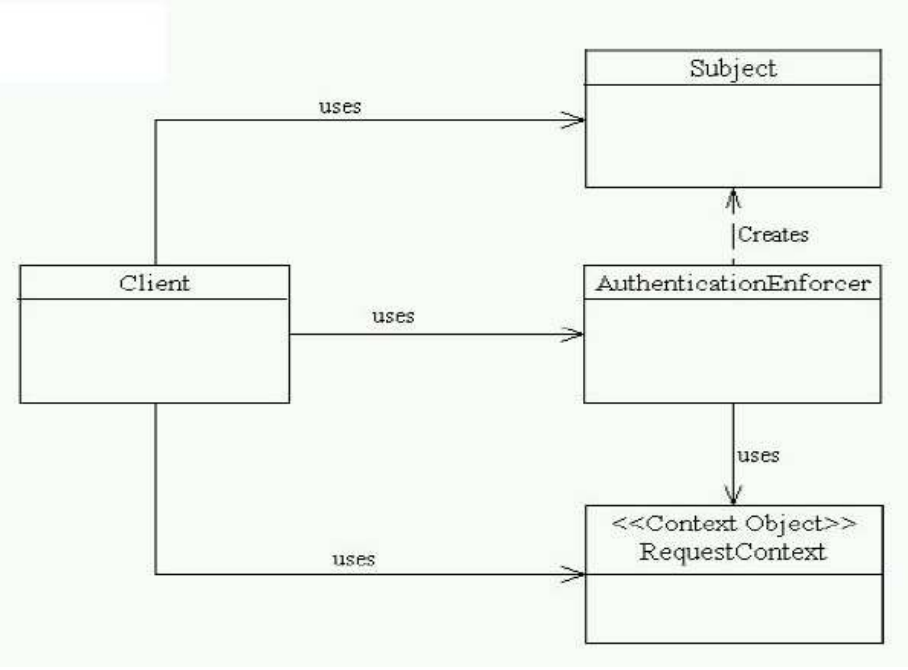

Abbildung 12: Authentication Enforcer Class Diagram nach Steel et al. [Ste+05, S. 537]

Konsequenzen:

Der Codeumfang wird reduziert und der Authentisierungsmechanismus wird konsolidiert [Ste+05, S. 544]. Durch die Kapselung und Zentralisierung sind Anwendungen vor Änderungen in den Authentisierungsmechanismen. Durch die Zentralisierung verringert sich auch die Zahl möglicher Angriffspunkte bzw. Sicherheitslücken [Ste+05, S. 545].

Beziehungen zu anderen Pattern:

Zwischen dem Authenticator Pattern und dem Intercepting Filter besteht eine Beziehung [Ste+05, S. 545].

7.2.2 Weitere Authentisierungsmuster

Varianten:

Das Muster Authenticator, ursprünglich 1999 von Brown et al. erarbeitet, verwendet Credentials, um die Authentisierung durchzuführen [Bro+99]; [Sch+06, S. 321]. Bevor über den Zugriff auf ein angefragtes Objekt entschieden wird, führt das Authenticator Pattern die Authentisierung das anfragenden Prozesses durch [Hal+06, S. 386]. Durch die Authenticatorklasse wird das “Least-Privilege”-Prinzip umgesetzt.

Authenticator für Public Key Infrastructures (PKI)

Das Authenticator Pattern kann geringfügig modifiziert werden, so dass es die Identität eines Subjekts mit Public Key Kryptographie verifiziert. Die Authenticator-Klasse führt die Authentisierung unter Verwendung eines Zertifikats durch, das mit dem Public Key einer Certificate Authority abgezeichnet ist. Das Ergebnis der Authentisierung könnte ein Authentisierungstoken sein [FeSi03, S. 5].

Yoder und Barcalow beschrieben 1997 in ihrer Arbeit über Security Patterns auch zwei Pattern, Single Access Point und Checkpoint, die dazu dienen, den Zugang eines Systems abzusichern [YoBa97, S. 4 / 7].

Single Access Point

Durch das Single Access Point Pattern wird eine einzige Schnittstelle für die gesamte Kommunikation mit dem System durch externe Entitäten definiert, um die Kontrolle und zu überwachen. Das Muster ist damit der einzige Weg für einen Zugang zum System, an dem kein Request vorbeikommt [Rom+06, S. 2].

Checkpoint

Die Struktur des Check Point Patterns prüft ankommende Eingaben auf Verstöße, um ggf.

Gegenmaßnahmen zu ergreifen [YoBa97, S. 7ff]. Neben der Authentisierung unterstützt das Check Point Pattern Logging und Überwachung [Rom+06, S. 2].

Im Security Pattern Buch von Schumacher et al. findet sich eine Pattern Language, die die Pattern Single Access Point und Checkpoint ergänzt [Sch+06, S. 297 / 473]. Dabei handelt es sich um die Muster Security Session und Front Door [Sch+06]. Die Pattern Single Access Point und Check Point Pattern können bspw. eingesetzt werden, um sicheren Zugang zu Betriebssystemen, Websites und verteilten Anwendungen zu gewährleisten. Security Session und Front Door liegen etwa den Single Sign-On Funktionalitäten wie Microsofts .NET framework, Federated Identity Management Funktionen und Authentisierungslösungen in Unternehmen zugrunde [Rom+06, S. 2].

In der Veröffentlichungen von Schumacher et al. finden sich auch eine Reihe von Analysemustern für Identification and Authentication (I&A), mit denen bspw. I&A Anforderungen erhoben oder Designalternativen bewertet werden können [Sch+06, S. 192 / 207]. Im Sinne der Zielsetzung dieser Arbeit werden Analysemuster jedoch nicht weiter vertieft.

7.3 Generisches Single Sign On und Spezialisierungen

Der Mechanismus, mit einer einmaligen Authentifizierung auf mehrere Systeme zugreifen zu können, wird als Single Sign On bezeichnet. Für dieses Konzept existiert eine allgemeines SSO-Muster, das Spezialisierungen erfahren kann [Mehl05b, S. 457]. Sicherheitsmuster, die generisches Single Sign On und seine Spezialisierungen unterstützen, werden auf der Systemebene und der Unternehmensebene eingruppiert [Mehl05a, S. 99].

Das Ziel, eine einmalige Authentifizierung über mehrere Systeme hinweg zu nutzen, lässt sich auf Basis unterschiedlicher Ansätze realisieren [Mehl05a, S. 131]. Für die allgemeine Zielsetzung existiert ein generisches Muster. Als Spezialisierungen kommen bspw. die Formen „Skriptbasiertes SSO“, „Agentenbasiertes SSO“, „Brokerbasiertes SSO“ und „Gatewaybasiertes SSO“ in Frage.

Die Ansätze können in dezentrale und zentrale Lösungen unterschieden werden. Bei dezentralen SSO-Lösungen verwendet das Betriebssystem eines Endgeräts die Authentifizierungsdaten des Benutzers dazu, um im Hintergrund weitere Anmelde- und Autorisierungsvorgänge im Netzwerk und an anderen Applikationen durchzuführen. Dies ist bei Gateway- und Skriptbasierten Lösungen der Fall.

Alternativ wird bei zentralen Ansätzen eine eigene Sicherheitsschicht mit SSO-Clients und Sicherheitsservern implementiert, die die Anmeldung für verschiedene Anwendungen und Systeme übernehmen [Mehl05a, S. 132].

Mustername:
Generisches Single Sign

Kontext:
Es werden verschiedene Systeme parallel eingesetzt, dabei ist für jedes System eine eigene Anmeldung erforderlich.

Problemabschnitt:
In einer Umgebung mit mehreren IT-Systemen führt das Erfordernis der wiederholten Anmeldung an jedem System zu Effizienzverlusten und einer Überforderung der Mitarbeiter. Die Usability ist gering. Durch strikte Passwortvorgaben wie häufige Änderungen und Beachtung von Groß- und Kleinschreibungen ist bei mehreren zu verwendenden Passworten die Zahl der Fehlanmeldungen hoch und die Belastung des User Help Desks steigt. Zu beachten sind auch die redundanten Verwaltungsaufwände bei den betroffenen Systemen [Mehl05a, S. 133]. Beim Ausscheiden oder Wechseln von Mitarbeitern besteht die Gefahr, dass Berechtigungen bestehen bleiben und die Datenqualität insgesamt abnimmt.

Lösungsabschnitt:
In einer möglichen Variante wird dem User die Arbeit der Anmeldung an verschiedenen Applikationen etc. durch ein System abgenommen. Da in diesem „Stellvertreter-System“ alle benötigten Anmeldeinformationen hinterlegt sind, müssen an den eigentlichen Anwendungen und anderen Zielsystemen nur geringe Änderungen durchgeführt werden. Die in den Zielsystemen bestehenden Mechanismen zur Authentifizierung und Benutzeradministration bleiben unverändert und müssen auch jedes für sich administriert werden.

Weitergehend ist die Lösung eines zentralen SSO-Dienstes, die allerdings auch Änderungen an den beteiligten Systemen erfordert. Der Prozess der Authentisierung ggf. auch Autorisierung wird jetzt nur durch ein System durchgeführt, das auch für die Benutzerverwaltung zuständig ist. Alle weiteren Systeme führen keine eigene Authentisierung mehr durch, sondern vertrauen jetzt dem zentralen Dienst, wenn ein im Vorfeld angemeldeter Benutzer auf sie zugreift [Mehl05a, S. 133].

Konsequenzabschnitt:
Für Nutzer wird durch diese Lösungen eine transparente Anmeldung auf allen Zielsys-

temen ermöglicht, bei unterschiedlichen Aufwänden für die Einführung und den Betrieb der Lösungen.

Nachteilig ist, dass durch die Konzentration sensibler Informationen auf ein System ein „Single Point of Failure“ bzw. auch Angriffspunkt entsteht. Gelangt ein unberechtigter Nutzer an die hinterlegten Daten, kann er u.U. auf alle angebundenen Systeme zugreifen bzw. den Zugang für alle anderen Benutzer unterbinden. Daher sind für zentrale Systeme hohe Anforderungen an die Vertraulichkeit und die Verfügbarkeit zu stellen. Durch eine redundante Auslegung kann bspw. Hochverfügbarkeit gewährleistet werden [Mehl05a, S. 134].

Abhängigkeiten:
Für das Sicherheitsmuster „Generisches Single Sign On“ existieren Spezialisierungen in Form von „De- und Zentralisiertes skriptbasiertes SSO“ [Mehl05a, S. 134/135], „Agentenbasiertes SSO“ [Mehl05a, S. 137], „Brokerbasiertes SSO“ [Mehl05a, S. 138] und „Gatewaybasiertes SSO“ [Mehl05a, S. 140], die an dieser Stelle aus Platzgründen nicht weiter vertieft werden sollen.

Steel et al. haben für die J2EE-Architektur ein Single Sign On Delegator Pattern vorgestellt. Das Pattern ist zwischen Client und Servicekomponenten des Identity Managements angesiedelt. Es sorgt für eine Entkopplung und kapselt Details der Dienstaufrufe [Ste+05, S. 777].

7.4 Autorisierung

Zugriffskontrollmuster haben die Aufgabe, bei Zugriff von Subjekten auf schutzwürdige Objekte die Berechtigung zu prüfen [Mehl05a, S. 100]. Die Art und Weise, wie die Kontrolle erfolgt, kann sich unterscheiden und wird im Zugriffskontrollmodell beschrieben. Im Hinblick auf den Grad der Abstraktion sind diese Muster der Architekturebene zuzuordnen.

Basis für Autorisierungsmuster ist nach einem Vorschlag von Mehlau die generische Zugriffskontrolle. Eine der mögglichen Spezialisierungen ist die rollenbasierte Zugriffskontrolle RBAC (Role Based Access Control). Für diese Spezialisierung werden Rechte in Rollen gebündelt, wobei ein Subjekt mehrere Rollen haben kann [Fer+03]. Eine weitere Spezialisierung stellt die metadatenbasierte Zugriffskontrolle dar, bei der die Attribute der Subjekte als Basis für die Entscheidung verwendet werden [Mehl05a, S. 101].

7.4.1 Generische Zugriffskontrolle

Mustername:

Generische Zugriffskontrolle [Mehl05a, S. 101]

Kontext:

Kontrolle der Nutzung von IT-Ressourcen durch personelle oder maschinelle Aufgabenträger in einem IT-System.

Problemabschnitt:

Die erlaubten Zugriffe von Subjekten (z.B. Personen, Programmagenten) auf zu schützende Objekte (etwa Dateien oder Datenbanktabellen) müssen beschrieben sein. Es sollen Kontextinformationen wie etwa die Tageszeit berücksichtigt werden. Die Delegation von Rechten soll ermöglicht werden. Durch das Zugriffskontrollmodell soll eine effiziente und konsistente Rechteadministration unterstützt werden.

Lösungsabschnitt:

Generelle Form: Alle Zugriffe von Subjekten auf Objekte werden durch einen Systemmonitor kontrolliert. Ein Recht besteht in der Autorisierung eines Aufrufs einer bestimmten Funktion auf einem Objekt [Mehl05a, S. 102].

Konsequenzabschnitt:

Nur mit einer entsprechenden Autorisierung kann auf Objekte zugegriffen werden. Zwischen der Lösung und der zu schützenden Ressource besteht keine technische Abhängigkeit. Welche Zugriffsarten neben Lese- und Schreibzugriffen möglich sind, wird individuell auf Systemebene definiert.

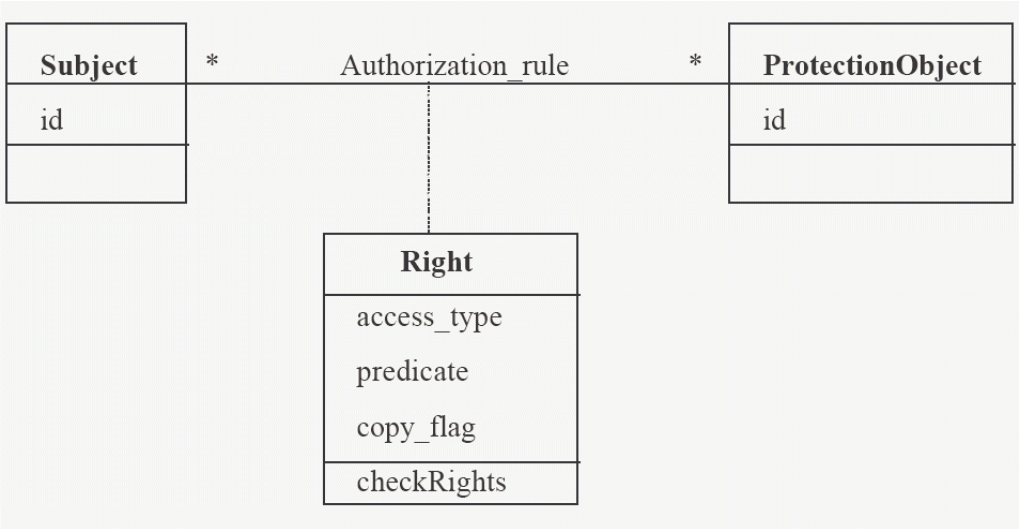

Abbildung 13: Authorization Pattern nach Fernandez und Pan [FePa01, S. 3]

Abhängigkeiten:

Das Muster stellt eine Generalisierung der spezifischen Implementierungsarten in Form der Muster „Rollenbasierte Zugriffskontrolle“, „Discretionary Access Control“ [Mehl05a, S. 102] und „Metadatenbasierte Zugriffskontrolle“ dar.

Die Spezialisierung der „Rollenbasierte Zugriffskontrolle“ RBAC wird direkt nach dem Sitzungsmuster dargestellt, da gerade die rollenbasierte Bündelung von Rechten in Unternehmen sehr häufig eingesetzt wird. Zunächst wird anschließend das Sitzungsmuster erläutert, auf dem das RBAC-Muster in Teilen auch beruht.

7.4.2 Sitzung

Mustername:

Sitzungsmuster [Mehl05a, S. 103]; [Sch+06, S.297]; [Ste+05, S.686]; [YoBa97, S. 14]; [Pri+04, S. 241]

Kontext:

Kontrolle der Nutzung von IT-Ressourcen durch personelle oder maschinelle Aufgabenträger in einem IT-System durch eine Umgebung mit beschränkbaren und kontrollierbaren Rechten.

Problemabschnitt:

Je nach Kontext nutzen Anwender oder Programme in der Regel nur einige der ihnen zugeordneten Rechte. Angreifer können bei unberechtigtem Zugriff ggf. alle Rechte nutzen. Bei berechtigten Usern besteht die Gefahr der Fehlbedienung. Ziel ist eine Minimierung des Schadens bei unberechtigter Nutzung oder berechtigtem, aber unbeabsichtigtem Fehlverhalten.

Lösungsabschnitt:

Generelle Form: Die Sitzungsverwaltung basiert auf den Klassen „Subjekt“ (Person, Programm), „Credential“ (den Subjekten zugeordnete, gekapselte Rechte) und die Klasse Sitzung [Mehl05a, S. 104]. Je Sitzung werden nur eine Teilmenge der dem Subjekt zugeordneten Rechte genutzt.

Konsequenzabschnitt:

Mit diesem Muster kommt das Prinzip „Least Privilege“ zur Anwendung. Da nur die jeweils benötigten Rechte genutzt werden können, ist bei Mißbrauch oder Fehlverhalten das Risiko verringert.

Abhängigkeiten:

Das Sicherheitsmuster „Sitzung“ kommt beim Muster „Rollenbasierte Zugriffskontrolle“ zum Einsatz.

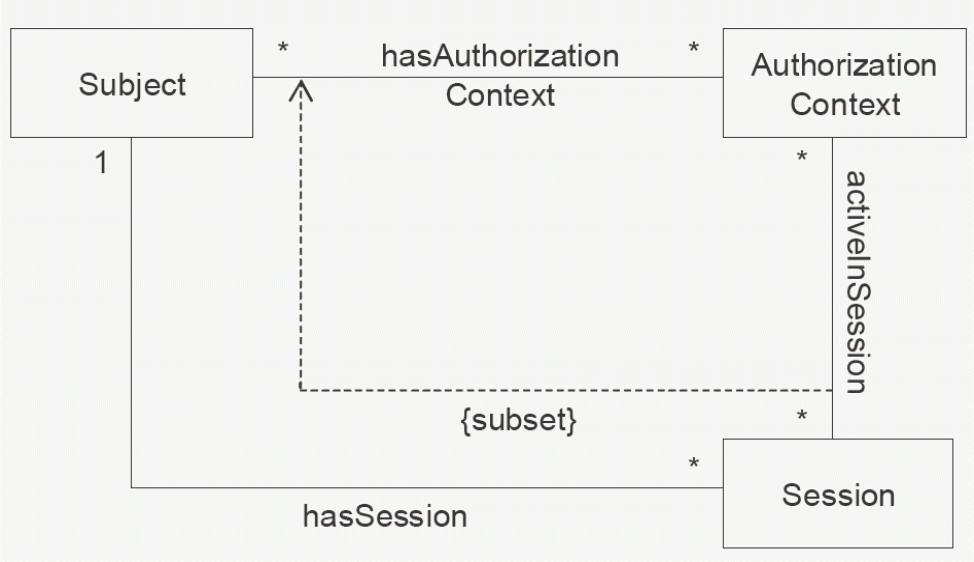

Abbildung 14: Session Pattern [Pri+04, S. 241]

7.4.3 Rollenbasierte Zugriffskontrolle (RBAC)

Mustername:

Rollenbasierte Zugriffskontrolle / Role Based Access Control RBAC [FePa01]; [Mehl05a, S. 105]; [Mas+04, S. 293]; [Pri+04, S. 8 / 243]; [Sch+06, S. 249]

Kontext:

Entspricht dem Kontext der „Generischen Zugriffskontrolle“. Es soll eine große Zahl von Nutzern zentral mit geringstmöglichem Aufwand administriert werden.

Problemabschnitt:

Auch hier entspricht der Problemabschnitt in Teilen dem der „Generischen Zugriffskontrolle“. Darüber hinaus ist die Administration von Rechten für eine große Zahl von Subjekten und Objekten unter Wahrung der Konsistenz sicherzustellen. Schließlich ist auch das „Least-Privileges“-Prinzip zu unterstützen. Ziel ist es, den Zugriff auf Ressourcen auf Basis der Rolle(n) eines Subjekts zu kontrollieren.

Lösungsabschnitt:

In der „Rollenbasierten Zugriffskontrolle“ werden die Muster „Generische Zugriffskontrolle“ und „Sitzung“ miteinander kombiniert. Die Klasse „Rolle“ tritt an die Stelle einer direkten Verbindung von Subjekt und Objekt, durch sie werden bspw. organisatorische Aufgabenprofile modelliert.

Subjekte können mehrere Rollen besitzen, die Berechtigungen sind mit den Rollen verknüpft. Ein Recht wird ebenfalls als Klasse modelliert. Das Teilmuster „Sitzung“ ermöglicht die Umsetzung des „Least Privileges“-Prinzips. Das Muster „Sitzung“ limitiert die Ausübung der Rechte auf diejenigen, die aktuell benötigt werden [Mehl05a, S. 106].

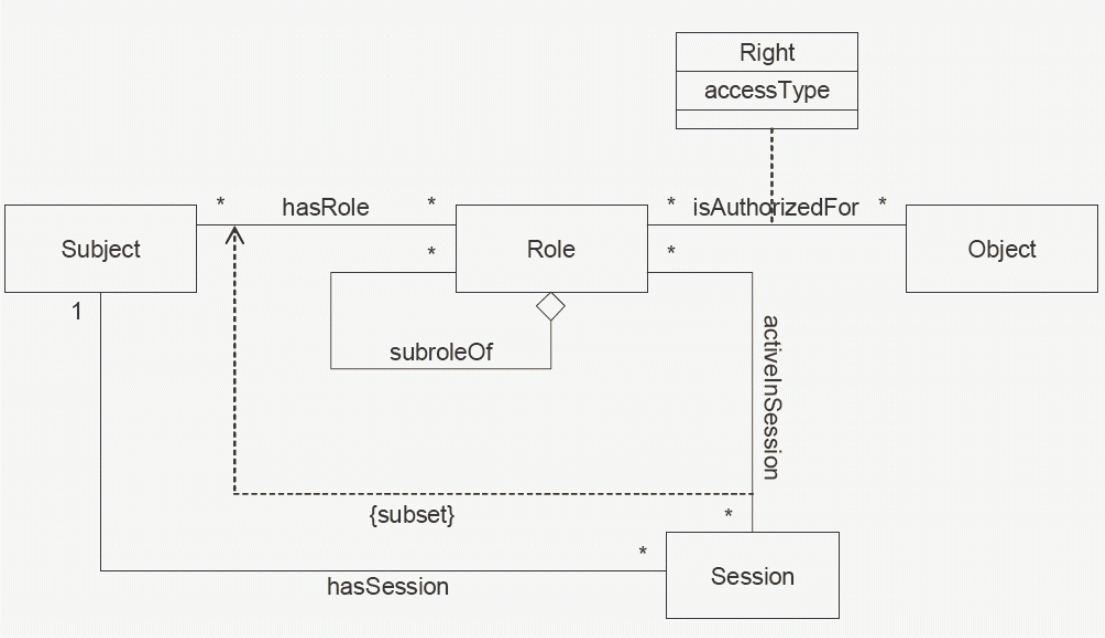

Abbildung 15: Pattern for Role Based Access Control (RBAC) nach Priebe et al. [Pri+04, S. 8 / 243]

Konsequenzabschnitt:

Allgemeine Aspekte ergeben sich bereits aus dem Konsequenzabschnitt der „Generischen Zugriffskontrolle“. Durch das Konzept der Rollen kann eine Reduktion des Administrationsaufwands erreicht werden, da in vielen Organisationen viele Personen gleichartige Aufgaben ausüben und daher auch gleiche Rechte benötigen, die zu Rollen gebündelt werden können. Nur diesen müssen eine entsprechend geringere Anzahl von Rechten zugeordnet werden.

Die Kontrolle der Vergabe der Rechte liegt zentral bei der jeweiligen Organisation, es können aber auch Rechte durch die Subjekte delegiert werden. Das Prinzip der benötigten Rechte („Least Privilege“) wird unterstützt.

Abhängigkeiten:

Das Sicherheitsmuster „Rollenbasierte Zugriffskontrolle“ ist ein Spezialfall des Musters „Generische Zugriffskontrolle“.

7.4.4 Discretionary access control (DAC)

Mustername:

Discretionary access control [Mehl05a, S. 101]

Kontext:

Gleicht dem Kontext der generischen Zugriffskontrolle. Das Muster soll es ermöglichen, dass Rechteinhaber Dritten selbständig Zugriffsrechte einräumen können (Delegation).

Problemabschnitt:

Der Problemabschnitt entspricht dem bereits oben aufgeführten Abschnitt bei „Generischer Zugriffskontrolle“.

Lösung

Generelle Form: Die Elemente entsprechen denjenigen des Muster Authenticators. Mit der Klasse „Subjekt“ wird die zugreifenden Entität beschrieben. Die zu schützende Ressource wird als Klasse „Objekt“ modelliert. Der Besitzer verfügt über sämtliche Rechte, die delegiert werden können. Ein Recht wird als assoziierte Klasse abgebildet. Hierüber werden die Zugriffsarten, der Kontext und die Möglichkeit der Weitergabe definiert. Es steht eine Operation „prüfeRechte“ zur Verfügung, die sowohl vom Subjekt als auch vom Objekt genutzt werden kann.

Einsatzgebiete: Diese Form der Zugriffskontrolle wird durch kommerzielle Produkte bspw. das Betriebssystem Microsoft Windows XP implementiert, allerdings noch um weitere Zugriffsmodelle ergänzt [Mehl05a, S. 103].

Konsequenzabschnitt:

Die Aussagen bzgl. der generischen Zugriffskontrolle treffen auch hier zu. Über das Attribut „Kontext“ ist es möglich, Zusammenhänge zum Zeitpunkt der Ausführung auszuwerten, bspw. die Uhrzeit des Zugriffs.

Die Vergabe und der Entzug von Rechten ist den Subjekten zugeordnet, ohne dass ein Administrator hinzugezogen werden müsste [Mehl05a, S. 103]. Dadurch ist jedoch auch kein spontaner Rechtestatus möglich.

Ein weiteres Problem ergibt sich innerhalb von Organisationen. Scheidet ein Mitarbeiter aus, ist eine Neuzuweisung der Rechte an neue Besitzer erforderlich. Wird dies versäumt, besteht ggf. kein Zugriff mehr auf bestimmte Objekte.

Abhängigkeit:

Das Sicherheitsmuster Discretionary Access ist eine Spezialisierung des Musters „Generische Zugriffskontrolle“.

7.4.5 Metadatenbasierte Zugriffskontrolle (MBAC)

Mustername:

Metadatenbasierte Zugriffskontrolle MBAC [Mehl05a, S. 106ff]

Kontext:

In Ergänzung des Kontextes der generischen Zugriffskontrolle ist eine sehr große Zahl von Anwendern bei hoher Fluktuation und bei einer sehr dynamischen Umgebung zu berücksichtigen.

Problemabschnitt:

Der Problemabschnitt für dieses Muster entspricht im wesentlichen dem der „Generischen Zugriffskontrolle“. Insbesondere wird aber der Aspekt der Effizienz der Administration besonders betont. In einem dynamischen Umfeld ist es denkbar, dass Rollen nicht mehr aktuell zu halten sind. Daher sollen Berechtigungen bei diesem Muster an fachliche Aspekte wie Attribute von Subjekten oder Objekten gebunden werden.

Lösungsabschnitt:

Generelle Form: Die Autorisierung erfolgt auf der Basis eines Abgleichs von Subjekt- und Objektattributen, durch die sowohl Subjekte und Objekte bestimmten Merkmalsgruppen zugeordnet werden können.. Letztlich entspricht dies der Bildung von Gruppen. Über die Gruppenzugehörigkeit entscheidet in erster Linie das Vorhandensein einer Merkmalskombination. Den Gruppen können wiederum Rechte zugewiesen werden. Damit werden die Rechte zwischen den Beschreibungen und nicht direkt zwischen den Subjekten und Objekten definiert.

Bekannte Einsatzgebiete: Zur Zeit sind hierfür keine Implementierungen bekannt [Mehl05a, S. 108].

Konsequenzabschnitt:

Auch hier gelten die Aussagen des Abschnitts zu den Konsequenzen der „Generischen Zugriffskontrolle“. Das Modell unterstützt eine vereinfachte, zentrale Vergabe und Entziehung von Rechten und bietet ein hohes Maß an Flexibilität [Mehl05a, S. 108].

Ändern sich Eigenschaften von Objekten und Subjekten, ändern sich auch die Zugriffsmöglichkeiten. Da jedoch ein Mustervergleich und ein Mengenvergleich ansteht, ist das Laufzeitverhalten nicht unbedingt optimal [Mehl05a, S. 109].

Abhängigkeiten:
Die Metadaten basierte Zugriffskontrolle MBAC stellt eine Spezialisierung der „generischen Zugriffskontrolle“ dar. Sie kann durch die Kombination mit dem Sitzungsmuster zur „Metadatenbasierten Zugriffskontrolle mit Sitzungskonzept“ spezialisiert werden [Mehl 05a, S. 109].

7.4.6 Weitere Autorisierungsmuster

Authorization Enforcer:
Bei den Pattern Authorization Enforcer von Steel et al. sowie Authorization Pattern von Schumacher et al. handelt es sich um Klassen zur Bearbeitung der Autorisierung von (http) Requests [Ste+05, S. 548]; [Schumacher 06, pp.245].

Secure Base Action:
Dieses Pattern zentralisiert und koordiniert sicherheitsrelevante Aufgaben in der Präsentationsschicht der Ebeneneinteilung von Steel et al. Secure Base Action steuert die Aktivitäten der Pattern Authentication, Authorization Enforcer und weiterer Pattern [Ste+05, S. 469].

Authorization Pattern:
Das Authorization Pattern von Fernandez und Pan beschreibt, welche Subjekte autorisiert sind, auf die Ressourcen eines Systems zuzugreifen [FePa01].

Access matrix
Durch das Muster Access Matrix wird ein Basismodell für Zugriffskontrolle spezifiziert. Es kann bspw. durch ein Role Based Access Control Modell oder einem Multilevel-Modell ersetzt werden [Sch+06, S.]. Das Multilevel Security Pattern von Fernandez und Pan stellt einen Mechanismus für das Zugriffsmanagement bereit [FePa01, S. 7]

Remote Authenticator /Authorizer
Dieses Pattern von Fernandez und Warrier stellt eine Anwendung des Single Access Point dar [FeWa03, S. 7] [YoBa97]. Es verwendet außerdem das Proxy Pattern als Komponente [Gam+94]. Die Benutzerrechte können unter Verwendung eines Role-Based Access Control Modells definiert werden [FePa01]; [YoBa97].

Reference Monitor Pattern

Das von Fernandez vorgestellte Reference Monitor Pattern dient ebenfalls der Autorisierung, wenn Prozessanfragen nach Ressourcen zu bearbeiten sind [Fern02].

7.5 Provisioning

Generell gibt es beim Provisioning verschiedene Varianten der Implementierung. Es gibt eine zentralisierte Variante, bei der vorgesehen ist, dass das Zielsystem in der Lage ist, in der Service Provisioning Markup Language SPML erstellte Requests zu verarbeiten. Die Requests stammen von einer zentralen Stelle, die als führendes System (Master) Anforderungen für Neuanlage, Änderungen oder Löschungen von Benutzerdaten an die Zielsysteme verschickt. Die Ablage der Identity-Informationen erfolgt repliziert und synchronisiert.

In der dezentralen Variante existiert ein führendes Verzeichnis. Auf Basis der Daten aus dieser Quelle werden Änderungsrequests etc. von einem zentralen Punkt aus versandt. Diese Anforderungen werden dann in den abhängigen Identity-Repositories umgesetzt.

Steel et al. stellen in ihrem Buch Core Security Pattern ein Pattern für Provisioning im Rahmen des J2EE-Architekturmodells vor, das Service Provisioning Pattern [Ste+05, S. 840].

Password Synchronizer Pattern [Ste+05, S. 840]

Das Password Synchronizer Pattern synchronisiert Benutzercredentials und Passworte über verschiedene Anwendungen hinweg [Ste+05, S. 840].

Problem:

In einer Anwendungslandschaft verwenden verschiedene Anwendungssysteme in der Regel eigene oder zumindest verschiedene Mechanismen zur Verwaltung von Benutzeraccounts. Das bringt erheblichen Aufwand und Kosten in der Administration von Veränderungen mit sich. Händische Administration ist überdies fehleranfällig. Durch Standardisierung und Zentralisierung kann das Problem gelöst werden.

Lösung:

Durch einen Password Synchronizer kann die Synchronisierung von Benutzeridentitätsdaten wie Passworte über verschiedene Applikationen hinweg zentralisiert werden [Ste+05, S. 842]. Der Password Snychronizer versendet Servicekommandos für Benutzeraccounts wie Passwortsetting, Rücksetzungen oder Änderungen. Die versendeten Requests werden

durch die empfangenden Systeme auf Echtheit geprüft und umgesetzt. Requests können geloggt werden, bei Verfügbarkeitsproblemen des Zielsystems werden sie erneut gesendet.

Struktur:

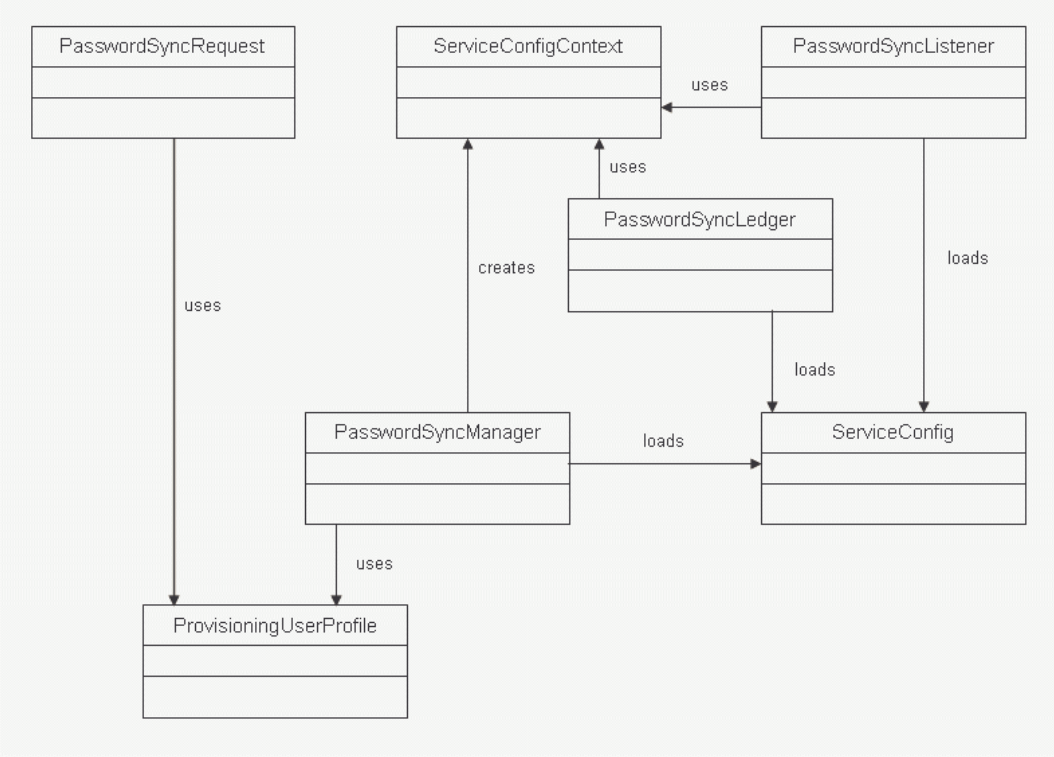

Abbildung 16: Service Provisioning Security Pattern nach Steel et al. [Ste+05, S. 844]

Der Kern des Passwortsynchronisierungsdienstes besteht aus vier Hauptklassen: PasswordSyncManager, ServiceConfig, PasswordSyncListener und PasswordSyncLedger. Die Klasse PasswordSyncManager ist für die Kernprozesse verantwortlich. Ein Request wird durch die Klasse PasswordSyncRequest erzeugt, die mit Hilfe der Klasse ProvisioningUserProfile ein Userprofil lädt [Ste+05, S. 844]. Der PasswordSyncManager erzeugt eine Sitzung, nimmt Verbindung mit den Zielsystemen auf und gibt einen Request aus. ServiceConfig sorgt u.a. für das Mapping der Benutzer-IDs und den unterschiedlichen Protokollen der Zielsysteme. Die Klasse PasswordSyncListener steht für die empfangenden Zielsysteme [Ste+05, S. 845]. Durch PasswordSyncLedger wird das Monitoring bzw. Logging der Abwicklung eines Requests zur Verfügung gestellt.

Konsequenzen:

Zum Nutzen dieser Lösung gehört die Möglichkeit, eine Schnittstelle für die Änderungen an Passworten nutzen zu können, die verschiedene Protokolle in gekapselter Form bereithält [Ste+05, S. 850]. Außerdem werden automatische Wiederholungen bei Nichtverfüg-

barkeit der Zielsysteme und eine sichere Speicherung von ID-Mappinginformationen, wie z.B. Secure LDAP, unterstützt [Ste+05, S. 851].

Realitätsprüfung:
Passwortsynchronisierungsfunktionen sollten nicht unbedingt selbst erstellt werden, da es ausreichende Anbieter am Markt gibt, die out-of-the-Box-Lösungen anbieten, darunter auch die OpenSPML Initiative [Ste+05, S. 890].

Beziehungen zu anderen Pattern:
Pattern, die zum eben beschriebenen Muster in Beziehung stehen, sind der „Single-Sign-On-Delegator“ und „BusinessDelegate“ [Ste+05, S. 890 / 891].

7.6 Identity Provider Pattern

Das von Delessy, Fernandez und Larrondo-Petrie entwickelte Identity Provider Pattern zentralisiert die Administration der Subjekte einer Sicherheitsdomäne [Del+07, S.34].

Kontext:
Den Kontext bildet eine Anwendungslandschaft einer Organisation mit mehreren Ressourcen wie z.B. Web Services, CORBA Services und Applikationen, auf die durch eine Menge von Subjekten zugegriffen wird.

Problem:
Dienste und Applikationen implementieren ggf. jeweils ihre Benutzerverwaltung. Das erhöht den Aufwand für die Erstellung und für die Wartung. Schließlich erhöhen die zusätzlichen Administrationsvorgänge, zumal wenn sie manuell durchgeführt warden, die Wahrscheinlichkeit von Fehlern und beeinträchtigen damit die Konsistenz über die verschiedenen Einheiten einer Organisation hinweg.

Lösung:
Die Verwaltung von Identitätsdaten einer Organisation wird beim Identity Provider zentralisiert, der für die Speicherung und auch die Verteilung (Propagierung) von Teilen der Benutzerinformationen zu den angebundenen Applikationen zuständig ist. In einem Teil der Aufgaben besteht eine Überschneidung zu denjenigen des Provisioning Pattern. Unter einer Sicherheitsdomäne ist in diesem Zusammenhang ein Set von Ressourcen zu verstehen, deren Nutzeridentitäten durch den Identity Provider verwaltet werden.

Das Pattern wurde für den Verwendungszusammenhang des Federated Identity Managements entwickelt, daraus erklärt sich, dass eine Sicherheitsdomäne bei diesem Pattern nur

ein Spezialfall eines Circle of Trust innerhalb einer Organisation ist. Das Identity Provider Pattern zentralisiert die Verwaltung der Subjekte der Sicherheitsdomäne [Del+07, S. 32].

Hierzu vergibt das Identity Provider Pattern an jeden Benutzer ein Set von Berechtigungsnachweisen (Credentials), die von den Ressourcen überprüft werden können. In der nachfolgenden Abbildung wird die Struktur der Lösung durch ein UML-Klassendiagramm veranschaulicht.

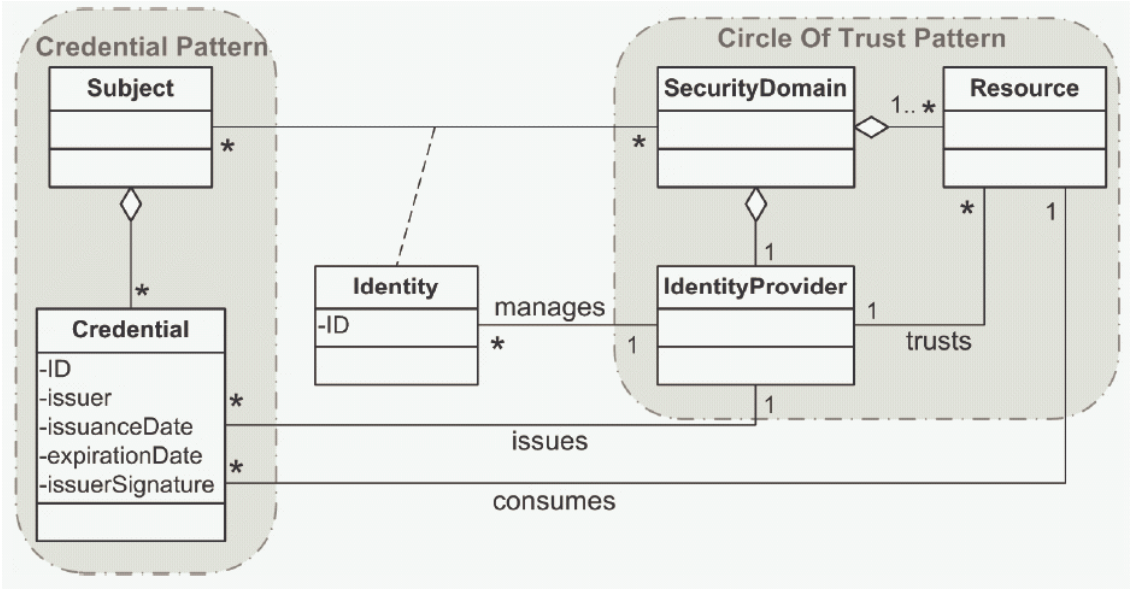

Abbildung 17: Identity Provider Pattern nach Delessy et al. [Del+07, S. 34]

Konsequenzen:

Mit dieser Lösung können die Betriebs- und Wartungskosten gesenkt werden, bei gleichzeitiger Wahrung der Konsistenz der Daten. Zu beachten ist, dass die Identity Provider Lösung so implementiert und ausgelegt ist, daß die Gefahr eines Engpasses im Betrieb der Organisation vermieden wird.

Bekannte Anwendungen:

Das Identity Provider Pattern kommt in Identity Management Produkten zum Einsatz, wie IBM Tivoli, Sun One Identity Server oder Netegrity's Siteminder [Del+07, S. 34].

Beziehungen zu anderen Pattern:

Das Identity Provider Pattern wird durch das Pattern "Identity Federation" verwendet. Außerdem besteht eine Verwendungsbeziehung zwischen dem Identity Provider Pattern und zwei von ihm genutzten Pattern, das „Credential Pattern" und das „Circle of Trust Pattern [Del+07, S.35].

7.7 Auditierungs- und Logfunktionen

Name:

Audit Interceptor [Ste+05, S. 624]

Problem:

Im Business-Tier (nach der Einteilung von Security Patterns durch Steel et al. [Ste+05, S. 624]) sind Requests und Antworten für Auditierungszwecke zu überwachen. Durch die Logaufzeichnungen wird die Prüfung auf Konformität mit den Policies einer Organisation oder nach Sicherheitsvorfällen ermöglicht. Hierzu sind geeignete Events und Nutzeraktionen aufzuzeichnen [Ste+05, S. 624].

Lösung:

Durch einen Audit Interceptor kann eine Auditierungsfunktion zentralisiert werden [Ste+06, S. 625]. Das Audit Interceptor Pattern ist für das Management des Log- und Audit-Prozesses im Back-End-Bereich zuständig. Durch dieses Pattern werden Requests und Antworten abgefangen und Audit-Ereignisse daraus erzeugt.

Dabei wird ein deklarativer Ansatz für die Definition der Auditierungsevents vorgeschlagen, damit Wartbarkeit und Anpassbarkeit gegeben sind. So wird es ermöglicht, dass sich verändernden Audit-Anforderungen angemessen berücksichtigt werden können, ohne dass die Applikation angepasst werden muss.

Struktur:

Wie auch dem UML-Klassendiagramm zu entnehmen ist, fängt der Audit Interceptor Requests ab, die vom Client an das Zielsystem abgesetzt werden. Anhand der Klasse AuditEventCatalog prüft der Audit Interceptor, ob ein Event aufgezeichnet werden sollte [Ste+06, S. 625].

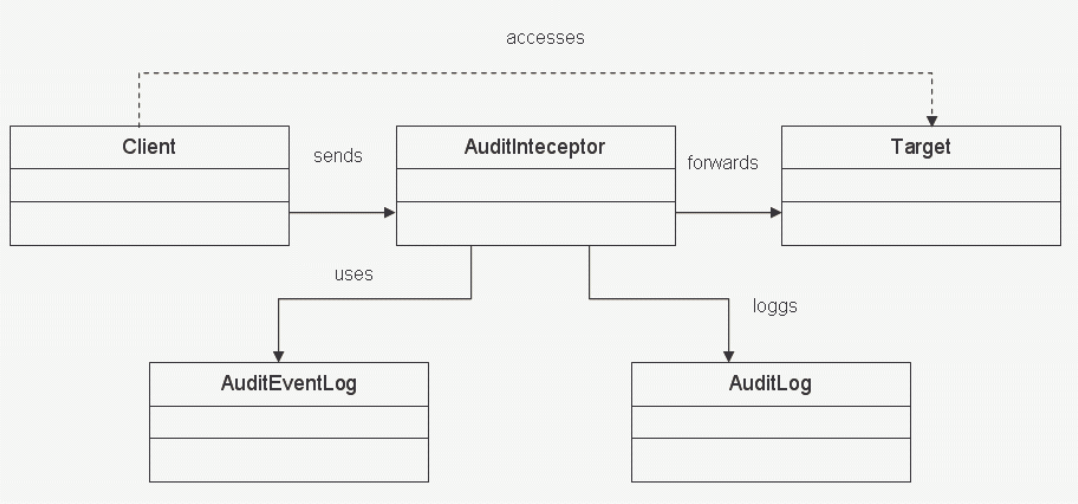

Abbildung 18: Audit Interceptor Class Diagram nach Steel et al. [Ste+05, S. 626]

Konsequenzen:

Auditfunktionen gehören zu den Schlüsselanforderungen für Applikationen der Kategorie "mission-critical". Durch die Auditfunktion werden Aufzeichnungen von Ereignissen erstellt, die für Überprüfungen erforderlich sind. Dadurch können keine gegenwärtigen, aber zukünftige Angriffe verhindert werden [Ste+05, S. 629].

Als Konsequenzen sind des weiteren zu nennen: Entkopplung der Auditfunktion von den eigentlichen Business Services, Unterstützung sich ändernder Eventanforderungen und –definitionen, aber auch Verringerung der Performance [Ste+05, S. 630].

Beziehungen zu anderen Pattern:

Der Audit Interceptor entspricht dem Pattern „Intercepting Filter" von Alur et al. [Alu+03], ist jedoch nicht ganz so komplex [Ste+05, S. 634]. Auch zum Pattern „Pipes and Filters" von Buschmann et al. besteht eine enge Beziehung [Bus+98]. Um bei Web Services Auditfunktionen einzusetzen, kann der Audit Interceptor durch das Pattern „Message Interceptor Gateway" verwendet werden [Ste+05, S. 634]. Darüber hinaus verweisen Steel et al zum Security event logging auf Romanosky [Ste+05, S. 462]; [Roma01, S. 8].

Secure Logger Pattern

Durch dieses Muster wird definiert, wie Ereignisse und Ausnahmen sicher und zuverlässig abgefangen werden könne, so dass sicheres Auditieren unterstützt wird [Ste+05, S. 469]. Das Secure Logger Pattern ist in der Einteilung von Steel et al. für Aspekte des Loggings im Front-End-Bereich zuständig. Mit dem Secure Logger Pattern werden Möglichkeiten der Implementierung für sichere Logfile-Einträge beschrieben, u. a. die Absicherung der zu loggenden Daten [Ste+05, S. 577].

7.8 Föderierungsdienste und Pattern für Web Services

Aufgrund des thematischen Fokus' dieser Arbeit und der Schwerpunktsetzung auf Aspekte des Enterprise Identity Management werden Pattern zu Themen wie Federated Identity Management und Web Services nur in den Grundzügen betrachtet. Gleichwohl sind sie bspw. für die Anbindung von Partnerunternehmen oder von Legacy Systemen von Relevanz.

Delessy, Fernandez und Larrondo-Petrie haben eine Mustersprache für Standards des Federated Identity Managements (z.B. Liberty Alliance) veröffentlicht [Del+07, S. 31]. Darin beschreiben sie unter anderem das bereits vorgestellte Identity Provider Pattern (s.o.), das die Administration der Subjekte einer Security-Domäne zentralisiert. Des weiteren wird durch das Circle of Trust Pattern die Bildung von Trust-Beziehungen unter Service Providern unterstützt. Dies ermöglicht die Föderierung multipler Identitäten über Organisationsgrenzen hinweg. Das SAML Assertion Pattern implementiert den Standard der Service Provisioning Markup Language, wodurch Identitätsinformationen zwischen Securitydomänen in einem einheitlichen Format ausgetauscht werden können [Del+07, S. 37].

Pattern für Web Services

Aus der Security Pattern Veröffentlichung von Steel et al. stammen drei Pattern, die Identity Management Funktionalitäten als Web Service bereitstellen [Ste+05, S. 477]. Dabei handelt es sich um das Muster Assertion Builder, den Credential Tokenizer sowie das SingleSignOn-Delegator-Pattern.

Assertion Builder

Durch das Muster Assertion Builder wird definiert, wie eine Identitätszusicherung (Assertion, z.B. nach der Security Assertion Markup Language SAML), für Zwecke der Authentisierung oder der Autorisierung aufgebaut ist. Das Muster bietet einen strukturierten und konsistenten Ansatz für die Verarbeitung von Benutzerdaten in Web Service Umgebungen [Ste+05, S. 756]. Im Muster wird die Kontrolllogik für die Verarbeitung von Assertion Statements gekapselt.

Credential Tokenizer

Durch dieses Pattern wird beschrieben, wie das Security Token eines Benutzers gekapselt und in eine SOAP-Nachricht eingebettet werden kann, damit Routing und Verarbeitung der Nachricht möglich werden [Ste+05, S. 477]. Ziel ist es, dass durch den Tokenizer verschiedene Arten von Credentials zu Token verarbeitet werden können, die bei unterschiedlichen Security Providern einsetzbar sind [Ste+05, S. 802].

Single Sign-on (SSO) Delegator Pattern

Dieses Muster kann bei der Anbindung von Legacy Systemen zum Einsatz kommen. Dadurch ist mit Hilfe dieses Pattern eine umfassende Nutzung von Single Sign On Funktionalitäten möglich. Hierzu wird die Security Assertion Markup Language SAML verwendet [Ste+05, S. 477]. [Ste+05, S. 776]

Security Assertion Coordination Pattern für WebServices

Das Security Assertion Coordination Pattern koordiniert Authentisierung und Autorisierung unter Verwendung eines Role -Based Access Control (RBAC) Modells beim Zugriff auf verteilte Ressourcen [Fern04, S. 2].

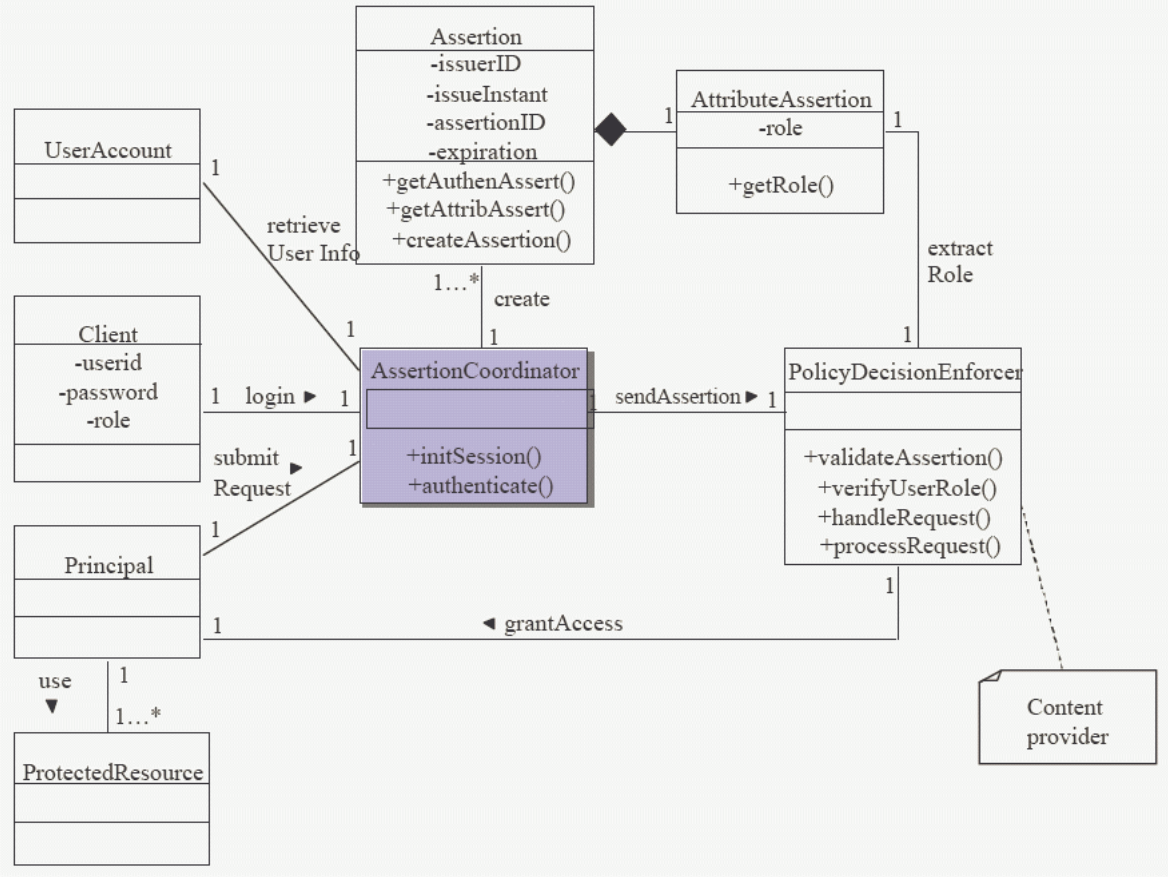

Abbildung 19: Security Assertion Coordination Pattern [Fern04, S. 4]

Message Inspector

Dieses Muster prüft die Qualität von Sicherheitsmechanismen auf Nachrichtenebene [Ste+05, S. 475]. Das umfasst sowohl Signaturen als auch Verschlüsselung in Verbindung mit Tokens. Dies gilt bspw. auch für die Überpürfung und Validierung von SOAP Nachrichten, die von mehreren beteiligten Parteien verarbeitet werden.

7.9 Weitere Security Pattern

Verzeichnisdienste und Directories

Das Thema Datenbankanbindung bzw. Verzeichnisdienste ist bislang nicht oder kaum im Bereich der Security Pattern repräsentiert. Fernandez verweist darauf, dass zukünftig noch Arbeit an Mustern für Datenbanken zu leisten ist [Fern07, S. 175].

8 Ein patternbasiertes Referenzmodell für Identity Management

Die im vorigen Kapitel vorgestellten Muster sollen jetzt in Form eines Sicherheitsmustersystems in den Ordnungsrahmen des Referenzmodells eingefügt werden. Ziel ist es dabei, Beziehungen zwischen den Pattern herauszuarbeiten und zu überprüfen, inwieweit das Ergebnis dem Sinn der in dieser Arbeit vorgestellten Referenzmodell-/ Patterninterpretation gerecht wird.

8.1 Annahmen und Beschreibung des Referenzmodells

Das hier vorgestellte Referenzmodell für die Domäne des Identity Management setzt sich aus dem in Kapitel 6 entwickelten Ordnungsrahmen mit Detaillierungen in Form von Pattern bzw. Patternsystemen zusammen. Konzeptionell besteht ein Zuordnungsverhältnis der Pattern zu Elementen bzw. Bereichen des Referenzmodells im Sinne der Interpretation nach Ferstl et al. [Fer+98]. Der Ansatz von Ferstl et al. stellt einen sinnvollen Vorschlag zur besseren Trennung der Konzepte dar. Die Autoren sehen bei Pattern gerade den Aspekt der Handlungsanleitung als dominierend an. Pattern können als gekapseltes Konstruktionswissen angesehen werden, das die Vorgehensweise bei der Erstellung von sicheren Systemen unterstützt. Aus den Überlegungen des 5. Kapitels zum Verhältnis von Pattern und Referenzmodellen ergibt sich auch, dass beide Konzepte sich in ihrem Scope, also in ihrer Reichweite bzw. ihrem Geltungsbereich unterscheiden.

Eine Annahme des Referenzmodells ist, dass eine Integration in ein Patternsystem innerhalb des Ordnungsrahmens möglich ist, obwohl die im Kapitel 7 aufgeführten Muster zum Teil sehr heterogen sind und auch auf verschiedenen Abstraktionsstufen operieren. Insofern werden sich einige dieser Pattern innerhalb des Systems daher nur mittelbar beeinflussen. Inhaltlich auf die Domäne bezogen wird z.B. angenommen, dass ein Subjekt und seine Identität bereits angelegt und vorhanden sind. Funktionen, die über diejenigen des Identity Provider Pattern von Delessy et al. hinausgehen, werden als gegeben angesehen [Del+07].

Das Referenzmodell und sein Ordnungsrahmen sind modular aufgebaut und können in mehrere Teilbereiche aufgeteilt werden, die wiederum durch verschiedene Bestandteile ausgeformt werden können. Der Ordnungsrahmen ordnet die abgebildeten Funktionen so an, dass zum einen die Abfolge im Prozess und zum anderen die Unterteilungen in Kategorien wie Kernfunktionen oder unterstützende Funktionen deutlich werden. Das Patternsystem ist demgegenüber feingranularer. Es unterstützt unter dem Gesichtspunkt der Navigation bzw. der Konfiguration die Auswahl von Pattern und berücksichtigt dabei die Bezüge zwischen den Security Pattern. Die Integration der Pattern in den Ordnungsrahmen wird auf einer mittleren Abstraktionsebene beschrieben, d.h. weniger abstrakt als der Ordnungsrahmen und nicht so detailliert wie auf der Ebene von Klassen einzelner Pattern.

Damit entspricht das Patternsysteme der Zielsetzung, Strukturierungshilfe für Teilprobleme zu bieten, indem mögliche Problemlösungskombinationen und ihrer Beziehungen dargestellt werden. Entlang der Beziehungen können Teillösungen durch Auswahl und Kombination einzelner Patterns erzeugt werden [Wolf01, S. 183].

Wendet man die aus dem Bereich der Referenzmodellierung bekannten Konfigurationsregeln auf die Patternbestandteile an, ergeben sich insbesondere bei alternativen Pattern oder Spezialisierungen Überschneidungen. Anhand der Patternbeziehungen können dann, je nach Kontext, geeignete Varianten ausgewählt werden.

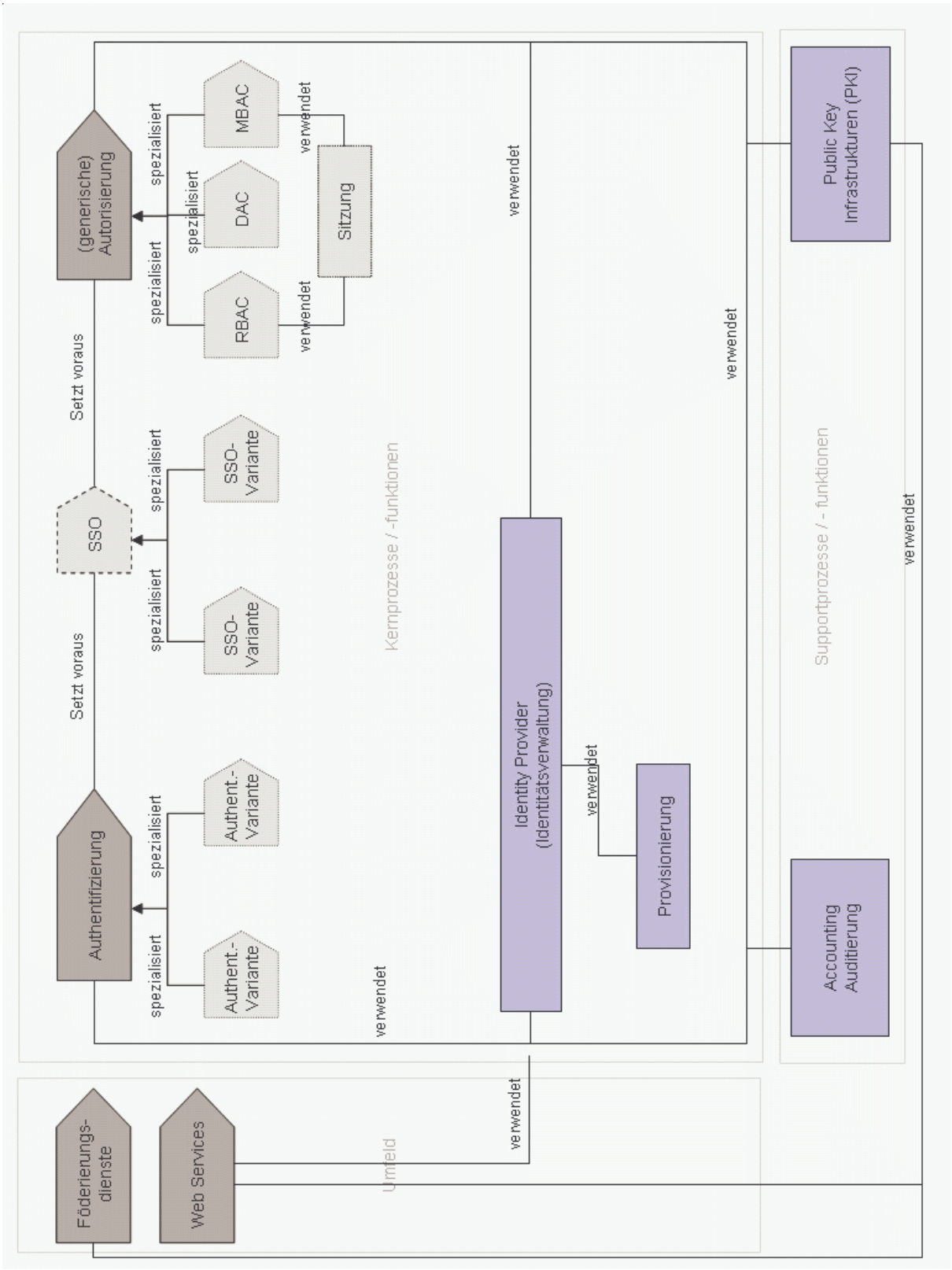

Abbildung 20: Patternsystem des Referenzmodells für Identity Management

In der Patternliteratur werden eine ganze Reihe von Beziehungsarten zwischen Pattern diskutiert. Als Beispiel für Abhängigkeiten unter Pattern weisen Yskout et al. darauf hin, dass das Muster Single Access Point für die Implementierung eines Authentication Enforcers Vorteile hat, der wiederum vom Vorhandensein eines Authorization Enforcer profi-

tiert [Ysk+ 06, S. 17]. Auch das korrekte Funktionieren des Audit Interceptor hängt vom Vorhandensein eines Secure Logger ab [Ysk+06, S. 16].

Yskout et al. schlagen als Arten der Beziehungen zwischen Pattern folgende Charakterisierungen vor: “depends”, “benefits”, “alternative”, „impairs” und „conflicts“. Im Hinblick auf eine Auswahl der Beziehungsarten hält Mehlau „Kombination“, „Spezialisierung, Konflikt und Beeinflussung für ausreichend [Mehl05a, S. 90]. Steel et al. haben ebenfalls Bezüge zwischen Pattern charkterisiert. Dies sind „Delegiert etwas“, „erzeugt“, „validiert“, verwendet und „ruft auf“. Delessy et al. nehmen folgende Position ein: Sie schlagen die Typen „enforces“, „uses“, „implemented as“ und „implements“ vor [Del+07, S. 32]

8.2 Vergleich mit anderen Patternsystemen und Referenzmodellen

Betrachtet man zum Vergleich andere existierende Patternsysteme, ist zu berücksichtigen, dass deren Ziele von der Schwerpunktsetzung dieser Arbeit abweichen. Mehrere Patternsysteme, bspw. das Hacker-Contest-Patternsystem nach Schumacher et al., das System der OpenGroup oder das Patternsystem von Mehlau erheben den Anspruch, mehrere IT-sicherheitsrelevante Funktionen in ihren Zusammenhängen abzubilden [Mehl05a, S. 65 / 97]. Funktionen des Identity Managements bilden in diesen Systemen nur einen Teil des abgedeckten Bereichs Security. Die größte Übereinstimung inhaltlicher Art besteht mit dem Sicherheitsmustersystem aus der Publikation von Steel et al [Ste+05, S. 479].

Die Übersicht von Steel et al. und die anderen Arbeiten werden in Sinne der Themenstellung dieser Arbeit betrachtet. Von Interesse ist neben dem Abstraktionsniveau und der Struktur auch, wie Zusammenhänge dargestellt werden und wo Auslassungen bestehen.

Beim Vergleich des Patternsystems des Hacker-Contests von Schumacher et al. ergibt sich, dass Muster teilweise unverbunden nebeneinander stehen. Im Schwerpunkt symbolisieren die Beziehungslinien, dass ein Muster ein anderes voraussetzt bzw. eine Spezialisierung darstellt, siehe auch [Mehl05a, S. 63].

Bei der Entwicklung seines Sicherheitsmustersystems hat Mehlau die Musterbeziehungen so ausgestaltet, dass diese sich bspw. gegenseitig verwenden, beeinflussen oder verfeinern können [Mehl05a, S. 97, 100]. Grundlage für Autorisationsmuster ist etwa das Sicherheitsmuster der generischen Zugriffskontrolle.

Wenn man den Patternsystementwurf von Steel et al. zum Vergleich heranzieht, kann man feststellen, dass dieser eine deutliche höhere Implementierungsnähe aufweist [Ste+05, S. 478]. Für Anwender ohne Expertise im Patternumfeld wird die Navigation jedoch eher

durch ein Abstraktionsniveau wie bspw. bei Mehlau oder Delessy et al. erleichtert, weil die Semantik klarer ist [Mehl05a, S. 63, 65, 100]; [Del+07, S. 32].

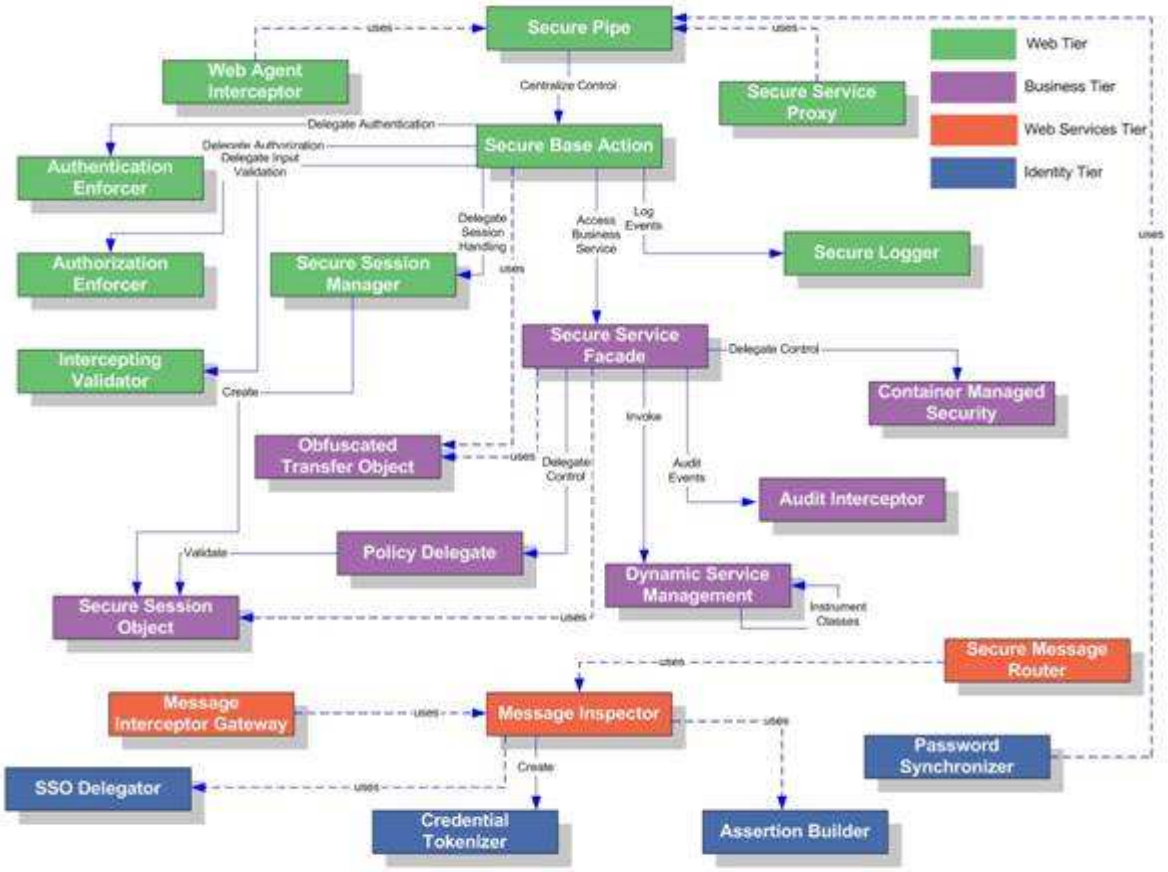

Abbildung 21: Sicherheitsmusterbeziehungen nach Steel et al. [Ste+05, S. 479]

Dies entspricht auch dem Standpunkt von Fernandez, Delessy et al., die Abbildungen auf mehreren Abstraktionsebenen an Anlehnung an den IT-Stack empfehlen [Del+07, S. 32]. Auf der obersten Ebene ist die Semantik klarer und vereinfacht die Navigation. Auf der bzw. den darunter liegenden Ebenen wird die Implementierung anhand von Pattern als Bausteine unterstützt.

Vergleicht man das in dieser Arbeit entwickelte Referenzmodell mit dem Vorschlag von Rottleb, dem Referenzmodell zur anwendungssystemübergreifend konsistenten Zugriffssteuerung, ergibt sich nur ein geringes Maß an Übereinstimmung, [Rott03, S. 122]. Auf der Ebene einzelner Pattern wie denen für Authentisierung, Autorisierung (z.B. Rollenbasierte Zugriffskontrolle, Role based Access Control (RBAC)), bestehen Übereinstimmungen z.B. in der Modellierung einiger Objekte gibt.

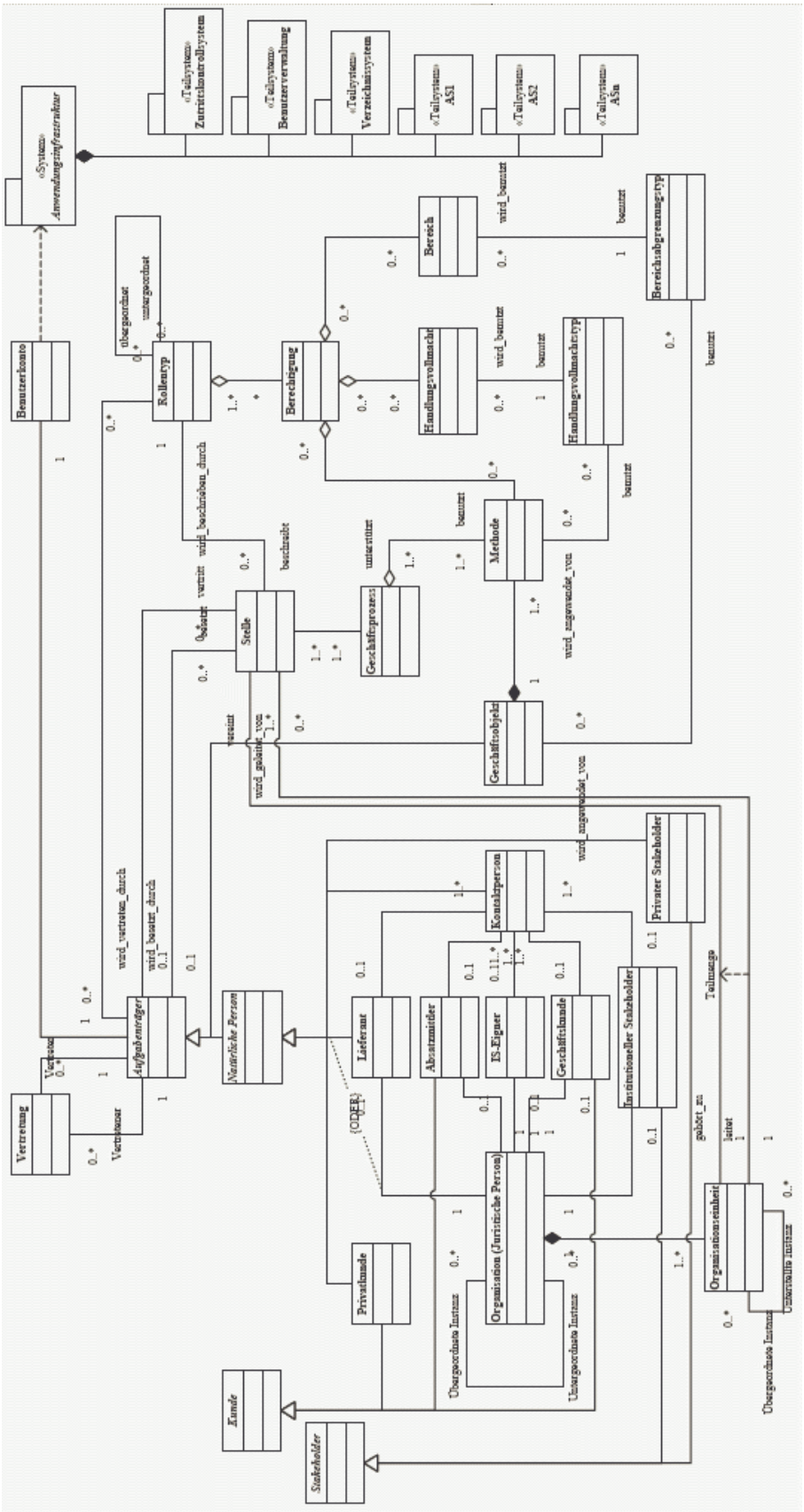

Abbildung 22: Referenzmodell zur anwendungssystemübergreifend konsistenten Zugriffssteuerung (MAKS) [Rott03, S. 122]

Im Ansatz von Rottleb steht schwerpunktmäßig die Modellierung der Objekte auf Fachkonzeptebene im Vordergrund. Er zieht für sein Modell bspw. Typen von Aufgabenträgern

(Benutzern), Organisationseinheiten, Stellen, Rollentypen, Geschäftsobjekte, Berechtigungen und Handlungsvollmachten heran.

Abweichungen gibt es in der Gestaltung der Beziehungen zwischen den Objekten, wobei diese natürlich nicht direkt verglichen werden können.

Rottleb verwendet, wie bereits in Kapitel 5 angemerkt, den Begriff Pattern oder Security Pattern nicht. Konzeptionell besteht aber eine gewissen Nähe, da er auf komponentenbasierte Softwareentwicklung verweist [Rott03, S. 148].

Das geringe Maß an Überschneidungen der beiden Modelle lässt keinen direkten Schluß zu, inwieweit nun Pattern in Elemente von Referenzmodellen überführt werden können. Deutlich wird jedoch, dass sie sich konzeptionell mit ihrem geringeren Scope und dem vorgehensorientierten Schwerpunkt von umfassenderen, fachkonzeptionell gelagerten Referenzmodellen unterscheiden.

8.3 Bewertung des Referenzmodells

Nach Becker et al. ist ein Referenzmodell vor seiner Publikation in seiner Gesamtheit abschließenden Tests oder einer Evaluation zu unterziehen. [Bec+02, S. 37]. Der für eine Erprobung und Evaluation erforderliche Aufwand würde den Rahmen dieser Arbeit jedoch sprengen. Zum jetzigen Zeitpunkt erfüllt das Referenzmodell damit die strengen Anforderungen des nutzungsorientierten Referenzmodellbegriffs nach Thomas nicht [Thom06, S. 23]. Anzumerken ist aber, dass es für einen nicht unerheblichen Teil der Pattern als Elemente des Referenzmodells durchaus Nachweise für eine Implementierung gibt. Die Bewertung beschränkt sich daher auf mit geringerem Aufwand prüfbaren Aspekte wie „Übereinstimmung mit den Grundsätze ordnungsmäßiger Modellierung“ und „Unterstützung bei der Patternauswahl“.

Für die Konstruktionsphase eines Referenzmodells sind drei der Grundsätze ordnungsmäßiger Modellierung von besonderer Bedeutung. Dies sind der Grundsatz der Richtigkeit, der Grundsatz der Relevanz und der Grundsatz des systematischen Aufbaus [Rose96, S. 98]). Bei diesen drei Grundsätzen ist eine Prüfung auf der Grundlage des Referenzmodell selbst möglich, ohne dass eine Evaluation anhand einer Umsetzung in spezifischen Modellen Voraussetzung wäre.

Der Grundsatz der Richtigkeit besagt, dass ein Modell syntaktisch und semantisch richtig ist. Auf Ebene der Klassenmodelle des vorliegenden Modells wird von einer grundsätzlichen Richtigkeit der Syntax ausgegangen. Die semantischen Richtigkeit ist stark von der

konkreten Auswahl der Pattern im Rahmen der Modellerstellung und einer späteren Modellüberprüfung abhängig.

Nach dem Grundsatz der Relevanz werden durch ein relevantes Modell die modellierungswerten Bestandteile der Diskurswelt angemessen priorisiert. Das Referenzmodell bildet das Feld des Identity Managements in ausreichender Breite ab. Grundlage hierfür waren die empirischen Quellen, deren Auswahl allerdings einem subjektiven Bias unterlag. Hier müssten ggf. Prüfungen durch Sicherheitsanalysen, Implementierung oder ein Experten-Review zeigen, ob diese Annahme zutreffend ist.

Schließlich fordert der Grundsatz des systematischen Aufbaus, dass bei einem Modell Sichten unterschieden werden und die Integration der Sichten sichergestellt wird. Das Modell weist noch Verbesserungsbedarf im Hinblick auf eine übergreifende Architektur auf, die sowohl Modellierungsergebnisse als auch das Vorgehen zur Erreichung dieser Ziele systematisch integriert. Für das komplexe Feld des Identity Managements und seine potentiell große Zahl an Informationsobjekten wird allerdings ein guter Überblick geliefert, womit Einarbeitungsaufwande für Anwender reduziert werden kann.

Ein Referenzmodell sollte Erfahrungswissen speichern. Dies tritt sowohl in Bezug auf die Modellierung der Gesamtdomäne zu, die empirisch basiert ist, als auch für die Detaillösungen, die auf bewährtem Erfahrungswissen aufbauen. In Bezug auf das integrierte Gesamt-Referenzmodell ist die Abbildung von Erfahrungswissen jedoch durch die fehlenden Anwendungen noch nicht gegeben.

Nach der Bewertung hinsichtlich der Grundsätze ordnungsmäßiger Modellierung wird nun noch die Frage erörtert, inwieweit das Referenzmodell die Auswahl von Pattern unterstützt. Meine Ausgangsposition war, dass Pattern vorgehens- und implementierungsorientierte Komponenten mit einem eingegrenzten Scope darstellen, die sich zur Detaillierung von Teilbereichen in umfassendere Referenzmodelle integrieren lassen. Dies gilt insbesondere für die Domäne des Identity Managements mit ihren zahlreichen sicherheitsrelevanten Teilbereichen.

Die Interpretation des Patternansatzes nach Ferstl et al. und die Bewertung der Qualität von Pattern nach Heyman et al. bieten eine Navigationshilfe bei der Patternauswahl, in dem beide eine Auswahl von Pattern nahe legen, die das Erreichen der Konstruktionsziele möglichst konkret und handlungsanleitend unterstützen [Fer+98]; [Hey+07].

Aus Sicht von Yskout et al. wird für die Unterstützung des Auswahlprozesses einzelner Pattern ein System aus Security Pattern benötigt [Ysk+06, S. 6]. Gerade hier besteht aber

noch Bedarf der Weiterentwicklung, da die Beziehungen noch unvollständig und nicht präzise genug abgebildet wurden. Teilweise sind sie mangels praktischer Erprobung auch noch nicht bekannt.

Problematisch ist, dass im Referenzmodell gegenwärtig die Auswahl geeigneter Pattern in hohem Maße dem Benutzer überlassen wird, ohne das das Gesamtmodell Stellung bezieht. Das vorliegende Referenzmodell nimmt zwar eine teilweise subjektiv begründete Auswahl von Pattern vor, bei alternativen Pattern oder Spezialisierungen fehlt es jedoch noch an einer weitergehenden Unterstützung des Nutzers. Damit läuft der Anwender Gefahr, dass die Beziehungen zwischen den Pattern ihre Geltung verlieren, weil sie sich auf ganz bestimmte Patternkombinationen beziehen. Streng genommen müsste je nach gewählter Kombination an Pattern sich auch das Verhältnis dynamisch ändern. Insoweit erfüllt das Referenzmodell die Forderung von Becker et al. nach einer Leitlinie für die Konfiguration von Elementen nur in Teilen [Bec+02, S. 48 / 65].

Auch bei der Ausgestaltung der Beziehungen zwischen Pattern bleiben Fragen offen. So können etwa zwei alternative Pattern ohne inhaltlich identisch zu sein, funktional als äquivalent angesehen werden [Ysk+06, S. 18]. Yskout et al. merken an, dass es schwierig ist, Identität in diesem Zusammenhang zu definieren, da Security Pattern nicht formal definiert sind. Die Autoren vertreten die pragmatische Auffassung, dass Security Pattern dann identisch sind, wenn sie versuchen, dasselbe Ziel auf dem gleichen Weg zu erreichen.

Die im Kapitel 5 formulierten Annahmen zum Verhältnis von Pattern zu Referenzmodellen“ haben sich, wenn man das integrierte Referenzmodell vorsichtig einschätzt, als tragfähig erwiesen. Zumindest haben sich in den im Kapitel 6 vollzogenen Entwicklungsschritten des Vorgehensmodells nach Becker et al. keine gegenteiligen Hinweise ergeben [Bec+02, S. 34ff].

Insgesamt kann man aus der Bewertung den Schluss ziehen, dass das vorliegende Referenzmodell des Identity Managements den bislang prüfbaren Teil der Anforderungen in den Grundzügen erfüllt, so dass eine weitere Überprüfung des Referenzmodells bspw. durch eine Ableitung spezifischer Modelle im Rahmen einer Softwareentwicklung sinnvoll erscheinen läßt. Als nützlich ist auch anzusehen, dass das Modell Informationsobjekte des Identity Managements umfassend darstellt, teilweise formalisiert und die Bedeutung vorgehensorientierter Patternbeschreibungen betont.

9 Schlußbetrachtung und Ausblick

Für Referenzmodelle gilt, dass sie nicht nur aus existierendem konkreten Modellen empirisch ermittelt, sondern auch auf der Basis theoretischer Erkenntnisse entworfen werden können. Um als Referenzmodell im strengen Sinne angesehen werden zu können, müssen sich derartige Entwürfe zunächst im Einsatz bewähren [Thom06, S. 23].

Betrachtet man die Phasen in der Entwicklung eines Referenzmodells, steht nach der Erstellung eine Evaluation anhand eines der geplanten Verwendungszwecke des Referenzmodells aus. Für das in dieser Arbeit vorgestellte Referenzmodell könnte eine konkrete Anwendung etwa in einer Sicherheitsanalyse bestehender Produkte oder Individuallösungen im Bereich des Identity Managements bestehen. Auch eine Neu- oder Weiterentwicklung existierender Identity Management Systeme käme in Frage.

Mit der Erprobung in der Anwendung ließe sich auch das von Thomas konstatierte Erkenntnisdefizit verringern, da es kaum empirische Studien zum Thema Referenzmodellierung gibt [Thom06, S. 157]. Gleichzeitig könnte dies auch der Forschung zur Anwendung von Security Pattern nützen, die bislang ebenfalls wenig oder heterogene Ergebnisse erbracht hat [Mehl05a, S. 53].

Das Referenzmodell in der hier vorgestellten Fassung deckt nicht alle durch den Ordnungsrahmen umrissenen Funktionen ab. Kandidaten für eine Weiterentwicklung sind z.B. solche Funktionen und Aufgaben, die in dem hier vorgestellten Referenzmodell bislang als gegeben vorausgesetzt werden. Z.B. geht das Referenzmodell von der Annahme aus, dass die Rechte, die bei der Zugriffskontrolle geprüft werden, bereits eingerichtet sind, dasselbe gilt für die eingesetzten Identitäten, die bereits etabliert sind.

Neben einer stärkeren Verknüpfung mit den Pattern für Federated Identity Management, wie Delessy et al. sie vorschlagen, könnte auch die Definition von Policies Gegenstand für weitere Pattern sein [Del+07]. Für die genannten Aspekte sind weitere Pattern oder Teilsysteme zu modellieren, was jedoch im Rahmen dieser Arbeit nicht näher betrachtet werden konnte.

Einige der in dieser Arbeit beschriebenen Pattern sind auf einen bestimmten Kontext hin ausgerichtet, wie eben die J2EE-Architektur. Dies stellt zum Teil eine Beschränkung dar, kann jedoch auch eine Erleichterung sein, wenn dies ohnehin dem gewählten Ansatz eines Projekts entspricht. Es wäre zu prüfen, ob ein Referenzmodell größeren Nutzen entfaltet, wenn es ausschließlich implementierungsneutrale Bestandteile anbietet, die erst auf den

Kontext eines Architekturmodells wie bspw. Dot.Net angepaßt werden müssen oder wenn es geeignete Pattern von vorneherein als Varianten beinhaltet.

Unabhängig davon sehen Heymann et al. Bedarf dafür, strenge Gütekriterien einzuführen, um so die Navigation in der unübersichtlicher werdenden Patternlandschaft zu erleichtern. Auch eine Verringerung der der Überlappung von Pattern wäre in diesem Zusammenhang wichtig [Hey+07, S. 4]. In Bezug auf den Abstraktionslevel von Pattern sehen Heymann et al. es als notwendig an, dass Muster eindeutiger positioniert werden [Hey+07, S. 7]. Dies ist eine Voraussetzung für eine verstärkte Untersuchung von Pattern im Hinblick auf ihre Beziehungen untereinander in Form von Patternsystemen [Hey+07, S. 7].

Bei der Entwicklung eines Sicherheitsmustersystems könnte es interessant sein, dass man Lösungen für Funktionsbereiche aggregiert auf abstrakter Ebene abbildet. Dies würde dem Ordnungsrahmen entsprechen. Hinzu müssten Detaillierungen auf den darunter liegendenden ein bis zwei Ebenen kommen, die Patternsysteme wiedergeben bzw. einzelne Klassen.

Mehrere Autoren, darunter Yskout et al. sehen Bedarf für Verbesserungen bei Methodiken zur Erstellung sicherer Software [Ysk+06, S. 7]. Heyman et al. fordern die Formulierung einer Methodologie für patternbasierten Softwareentwurf [Hey+07, S. 7]. Speziell in Bezug auf Pattern sollte ihres Erachtens eine unterstützende Methodologie die Designer durch eine Sequenz standardisierter und wiederholbarer Schritte bei der Auswahl von Pattern leiten [Ysk+06, S. 7]. Dies würde einem Referenzvorgehensmodell entsprechen [Beck04, S. 325]; [Bec+02, S. 36].

Fernandez et al. schlagen eine Methodologie zur Entwicklung sicherer Systeme unter Verwendung von Pattern vor [Fer+06]. Der Ansatz muss jedoch noch im Hinblick auf die Integration von Pattern präzisiert werden [Del+07, S. 37].

Der Grad an Unterstützung durch ein statisches „Papierreferenzmodell“ ist natürlich begrenzt. Für die Navigation zwischen Pattern oder generell Teilbereichen auf hohem und niedrigerem Abstraktionsniveau wäre eine Werkzeugunterstützung hilfreich. Ziel sollte es außerdem sein, die verwendeten Modellierungstechniken im Sinne einer methodischen Informationssystemarchitektur miteinander zu verbinden. [Bec+02, S. 48].

Hier wären Ansätze auf Basis von Modellierungs- und Verwaltungstools zu nennen, wie bspw. Brocke oder Thomas sie beim Referenzmodellmanagement verfolgen [Broc03]; [Thom06]. Dieser Weg wird teilweise auch im Bereich der Katalogisierung von Pattern beschritten [FeLo 02b, S. 26].

Die von Brocke vorgeschlagenen komponentenorientierten und modularen Referenzmodelle könnten ein weiterer Ansatzpunkt für die Verbesserungen der konzeptionellen Integration von Pattern und Referenzmodellen sein [Broc03, S. 351]. Diesbezüglich gilt es, eine Abbildung von Referenzmodellkomponenten auf Pattern zu untersuchen und ihre Bedingungen zu präzisieren. Denkbar wäre, dies anhand eventueller Bezüge zwischen einzelnen Pattern und Ausschnitten des Referenzmodells von Rottleb und anderer Referenzmodelle für Identity Management zu prüfen.

Interessant ist auch die Entwicklung, die sich in den Arbeiten von Emig et al. widerspiegelt [Emi+07]. Die Kapselung von Funktionen in Pattern, ihre Wiederverwendbarkeit und ihre inhärente Modularisierung lassen sie für virtualisierte Einsatzszenarien wie serviceorientierte Architekturen, unabhängig von einer stark abgegrenzten Umgebung eines Unternehmens recht gut geeignet erscheinen. Je nachdem, wie sich der Trend des verstärkten Einsatzes von Web Services entwickeln wird, sollte das Referenzmodell diesen aktuellen Erfordernissen angepasst werden, falls das Szenario eines Enterprise Identity Managements an Bedeutung verliert. Generell sind bei Änderungen an den Rahmenbedingungen durch Aufkommen neuer Technologien und Standards die enthaltenen Muster anzupassen, zu ergänzen oder auch zu ersetzen.

Die Eingangs dieser Arbeit gestellte Frage, welches das richtige Pattern für ein gegebenes Problem ist, konnte auf den zurückliegenden Seiten vielleicht nicht erschöpfend geklärt werden. Vielleicht ist aber deutlich geworden, wie dieses Problem der Praxis in einem komplexen Gebiet wie dem Identity Management zukünftig mit Hilfe einer Integration in den Referenzmodellansatz noch besser unterstützt werden kann.

Literaturverzeichnis

[Ale+77] ALEXANDER, Christopher; ISHIKAWA, Sara und SILVERSTEIN, Murray: *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press, 1977

[Alex79] ALEXANDER, Christopher: *The Timeless Way of Building*. New York: 1979

[Alu+03] ALUR, Deepak; CRUPI, John; MALKS, Dan: *Core J2EE Patterns. Best Practices and Design Strategies (Core)*. Upper Saddle River, N.J: Prentice Hall, 2003

[AvZd05] AVGERIOU, Paris; ZDUN, Uwe: Architectural Patterns revisited - A Pattern Language. In: *Proceedings of 10th European Conference on Pattern Languages of Programs (EuroPlop 2005)*. 1-39. http://www.infosys.tuwien.ac.at/Staff/zdun/publications/ArchPatterns.pdf Zugriff am 12.11.2006

[Baie05] BAIER, Tobias: *Persönliches digitales Identitätsmanagement Untersuchung und Entwicklung von Konzepten und Systemarchitekturen für die kontrollierte Selbstdarstellung in digitalen Netzen*. Universität Hamburg, Dissertation. 2005. http://www.sub.uni-hamburg.de/opus/volltexte/2006/2746/pdf/TBaier-Diss-IDM.pdf Zugriff am 11.11.2006

[Bas+05] BASIN, David; DOSER, Jürgen; LODDERSTEDT, Torsten: *Model Driven Security*. http://kisogawa.inf.ethz.ch/WebBIB/publications/papers/2005/marktoberdorf.pdf Zugriff am 12.11.2006 - ETH Zürich

[BeCu87] BECK, Kent; CUNNINGHAM, Ward: *Using Pattern Languages for Object-Oriented Programs*. Technical Report No. CR-87-43. 17. September 1987. Submitted to the OOPSLA-87 workshop on the Specification and Design for Object-Oriented Programming. http://c2.com/doc/oopsla87.html Zugriff am 27.03.2007

[Beck01] BECKER, Jörg: Referenzmodell. In: MERTENS, Peter (Haupt-Hrsg.). BACK, A.; BECKER, J.; KÖNIG, W.; KRALLMANN, H.; RIEGER, B.; SCHEER, A.-W.; SEIBT, D.; STAHLKNECHT, P.; STRUNZ, H.; THOME, R.; WEDEKIND, H. (Hrsg.): *Lexikon der Wirtschaftsinformatik*. Vierte Auflage. Berlin: Springer, 2001, S. 399 - 400

[Beck04] BECKER, Jörg: Referenzmodellierung - Aktuelle Methoden und Modelle. In: *WIRTSCHAFTSINFORMATIK* 46 (2004) 5, S. 325-326 http://www.wirtschaftsinformatik.de/dateien/beitraege/wi2004_5_325_326.pdf Zugriff am 17.08.2006

[Bec+02] BECKER, J.; DELFMANN, P.; KNACKSTEDT, R.; KUROPKA, D.: Konfigurative Referenzmodellierung. In: BECKER, J.; KNACKSTEDT, R. (Hrsg.): *Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem und Organisationsgestaltung*. Heidelberg: Physica-Verlag, 2002, S. 25-144

[Bec+06] BECKER, Jörg; DELFMANN, Patrick; RIEKE, Tobias: *RefMod06 – Wiederverwendung fachkonzeptioneller Softwaremodelle für kleine und mittlere Softwareunternehmen durch adaptive, komponentenorientierte Referenzmodellierung*. http://www.softwarefoerderung.de/projekte/refmod06/ beitrag_REFMOD06.pdf Zugriff am 03.12.2006 – Universität Münster

[Bec+00] BECKER, J.; HOLTEN, R.; KNACKSTEDT, R.; SCHÜTTE, Reinhard: Referenz-Informationsmodellierung. In: BODENDORF, F., GRAUER, M. (Hrsg.): *Verbundtagung Wirtschaftsinformatik* 2000. Aachen 2000, S. 86-109.

[BeSc96] Becker, J.; Schütte, R.: Handelsinformationssysteme. Landsberg/Lech, 1996

[Beed97a] BEEDLE, M.A.: Pattern *Based Reengineering*. URL: http://www.e-architects.com/users/beedlem/papers.html

[Beed97b] BEEDLE, M.A.: *cOOherentBPR – a Pattern Language to Build Agile Organizations*. URL: http://www.e-architects.com/users/beedlem/papers.html

[Bena06] Benantar, Messaoud: Access Control Systems - Security, Identity Management and Trust Models. Springer, 2006

[Bish02] BISHOP, Matt: *Computer Security: Art and Science*. Reading, Mass.: Addison-Wesley, 2002

[BlHe04] BLAKLEY, B.; HEATH, C. and Members of the Open Group Secu-rity Forum: *Security Design Patterns*. Open Group Technical Guide, 2004

[Bra+98] BRAGA, A.; RUBIRA, C.; DAHAB, R.: Tropyc: A Pattern Language for cryptographic Software. In: *Proceedings of the fifth Conference on Pattern Languages of Programming (PLoP '98)*. 1998

[Broc03] BROCKE, Jan vom: *Referenzmodellierung - Gestaltung und Verteilung von Konstruktionsprozessen*. Reihe: Advances in Information Systems and Management Science, Band 4, Berlin: Logos Verlag 2003. – Zugl.: Münster (Westfalen), Univ., Diss., 2002

[Broc04] BROCKE, Jan vom: Internetbasierte Referenzmodellierung - State-of-the-Art und Entwicklungsperspektiven. In: *Wirtschaftsinformatik* 46 (2004) 5, S. 390-404

[Bro+98] BROWN, W. J., R. MALVEAU, H. MCCORMICK III, T. MOWBRAY: *Anti Patterns*. Chichester: Wiley & Sons, 1998

[Bro+99] BROWN, F. Lee; DI VIETRI, J.; DIAZ DE VILLEGAS, G,; FERNANDEZ, E.: *The authenticator pattern*. In: Proceedings of the Sixth conference on pattern languages of programming (PLoP '99); 1999. http://jerry.cs.uiuc.edu/~plop/plop99/proceedings/Fernandez4/Authenticator3.PDF Zugriff am 03.05.2007

[Buc+07] BUCKL, S.; ERNST, A.; LANKES, J.; SCHNEIDER, K.; SCHWEDA, C.: A Pattern based Approach for constructing Enterprise Architecture Management Information Models. In: OBERWEIS, Andreas; WEINHARDT, Christof; GIMPEL, Henner; KOSCHMIDER, Agnes; PANKRATIUS, Victor; SCHNIZLER, Björn (Hrsg.): *eOrganisation: Service-, Prozess-, Market-Engineering – 8. Internationale Tagung Wirtschaftsinformatik*. Karlsruhe, Germany : Universitätsverlag Karlsruhe, Februar 2007. S. 145–162

[BuMe95] BUSCHMANN, F.; MEUNIER, R.: A System of Patterns. In: COPLIEN, J. O.; SCHMIDT, D. C. (Hrsg.): *Pattern Languages of Program Design*. Reading, Mass.: Addison-Wesley, 1995

[Bus+98] BUSCHMANN, F; MEUNIER, R.; ROHNERT, H.; SOMMERLAD, P.; STAL, M.: *Pattern-orientierte Software-Architektur. Ein Pattern-System*. Bonn et al.: Addison-Wesley, 1998

[Cas+03] CASASSA MONT, Marco; BRAMHALL, Pete ; PATO, Joe: *Adaptive Identity Management: The Next Generation of Identity Management Technologies*. Trusted Systems Laboratory, HP Laboratories Bristol HPL-2003-149 July 23rd , 2003* http://www.hpl.hp.com/techreports/2003/HPL-2003-149.pdf 27.12.2005 - HP Hewlett Packard

[CERT07] Aktuelle Statistik http://www.cert.org/stats/

[Copl92] COPLIEN, James O.: *Advanced C++ programming styles and idioms*. Reading, Mass.: Addison-Wesley, 1992

[Corf98] CORFMAN, Russel: An overview of patterns. In: RISING, Linda (Hrsg.): *The Patterns Handbook. Techniques, Strategies and Applications*. Cambridge: Cambridge University Press, 1998. S. 19-29

[Del+07] DELESSY, Nelly; FERNANDEZ, Eduardo B.; LARRONDO-PETRIE, Maria M.: "A Pattern Language for Identity Management," In: *Proceedings of the iccgi, International Multi-Conference on Computing in the Global Information Technology (ICCGI'07)*, 2007, S. 31 - 37.

[Emi+06] EMIG, Christian; SCHANDUA, Heiko; ABECK, Sebastian: SOA-aware Authorization Control. In: *Proceedings of the International Conference on Software Engineering Advances (ICSEA 2006)*, October 28 - November 2, 2006, Papeete, Tahiti, French Polynesia. IEEE Computer Society 2006. S. 62

[Empr06] EMPRISE: *BONAPART - Identity Management*. 2006 http://www.emprise.de/ steckbriefdownload.shtml? dbAlias=emprise&identifier=3128&version=1&content=file.pdf Zugriff am 15.07.2006

[Fern02] FERNANDEZ, E. B.: *Patterns for Operating Systems Access Control*. In: Proceedings of the 9th Conference on Pattern Languages of Programs, PLoP 2002, Allerton Park, Illinois, USA, 2002. http://jerry.cs.uiuc.edu/~plop/plop2002/proceedings.html Zugriff am 24.04.2007

[Fern04] FERNANDEZ, E. B.: "*Two patterns for web services security*". In: 2004 Intl. Symposium on Web Services and Applications (ISWS'04), Las Vegas, NV, June 21-24, 2004. http://www.cse.fau.edu/~ed/LasVegas3.pdf Zugriff am 11.11.2006

[Fern07] FERNANDEZ, Eduardo B.: *Security patterns and secure systems design using UML*. ACMSE 2007: 45th ACM Southeast Conference, Winston-Salem, North Carolina, USA March 23-24, 2007 http://acmse2007.wfu.edu/ materials/SecPattsACMSE07.ppt Zugriff am 02.04.2007]

[Fer+05] FERNANDEZ, E.B.; LARRONDO-PETRIE, M. M.; SELIYA, N.; DELESSY-GASSANT, N. and Schumacher, M.: *A pattern language for firewalls*.

http://www.cse.fau.edu/~ed/pubs.html_FirewallPatternv6.pdf Zugriff am 15.11.2006 – Florida Atlantic University / Siehe auch: [Sch+06]

[FeLa06] FERNÁNDEZ, Eduardo B.; LARRONDO PETRIE, María M.: Security Patterns and Secure Systems Design. In: *Proceedings of Fourth LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCET'2006)* 21-23 June 2006, Mayagüez, Puerto Rico

[Fer+06] FERNANDEZ, E. B.; LARRONDO-Petrie, M.M.; SORGENTE, T. and VANHILST, M.: A methodology to develop secure systems using patterns. In: MOURATIDIS, H. and GIORGINI, P. (Eds.): *Integrating Security and Software Engineering: Advances and Future Vision*. Hershey, PA: IDEA Press, 2006, Chapter 5

[FePa01] FERNANDEZ, Eduardo B.; PAN, Rouyi: *A pattern language for security models*. Procs. of PLoP 2001. http://jerry.cs.uiuc.edu/~plop/plop2001/ accepted_submissions/PLoP2001/ebfernandezandrpan0/PLoP2001_ebfernandezandrpan0_1.pdf Zugriff am 5.11.2006

[FeSi03] FERNANDEZ, Eduardo B.; SINIBALDI, J. C.: More patterns for operating systems access control. In: *Proceedings of the European Conference on Patterns Language of Programming (Euro-PLoP'03)*, 2003

[FeWa03] Fernandez, Eduardo B.; Warrier, Reghu: Remote Authenticator /Authorizer. Proceedings of PLoP 2003. http://jerry.cs.uiuc.edu/~plop/plop2003/Papers/Fernandez-remote-authenticator.pdf Zugriff am 10.05.2007

[Fer+03] FERRAIOLO, David F.; KUHN, D. Richard; CHANDRAMOULI, Ramaswamy: *Role-Based Access Control*. Norwood, MA: Artech House Inc., 2003

[Fer+98] FERSTL, O.K.; SINZ, E.J.; HAMMEL, C.; SCHLITT, M.; WOLF, St.; POPP, K.; PFISTER, A.: Wiederverwendbare und erweiterbare Geschäftsprozeß- und Anwendungssystem-Architekturen. In: PROJEKTTRÄGER INFORMATIONSTECHNIK DES BMBF BEIM DLR E.V.; GROTE, Ursula (Hrsg.): *Statusseminar des BMBF zur Softwaretechnologie*, Bonn, 1998.

[Fett01] FETTKE, Peter: *Eine Ordnungslehre für Informationsmodelle*. In: Doctoral Consortium im Vorfeld der WI-IF 2001 - Kolloquium für Doktoranden der Wirtschaftsinformatik - 18. September 2001, Schloß Reisensburg bei Günzburg. http://archiv.tu-chemnitz.de/pub/2001/0077/data/

2001_fettke_eine_ordnungslehre_fuer_informationsmodelle_dc_wi-if-2001.pdf Zugriff am 09.11.2006 - TU Chemnitz

[Fett06] FETTKE, Peter: *Referenzmodellevaluation. Konzeption der strukturalistischen Referenzmodellierung und Entfaltung ontologischer Gütekriterien*. Berlin: Logos-Verlag, 2006

[FeLo02a] FETTKE, Peter; LOOS, Peter: Der Referenzmodellkatalog als Instrument des Wissensmanagements: Methodik und Anwendung. In: BECKER, J.; KNACKSTEDT, R. (Hrsg.): *Wissensmanagement mit Referenzmodellen. Konzepte für die Anwendungssystem und Organisationsgestaltung*. Heidelberg: Physica-Verlag, 2002, S. 1 - 24

[FeLo02b] FETTKE, Peter; LOOS, Peter: Methoden zur Wiederverwendung von Referenzmodellen - Übersicht und Taxonomie. In: BECKER, Jörg; KNACKSTEDT, Ralf (Hrsg.): *Referenzmodellierung 2002. Methoden – Modelle – Erfahrungen*. http://www.wi.uni-muenster.de/inst/arbber/ab90.pdf Zugriff am 03.05.2007

[FeLo04] FETTKE, Peter; LOOS, Peter: Referenzmodellierungsforschung. In: *WIRTSCHAFTSINFORMATIK* Bd.: 46 (2004) Nr. 5, S. 331-340

[Fet+05] FETTKE, P.; LOOS, P.; ZWICKER, J.: Business Process Reference Models - Survey and Classification. In: Proceedings of the Workshop on Business Process Reference Models. In: KINDLER, E.; NÜTTGENS, M. (Hrsg.): *Business Process Reference Models (BPRM) - Proceedings of the Workshop on Business Process Reference Models (BPRM 2005)*, Satellite workshop of the Third International Conference on Business Process Management (BPM), Nancy, France, September 5, 2005, S. 1-15. http://www.staff.uni-mainz.de/fettke/free/fettke_2005_reference_process_models.pdf

[Fowl97] FOWLER, Martin: *Analysis Patterns – Reusable Object Models*. Menlo Park, Calif.: Addison-Wesley, 1997

[FuSa05] FUMY, Walter; SAUERBREY, Jörg: Identity & Access Management - Faster ROI and improved security through efficient asignment of rights and access control. In: KUHLIN, Bernd; THIELMANN, Heinz (Eds.): *The Practical Real-Time Enterprise: Facts and Perspectives*. Berlin: Springer, 2005, S. 259 - 274

[Gae+05] GAEDKE, M., MEINECKE, J., and NUSSBAUMER, M.: A modeling approach to federated identity and access management. In: *Special interest Tracks and Posters of the 14th international Conference on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05*. New York, NY: ACM Press, 2005, S. 1156-1157. DOI= http://doi.acm.org/10.1145/1062745. 1062916]

[Gamm92] GAMMA, Erich: *Objektorientierte Software-Entwicklung am Beispiel von Et++: Design-Muster, Klassenbibliothek, Werkzeuge*. Berlin, Heidelberg: Springer-Verlag,1992

[Gam+94] GAMMA, E.; HELM, R.; JOHNSON, R. und VLISSIDES, J.: *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Mass.: Addison-Wesley, 1994.

[GrEs01] GREIFFENBERG, S.; ESSWEIN, W.: *Stand der Entwicklung einer Methode zur Metamodellierung*. Arbeitsbericht des Lehrstuhls Wirtschaftsinformatik, insbes. Systementwicklung, Technische Universität Dresden, Fakultät Wirtschaftswissenschaften, Dresden, 2001 http://wiseweb.wiwi.tu-dresden.de/gme/arbeitsbericht.pdf - Zugriff am 01.04.2007 **NICHT** möglich

[Hars94] HARS, A.: *Referenzdatenmodelle - Grundlagen effizienter Datenmodellierung*. Wiesbaden: Gabler, 1994

[HaJo06] HAFIZ, Munawar; JOHNSON, Ralph E.: *Security Patterns and their Classification Schemes.* University of Illinois at Urbana-Champaign. August 15, 2006. https://netfiles.uiuc.edu/mhafiz/www/ResearchandPublications/secpatclassify.pdf Zugriff am 02.11.2006

[Hal+06a] HALKIDIS, Spyros T.; CHATZIGEORGIOU, Alexander; STEPHANIDES, George: A qualitative analysis of software security patterns. In: *Computers & Security* Vol. 25 (2006), Nr. 5, S. 379 – 392

[Hal+06b] HALKIDIS, Spyros T.; CHATZIGEORGIOU, Alexander; STEPHANIDES, George: *A Practical Evaluation of Security Patterns*. University of Macedonia, Thessaloniki, Greece. Proceedings of the The 6th International Conference on ARTIFICIAL INTELLIGENCE and DIGITAL COMMUNICATIONS AIDC 2006 http://www.inf.ucv.ro/~aidc/proceedings/2006/5%20shalkidis.pdf Zugriff am 03.12.2006

[Hey+07] HEYMAN, Thomas; YSKOUT, Koen; SCANDARIATO, Riccardo; JOOSEN, Wouter: An Analysis of the Security Patterns Landscape. In: *IEEE Workshop on Software Engineering for Secure Systems (SESS)*, Minneapolis, MN, USA, May 2007 http://www.cs.kuleuven.be/~riccardo/uploads/docs/sess07-heyman.pdf Zugriff am 22.04.2007

[Hoc+04] HOCHSTEIN, A.; ZARNEKOW, R.; BRENNER, W.: ITIL als Common-Practice-Referenzmodell für das IT-Service-Management: Formale Beurteilung und Implikationen für die Praxis, in: *Wirtschaftsinformatik*, Vol. 46 (2004), Nr. 5, S. 382 - 389

[Hog+06] HOGG, Jason; SMITH, Don; CHONG, Fred; TAYLOR, Dwayne; WALL, Lonnie; SLATER, Paul: *Web Service Security. Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0* http://www.microsoft.com/downloads/details.aspx?familyid=3E02A6C8-128A-47C2-9F39-4082582F3FE1&displaylang=en Zugriff am 4.05.2007

[Jürj04] JÜRJENS, Jan: *Secure Systems Development with UML*. Berlin: Springer Academic Publishers, 2004

[Kie+02] KIENZLE, M D.; ELDER, M. C.; TYREE, D. and EDWARDS-HEWITT, J.: *Security patterns repository. Version 1.0.* http://www.scrypt.net/~celer/securitypatterns/, 2002. Zugriff am 06.06.2007

[Knac01] KNACKSTEDT, R.: Konfigurative Referenzmodelle als Instrumente des Wissensmanagements bei der Data-Warehouse-Entwicklung. In: SCHNURR, H.-P.; S. STAAB, R. STUDER, G. STUMME, Y. SURE (Hrsg.): *Professionelles Wissensmanagement. Erfahrungen und Visionen*. Aachen 2001, S. 113-128.

[Lang97] LANG, K. (1997): *Gestaltung von Geschäftsprozessen mit Referenzprozeßbausteinen*. Wiesbaden : DUV (Gabler Edition Wissenschaft). – Zugl.: Erlangen, Nürnberg, Univ., Diss., 1996

[Loos06] LOOS, Peter: Geleitwort. In: FETTKE, Peter: *Referenzmodellevaluation. Konzeption der strukturalistischen Referenzmodellierung und Entfaltung ontologischer Gütekriterien*. Berlin : Logos-Verlag, 2006, S. V.

[Maie96] MAIER, R.: *Qualität von Datenmodellen*. Deutscher Universitäts Verlag 1996

[Mas+04] MASOVIC, E.; MEHLAU, J.-I.; PRIEBE, T.; REMUS, U. Sicherheitsmuster für Zugriffskontrolle in überbetrieblichen Portalen. In: BARTMANN, Dieter; MERTENS, Peter; SINZ, Elmar J. (Hrsg.): *Überbetriebliche Integration von Anwendungssystemen. FORWIN-Tagung 2004*. Shaker, Aachen 2004, S. 289 - 303.

[Mehl05a] MEHLAU, Jens: *Metamodellbasiertes Sicherheitsmustersystem zur Planung und Umsetzung von Sicherheitsmaßnahmen*. Dieter Bartmann (Hrsg.) Bankinnovationen, Bd. 17. Regensburg: Universitätsverlag, 2005

[Mehl05b] MEHLAU, Jens: Single Sign-On Sicherheitskonzept für ein Finanzportal auf Basis von Sicherheitsmustern. In: BARTMANN, Dieter (Hrsg.) *Innovationen im Retail Banking - Der Weg zum erfolgreichen Privatkundengeschäft*. 1. Auflage. Weinheim: Wiley-VCH, Mai 2005, S. 453 - 474

[Mei+05] MEINECKE, Johannes; NUSSBAUMER, Martin; GAEDKE, Martin: Building Blocks for Identity Federations. In: *Proceedings of the Fifth International Conference on Web Engineering (ICWE 2005)*, Pages 203-208, Sydney http://mwrg.tm.uni-karlsruhe.de/DownloadCenter/publications/dcpub/2005/2005-ICWE2005-MeineckeNussbaumerGaedke-BuildingBlocks ForIdentityFederations_pdf Heruntergeladen am 31.03.2007

[Meis01] MEISE, Volker: *Ordnungsrahmen zur prozessorientierten Organisationsgestaltung - Modelle für das Management komplexer Reorganisationsprojekte*. Studienreihe Wirtschaftsrechtliche Forschungsergebnisse, Bd. 10, Hamburg: Verlag Dr. Kovac, 2001, http://www.verlagdrkovac.de/pdf/0354/0354_3.pdf Zugriff am 13.08.2006

[MeHo92] MERTENS, P.; HOLZNER, J.: Eine Gegenüberstellung von Integrationsansätzen der Wirtschaftsinformatik. In: *Wirtschaftsinformatik*, Vol. 34 (1992), Nr. 1, S. 5 - 25

[MITR07] MITRE http://cve.mitre.org/

[MoMa97] MOWBRAY, Thomas J.; MALVEAU, Raphael C.: *Corba Design Patterns*. Chichester: Wiley & Sons, 1997

[Nag+05] NAGARATNAM, N.; A. NADALIN; M. HONDO; M. MCINTOSH; P. AUSTEL: Business-driven application security: From modeling to managing secure applications. In: *IBM Systems Journal*, Vol 44 (2005), Nr 4, S. 847-867

[PeAa05] PESIC, M. and AALST, W.M.P. van der: Towards a Reference Model for Work Distribution in Workflow Management Systems. In: KINDLER, E.; NÜTTGENS, M. (Hrsg.): *Business Process Reference Models (BPRM) – Proceedings of the Workshop on Business Process Reference Models (BPRM 2005)*, Satellite workshop of the Third International Conference on Business Process Management (BPM), Nancy, France, September 5, 2005, S. 30-44

[Pri+04] PRIEBE, Torsten; FERNANDEZ, Eduardo B.; MEHLAU, Jens I.; PERNUL, Günther: A Pattern System for Access Control. In: FARKAS, Csilla; and SAMARATI, Pierangela (Eds.): *Research Directions in Data and Applications Security* XVIII, IFIP TC11/WG 11.3 Eighteenth Annual Conference on Data and Applications Security, July 25-28, 2004, Sitges, Catalonia, Spain (DB Sec). Dordrecht: Kluwer, 2004, S. 235-249 http://www.cse.fau.edu/~ed/138_PatternSystem4AC_Priebe.pdf Zugriff am 05.11.2006

[Prob03] PROBST, Christian: *Referenzmodell für IT-Service-Informationssysteme* Berlin: Logos-Verlag, 2003

[Rens03] RENSING, Christoph: Hacker-Praktikum. In: GÖRTS, Wim (Hrsg.): *Projektveranstaltungen in Mathematik, Informatik und Ingenieurwissenschaften*. Bielefeld: UniversitätsVerlagWebler, 2003

[Roma01] ROMANOSKY, Sasha: *Security design patterns, Part 1*. http://citeseer.ist.psu.edu/575199.html, November, 2001. Zugriff am 06.06.2007

[Rom+06] ROMANOSKY, Sasha; ACQUISTI, Alessandro; HONG, Jason; CRANOR, Lorrie Faith; FRIEDMAN, Batya: Privacy Patterns for Online Interactions. In: *Proceedings of the Conference on Pattern Languages of Programs, PLoP 2006*. http://hillside.net/plop/2006/Papers/Library/romanosky_privacy_patterns_plop06.pdf Zugriff am 30.04.2007].

[Ros+06] ROSADO, David G.; FERNÁNDEZ-MEDINA, Eduardo; PIATTINI, Mario; GUTIERREZ, Carlos: Comparison of Security Patterns. In: *IJCSNS International Journal of Computer Science and Network Security*, Vol.6 (2006) No. 2B, February. http://paper.ijcsns.org/07_book/200602/200602C06.pdf Zugriff am 30.04.2007

[Rose96] ROSEMANN, M.: *Komplexitätsmanagement in Prozeßmodellen: Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung*. Wiesbaden: Gabler, 1996.

[Rott03] ROTTLEB, René: *Das Paradigma des homogenen Enterprise Access Managements sowie ein Vorschlag zur unternehmensweit konsistenten Zugriffssteuerung*. Dresden, 2003 http://hsss.slub-dresden.de/pub2/ dissertation/ 2004/wirtschaftswissenschaften/1071583567796-8444/1071583567796-8444.pdf http://nbn-resolving.de/urn:nbn:de:swb:14-1071583567796-84449 15.01.2006 – Universität Dresden

[Rüff99] RÜFFER, T.: Referenzgeschäftsprozeßmodellierung eines Lebensversicherungs- unternehmens. In: SINZ, E.J. (Hrsg.): *Modellierung betrieblicher Informationssysteme*. Proceedings der MobIS-Fachtagung 1999, Rundbrief der GI-Fachgruppe 5.10, 6. Jg., Heft 1, Oktober 1999, S. 86-107

[San+96] SANDHU, Ravi S.; COYNE, Edward J.; FEINSTEIN, Hal L. and Youman, Charles E.: Role-based access control models. In: *IEEE Computer*, Vol. 29 (1996) Nr. 2, 38–47, http://www.list.gmu.edu/journals/computer/ i94rbac(org).pdf 08.01.2006

[Sand98] SANDHU, Ravi: Role-Based Access Control, In: ZERKOWITZ, M. (Hrsg.), *Advances in Computers*, Vol.46, Academic Press 1998 http://www.list.gmu.edu/ articles/advcom/a98rbac.pdf 08.01.2006

[Sche97] SCHEER, August-Wilhelm : *Wirtschaftsinformatik : Referenzmodelle für industrielle Geschäftsprozesse* Berlin [ua] : Springer, 1997

[Schl00] SCHLAGHECK, Bernhard: *Objektorientierte Referenzmodelle für das Prozess- und Projektcontrolling. Grundlagen - Konstruktionen – Anwendungsmöglichkeiten*. Wiesbaden: Deutscher Universitäts-Verlag, Gabler 2000

[Schl03] SCHLITT, Michael: *Grundlagen und Methoden für Interpretation und Konstruktion von Informationssystemmodellen*. Dissertation Universität Bamberg. 2003 http://deposit.ddb.de/cgi-bin/dokserv?idn=975205722& dok_var=d1&dok_ext=pdf&filename=975205722.pdf Zugriff am 03.12.2006

[Schm95] SCHMALZL, J.: *Architekturmodelle zur Planung der Informationsverarbeitung von Kreditinstituten*. Physica-Verlag 1995

[Schü98a] SCHÜTTE, Reinhard: *Grundsätze ordnungsmäßiger Referenzmodellierung : Konstruktion konfigurations-und anpassungsorientierter Modelle*. Wiesbaden: Gabler, 1998. – Zugl.: Münster Westfalen, Univ., Diss., 1997

[Schu01] SCHULZE, Dirk: *Grundlagen der wissensbasierten Konstruktion von Modellen betrieblicher Systeme* . Aachen: Shaker, 2001 (Zugl. Bamberg, Diss., 2001.)

[Schu03] SCHUMACHER, Markus: *Security Engineering with Patterns. Origins, Theoretical Models, and New Applications,* LNCS 2754. 1. Auflage. Berlin: Springer Verlag, September 2003

[Sch+03] SCHUMACHER, Markus; RÖDIG, Utz; MOSCHGATH, Marie-Luise: *Hacker Contest - Lösungen, Beispiele*. Xpert.press XI. Berlin: Springer, 2003

[Sch+06] SCHUMACHER, Markus; FERNANDEZ-BUGLIONI, Eduardo; HYBERTSON, Duane; BUSCHMANN, Frank; SOMMERLAD, Peter: *Security Patterns – Integrating Security and Systems Engineering*. Chichester: Wiley, 2006

[Schw99] SCHWEGMANN, A.: *Objektorientierte Referenzmodellierung – Theoretische Grundlagen und praktische Anwendung*. Wiesbaden 1999. (Zugl.: Diss., Münster 1999.)

[Secu07]. *Welcome to SecurityPatterns.Org*. http://www.securitypatterns.org 30.11.2006

[Simo98] SIMONEIT, Monika: *Informationsmanagement in Universitätsklinika – Konzeption und Implementierung eines objektorientierten Referenzmodells*. Diss. 1998. : Deutscher Universitätsverlag, 1998

[Sinz97] SINZ, E.J.: Architektur betrieblicher Informationssysteme. In: *Bamberger Beiträge zur Wirtschaftsinformatik*, Nr. 40, Bamberg 1997

[Smit02] SMITH, Richard E.: *Authentication: From Passwords to Public Keys*. Boston: Addison-Wesley Longman, 2002

[Stac73] STACHOWIAK, Herbert: *Allgemeine Modelltheorie*. Wien: Springer, 1973

[Ste+05] STEEL, Christopher; NAGAPPAN, Ramesh; LAI, Ray: *Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management*. Upper Saddle River, NJ: Prentice Hall, 2005

[Thom05] THOMAS, Oliver: Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation. In: KINDLER, Ekkart; NÜTTGENS, Markus (Hrsg.): *Business Process Reference Models : Proceedings of the Workshop on Business Process Reference Models (BPRM 2005)* ; Satellite workshop of the Third International Conference on Business Process Management (BPM), Nancy, France, September 5, 2005. Nancy, 2005, S. 16-29 http://wwwcs.uni-paderborn.de/cs/kindler/events/BPRM05/PDF/BPRM05_Proceedings.pdf

[Thom06] THOMAS, Oliver: *Management von Referenzmodellen : Entwurf und Realisierung eines Informationssystems zur Entwicklung und Anwendung von Referenzmodellen*. Berlin : Logos-Verl., 2006. Zugl.: Saarbrücken, Univ., Diss., 2006

[Urba04] URBAN, Christoph: *Das Referenzmodell PECS - Agentenbasierte Modellierung menschlichen Handelns, Entscheidens und Verhaltens*. Dissertation. Universität Passau. April 2004 http://www.opus-bayern.de/uni-passau/ volltexte/2005/47/pdf/Dissertation_Ch_Urban_Hauptteil.pdf Zugriff am 12.11.2006

[Wild06] WILDGRUBER, Rudolf: Identity and Access Management (IAM). In: FUMY, Walter; SAUERBREY, Jörg (Hrsg.): *Enterprise Security. IT Security Solutions: Concepts, Practical Experiences, Technologies* Publicis Corporate Publishing, Erlangen. 1. Auflage - Januar 2006 S. 70 - 83 http:// www.competence-site.de/ itsecurity.nsf/ AC01E037227ED7B4C12571DB004BA9DB/$File/ identity_access_management_enterprise%20security.pdf Zugriff am 15.09.2006

[Wind05] WINDLEY, Phillip J.: *Digital Identity Management.* 1. Auflage. Köln : O'Reilly, 2005

[Wolf01] WOLF, Stefan: *Wissenschaftstheoretische und fachmethodische Grundlagen der Konstruktion von generischen Referenzmodellen betrieblicher Systeme*. Dissertation, Aachen: Shaker, 2001 (Zugl.: Diss., Bamberg 2001.)

[YoBa97] YODER, Joseph und BARCALOW, Jeffrey: Architectural patterns for enabling application security. *Proceedings of PLOP'97*. Oder Kapitel 15 in *Pattern Languages of Program Design*, Vol. 4 (N. Harrison, B. Foote, and H. Rohnert, Eds.), Menlo Park, Calif.: Addison-Wesley, 2000

[Ysk+06] YSKOUT, Koen; HEYMAN, Thomas; SCANDARIATO, Riccardo; JOOSEN, Wouter: *A system of security patterns*. Report CW469, December 2006. Katholieke Universiteit Leuven, Department of Computer Science http://www.cs.kuleuven.ac.be/publicaties/rapporten/cw/CW469.pdf Zugriff am 22.04. 2007

Eidesstattliche Versicherung

Ich versichere an Eides statt durch meine Unterschrift, dass ich die Masterarbeit „*Ein patternbasiertes Referenzmodell für Identity Management*“ selbständig und ohne fremde Hilfe angefertigt und alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe, mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Die Arbeit hat in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Berlin, 12.06.2007

Ort, Datum	Unterschrift