Karlsruhe Institute of Technology

KIM-Identitätsmanagement

Projektdokumentation

Dipl.-Inform. Thorsten Höllrigl
Dipl.-Inform. Sebastian Labitzke
Dipl.-Inform. Frank Schell
Dr.-Ing. Jochen Dinger
Dipl.-Inform. Axel Maurer
Prof. Dr. Hannes Hartenstein

STEINBUCH CENTRE FOR COMPUTING (SCC)

KIT – die Kooperation von
Forschungszentrum Karlsruhe GmbH
und Universität Karlsruhe (TH)

Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Executive Summary

Das Projekt KIM-IDM (Phase 2006 bis 2008) ist mit der Konzeption und technische Realisierung eines zunächst Universitäts-weiten (später KIT-weiten) integrierten Identitäts- und Zugriffsmanagements betraut worden. Das vorgeschlagene Konzept verfolgt eine Ausgewogenheit zwischen zentralen und dezentralen Verfahren und betrachtet die Universität bzw. das Karlsruher Institut für Technologie als eine Föderation von Einrichtungen, in der Identitätsmanagementdienste im Rahmen einer Dienste-orientierten Architektur angeboten werden. Eines der hierbei verfolgten Ziele war es, eine Datenkonsistenz kongruenter personenbezogener Daten zu erreichen, ohne die Autarkie einzelner Einheiten einzuschränken. Zunächst wurde ein Identitätsmanagement, das sowohl Institutionen an der Universität als auch am Forschungszentrum integriert, in Betrieb genommen und damit das Fundament für KIT-weite Diensterbringung gelegt.

Das Projekt KIM-IDM ist zunächst von der Universität Karlsruhe mit Mitteln für zwei akademische Mitarbeiter über den Zeitraum von drei Jahren ausgestattet worden. Für weitere zwei Jahre (2008 bis 2010) sind ebenfalls Mittel für zwei akademische Mitarbeiter zu Verfügung gestellt worden, um die Projektergebnisse in einen geordneten Betrieb zu überführen.

Mit dem vom Projekt KIM-IDM und dem Steinbuch Centre for Computing aufgebauten Identitätsmanagementsystem konnten bislang folgende Ergebnisse erzielt werden:

- Das Identitätsmanagementsystem bildet die Grundlage für die Nutzung des Studierendenportals durch alle Studierenden und Lehrenden des KIT. Zusätzlich wurde der Immatrikulationsprozess erheblich verbessert: so kann etwa die Einwilligung in die Nutzungsbedingungen online erteilt und können die IT-Dienste für die Studierenden unmittelbar nach der Einschreibung genutzt werden.
- Das Identitätsmanagementsystem bildet die Grundlage für die Nutzung des Mitarbeiterportals durch die KIT-Mitarbeiter. Derzeit nutzen ca. 60 % aller KIT-Mitarbeiter das Portal. Das Portal bietet u.a. Dienste für die Einordnung der Mitarbeiter in die KIT-Kompetenzfelder und -bereiche.

Dieser Bericht legt die Grundüberlegungen zur Konzeption sowie die Prozesse, Dienste und technischen Komponenten des Identitätsmanagementsystems dar. Mit dem Projekt KIM-IDM sind auch

- der Aufbau des Namensraums kit.edu sowie des zugehörigen Arbeitsstabs „KIT Namen und Nummern“ (KNN)
- der Aufbau des Arbeitsstabs IT-Sicherheit, Datenschutz und IT-Rechtskonformität (ASDUR)
- der Aufbau eines SCC-internen „Stakeholder Forum“

verbunden, die an anderer Stelle beschrieben werden.

Karlsruhe, im August 2009

Titel	**Autoren**	**Quelle**
Performance Evaluation of Identity and Access Management Systems in Federated Environments	F. Schell, J. Dinger, H. Hartenstein	Proceedings of the 4th International ICST Conference on Scalable Information Systems (INFOSCALE 2009), Hong Kong, China, Juni 2009
Towards Systematic Engineering of Service-Oriented Access Control in Federated Environments	T. Höllrigl, F. Schell, S. Suelmann, H. Hartenstein	Proceedings of IEEE Congress on Services Part II (SERVICES-2 2008), S. 104-111, Peking, China, September 2008
Integriertes Service-Portal zur Studienassistenz	F. Allerding, J. Buck, P. Freudenstein, B. Klosek, T. Höllrigl, W. Juling, B. Keuter, S. Link, F. Majer, A. Maurer, M. Nussbaumer, D. Ried, F. Schell	38. Jahrestagung der Gesellschaft für Informatik (INFORMATIK 2008), S. 596-601, München, Deutschland, September 2008
Federated and Service-Oriented Identity Management at a University	F. Schell, T. Höllrigl, H. Hartenstein	In Proceedings of 14th European University Information Systems (EUNIS 2008), S. 59, Aarhus, Dänemark, Juni 2008
Föderatives und dienstorientiertes Identitätsmanagement im universitären Kontext	T. Höllrigl, F. Schell, H. Wenske, H. Hartenstein	Proceedings of the 1st Workshop Integriertes Informationsmanagement an Hochschulen (IIM 2007), S. 75-90, Karlsruhe, Deutschland, März 2007
Dienstorientiertes Identitätsmanagement für eine Pervasive University	T. Höllrigl, A. Maurer, F. Schell, H. Wenske, H. Hartenstein	Jahreskonferenz der GI - Lecture Notes in Informatics, S. 70-74, Dresden, Deutschland, September 2006
Integriertes Management von Identitäten im fakultativen und universitätsweiten Kontext	K. Scheibenberger, H. Wenske, H. Hartenstein, O. Hopp	20. DFN-Jahrestagung (DFN2006), S. 206-218, Heilbronn, Deutschland, März 2006

Tabelle 1: Konferenzbeiträge des Projektes KIM im Bereich Identitätsmanagement

Titel	**Autoren**	**Quelle**
Federated Identity Management as a Basis for Integrated Information Management	F. Schell, T. Höllrigl, H. Hartenstein	it-Information Technology, Jahrgang 51 (2009), Heft 1, Oldenbourg Wissenschaftsverlag, S. 14-23, März 2009
Karlsruher Integriertes Informations-Management - KIM	H. Hartenstein, W. Juling, A. Maurer	Medien in der Wissenschaft, Band 46. E-Strategy, Strategisches Informationsmanagement für Forschung und Lehre: Gesellschaft für Medien in der Wissenschaft, Waxmann Verlag, S. 99-114, Juli 2008
Föderatives und dienstorientiertes Identitätsmanagement: Konzept und Erfahrungen	T. Höllrigl, F. Schell, H. Wenske, H. Hartenstein	Praxis der Informationsverarbeitung und Kommunikation (PIK), Jahrgang 30, Heft 4, S. 156-162, Oktober 2007
Integriertes Informationsmanagement und zugehörige Dienstestruktur	H. Hartenstein, W. Juling, A. Maurer	Education Quality Forum, Band 2006. E-University - Update Bologna, Waxmann Verlag, S. 161-172, September 2007
Karlsruher Integriertes Informations-Management - KIM	H. Hartenstein, W. Juling, A. Maurer	Informationsinfrastrukturen im Wandel, Informationsmanagement an deutschen Universitäten, DINI Deutsch Initiative für Netzwerkinformationen e.V, S. 194-205, April 2007

Tabelle 2: Zeitschriftenartikel und Buchbeiträge des Projekts KIM im Bereich Identitätsmanagement

Veranstaltung	**Datum**	**Veranstaltungsort**
Tagung der DFN-Nutzergruppe Hochschulverwaltung	11.05.09	Universität Leipzig
Arbeitssitzung Serviceverbund KIM der Universität Konstanz	12.12.08	Universität Konstanz
SUN Summit Hochschulverwaltung	26.11.08	Universität Göttingen
Präsidium Universität des Saarlandes	14.08.08	Universität Saarbrücken
ZKI Verzeichnisdienste	10.03.08	FU Berlin
BGNW Arbeitskreis Security	05.11.07	Universität Karlsruhe (TH)
Arbeitskreis Windows-/Linux PCs in der HGF	24.10.07	DKFZ Heidelberg
Softwarehouse IT-Solution Forum 2007	11.10.07	GWDG Göttingen
Präsidium Universität Hamburg	01.10.07	Universität Hamburg
VPS Kundentag	22.06.07	Ettlingen
SUN DAY Karlsruhe	11.05.07	Universität Karlsruhe (TH)
Treffen der CIOs der TU9	27.03.07	Universität Karlsruhe (TH)
Hochschultag Microsoft	15.-21.03.07	CeBIT 2007, Hannover

Tabelle 3: Ausgewählte Vorträge des Projektes KIM im Bereich Identitätsmanagement

Inhaltsverzeichnis

1. Einleitung

In diesem Kapitel wird zunächst die grundsätzliche Motivation eines integrierten Identitäts- und Zugriffsmanagements dargelegt. Es folgt die Beschreibung der Ziele und Ergebnisse des Projekts KIM-Identitätsmanagement (KIM-IDM) im Rahmen dessen die Basis für ein KIT-weites Identitätsmanagements gelegt und realisiert wurde. Wesentliche Kennzahlen des produktiven Identitätsmanagementsystems sowie eine Gliederung bilden den Abschluss dieses Kapitels.

1.1 Motivation für ein Identitäts- und Zugriffsmanagement

Zugriffe auf Dienste und Ressourcen einer IT-Infrastruktur dürfen in der Regel nur hierfür berechtigten Personen oder Prozessen gewährt werden. Um sich gegenüber einem IT-System auszuweisen, muss die Identität des Anfragers zweifelsfrei feststellbar sein, d.h. eine vorgebliche Identität muss authentifiziert werden. Somit muss ein IT-System „digitale Identitäten“ seiner Nutzer und die zugehörigen Attribute verwalten, um Authentifikation und Autorisation durchführen zu können.

Wenn nun verschiedene Teilsysteme der IT-Infrastruktur eines Unternehmens oder einer Universität jeweils für sich eigenständig ihre Benutzer bzw. Identitäten verwalten, entstehen verschiedene Probleme: i) ein Benutzer muss sich gegenüber jedem Teilsystem in einer anderen Art und Weise ausweisen, ii) ein Geschäftsablauf über verschiedene Teilsysteme hinweg ist unmöglich oder aufwendig, iii) die Angaben zur Identität und daraus resultierende Autorisationen sind über die verschiedenen Teilsysteme hinweg betrachtet inkonsistent.

Um nun einem Benutzer die Authentifikation zu erleichtern, um IT-gestützte Geschäftsabläufe über Einrichtungs- und Systemgrenzen hinweg zu ermöglichen und um Identitätsinformation konsistent zu halten, insbesondere auch im Hinblick auf Sicherheit und Datenschutz, wird ein integriertes Identitäts- und Zugriffsmanagement

über die Organisationseinheiten hinweg benötigt. Durch den Aufbau eines integrierten Identitätsmanagements sollen sowohl die Produktivität in Forschung und Lehre erhöht als auch Sicherheits- und Datenschutzaspekte verbessert werden. Demnach ist das Identitätsmanagement als „Enabler“ für effizientes und effektives Arbeiten zu verstehen und liefert die Basis für sicherheitsbezogene Vorgänge.

Die wesentlichen Ziele eines Identitäts- und Zugriffsmanagements aus Betreibersicht sind hierzu nachfolgend aufgeführt [Schell et al. 2009].

- *Verwaltung von Nutzerkonten.* Das Identitätsmanagement soll durch automatisiertes Anlegen, Pflegen und Entfernen von Benutzerkonten auf den unterschiedlichen Systemen eine fehlerfreie und konsistente Verwaltung von Nutzerkonten sicherstellen.

- *Aktualität von Zugriffsberechtigungen.* Auf schützenswerte Ressourcen dürfen nur hierzu berechtige Nutzer zugreifen, was durch die Wahrung der Aktualität von Zugriffsberechtigungen sichergestellt werden soll. Dies betrifft die Vergabe, Anpassung und den Entzug von Zugriffsrechten durch konsistente Rechtevergabe.

- *Aufwandsreduktion.* Das Identitätsmanagement soll Mitarbeiter bei der Eingabe und Pflege von Daten unterstützen und den administrativen Aufwand reduzieren. Die Verringerung der Anzahl der Systeme, an denen die Daten gepflegt werden müssen und die automatische Belieferung soll hierbei zu einer möglichen Refokussierung des Personals auf die eigentlichen Kernaufgaben führen.

- *Nutzerfreundlichkeit.* Der Umgang des Nutzers mit Diensten soll durch Self Services, bspw. Änderung des Passworts, und Single Sign-On vereinfacht werden.

- *Vermeidung von Redundanz und Erhöhung der Datenqualität.* Typischerweise werden in einem Unternehmen an vielen Stellen Benutzerverzeichnisse geführt. Hierdurch sind Namen, Räume, Telefonnummern, etc. fortwährend in einer Vielzahl von Systemen konsistent zu halten. Durch die redundante Pflege entstehen in der Regel höhere Kosten und die Qualität der Daten leidet. Durch Konsolidierung dieser Benutzerverwaltungen sollen Mehraufwand, Inkonsistenzen und Kosten reduziert werden.

- *IT-Compliance und Audit.* Die Einhaltung sowie der Nachweis der Einhaltung der gesetzlichen Anforderungen wird unter dem Begriff *IT-Compliance* verstanden. Identitätsmanagement soll die Grundlage zur Auditierung von Zugriffen schaffen und somit die Basis zur Wahrung der IT-Compliance dienen.

Darüber hinaus bedarf der Umgang mit sensiblen Daten, wie es personenbezogene Informationen darstellen, der Einhaltung von Richtlinien. Neben den geltenden datenschutzrechtlichen Regelungen, die in Kapitel 3 erläutert werden, definiert bspw.

die Liberty Alliance – ein Zusammenschluss von über 150 Organisationen mit dem Ziel offene Standards und Leitfäden für das Identitätsmanagement zu erstellen und zu etablieren – folgende Grundsätze, die dem Schutz von Identitätsinformationen vor der Willkür eines Betreibers dienen sollen [WWW Liberty Alliance 2009].

- *Benachrichtigung.* Der Benutzer muss darüber aufgeklärt werden, wer die personenbezogenen Daten erhält und speichert, welche Daten gespeichert werden, wie diese gespeichert werden und ob die Daten an andere weitergegeben werden oder nicht.

- *Wahlmöglichkeit.* Falls keine ausdrückliche gesetzliche Grundlage zur Erhebung und Verarbeitung von personenbezogenen Daten existiert, muss der Benutzer die Wahl haben, selbst zu entscheiden, für welchen Zweck und an wen seine Daten weitergegeben werden. Darüber hinaus muss der Benutzer jederzeit die Möglichkeit haben, Einwilligungen, welche zuvor gegeben oder verweigert wurden, zu überprüfen, zu berichtigen und gegebenenfalls anzupassen.

- *Benutzerzugang zu Identitätsinformationen.* Dem Benutzer muss eine Möglichkeit eröffnet werden, so dass er sämtliche über ihn gespeicherten personenbezogenen Daten einsehen kann.

- *Beschwerdemöglichkeit.* Der Benutzer muss die Möglichkeit haben, bei Verdacht auf Missbrauch seiner personenbezogenen Daten Beschwerde einzureichen.

- *Zweckbindung.* Personenbezogene Daten dürfen grundsätzlich nur für den Zweck eingesetzt werden, für welchen sie ursprünglich vorgesehen wurden, bzw. zu welchem der Benutzer seine Zustimmung gegeben hat.

- *Qualität.* Der Benutzer muss eine angemessene Möglichkeit haben, seine Daten jederzeit zu korrigieren.

- *Zeitbeschränkungen.* Identitätsinformationen dürfen nur so lange gehalten werden, wie sie benötigt werden bzw. wie der Benutzer in einer entsprechenden Erklärung der Nutzung zugestimmt hat.

- *Sicherheit.* Die Speicherung und Übertragung von Identitätsinformationen muss durch entsprechende Maßnahmen vor unberechtigtem Zugriff oder Verlust geschützt werden.

1.2 Zielsetzung und Ergebnisse des Projekts KIM-IDM

In dem Projekt KIM-Identitätsmanagement (KIM-IDM) – Teil des Gesamtvorhabens KIM (Karlsruher Integriertes InformationsManagement) – sollte ursprünglich

für die Universität Karlsruhe (TH) die Basis für ein integriertes Identitätsmanagement geschaffen werden. Ziel war es dabei, in einem Zeitraum von drei Jahren den Produktionsbetrieb für die Gesamtuniversität und die zugehörigen Arbeitsabläufe zu etablieren. Nach Bekanntgabe des Zusammenschlusses der Universität Karlsruhe (TH) mit dem Forschungszentrum Karlsruhe (FZK) zum Karlsruher Institut für Technologie (KIT) wurde die Zielsetzung des Projekts KIM-IDM auf beide Teilorganisationen ausgeweitet[1]. Das Projekt KIM-IDM widmete sich sowohl den technischen als auch den organisatorischen Grundlagen und arbeitete eng mit dem Vorhaben KIM-LPS (Lehrveranstaltungsmanagement, Prüfungsmanagement, Studienassistenz) zusammen.

Das Projekt KIM-IDM folgt dem Grundsatz, dass eine Zentralisierung soweit erfolgen sollte, dass Einrichtungen entlastet, aber in der Durchführung ihrer spezifischen Aufgaben nicht behindert werden. So möchte eine dezentrale Einrichtung meist selbständig über Berechtigungen an eigenen Ressourcen verfügen. Eine Authentifikation und zugehöriges Passwort-Management kann jedoch als Dienstleistung einer zentralen Einrichtung durchgeführt werden. Die zu einer Identität zugehörigen Daten, die in einem integrierten Informationsmanagement verwaltet werden, sind in der Regel als personenbezogen zu klassifizieren und stellen somit sensible Daten dar, die strengen Datenschutzrichtlinien unterliegen.

Ziel von KIM-IDM war es deshalb auch, gemeinsam mit den betroffenen Organisationseinheiten die Arbeitsabläufe zu definieren, in denen über Zugriff auf Identitätsattribute entschieden werden soll. Insbesondere war die Nachvollziehbarkeit von Zugriffsberechtigungen auf Identitätsattribute sowie die Nachvollziehbarkeit des Austauschs von Identitätsinformation zwischen Einrichtungen ein zentraler Aspekt der Projektarbeit.

Nachfolgend sind die wesentlichen erzielten Ergebnisse des Projekts KIM-IDM aufgeführt:

- *Erarbeitung eines föderativen Konzepts.* Zur Realisierung eines Identitäts- und Zugriffsmanagements wurde ein föderatives Konzept entwickelt. Eine detaillierte Beschreibung dieses Konzepts findet sich in Kapitel 2.

- *Klare Prozesse und Zuordnung von Verantwortlichkeiten.* Die Prozesse zur Versorgung von Studierenden nach der Immatrikulation und der Mitarbeiter nach ihrer Einstellung mit E-Mail-Adressen und sicheren Nutzerkontendaten und die konsequente Erfassung aller Lehrbeauftragten am KIT über die Fakultäten wurde durch das Projekt KIM-IDM verbessert. Die organisatorischen Festlegungen werden in Kapitel 3 erläutert und die einzelnen Prozesse werden in Abschnitt 4.3 detailliert beschrieben.

- *Synchronisation der Personendaten.* Es findet eine KIT-weite Synchronisation von Identitätsdaten zwischen den Einrichtungen statt, wodurch alle Studierende und Mitarbeiter des KIT identifiziert werden können. Darüber hinaus erhält

[1] Aufgrund ihrer geographischen Lage wird die Universität auch als Campus Süd und das Forschungszentrum als Campus Nord bezeichnet.

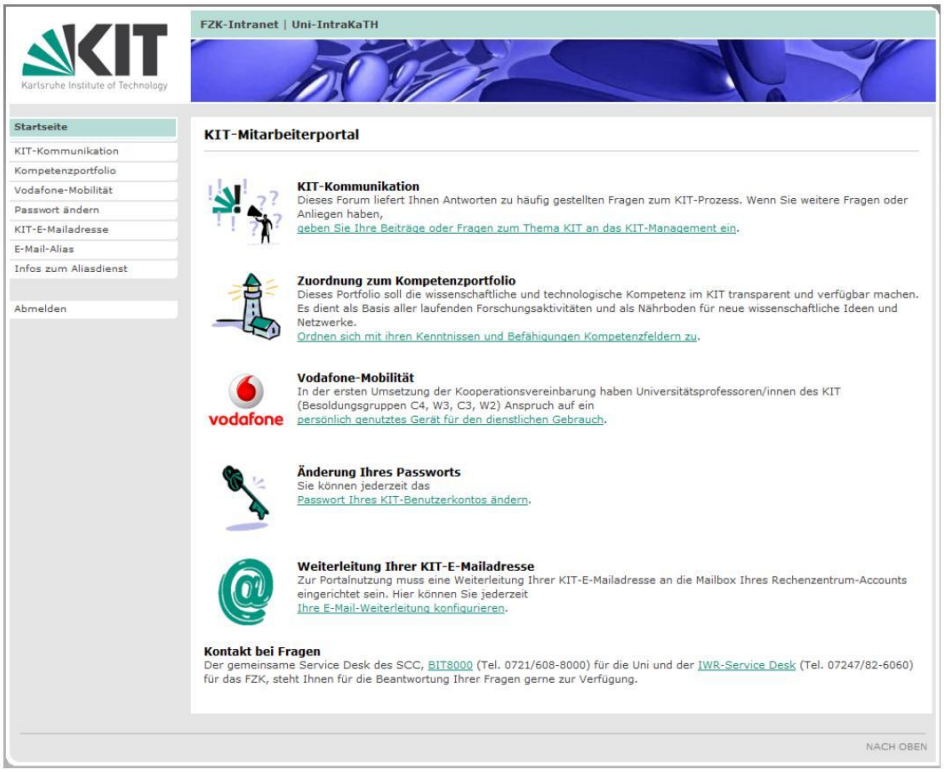


Abbildung 1.1: Startseite des Mitarbeiterportals

jedes Mitglied des KIT automatisiert eine *kit.edu*-E-Mail-Adresse. Das hierzu realisierte Provisionierungssystem wird ausführlich in Kapitel 4 erläutert.

- *Belieferung von Authentifikationsdiensten.* Zur Authentifikation von Mitarbeitern und Studierenden am KIT werden insbesondere dem KIT-weiten Active Directory (siehe Abschnitt 4.2.4) sowie Shibboleth (siehe Abschnitt 6.2) die notwendigen Daten zur Verfügung gestellt.

- *Realisierung von Portaldiensten.* Die Entwicklung des Mitarbeiter- und Studierendenportals wurde durch das Projekt KIM-IDM mitgetragen. Abbildung 1.1 zeigt exemplarisch die Startseite des Mitarbeiterportals. Als Funktionen sind in diesem Portal Selbstbedienungsfunktionen wie ein E-Mail-Alias-Dienst, Vodafone-Abwicklung, Zuordnung zum Kompetenzportfolio, Änderung des Passworts oder E-Mail-Weiterleitung implementiert, die in Kapitel 5 erläutert werden. Zur Realisierung von Portaldiensten sind Infrastrukturdienste notwendig, die als Web Services realisiert wurden und in Kapitel 6 näher beschrieben werden.

- *Datenschutzgerechter Umgang mit personenbezogenen Daten.* Durch die organisatorischen Festlegungen in Verbindung mit dem realisierten föderativen Identitätsmanagement (vgl. Kapitel 2 bzw. 3) wurde auch dem Datenschutz Rechnung getragen, da durch Automatisierung weniger Mitarbeiter mit diesen Daten in Berührung kommen. Ferner wird der Datenaustausch sowie das Sperren von Konten durch das Identitätsmanagementsystem kontrolliert und protokolliert und ist somit nachvollziehbar (vgl. Kapitel 4). Zusätzlich wird in

Administrierte Mitarbeiter Campus Süd	5.202
Administrierte Mitarbeiter Campus Nord	4.454
Administrierte Studierende	18.916
Attribute Mitarbeiter Campus Süd	28
Attribute Mitarbeiter Campus Nord	29
Attribute Studierende	23
Angeschlossene Ressourcen	9
Provisionierungs-Prozesse	18
Portaldienste	11
Infrastrukturdienste (Web Services)	8

Tabelle 1.1: Kennzahlen des produktiven Identitätsmanagements (Stand: 20.01.2009)

Browser-basierten Szenarien via Shibboleth der Austausch von Nutzerattributen durch den Nutzer selbst reglementiert (vgl. Abschnitt 6.2).

1.3 Kennzahlen des Projekts KIM-IDM

In Tabelle 1.1 wird ein Überblick wesentlicher Fakten für das KIT-weite Identitätsmanagementsystem, das im Rahmen des Projekts KIM-IDM umgesetzt wurde, gegeben. Durch das Identitätsmanagementsystem werden mittlerweile mehr als 9.500 Mitarbeiter am Campus Süd und Campus Nord administriert. Zusätzlich werden auch ca. 19.000 Studierende verwaltet. Für die Mitarbeiter des KIT werden am Campus Nord 29 und am Campus Süd 28 personenbezogene Attribute wie Name und Geburtsdatum etc. zwischen den Quell- und Zielsystemen synchronisiert. Bei Studierenden sind es 23 personenbezogene Attribute. Die Anzahl der Attribute ergibt sich jeweils aus der Summe aller für die entsprechende Gruppe verfügbaren Attribute der angeschlossenen Ressourcen. Die Beschreibung der Attribute aller am Identitätsmanagementsystem angekoppelten Ressourcen findet sich in Kapitel 4.

An das Identitätsmanagement sind 9 Ressourcen angeschlossenen, die sowohl Quelle als auch Ziel von Datenflüssen sein können. Diese aus unterschiedlichen Instituten stammenden, integrierten Systeme werden in Kapitel 4 näher erläutert. Für den Austausch der Daten respektive die Provisionierung wurden 18 Provisionierungs-Prozesse realisiert, die detailliert in Abschnitt 4.1.3 erläutert werden. Darüber hinaus wurden im Projekt 11 Dienste für das Studierenden- und Mitarbeiterportal (siehe Kapitel 5) sowie 8 Infrastrukturdienste (siehe Kapitel 6), die als Web Services umgesetzt wurden, realisiert.

Neben der oben genannten Anzahl administrierter Mitarbeiter kann die Anzahl der Mitarbeiter unterschieden werden, die ihr Nutzerkonto im Mitarbeiterportal aktiviert haben. Aus Mitarbeitersicht ist die Aktivierung nötig, um verschiedene Dienste wie den E-Mail-Alias-Dienst (vgl. Abschnitt 5.2.4) oder den Vodafone-Beantragungsdienst (vgl. Abschnitt 5.2.7) zu nutzen. Aus dem Blickwinkel des Identitätsmanagement ist eine Aktivierung im Wesentlichen aus organisationstechnischen Gründen

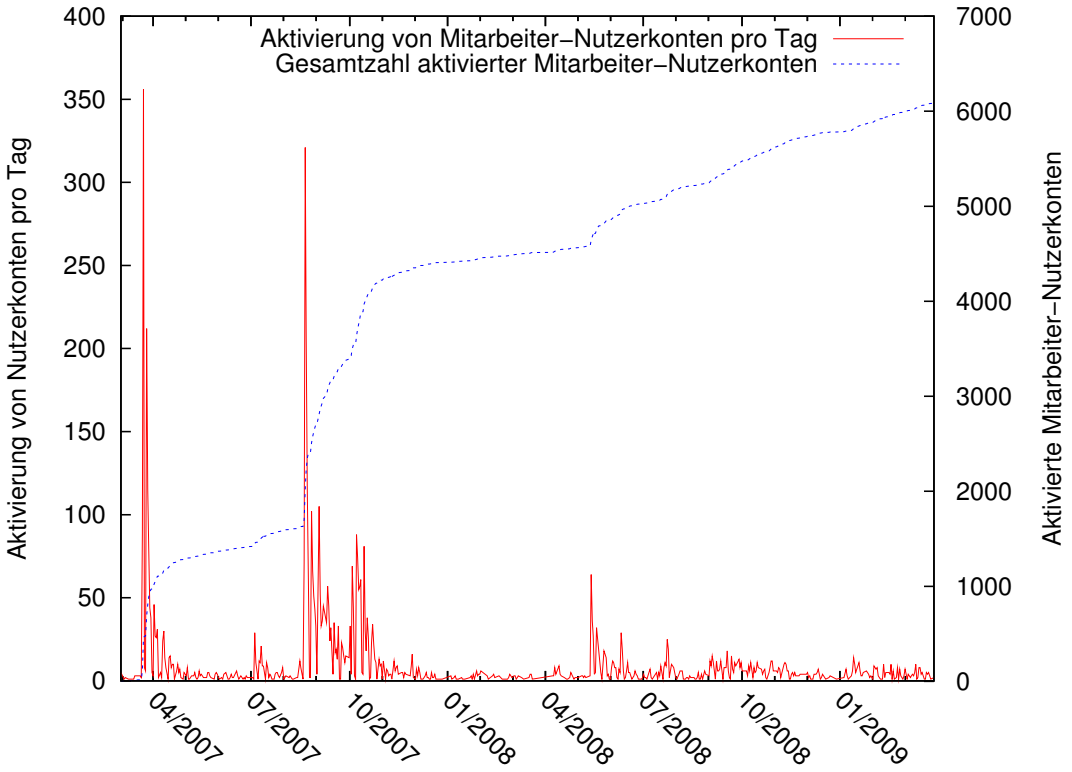


Abbildung 1.2: Anzahl der aktivierten Mitarbeiter im Mitarbeiterportal

nötig. So wurden z.B. die Mitarbeiterdaten des Campus Süd und die Nutzerkonten des Steinbuch Centre for Computing (SCC) durch Mithilfe der Mitarbeiter verknüpft, da eine voll automatisierte Verknüpfung mangels eindeutigem Merkmal nicht möglich war (vgl. Abschnitt 5.2.2). Insofern kann diese Aktivierung als Indikator herangezogen werden, um zu beurteilen ob die Identitätsmanagementdienste von den Mitarbeitern des KIT angenommen und genutzt werden.

Die Aktivierung von Nutzerkonten über das Mitarbeiterportal ist in Abbildung 1.2 dargestellt. Zum einen ist dort die Anzahl der Aktivierungen von Mitarbeiter-Nutzerkonten pro Tag bezogen auf die linke Y-Achse (Wertebereich 0 bis 400) verzeichnet. Zum anderen ist die Gesamtzahl aktivierter Nutzerkonten bezogen auf die rechte Y-Achse (Wertebereich 0 bis 7.000) aufgeführt. Zu mehreren Zeitpunkten ist ein markanter Anstieg zu verzeichnen. Nach der Freischaltung des Portals im März 2007 wurde über das Portal am Juli 2007 die Möglichkeit geschaffen, Vodafone-Handys zu ordern und Vodafone-Verträge abzuschließen. Die Bekanntmachung dieses Features führte in den Folgemonaten August und September 2007 zu einem Anstieg auf über 2.000 aktivierte Nutzerkonten. Durch die Freischaltung des Portalfeatures Kompetenzfeldzuordnungsdienst, das es wissenschaftlichen Mitarbeitern erlaubt, die Zuordnung zu ihren Kompetenzfeldern vorzunehmen, im Oktober 2007 stieg in den darauffolgenden Monaten die Gesamtzahl der aktivierten Nutzerkonten auf über 4.000 an. Die Bekanntgabe der Kompetenzfeldsprecherwahl und die Ankündigung, dass die Wahlberechtigung für ein bestimmtes Kompetenzfeld nur über das im Mitarbeiterportal zugeordnete Kompetenzfeld erlangt werden kann, im Sommer 2008 führte

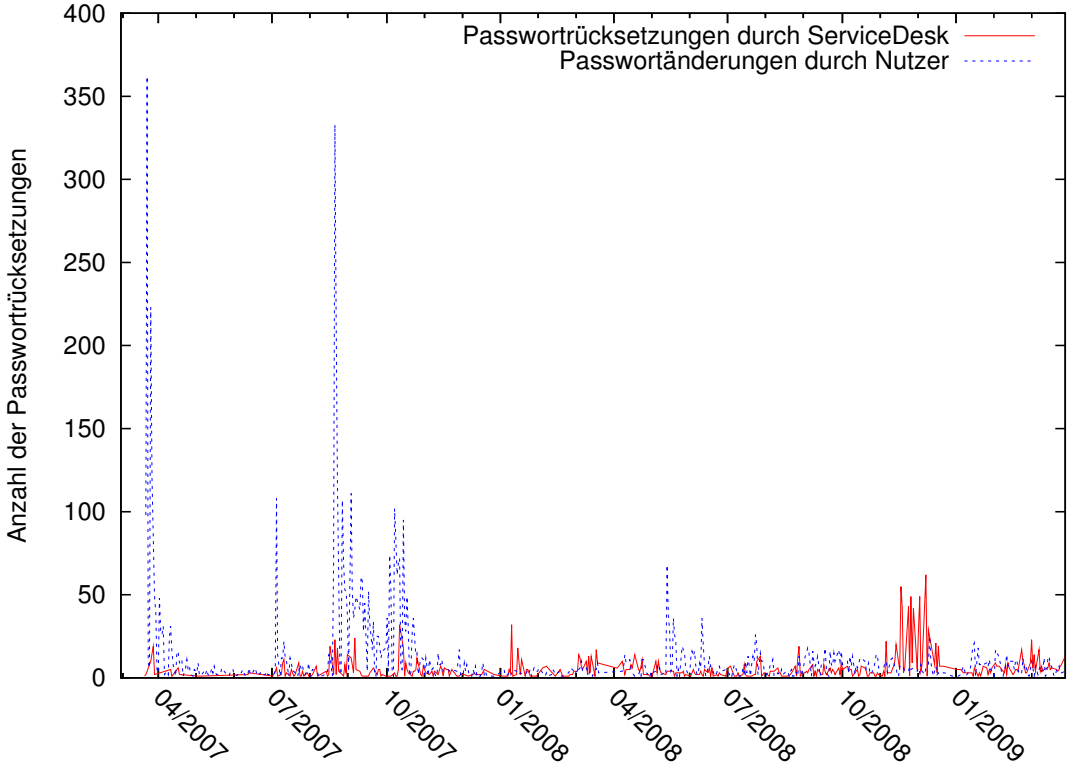


Abbildung 1.3: Häufigkeit der Änderung von Passwörtern

zu einem weiteren Zuwachs auf nun über 6.000 aktivierte Nutzerkonten. Insgesamt wurde das Portal somit von mehr als 60 % aller KIT-Mitarbeiter genutzt, wobei anzumerken bleibt, dass die bislang existierenden Dienste teilweise nur von einem Teil der mehr als 9.500 administrierten Mitarbeiter genutzt werden können. So darf eine Kompetenzfelderzuordnung bspw. nur wissenschaftlichen Mitarbeitern möglich sein. Ferner zeigt sich, dass durch entsprechende Anreize wie z.B. die *kit.edu*-E-Mail-Adresse eine hohe Aufmerksamkeit und letztlich auch Teilnahme erreicht werden kann.

Die Häufigkeit der Passwortänderungen pro Monat von März 2007 bis März 2009 ist in Abbildung 1.3 dargestellt. Hierbei werden zum einen die Passwortänderungen gezeigt, die durch die Benutzer selbst durchgeführt wurden. Zum anderen sind in der Abbildung auch die von Nutzern gewünschte Passwortrücksetzungen mit dem anschließenden Versand eines neuen Passworts per Brief durch den ServiceDesk dargestellt. Deutlich sind hier die Anstiege zum Zeitpunkt der Inbetriebnahme des Mitarbeiterportals im März 2007 und der Freischaltung der Portalfeatures Vodafone-Beantragung und Kompetenzfeldzuordnung im Juli und Oktober 2007 erkennbar (vgl. auch Abbildung 1.2). Dies ist darauf zurückzuführen, dass jeder Benutzer zur Aktivierung des Nutzerkontos im Mitarbeiterportal sein Passwort zunächst ändern muss, um daraufhin diese Dienste nutzen zu können.

Anhand der Abbildung 1.3 wird zum einen deutlich, dass sich der Aufwand für den ServiceDesk trotz tausender Nutzer in einem akzeptablen Rahmen bewegt. Des Weiteren ist aber auch erkennbar, dass die Kompetenzfeldsprecherwahl zu einem Anstieg der ServiceDesk-Anfragen zwischen Oktober 2008 und Januar 2009 geführt hat.

Dies ist darauf zurückzuführen, dass die Dienste des Mitarbeiterportals keine ständige Nutzung erfordern und für Mitarbeiter daher keine unmittelbare Veranlassung besteht sich häufig am Mitarbeiterportal anzumelden. Insofern sollte den Nutzern zukünftig die Möglichkeit eröffnet werden, die bislang immer noch existierenden unterschiedlichen Nutzerkonten des SCC, der KIT-Bibliothek, des Mitarbeiterportals etc. durch ein gemeinsames Konto zu ersetzen, so dass sich Nutzer nur noch *eine* Kennung und *ein* Passwort merken müssen.

1.4 Gliederung der Arbeit

Nach diesem einleitenden Kapitel erfolgt im Kapitel 2 die Beschreibung des technischen Konzepts, der im Projekt KIM-IDM entstandenen Lösung. In Kapitel 3 wird auf organisatorische Festlegungen, die für das KIT-weite Identitätsmanagement von Bedeutung sind, eingegangen. Im hierauf folgenden Kapitel 4 wird nach einem Überblick über das Provisionierungssystem auf die technischen Details der angebundenen Ressourcen eingegangen und deren Abhängigkeiten zu relevanten Identitätsmanagement-Prozessen dargelegt. Kapitel 5 erläutert die technische Realisierung der Portaldienste. Hierbei werden jeweils für das Mitarbeiter- und das Studierendenportal die einzelnen zur Verfügung stehenden Dienste näher erläutert. Als Fundament für die realisierten Portalfeatures werden Web Services als Infrastrukturdienste verwendet, die neben dem Dienst Shibboleth im Kapitel 6 näher beschrieben werden.

2. Technische Konzeption

Das im Rahmen des Projekts KIM-IDM realisierte Identitätsmanagement (IDM) dient als Grundlage für die im Zuge des Gesamtprojekts KIM entwickelten Dienste. Des Weiteren stellt das Projekt KIM-IDM eigene Dienste bereit. Diese Dienste sind in die Dienstelandschaft des Gesamtvorhabens KIM eingebunden und werden darüber hinaus den Mitarbeitern und Studierenden über integrierte Portaldienste zur Verfügung gestellt. Demnach verfolgt das Projekt KIM-IDM das Prinzip der Dienstorientierung, im Sinne der Bereitstellung von Funktionalität über dedizierte Dienste [Höllrigl et al. 2007]. Wesentliche Identitätsdienste werden weiter unten ausgeführt.

Fundamental für die vorgestellte Architektur ist die Betrachtung des KIT als föderativer Verbund seiner organisatorischen Einheiten [Höllrigl et al. 2006]. Ein föderativer Verbund wird hierbei als eine Ansammlung verschiedener Organisationseinheiten in unterschiedlichen Sicherheits- und Vertrauensdomänen verstanden, welche eine Vertrauensbeziehung zueinander aufgebaut haben. Das Föderationsprinzip wird in diesem Zusammenhang keinesfalls als Dogma oder Rechtsform verstanden, sondern vielmehr als hilfreiches Instrument. Geeignet für den Einsatz an deutschen Universitäten ist der föderative Ansatz dadurch, dass deutsche Universitäten strukturell oftmals einer Holding-Struktur gleichen [Bazijanec et al. 2007], welche sich durch evolutionär unabhängig gewachsene organisatorische Einheiten auszeichnen. Als organisatorische Einheiten, welche wir auch als „Satelliten“ bezeichnen, werden bspw. die Zentrale Universitätsverwaltung, die Universitätsbibliothek und das SCC sowie die Fakultäten und Institute betrachtet. Verbunden mit der föderativen Sicht auf das KIT ist die Frage der Informationshaltung. Im Gegensatz zu einem verteilten Ansatz, wie er im Projekt KIM-IDM verfolgt wird, ist ein zentraler Ansatz als Basis für ein Identitätsmanagement im universitären Kontext weit verbreitet. Hierbei wird auf der Basis eines Meta Directory [Jahn & Stamms 2004], ein Identitätsmanagementsystem entwickelt. Dies setzt das Vorhandensein eines zentralen Schemas voraus. Durch die im KIT im Laufe der Jahre unabhängig voneinander gewachse-

nen IT-Infrastrukturen der organisatorischen Einheiten besteht bei dem zentralen Ansatz die Schwierigkeit, eine Harmonisierung der Daten durchzuführen.

Im Gegensatz zum zentralen Ansatz, bei welchem Anwendungen prozessbezogene Daten aus dem Meta Directory beziehen, erhalten Anwendungen bei einem föderativen Ansatz die Daten über Dienstschnittstellen oder auch über lokal vorliegende Datenquellen, welche durch ein Provisionierungssystem gespeist werden. Ein Dienst, der hier beispielhaft genannt werden kann, ist ein durch die Zentrale Universitätsverwaltung bereitgestellter Web Service, welcher im Rahmen des KIM-Projekts entwickelt wurde. Dieser Dienst kapselt die Informationssysteme, die zur Verwaltung der Mitarbeiter- und Studierenden in der Universitätsverwaltung eingesetzt werden. Ein Vorteil des föderativen Ansatzes liegt demnach in der Möglichkeit der dezentralen Datenhaltung. Da die einzelnen organisatorischen Einheiten in dem hier vorgeschlagenen Ansatz die Hoheit über ihre Daten behalten, ist es, von der Replikation weniger identitätsbezogener Attribute aus technischen Gründen abgesehen, nicht notwendig, die Daten in ein zentrales System zu replizieren. Darüber hinaus behalten die Satelliten die Kontrolle über sämtliche Zugriffsrechte und eine vollständig zentralisierte Zugriffskontrolle entfällt. Da die Satelliten ihre Datenbasen weiterhin lokal vorhalten, entfällt die direkte Abhängigkeit von einem zentralen System und die Autarkie der Satelliten bleibt erhalten. Ein weiterer entscheidender Vorteil, der für eine dezentrale Datenhaltung spricht, ist das Vorhandensein von Altsystemen. Diesen ist es oftmals nicht möglich benötigte Daten über ein zentrales System zu beziehen. Bei dem föderativen Ansatz können bestehende Systeme der Satelliten ohne tiefgreifende Veränderungen integriert werden, indem dedizierte Schnittstellen eingesetzt werden, über welche ein Datenzugriff ermöglicht wird.

Bevor das integrierte IDM detailliert beschrieben wird, gibt Abbildung 2.1 einen Überblick über das Zusammenspiel der zentralen Identitätsmanagementdienste mit den einzelnen Interaktionspartnern sowie über die „vertragliche“ Beziehung der Föderation. Für einen Benutzer stehen Identitätsmanagementdienste über ein Portal zur Verfügung, welches bspw. den Selbstbedienungsdienst des Identitätsmanagementsystems nutzt.

Aufgrund der gewählten Architektur kann das System sukzessiv erweitert und die Integration weiterer organisatorischen Einheiten schrittweise fortgeführt werden. Hierbei wird das föderative Identitätsmanagement als fortlaufender Prozess verstanden; die Föderation kann sich in diesem Kontext kontinuierlich weiterentwickeln.

Für die Integration der bestehenden Systeme wurden in der ersten Entwicklungsphase Identitätsmanagementdienste identifiziert, welche die verteilten Dienste des KIT unterstützen. Diese Dienste lassen sich in die Klassen der Selbstbedienungs- und Infrastrukturdienste unterteilen (Abbildung 2.2). Die Selbstbedienungsdienste („Self Services“) erlauben den Mitgliedern des KIT die individuelle Konfiguration und Änderung ihrer persönlichen Einstellungen. In diese Klasse fallen das Passwortmanagement, das nutzerkontrollierbare Single Sign-on bei Shibboleth und Dienste zur Administration nutzerbezogener Daten, welcher aktuell jedoch nur für Studierende angeboten wird. Diese Dienste ermöglichen sowohl Studenten als auch Mitar-

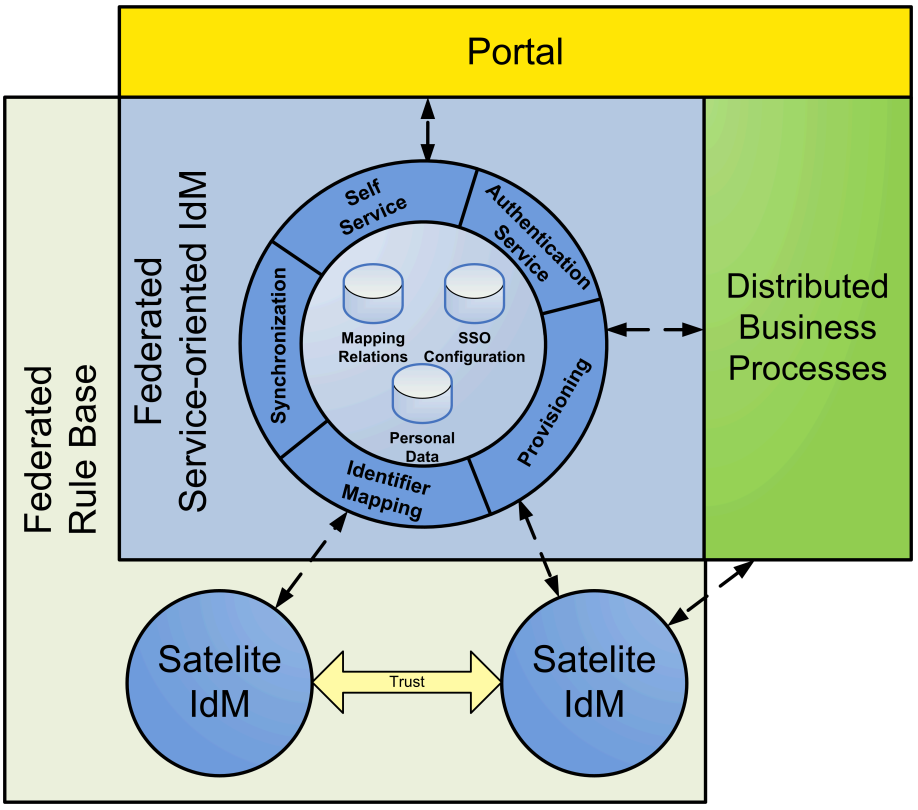


Abbildung 2.1: Übersicht der Dienste des IDM und deren Interaktionspartner

beitern schnell und flexibel Anpassungen vorzunehmen wie z.B. die Änderung des Passwortes für den KIT-weiten Portalzugang. Dadurch wird letztlich die Effizienz des Supports gesteigert sowie der bürokratische Aufwand gesenkt.

Die Infrastrukturdienste („Infrastructure Services“) bilden das Fundament für weitere Dienste und können sowohl von Diensten des Identitätsmanagements als auch von den Satelliten oder von weiteren Diensten, die das KIT anbietet, eingesetzt werden. Zu der Klasse der Infrastrukturdienste zählen der Abbildungsdienst („Identifier Mapping“, siehe Kapitel 6), der Authentifikationsdienst („Authentication Service“, siehe Kapitel 6) und die (De-)Provisionierung (siehe Kapitel 4).

Ein integraler Bestandteil der Infrastrukturdienste ist der Identifier Mapping Dienst, der die Grundlage dafür bietet, dass lokale Identitäten der einzelnen organisatorischen Einheiten aufeinander abgebildet werden können. Ein Mitglied des KIT verfügt typischerweise bei jeder Organisationseinheit über ein Nutzerkonto, die in der Regel untereinander nicht verknüpft sind. Somit ist es bspw. nicht möglich, dass ein Nutzer mit einem Nutzerkonto sowohl im SCC seine E-Mails verwalten als auch in der Universitätsbibliothek Bücher ausleihen kann. Um nun Dienste, die diese Grenzen von Organisationseinheiten überschreiten, anbieten zu können, wird eine Verbindung zwischen diesen über das KIT verteilte Accounts benötigt. Der Aufbau der hierfür notwendigen Mapping Relations, welche in der Literatur auch als Account Linking [Klingenstein 2007] bezeichnet werden, ist einer der grundlegendsten und ersten Schritte bei der Einführung eines Identitätsmanagementsystems. Wir begegnen dieser Herausforderung durch Einbindung des Nutzers in diesen Vorgang. Der

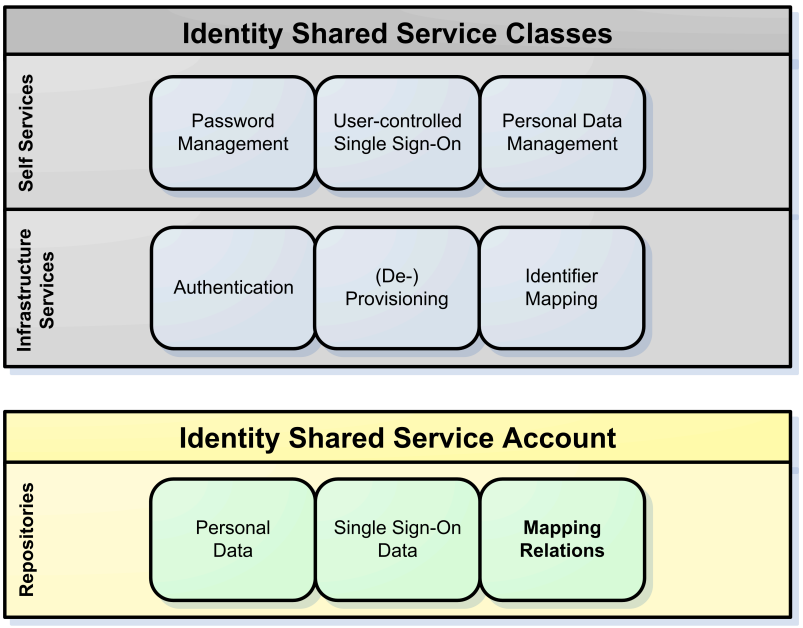


Abbildung 2.2: Klassifizierung der Identitätsmanagementdienste

Benutzer wird hierfür in den IDM-Prozess integriert mit der Perspektive, dass dieser bspw. zukünftig nur noch an einer Stelle seine personenbezogenen Daten ändern muss. In unserem Fall diente zunächst die Generierung einer organisationsweiten eindeutigen E-Mail-Adresse (vorname.nachname@kit.edu) als zusätzlicher Motivator. Das nutzerbasierte Identifier Mapping stützt sich auf mehrere Komponenten, auf die wir im Folgenden näher eingehen. Eine zentrale Komponente ist das Portal, welches zur Interaktion mit dem Benutzer dient und verschiedene Selbstbedienungsfunktionalitäten zur Verfügung stellt. Für die Nutzung des Portals wurden persönliche Accounts generiert, um eine eindeutige Authentifikation eines Nutzers sicherzustellen. Eine weitere Komponente ist eine Workflow Engine, die im Portal integriert wurde, um prozessgesteuerte Identitätsmanagementdienste zu realisieren. Diese prozessgesteuerten Identitätsmanagementdienste interagieren über das Portal mit dem Nutzer und über Web Service Schnittstellen mit den bereits vorhandenen Systemkomponenten in den Satelliten.

Die Synchronisation, die Provisionierung und die Deprovisionierung sind Dienste, die es ermöglichen, Datensätze über Satellitengrenzen hinaus gegebenenfalls zu verteilen und anschließend miteinander synchron zu halten. Durch die Nutzung dieser Dienste kann sichergestellt werden, dass z.B. beim Ausscheiden eines Nutzers aus dem KIT etwa gewisse Datensätze gelöscht und seine Zugriffsrechte entsprechend entzogen werden. Dies erleichtert sowohl dem Nutzer, der nun seine Daten nur noch an einer Stelle angeben und aktualisieren muss, als auch dem KIT die Administration der Benutzerdaten. Hierdurch werden auch verwaiste Nutzerkonten sowie mögliche unrechtmäßige Zugriffe vermieden.

3. Organisatorische Festlegungen

Identitätsmanagement erfordert neben einer IT-Architektur und entsprechenden technischen Lösungen erhebliche Anstrengungen im Bereich der Prozesse, der Organisation und der zu Grunde liegenden Regeln. Oft gilt es dabei eine Lösung zu finden, um die existierenden einfachen, teilweise zu einfachen, Prozesse in Einklang zu bringen mit den Regeln, die durch interne Vorgaben (engl. Governance) und rechtliche Regelungen vorgegeben sind. Aufgrund der hohen Relevanz der Rechtskonformität (engl. Compliance) sollen an dieser Stelle kurz auf rechtliche Regelungen eingegangen werden, die im Rahmen des Identitätsmanagement wesentlich sind. Erschwerend kommt hinzu, dass die derzeitige Rechtssituation (Stand: vor Oktober 2009) es erfordert, insbesondere bei übergreifenden Prozessen, die Regelungen für das Forschungszentrum als bundeseigene GmbH und der Universität als Körperschaft des öffentlichen Rechts des Landes Baden-Württemberg zu berücksichtigen. So war es häufig notwendig zunächst einmal festzustellen, für welchen Bereich welche Regelung angewandt werden muss und ob es nicht doch eine Lösung gibt, die allen Ansprüchen gerecht wird. Tabelle 3.1 zeigt exemplarisch einige rechtliche Regelungen, die von den Kooperationspartnern Beachtung finden müssen, und verantwortliche sowie beratende Gremien, welche für die Einhaltung zuständig ist.

Zusätzlich war es ein Anliegen des Projektteams, zu jedem Zeitpunkt auch die Anliegen der betroffenen Einrichtungen und Experten zu berücksichtigen. Zur Förderung dieses Austausches wurde ein so genannter IDM-Expertenkreis installiert. Ausgehend von diesem mehr IT-geprägtem Gremium wurde später durch die Projektleitung die Gründung des Arbeitskreises für IT-Sicherheit, Datenschutz und Rechtskonformität initiiert, der inzwischen ein wichtiges Beratungsgremium des Chief Information Officers (CIO) des KIT darstellt. Darüber hinaus wurden auch alle Interessengruppen immer wieder von dem Vorhaben und dessen Fortschritt informiert. So wurden bspw. bereits im Vorfeld des Projektes der Personalrat sowie das Studierendenparlament von Zielen des Vorhabens in Kenntnis gesetzt.

	Regelungen	**Gremien**
Universität	Landeshochschulgesetz Hochschuldatenschutzverord. Landesdatenschutzgesetz bereichsspezifische Regelungen	Beauftragter für den Datenschutz an der Universität / Zendas (Zentrale Datenschutzstelle der baden-württembergischen Universitäten)
	Personalvertretungsgesetz	Personalrat der Universität / Hauptpersonalrat im Geschäftsbereich des Wissenschaftsministeriums
Forschungszentrum	Bundesdatenschutzgesetz bereichsspezifische Regelungen	Datenschutzbeauftragter
	Betriebsverfassungsgesetz	Betriebsrat

Tabelle 3.1: Exemplarische Übersicht von rechtlichen Regelungen sowie verantwortlichen und beratenden Gremien zum Datenschutz und zur Mitbestimmung

Wesentlich beim Aufbau eines Identitätsmanagements ist die Identifikation der Personenkreise, die aufgenommen werden sollen sowie die autoritativen Quellen der Daten zu diesen Personenkreisen. Bereits sehr früh im Projekt wurde daher eine Analyse dieser Problematik angegangen. Es wurden dabei folgende Personenkreise und die für die entsprechende Datenhaltung relevanten Systeme zunächst nur für den Bereich Universität des KIT identifiziert, da zu diesem Zeitpunkt die Planungen für das KIT noch nicht gegenständlich waren:

1. Landesbeschäftigte
   - Beamte
   - Beschäftigte auf Landesstellen
   - Drittmittelfinanzierte Beschäftigte
2. Geprüfte wissenschaftliche Hilfskräfte
3. Studierende
   - Eingeschriebene Studierende
   - Gast- und Partnerstudierende
4. Doktoranden
5. Lehrbeauftragte
6. Abgeordnete von anderen Einrichtungen
7. Personen, die auf der Grundlage von Werkverträgen tätig sind
8. Stipendiaten

Die Zusammenstellung der Gruppen ergibt sich aus den Vorgaben des Landeshochschulgesetztes und der Grundordnung der Universität. Als autoritative Systeme wurden auf Seiten der Universität die IT-Systeme der Zentralen Verwaltung zugrunde gelegt. Die Studierendendaten werden aus dem HIS SOS und die Mitarbeiterdaten aus dem HIS SVA System bezogen. Bei der Ergänzung des Datenbestandes um die Mitglieder des Forschungszentrums wurde ein sehr pragmatischer Ansatz gewählt und eine möglichst umfassende und gleichzeitig mit einfachen, aber nachvollziehbaren Regeln versehene Quelle angebunden. Diese Quelle bestand im Nutzerverzeichnis des damaligen Instituts für Wissenschaftliches Rechnen (IWR) des FZK (heute SCC). Aus technischer Sicht handelt es sich dabei um ein Active Directory. Weiterführende technische Ausführungen finden sich in Kapitel 4.

Noch gibt es nicht für alle Personengruppen eindeutige Regelungen und autoritative Systeme, jedoch wird im Rahmen des KIT-Gründungsprozesses auch ein entsprechendes Projekt gestartet, in dem alle Personengruppen des KIT, sowohl im Bereich der Universität als auch im Bereich Großforschung abgedeckt werden. Aus technischer Sicht wäre ein eindeutiger personenbezogener Kennzeichner für alle Angehörigen des KIT wünschenswert, jedoch muss dieses Vorhaben auch mit den Persönlichkeitsrechten und dem Datenschutz der betroffenen Personen konform gehen. Organisatorisch hätte dies einen Paradigmenwechsel bei den datenverarbeitenden Stellen zur Folge, die derzeit mit der Verwaltung von personenbezogenen Daten befasst sind. Die diesbezüglichen Prozesse müssten von der derzeit praktizierten Vorgangsorientierung zu einer Personenorientierung umgestellt werden.

Sollen über das Identitätsmanagement auch Berechtigungs- bzw. Zugriffsentscheidungen getroffen werden, so wird das in der Regel über vorhandene Attribute geführt. So gilt z.B. am KIT die Regelung, dass Universitätsprofessoren ein Mobiltelefon aus einer Kooperation mit Vodafone erhalten oder dass nur wissenschaftliche Mitarbeiter sich zu Kompetenzfeldern eintragen dürfen. Das setzt aber ein über die gesamte Einrichtung gemeinsames Verständnis dieser Attribute voraus. War das im Fall der Universitätsprofessoren recht einfach, da es gesetzliche Vorgaben zu diesem Begriff gibt, so musste am KIT für den Begriff des Wissenschaftlers erst einmal ein gemeinsames Verständnis gebildet werden. Dieser Prozess war durchaus komplex und ist bis heute nicht endgültig abgeschlossen. So gibt es derzeit keine eindeutigen und damit für ein Berechtigungssystem nachvollziehbaren Regeln, wie in diesem Zusammenhang mit Stipendiaten umgegangen wird. Selbst wenn ein über die Organisation hinweg einheitliches und eindeutiges Verständnis vorhanden ist, wie bspw. bei der Einordnung von Doktoranden als Wissenschaftler, so muss immer anschließend geklärt werden, welches System die autoritative Quelle für diese Information ist und wie die Abbildung der dort geführten Identitäten im IDM stattfinden kann. Dies ist in der Regel auch kein einmaliger Vorgang, sondern es müssen die Prozesse so umgestaltet werden, dass diese Beziehung dauerhaft und nachvollziehbar gepflegt wird. Grundsätzlich muss jedes Attribut, das zu einer Berechtigungsentscheidung dient hinsichtlich dieser Entscheidung hinterfragt werden, da häufig vorhandene Attribute verwendet werden sollen, die aber bei genauerer Betrachtung sich als ungeeignet

herausstellen, entweder mangels der erforderlichen Qualität in der Pflege oder weil die Überdeckung mit dem Personenkreis nicht ausreicht.

Im Rahmen des Projektes KIM-IDM wurden Prozesse für folgende Personengruppen festgelegt und umgesetzt:

1. *Provisionierung und Deprovisionierung der Mitarbeiter am Campus Süd (Universitätsteil) und des Campus Nord (FZK-Teil)*
   Die technische Realisierung der Provisionierung und Deprovisionierung wird in Abschnitt 4.1.3 beschrieben. Am Campus Süd mussten dabei insbesondere folgende organisatorischen Änderungen vorgenommen werden:

   - Alle Lehrbeauftragten werden über das Personalsystem der Universität gepflegt.
   - Es wurde eine Regelung für Partnerinstitutionen, wie bspw. das „International Department“ gefunden.
   - Derzeit wird ein Verfahren eingeführt, wie Mitarbeiter, die nicht in einem Beschäftigungsverhältnis stehen, in das IDM aufgenommen werden.

2. *Provisionierung und Deprovisionierung der Studierenden*
   Hier war es notwendig den Provisionierungs-Prozess für die Vergabe des Accounts zu den Dienstleistungen des SCC zu verlagern. Bisher war es so, dass die Studierenden persönlich erscheinen mussten um die Akzeptanz der Nutzungsbedingungen durch Unterschrift zu quittieren. Durch die Umstellung auf das Studierendenportal müssen nun auch Studierende unmittelbar nach der Immatrikulation, zu dem Zeitpunkt zu dem sie sich noch nicht am Studienort aufhalten, Zugriff auf das Portal und damit einen gültigen Account erhalten. Dazu wurde ein Zweibrief-Verfahren eingeführt. In zwei unabhängig voneinander versendeten Briefen erhält der Studierende jeweils Informationen zu seinem Account. Diese Informationen müssen in einem Registrierungsvorgang zusammen geführt werden. Gleichzeitig müssen in diesem Vorgang auch die Nutzungsbedingungen akzeptiert werden, erst dann erfolgt der Zugang.

   Als problematisch stellte sich die Deaktivierung des Accounts heraus. Bedingt durch die Studiengebühren exmatrikulieren sich die Studierenden häufig bereits zu einem Zeitpunkt zu dem noch nicht alle Prüfungsleistungen bewertet sind. Zunächst wurde die Exmatrikulation als Zeitpunkt für die Deaktivierung und damit in der Folge die Deprovisionierung des Accounts festgelegt. Aus obigen Erwägungen ist diese Vorgehensweise nicht mehr haltbar. Es wird derzeit eine Regelung erarbeitet, die auf der einen Seite den geänderten Verhältnissen Rechnung trägt und gleichzeitig den gesetzlichen Vorgaben durch Datenschutz und Hochschulgesetz gerecht wird.

Es stehen nun im Rahmen eines Folgeprojektes weitere organisatorische Änderungen an, die zum einen die erreichten Änderungen weiterhin verfeinert. Insgesamt müssen sich alle personenbezogene Verwaltungsprozesse dahin gehend organisieren, dass nicht mehr der Verwaltungsvorgang sondern die Person im Vordergrund steht.

4. Identitätsmanagementsystem

In diesem Kapitel wird zunächst im Abschnitt 4.1 ein Überblick über die im Projekt KIM-IDM realisierte Lösung gegeben. Danach werden in Abschnitt 4.2 die angebundenen Ressourcen beschrieben. Basierend auf diesen an das KIT-weite Identitätsmanagementsystem angekoppelten Systemen werden abschließend in Abschnitt 4.3 die implementierten Prozesse zur Durchführung der Provisionierungsaufgaben erläutert.

4.1 Überblick

Abbildung 4.1 gibt einen Überblick über das im Projekt KIM-IDM realisierte, produktive Identitätsmanagementsystem und die angekoppelten Systeme, die in Quell- und Zielsysteme eingeordnet werden können. Die grauen Pfeile deuten derzeit noch nicht produktiv geschaltete Verbindungen an. Als Basis für die Realisierung des KIT-weiten Identitätsmanagementsystem (*KIT-IDM*, zentral skizziert) wird der Sun Identity Manager in der Version 7.0 (20061020) genutzt.

4.1.1 Quellsysteme

Das KIT-weite Identitätsmanagementsystem (*KIT-IDM*, zentral skizziert) bezieht die Daten für Mitarbeiter des Campus Nord, Mitarbeiter des Campus Süd und Studierende aus jeweils einer eigenen Datenquelle. Diese 3 Datenquellen sind somit autoritativ für jeweils eine dieser Nutzergruppen am KIT und werden tagesaktuell nach Änderungen abgefragt.

Aus dem HIS-System werden die Daten der Mitarbeiter des Campus-Süd (*HIS SVA*) und Studierendendaten (*HIS SOS*) bezogen. Die Daten für Mitarbeiter des Campus Nord (Forschungszentrum Karlsruhe, FZK) werden aktuell aus dem dortigen Active Directory (*FZK AD*) bezogen. Eine Anbindung des SAP-HR-Systems des Campus Nord soll hier in Zukunft das FZK-AD als autoritative Quelle ablösen. Dies wird derzeit vorbereitet und die Fertigstellung wird bis zum Ende des Jahres 2009 erwartet.

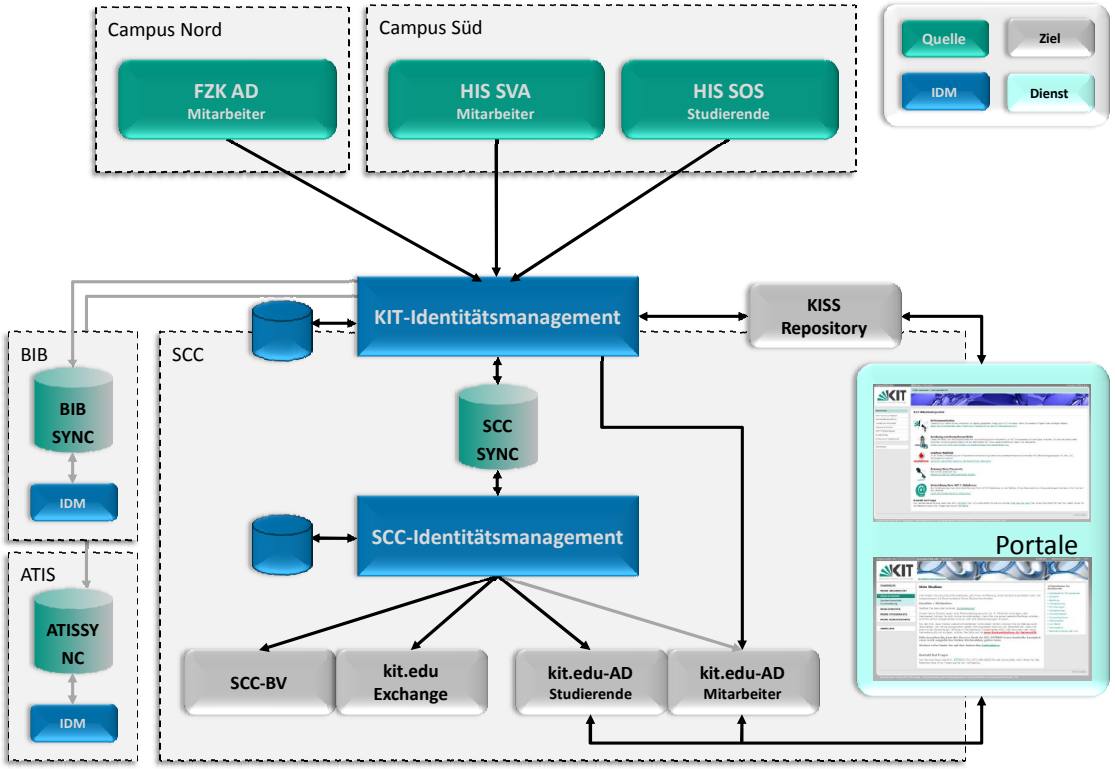


Abbildung 4.1: Übersicht der Datenflüsse des KIT-Identitätsmanagementsystems

4.1.2 Zielsysteme

Für jede Institution, respektive jeden Satelliten, ist eine ausgewiesene Datenbank als Schnittstelle zum KIT-IDM (*SYNC*) vorgesehen. Exemplarisch ist dies in Schaubild 4.1 für das am Steinbuch Centre for Computing (SCC) in diesem Jahr in Betrieb genommene Identitätsmanagementsystem (*SCC-IDM*) dargestellt. Das SCC-IDM wurde auf Basis der im Projekt KIM-IDM gewonnenen Erfahrungen erstellt.

Hierbei werden die vom KIT-IDM in der Schnittstelle des SCC (*SCC SYNC*) angelegten Nutzerdaten vom SCC-IDM ausgelesen, weiter verarbeitet, mit weiteren Daten angereichert (bspw. Unix-spezifischen Nutzerkontodaten) und in die am SCC-IDM angekoppelten Zielsysteme überführt. Weiterhin können vom SCC-IDM wiederum Daten, die über das KIT-IDM an weitere Systeme anderer Institutionen bzw. Satelliten verteilt werden müssen wie bspw. eine *kit.edu*-E-Mail-Adresse, in die SCC-SYNC Datenbank provisioniert werden. Demnach kann aus Sicht des KIT-IDM eine SYNC-Datenbank sowohl eine Quelle als auch ein Ziel für Identitätsdaten darstellen.

Ähnliche Entwicklungen führen zur Integration weiterer Satelliten wie der Abteilung Technische Infrastruktur (ATIS) und Universitätsbibliothek Karlsruhe (BIB), wobei sich die Anbindung der Bibliothek noch in der Planungsphase befindet. Die Anbindung der ATIS ist bereits in einer Testumgebung erfolgreich umgesetzt.

Zwei Systeme sind als Zielsysteme gesondert zu betrachten. Zum einen nimmt die als KISS-Repository bezeichnete Datenbank eine besondere Stellung ein. Das KISS-Repository enthält Nutzerinformationen, die KIT-weite Prozesse ermöglichen, wie sie im Mitarbeiterportal *intra.kit.edu* und Studierendenportal *studium.kit.edu* bereits

realisiert wurden. Zum anderen werden zurzeit die Mitarbeiter-Konten im *kit.edu*-Active Directory (*kit.edu*-AD Mitarbeiter) direkt durch das KIT-IDM angelegt, aktualisiert und gelöscht. Zukünftig wird diese Aufgabe vom SCC-eigenen Identitätsmanagementsystem übernommen.

4.1.3 Provisionierungs-Prozesse

Nachfolgend werden die Prozesse erläutert, die prinzipiell im Identitätsmanagementsystem auf einzelne Identitäten angewendet werden können:

- **Create.** Falls in einer angekoppelten Ressource ein neuer Datensatz erkannt wird, bspw. ein neuer Eintrag in einer angekoppelten Datenbank, kann im Identitätsmanagementsystem ein Create-Prozess gestartet werden, der in den hierfür definierten Zielsystemen ein Nutzerkonto mit den entsprechenden Identitätsdaten und Rechten anlegt.

- **Update.** Änderungen von Nutzerdaten in einem Quellsystem an bereits im Identitätsmanagementsystem bekannten Nutzern werden durch einen Update-Prozess behandelt. Dieser sorgt für die Aktualisierung dieser Daten basierend auf den im Identitätsmanagementsystem hinterlegten Regeln.

- **Delete.** Die Detektion eines gelöschten Nutzers in einem Quellsystem hat meist das sofortige Löschen der entsprechenden Nutzerdaten oder zumindest den Entzug der Rechte eines Nutzers in angeschlossenen Systemen durch einen Delete-Prozess zur Folge. Anschließend ist dem Identitätsmanagementsystem dieser Nutzer nicht mehr bekannt.

- **Disable.** Anstatt einen Nutzer sofort nach Erkennen eines Löschvorgangs in einem Quellsystems von allen Systemen zu entfernen, können zunächst in einem Disable-Prozess die Konten des Nutzers deaktiviert werden. Hierbei werden einem Nutzer bestimmte Zugriffsrechte entzogen, ohne dass alle Daten gelöscht werden müssen. Das Auslösen eines Delete-Prozesses kann anschließend basierend auf Regeln, bspw. der Ablauf einer bestimmten Frist, durchgeführt werden.

- **Enable.** Falls ein Nutzer durch einen Disable-Prozess deaktiviert wurde, kann mit Hilfe eines Enable-Prozesses der Nutzer wieder aktiviert werden, indem der Nutzer die entzogenen Rechte in den angekoppelten Systemen wieder zugewiesen bekommt.

Die im KIT-IDM implementierten Prozesse Create (C), Update (U), Disable (Di), Enable (E), Delete (D) für die angekoppelten Quell- und Zielsysteme stellt Tabelle 4.1 als Übersicht dar. Bspw. werden beim Erkennen einer Änderung eines Attributes im Quellsystem HIS SOS entsprechend die Nutzerdaten in der SCC- und ATIS-Schnittstelle und im KISS-Repository aktualisiert oder beim Anlegen eines

	Zielsysteme				
Quellsysteme	kit.edu Active Directory	kit.edu Exchange Server	SCC-IDM	ATIS-IDM	KISS-Repository
HIS SOS	-	-	C,U,Di,E	U,Di,E,D	C,U,Di,E
HIS SVA	C,U,Di,E	C,U	C,U,Di,E	-	C,U,Di,E
FZK AD	C,U,Di,E	C,U	C,U,Di,E	-	C,U,Di,E
SCC-IDM	-	-	-	-	U
Portale (via SPML)	C,U	C,U	U	-	U
ATIS-IDM (via SPML)	-	-	-	C,U	-

Tabelle 4.1: Angekoppelte Quell- und Zielsysteme in KIT-IDM und implementierte Provisionierungs-Prozesse

Nutzers im FZK AD wird ein Konto bzw. die Nutzerdaten im *kit.edu*-Active Directory, im *kit.edu*-Exchange Server, in der SCC-IDM-Schnittstelle und im KISS-Repository angelegt.

Darüber hinaus steht mit einer Service Provisioning Markup Language (SPML)-Schnittstelle[1] eine standardisierte Möglichkeit zum Anstoßen von Prozessen zur Verfügung. Diese Schnittstelle wird vom Mitarbeiter- und Studierendenportal und vom Identitätsmanagementsystem der ATIS verwendet, um Prozesse im KIT-IDM auszulösen, siehe Abschnitt 6.1.8. Zur Wahrung der Nachweisbarkeit und Konsistenz werden bisher Nutzer nur automatisiert deaktiviert. Das Löschen der Daten ist technisch unproblematisch, jedoch bedarf es hier noch der Klärung entsprechender organisatorischer Festlegungen.

Die vorgestellten Prozesse können jeweils grundsätzlich in 2 getrennte Phasen unterteilt werden. Zunächst können Bestätigungen (so genannte *Approvals*) von den zuständigen *Approvern*, bspw. Managern oder Vorgesetzten, eingeholt werden, so dass für diese Person tatsächlich in den gewünschten Systemen ein Nutzerkonto mit entsprechenden Identitätsdaten und Rechten angelegt werden darf. Hierfür werden die Approver benachrichtigt, worauf diese sich im Identitätsmanagementsystem anmelden und den Approval akzeptieren oder ablehnen können. Diese Phase kann bspw. durch die Berücksichtigung von Urlaubsvertretungen für die Approver komplexer gestaltet werden. Diese Phase wird zurzeit in den Prozessen des KIT-weiten Identitätsmanagementsystems noch nicht implementiert. Die zweite Phase ist die

[1] SPML ist ein XML-basiertes Framework, um Provisionierungsaufgaben interoperabel zu gestalten und diese Standard-basiert integrieren zu können. Hierfür werden Nachrichtentypen und ein Protokoll zum Austausch dieser Nachrichten spezifiziert. Die Entwicklung von SPML wird durch das Standardisierungsgremium OASIS getrieben [WWW SPML 2009].

eigentliche Durchführung der Anlege-, Änderungs- oder Lösch-Operationen, die im Fehlerfall mehrmals Durchlaufen werden kann. Bei einer dauerhaften Störung sollte dementsprechend der Prozess angehalten werden, eine Lösung durch den Support des Zielsystems erfolgen und eine Wiederholung der fehlgeschlagenen Operationen durchgeführt werden.

Entscheidend für die korrekte Durchführung der Prozesse ist das Vorhandensein einer Abbildung der Informationsmodelle der angekoppelten Ressourcen. Hierfür werden in Identitätsmanagementsystemen Abbildungen zwischen den lokalen Nutzerattributen definiert. Dies kann auch die Umwandlung eines Attributs in einen anderen Datentyp beinhalten.Die Abbildung der Attribute für das KIT-weite Identitätsmanagementsystem wird im nachfolgenden Abschnitt 4.2 detailliert erläutert.

4.1.4 Entwicklungsprozess

Das Projekt KIM-IDM verfolgt einen drei-stufigen Ansatz für das Deployment entwickelter Funktionalitäten. Eine Neuentwicklung, unter welcher auch ein neu angeschlossenes System zu verstehen ist, wird zunächst prototypisch auf dem Entwicklungssystem eingebunden und entwickelt. Eine Release-fähige Entwicklung wird daraufhin auf das Testsystem umgezogen. Nach der erfolgreichen Durchführung von Tests auf dem Gesamtsystem wird der aktuelle Stand des Testsystems auf das Produktivsystem eingespielt. Eine Voraussetzung für dieses Vorgehen ist die Vergleichbarkeit des Entwicklungs-, Tests- und Produktivsystem, hinsichtlich der angebundenen Ressourcen, umgesetzten Prozessen und Diensten, als auch deren Konfiguration.

4.2 Angebundenen Ressourcen

Es folgt eine Beschreibung aller an das KIT-IDM, d.h. aus technischer Sicht den eingesetzten Sun Identity Manager (Sun IdM), gekoppelten Ressourcen, die als Quelle oder Ziel der Provisionierung dienen, mit jeweils folgenden Elementen:

- Beschreibung der jeweiligen Ressource
- Beschreibung aller betroffenen Attribute
- Verweise auf die zugehörigen Prozesse
- Konfiguration im Sun Identity Manager

4.2.1 HIS SOS

4.2.1.1 Beschreibung

Die hauptsächliche Quelle für Studierendeninformationen ist das System HIS SOS der Universitätsverwaltung. In diesem System werden die Identitätsdaten von Studierenden nach deren Immatrikulation von Mitarbeitern der Zentralen Universitätsverwaltung eingepflegt. Der Sun Identity Manager hat nicht direkten Zugriff auf die SQL-Datenbank des HIS-Systems, sondern es wird regelmäßig ein Abzug der in Tabelle 4.2 beschriebenen Attribute in eine .csv-Datei erzeugt. Diese Datei dient als eigentliche Quelle für den Sun Identity Manager.

Attribut	Beschreibung	Beispiel
Matrikelnummer	Identifikator eines Studierenden an der Universität Karlsruhe	1234567
Geschlecht	Geschlecht eines Studierenden	männlich
Vorname	Alle Vornamen eines Studierenden	Hans Joachim
Nachname	Alle Nachnamen eines Studierenden, auch Adelstitel	von Rechtenfeld
Studienfach	Studienfach eines Studierenden	1
Status	Beschreibt den aktuellen Zustand eines Studierenden an der Universität. Ein Studierender der sich nicht bzw. nicht mehr in der aktuellen Datenlieferung befindet, gilt als exmatrikuliert. Es gibt hierbei keine Verzögerung seitens der Verwaltung. Mögliche Werte sind Beurlaubung, Ersteinschreibung, Neueinschreibung, Rückmeldung	Rückmeldung
Straße	Straße inklusive Hausnummer der Postadresse eines Studierenden	Zirkel 2
Postleitzahl	Postleitzahl des Wohnsitzes	76128
Stadt	Ortsnamen des Wohnsitzes	Karlsruhe
Land	Länderkennung des Wohnsitzes	D
Adressnachtrag	Zusätzliche Informationen zur postalischen Adresse	K1 E201
Fricard-ID	Intern unverschlüsselt gespeicherte Nummer der FriCard eines Studierenden	2306304276
Fricardchip-ID	Aufgedruckte Nummer der FriCard eines Studierenden	158001131941

Tabelle 4.2: Attribute der Ressource HIS SOS

4.2.1.2 Attribute

Eine Übersicht über die Attribute dieser Ressource bietet Tabelle 4.2.

4.2.1.3 Prozesse

Wie bereits erwähnt ist die HIS SOS die autoritative Datenquelle für die in Tabelle 4.2 aufgeführten Attribute. Bei den Prozessen ist nur die Provisionierung der Studierenden unmittelbar von diesem System abhängig.

4.2.1.4 Konfiguration im Sun Identity Manager

Abbildung 4.2 gibt eine Übersicht der Resource Parameters der Datenquelle HIS SOS. Da diese Datenquelle über ein Flatfile angebunden wird, muss hier im Wesentlichen nur der Speicherort, das Format und optional das Encoding angegeben werden. Das Default Encoding entspricht den Einstellungen der Java Virtual Maschine(JVM), in unserem Fall UTF-8.

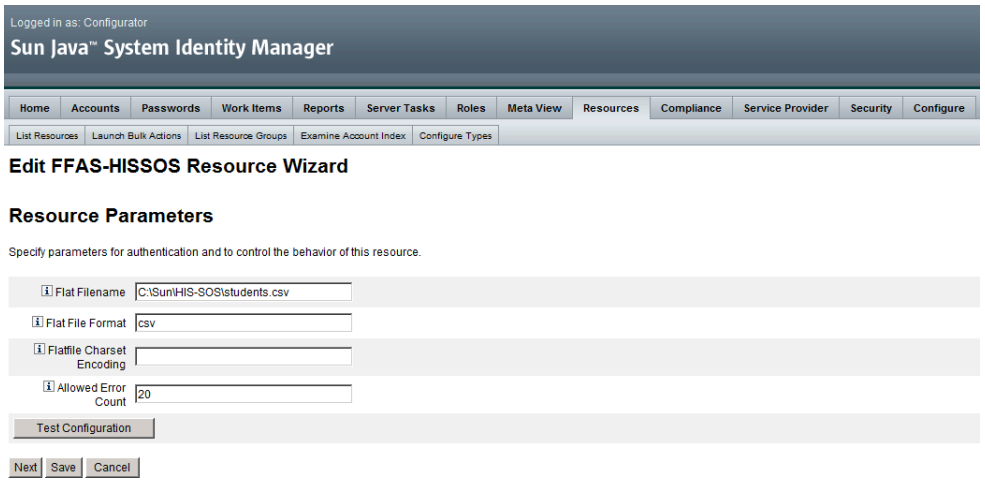


Abbildung 4.2: Resource Parameters der Datenquelle HIS SOS

Abbildung 4.3 gibt eine Übersicht der Account Attribute der HIS SOS Datenquelle. Auf der rechten Seite stehen die Attributbezeichner der .csv-Datei. Diese werden mit den auf der linken Seite stehenden Identity System Attribute des Sun IdM korreliert. Zusätzlich ist es möglich ergänzende Angaben zu den einzelnen Attributen zu machen.

Die Identity System Parameters (siehe Abbildung 4.4) ermöglichen die genaue Konfiguration der Ressource HIS SOS im Sun IdM. Hierbei ist es bspw. möglich einen „Retry Mechanismus“ zu konfigurieren, eine Passwort Richtlinie anzugeben, usw.

Abbildung 4.5 ermöglicht die Konfiguration der Synchronisation. Hierbei kann bspw. die Häufigkeit, das für einen Diff benötigte eindeutige Attribut aus der Datei oder die Lokation des Log-Files und dessen Log-Level angegeben werden.

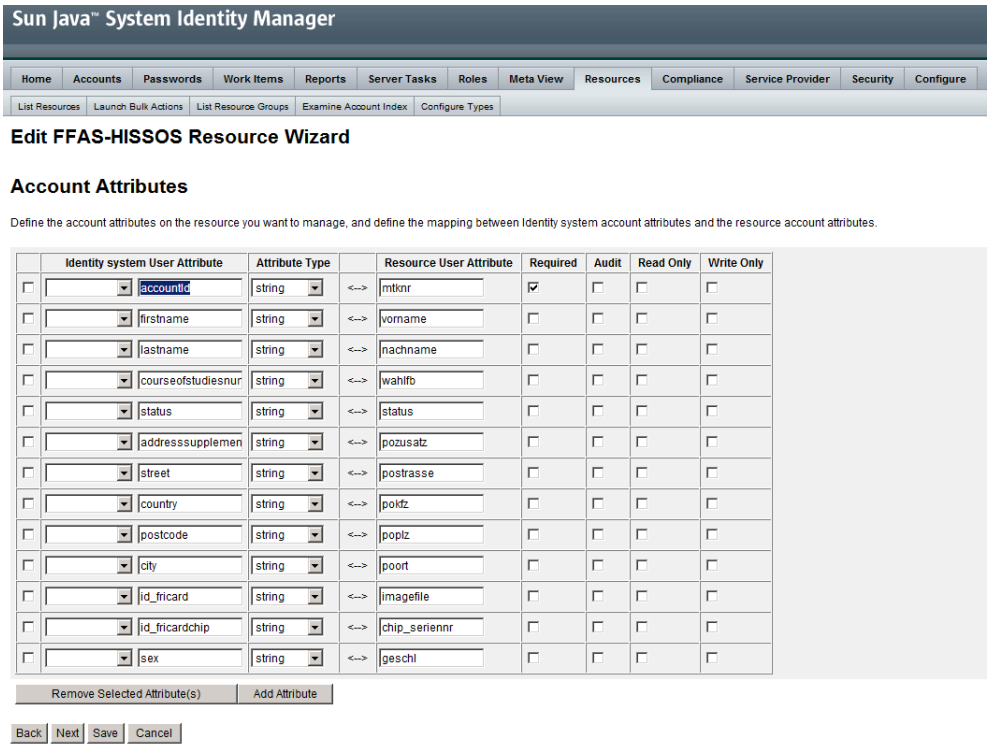


Abbildung 4.3: Abbildung der Account Attribute auf die IDM-internen Attribute der Datenquelle HIS SOS

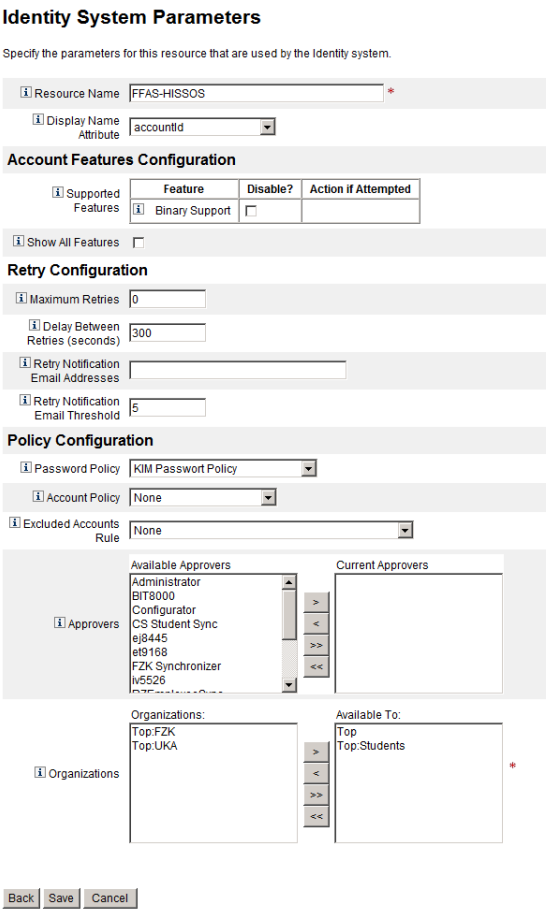


Abbildung 4.4: Parameter für das Identity System der Datenquelle HIS SOS

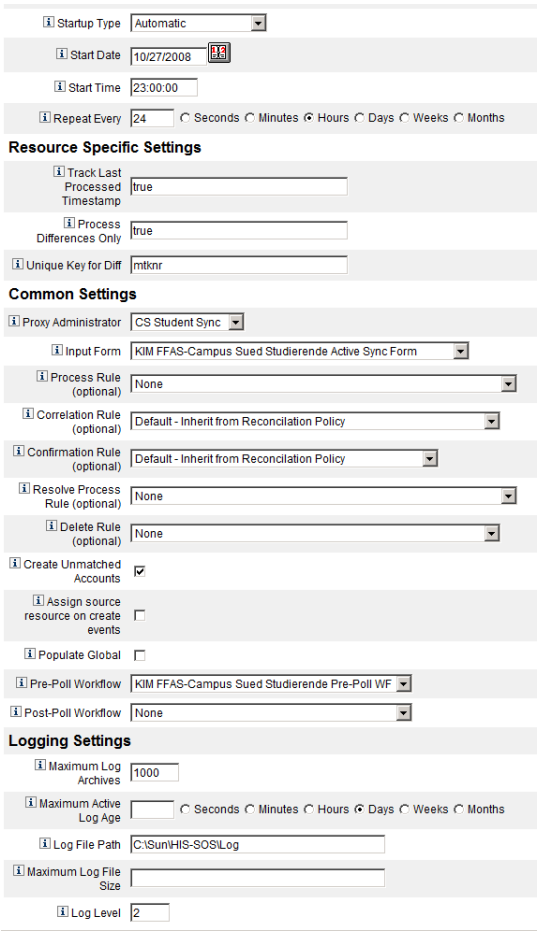


Abbildung 4.5: Synchronization Policy der Datenquelle HIS SOS

Attribut	Beschreibung	Beispiel
SVA-Identifikator	Identifikator eines Mitarbeiters an der Universität Karlsruhe	123456
Geschlecht	Geschlecht eines Mitarbeiters	M
Anrede	Anrede eines Mitarbeiters	Prof.Dr.-Ing.
Vorname	Alle Vornamen eines Mitarbeiters	Hans Joachim
Nachname	Alle Nachnamen eines Mitarbeiters, auch Adelstitel	von Rechtenfeld
Instituts-Nummer	Nummer des Instituts an dem ein Mitarbeiter beschäftigt ist	41003000
Adressinformationen	Adresse des Instituts oder der Einrichtung in der der Mitarbeiter beschäftigt ist	Universitätsbibliothek
Kostenstelle	Kostenstelle eines Mitarbeiters	41003011

Tabelle 4.3: Attribute der Ressource HIS SVA

4.2.2 HIS SVA

4.2.2.1 Beschreibung

In der Ressource HIS SVA werden die Identitätsdaten von Mitarbeitern der Universität gepflegt.

4.2.2.2 Attribute

Tabelle 4.3 zeigt die relevanten Attribute der HIS SVA Datenquelle gemeinsam mit deren Beschreibung und Beispielen.

4.2.2.3 Prozesse

Die HIS SVA Datenquelle ist für die in Tabelle 4.3 aufgeführten Attribute die autoritative Quelle. Der einzige abhängige Prozess ist die Provisionierung der Campus Süd Mitarbeiter.

4.2.2.4 Konfiguration im Sun Identity Manager

Die Mitarbeiter der Universität sind analog zu HIS SOS (vgl. Abschnitt 4.2.1) über einen FlatFileAdapter angebunden. Abbildung 4.6 stellt die Einstellungen für die Ressourcen Parameter dar.

Abbildung 4.7 gibt eine Übersicht der Account Attributes der Datenquelle HIS SVA. Auf der rechten Seite stehen die Attributbezeichner der .csv-Datei. Auf der linken Seite sind die entsprechenden Attribute innerhalb des Sun IdM aufgeführt.

Die Konfiguration der Identity System Parameter, siehe Abbildung 4.8, spezifiziert die Passwort Richtlinie, Retry Mechanismen usw. der Ressource HIS SVA.

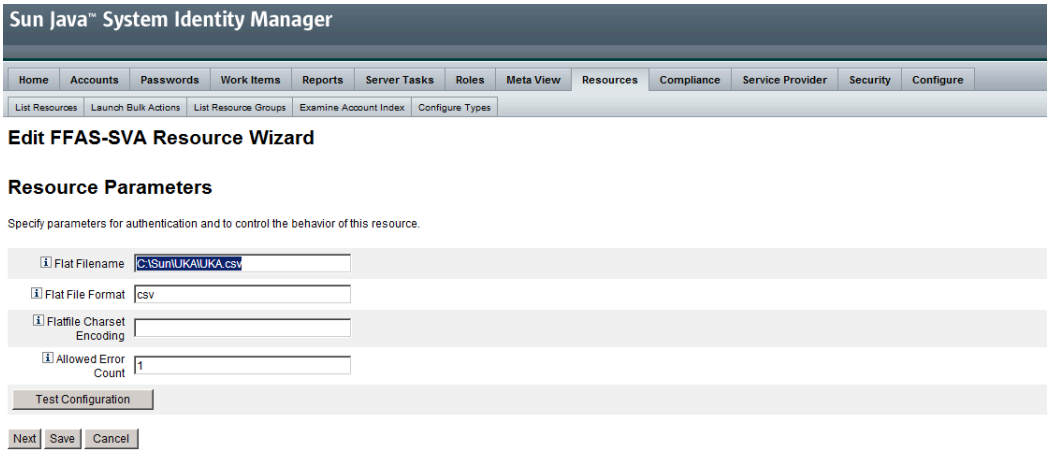


Abbildung 4.6: Resource Parameters der Datenquelle HIS SVA

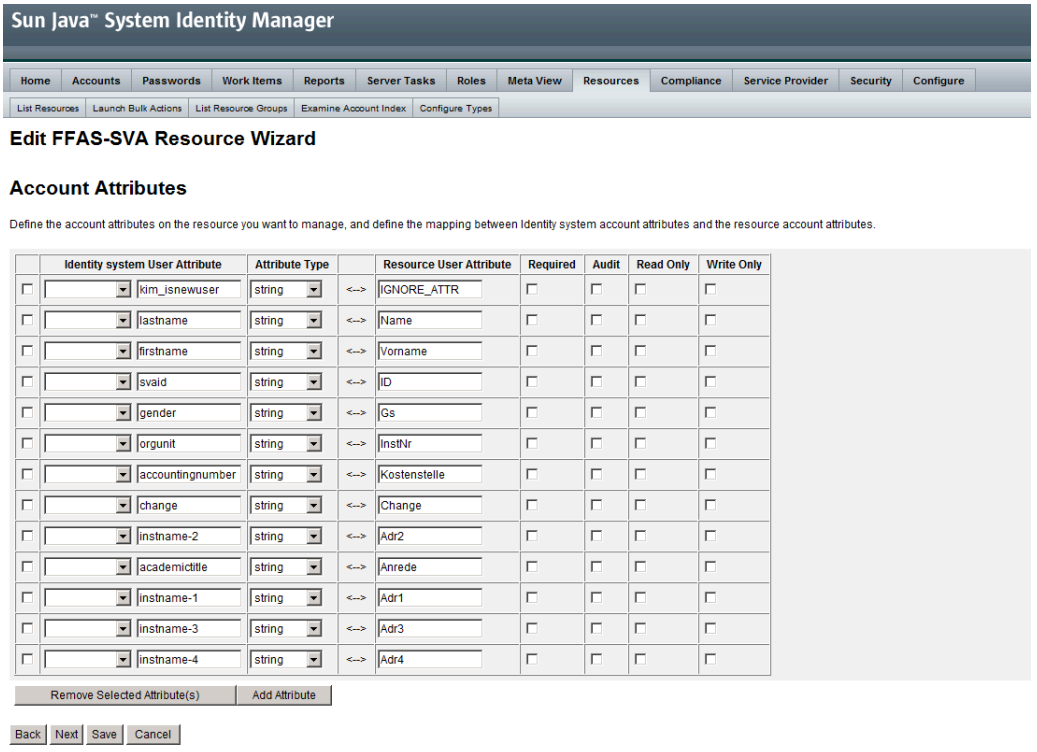


Abbildung 4.7: Account Attribute der Datenquelle HIS SVA

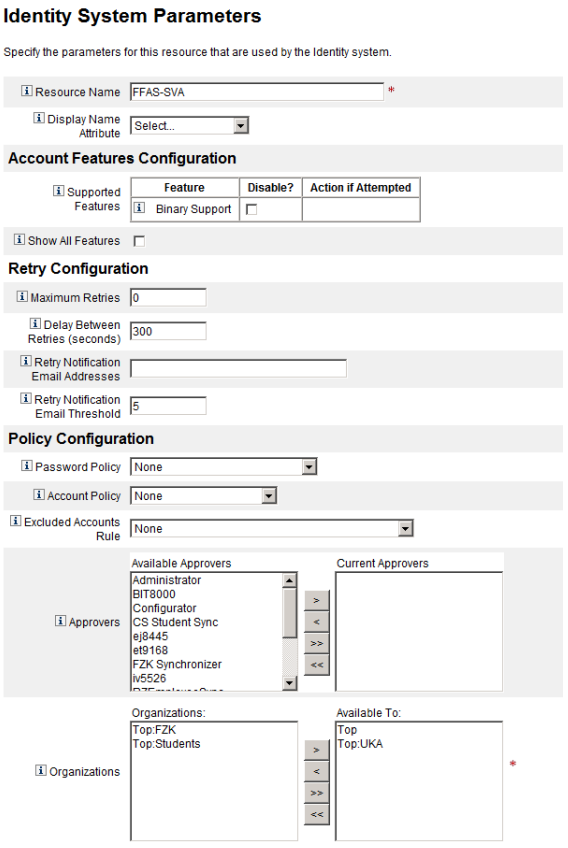


Abbildung 4.8: Identity System Parameters der Datenquelle HIS SVA

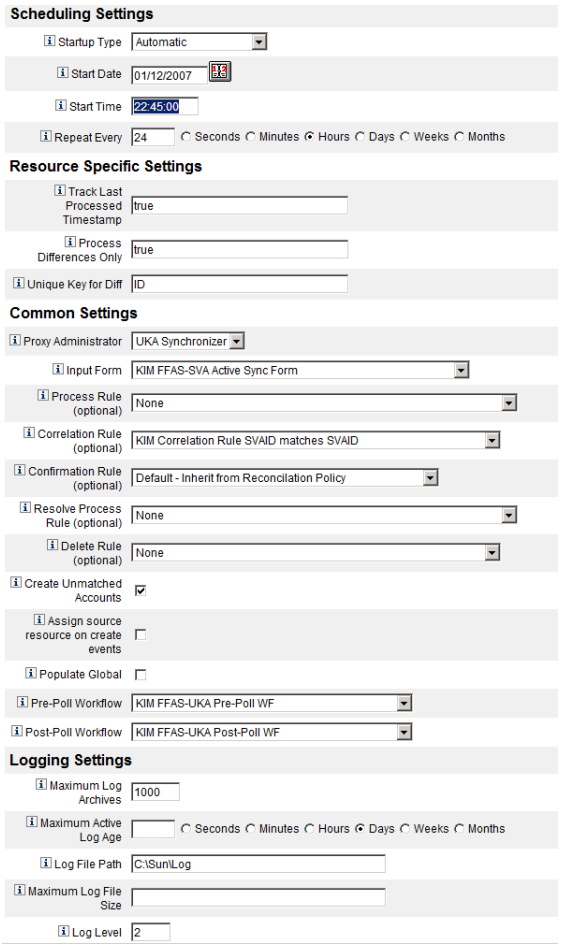


Abbildung 4.9: Synchronization Policy der Datenquelle HIS SVA

Abbildung 4.9 zeigt die Einstellungen für die Synchronisation. Neben dem Zeitplan für die Synchronisation werden hier auch bspw. die Log-Einstellungen angegeben.

4.2.3 FZK Active Directory

4.2.3.1 Beschreibung

Das FZK Active Directory dient als Quelle für die Mitarbeiter des Forschungszentrums. Es werden die Domänen ka.fzk.de, bti.fzk.de, gap.fzk.de, hs.fzk.de, iai.fzk.de und irs.fzk.de ausgelesen und alle Benutzerkonten mit Postfach aus dem FZKA-Forest synchronisiert. Der Export der Daten erfolgt über ein Script, das eine .csv-Datei anlegt, die als eigentliche Quelle für den Sun IdM dient.

4.2.3.2 Attribute

Tabelle 4.4 gibt eine Übersicht der Attribute der Datenquelle FZK Active Directory.

4.2.3.3 Prozesse

Die Ressource FZK Active Directory ist die autoritative Datenquelle für die in Tabelle 4.4 aufgeführten Attribute. Die Provisionierung der Campus Nord Mitarbeiter ist von dieser Datenquelle abhängig.

Attribut	Beschreibung	Beispiel
Distinguished Name	Distinguished Name eines Mitarbeiters	CN=Nach,Vor,...,DC=de
Nachname	Nachname eines Mitarbeiters	von Rechtenfeld
Vorname	Vorname eines Mitarbeiters	Hans
Telefonnummer	Telefonnummer eines Mitarbeiters	8000
DisplayName	Vornamen und Nachnamen eines Mitarbeiters zur Darstellung in Applikationen	Schell, Frank
ObjectSid	Eindeutige Active Directory-Schlüssel eines Mitarbeiters	010...ab2000
SAMAccountName	Bisheriger SAMAccountName eines Mitarbeiters	frank.schell
UserPrincipalName	Bisheriger Userprincipalname eines Mitarbeiters	frank.frueh@iai.fzk.de
E-Mail	Bisherige E-Mail-Adresse eines Mitarbeiters	frank.schell@extern.fzk.de
Unix-Informationen	Mapping der Unix-Attribute der Domänen ka.fzk.de und irs.fzk.de (msSFU30UidNumber, msSFU30GidNumber und msSFU30LoginShell)	/bin/ksh

Tabelle 4.4: Attribute der Ressource FZK Active Directory

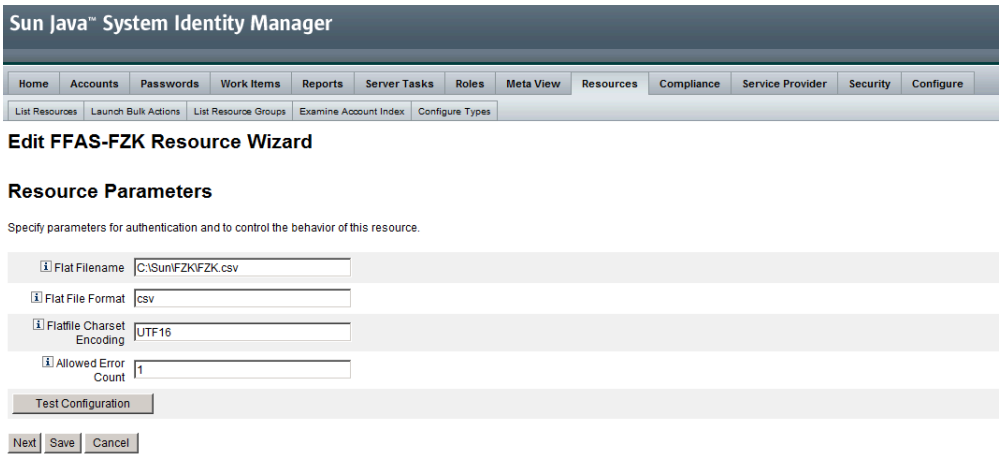


Abbildung 4.10: Resource Parameters der Datenquelle FZK Active Directory

4.2.3.4 Konfiguration im Sun Identity Manager

Die Mitarbeiter des Forschungszentrums Karlsruhe sind analog zu den Mitarbeitern der Universität über einen FlatFileAdapter angebunden. Abbildung 4.10 zeigt die Einstellungen für die Ressourcen Parameter.

Abbildung 4.11 gibt eine Übersicht der Account Attributes des FZK Active Directory. Auf der rechten Seite stehen die Attributbezeichner der .csv-Datei. Auf der linken Seite sind die entsprechenden Attribute innerhalb des Sun IdM aufgeführt.

Die Konfiguration der Identity System Parameter, siehe Abbildung 4.12, spezifiziert die Passwort Richtlinie, Retry Mechanismen usw. der Ressource FZK AD.

Abbildung 4.13 zeigt die Einstellungen für die Synchronisation. Neben dem Zeitplan für die Synchronisation werden hier auch die Log-Einstellungen angegeben.

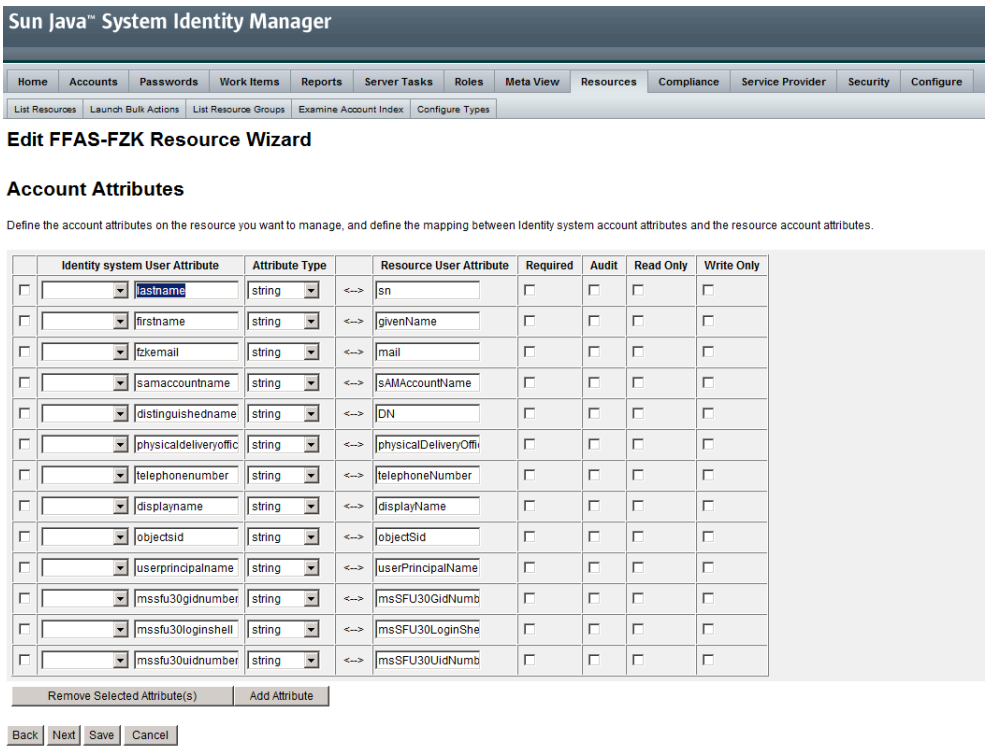


Abbildung 4.11: Account Attribute der Datenquelle FZK Active Directory

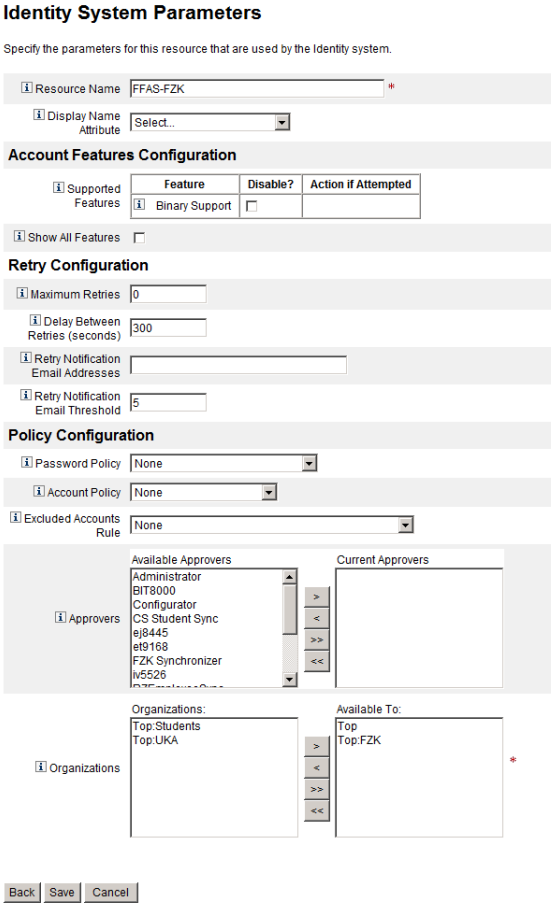


Abbildung 4.12: Identity System Parameters der Datenquelle FZK Active Directory

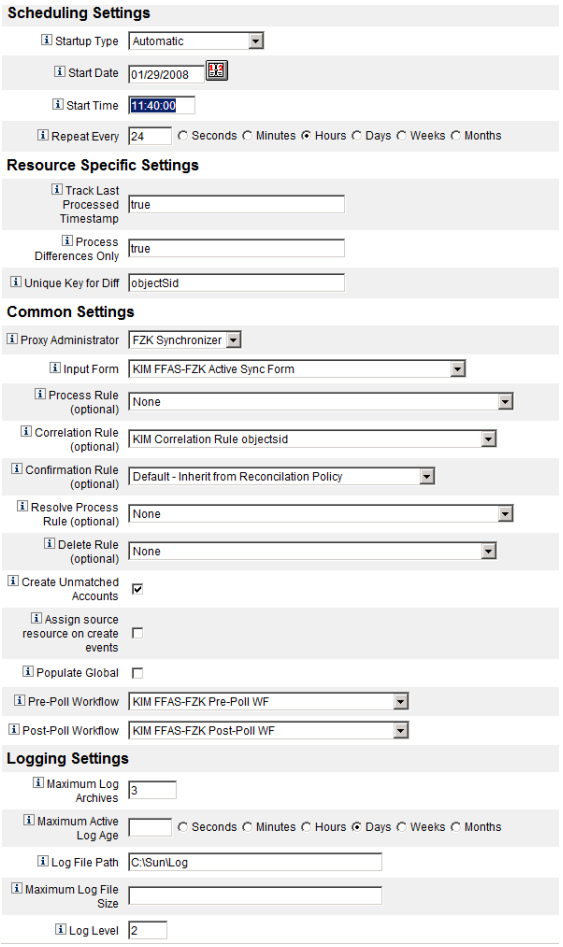


Abbildung 4.13: Synchronization Policy der Datenquelle FZK Active Directory

4.2.4 kit.edu-Active Directory

4.2.4.1 Beschreibung

Das *kit.edu*-Active Directory (*kit.edu*-AD) dient als Authentifikationsdienst am KIT, insbesondere für das Mitarbeiter- und Studierendenportal. Demnach ist das *kit.edu*-Active Directory die Quelle für das *kit.edu*-Benutzerkonto. Jeder Mitarbeiter des KIT wird sobald er entweder bei der Zentralen Universitätsverwaltung oder im Falle des FZK im FZK Active Directory als aktiver Mitarbeiter geführt wird in das *kit.edu*-AD provisioniert. Analog hierzu wird dieses Konto auch unmittelbar nach dem Ausscheiden wieder deaktiviert. Des Weiteren dient das *kit.edu*-AD dem *kit.edu*-Exchange Server als Hintergrundspeicher. Das *kit.edu*-Benutzerkonto ist somit Voraussetzung für die *kit.edu*-E-Mail-Adresse.

4.2.4.2 Attribute

Da sich die Benutzerattribute für einen Campus Süd Mitarbeiter von denen eines Campus Nord Mitarbeiter unterscheiden, werden diese hier gesondert betrachtet. Tabelle 4.5 gibt eine Übersicht, der durch das Identitätsmanagement verwalteten Benutzerattribute eines Campus Süd Mitarbeiters. Neben diesen Attributen werden für einen Campus Nord Mitarbeiter noch weitere Attribute gespeichert. Diese oder abweichende Attribute sind in Tabelle 4.6 aufgeführt.

4.2.4.3 Prozesse

Das *kit.edu*-AD ist für zahlreiche Prozesse für Mitarbeiter als auch für Studierende notwendig. Konkret sind das die Provisionierung der Universitätsmitarbeiter, FZK Mitarbeiter und Studierenden. Darüber hinaus sind folgende Prozesse des Mitarbeiterportals abhängig vom *kit.edu*-AD:

- Aktivierung eines Mitarbeiter-Benutzerkontos
- Einrichtung der E-Mail-Weiterleitung für die Campus Süd Mitarbeiter
- KIT-E-Mail-Alias-Dienst
- Passwortänderungsdienst

4.2.4.4 Konfiguration im Sun Identity Manager

Das *kit.edu*-AD dient dem Sun Identity Manager ausschließlich als Datensenke. Dies bedeutet, dass in das *kit.edu*-AD nur geschrieben wird. Es ist allerdings für den Sun Identity Manager keine autoritative Quelle für Attribute. Dies gilt wie in Abschnitt 4.2.4.3 gezeigt für andere Komponenten durchaus. Dies ist auch der Grund, warum es für diese Ressource keine Synchronization Policy gibt. Abbildung 4.15 zeigt die Resource Parameters des *kit.edu*-ADs. Unter anderem wird hier der Host, der TCP Port, der Benutzer mit welchem auf das AD zugegriffen wird, dessen Credentials und weitere Angaben zu bspw. der Verschlüsselung der zu übertragenden Daten angegeben.

Attribut	Beschreibung	Beispiel
givenName	Vorname eines Mitarbeiters	Thorsten
sn	Nachname eines Mitarbeiters	von Rechtenfeld
sAMAccountName	Durch das KIT-weite Identitätsmanagement generierter Identifier der Form `[a-zA-Z]^{2}[0-9]^{4}`. Dieser ist AD-weit eindeutig und eine zusätzliche Benutzerkennung	ab1234
userPrincipalName	Mit der *kit.edu*-E-Mail-Adresse des Benutzers identisch und dient als Kennung für das *kit.edu*-Benutzerkonto	hans.muster@kit.edu
extensionAttribute1	„Hilfsattribut“, das den SCC Benutzeraccount eines Mitarbeiters beinhaltet. Dies ist die Adresse der E-Mail-Weiterleitung.	rz02@rz.uni-karlsruhe.de
extensionAttribute4	Dient ebenfalls als Hilfsattribut	010...ab2000
mail	*kit.edu*-E-Mail-Adresse eines Mitarbeiters. Diese hat die Form [vorname.nachname]@kit.edu.	hans.muster@kit.edu
displayName	Besitzt die Form Nachname, Vorname.	Höllrigl, Thorsten
extensionAttribute10	Hilfsattribut, welches den localPart, d.h. den Teil vor dem „@“ der *kit.edu*-E-Mail-Adresse beinhaltet.	hans.muster
cn	commonName und beinhaltet bei den Campus Süd Mitarbeitern ebenfalls den localPart der *kit.edu*-E-Mail-Adresse	hans.muster

Tabelle 4.5: Attribute der Ressource *kit.edu*-Active Directory Campus Süd

Attribut	Beschreibung	Beispiel
ipPhone	In diesem Attribut wird die ObjectSID der Quelle dieses Mitarbeiters gespeichert	X'0....230'
Uid	SAMAccountName der Quelle	Mueller-T
Pager	Enthält den userPrinzipalName der Quelle	hans.meier@iwr.fzk.de
gidNumber	msSFU30GidNumber	
uidNumber	msSFU30GidNumber	7519
loginShell	msSFU30LoginShell	
physicalDeliveryOfficeName	physicalDelivery OfficeName der Quelle	SCC-DMK
telephoneNumber	Telefonnummer des Mitarbeiters	6113
cn	commonName und beinhaltet bei den Campus Nord Mitarbeitern die *kit.edu*-E-Mail-Adresse	hans.muster@kit.edu

Tabelle 4.6: Zusätzliche oder abweichende Attribute der Ressource *kit.edu*-Active Directory Campus Nord

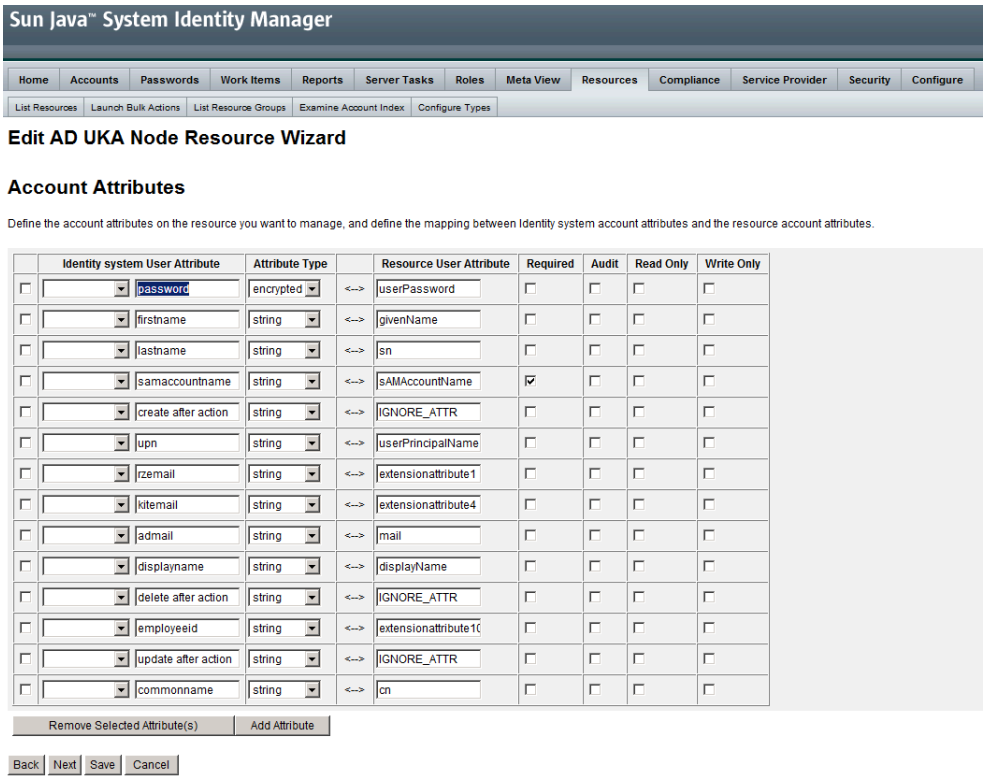


Abbildung 4.14: Account Attribute der Datenquelle *kit.edu*-AD

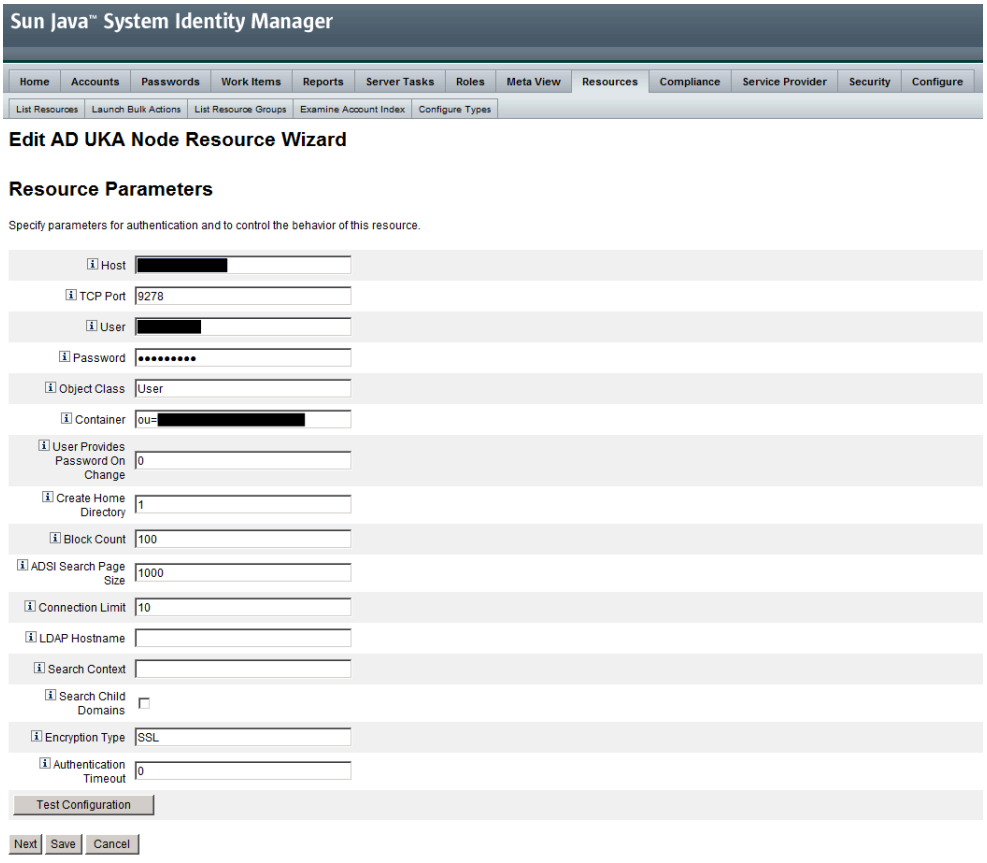


Abbildung 4.15: Resource Parameters der Datenquelle *kit.edu*-AD

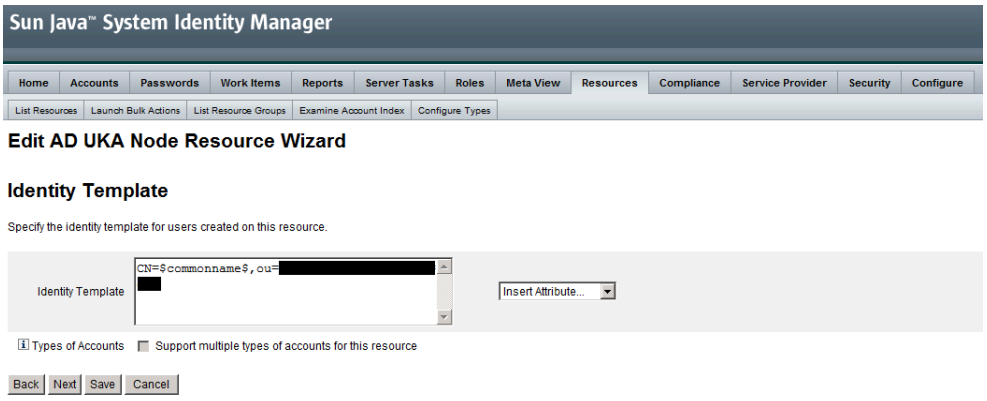


Abbildung 4.16: Identity Template der Datenquelle *kit.edu*-AD

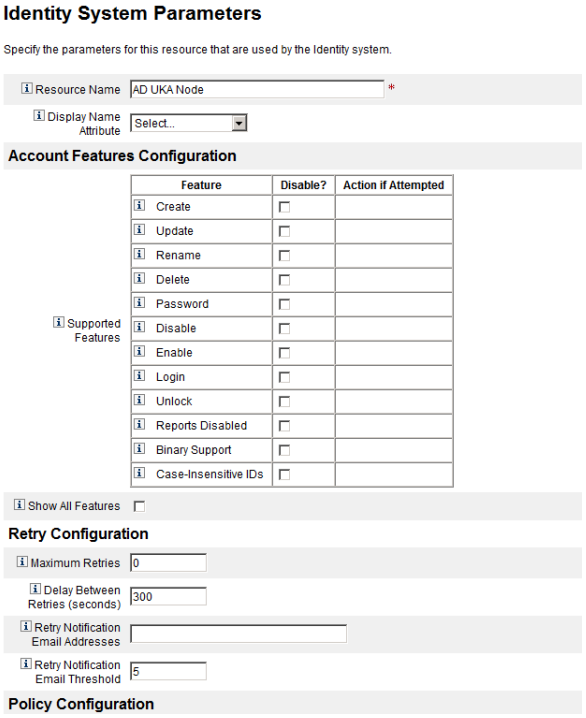


Abbildung 4.17: Identity System Parameters (1) der Datenquelle *kit.edu*-AD

Abbildung 4.14 gibt eine Übersicht der Account Attributes des *kit.edu*-AD. Auf der rechten Seite stehen die Attributbezeichner der .csv-Datei. Auf der linken Seite sind die entsprechenden Attribute innerhalb des Sun IdM aufgeführt.

Abbildung 4.16 zeigt die Einstellungen für das Identity Template. Dieses wird benötigt, falls auf einer Ressource Benutzer angelegt werden sollen. Da dies für die zuvor beschriebenen Ressourcen nicht der Fall war, war diese Einstellung dort nicht notwendig.

Die Konfiguration der Identity System Parameter, siehe Abbildung 4.17 und 4.18, spezifiziert die Passwort Richtlinie, Retry Mechanismen usw. der Ressource *kit.edu*-AD.

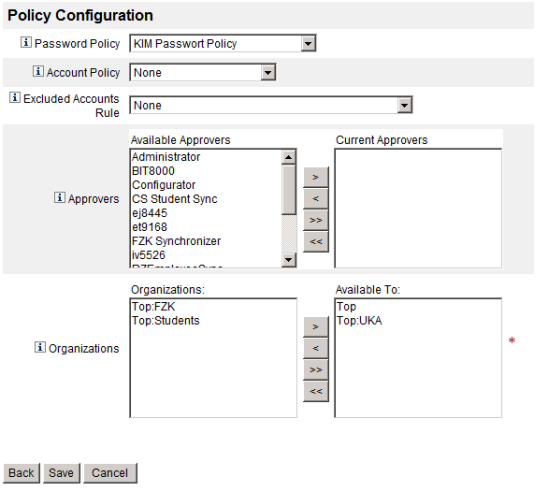


Abbildung 4.18: Identity System Parameters (2) der Datenquelle *kit.edu*-AD

4.2.5 kit.edu-Exchange Server

4.2.5.1 Beschreibung

Der *kit.edu*-Exchange Server dient den Mitarbeitern des KIT als Host für die E-Mail-Weiterleitungen. Hierbei werden sowohl für die Campus Süd als auch für die Campus Nord Mitarbeiter E-Mail-Weiterleitungskonten im *kit.edu*-Exchange Server gehalten.

4.2.5.2 Prozesse

Da der *kit.edu*-Exchange Server für die E-Mail-Konten der Mitarbeiter zuständig ist, ist er im Wesentlichen für die Einrichtung der E-Mail-Weiterleitung bei den Campus Süd Mitarbeitern (siehe Abschnitt 5.2.3) notwendig. Darüber hinaus wird für jeden Campus Nord Mitarbeiter automatisch nach der Provisionierung (siehe Abschnitt 4.3.2) ein Weiterleitungskonto eingerichtet. Demnach ist auch dieser Prozess vom *kit.edu*-Exchange Server abhängig.

4.2.5.3 Konfiguration im Sun Identity Manager

Der *kit.edu*-Exchange Server ist nur indirekt an den Sun Identity Manager angeschlossen. Dieser ruft über eine sogenannte Resource Action einen Dienst auf, welcher wiederum das E-Mail-Weiterleitungskonto im Exchange anlegt. Dies ist ein asynchroner Vorgang und liefert keine Rückmeldung über das Ergebnis des Dienstes. Der Dienst setzt hierbei ein Powershell-Skript, siehe Anhang A.4 um. Die Übergabeparameter seitens des Sun Identity Managers sind:

- *kit.edu*-E-Mail-Adresse
- Weiterleitungsadresse
  - SCC-E-Mail-Adresse im Fall der Campus Süd Mitarbeiter
  - FZK-E-Mail-Adresse im Fall der Campus Nord Mitarbeiter

4.2.6 KISS-Repository

4.2.6.1 Beschreibung

Das KISS (KIM Identity Shared Services) Repository ist eine Art Hilfsdatenbank für KIM-IDM. In dieser Datenbank werden neben unterschiedlichen Benutzerinformationen (siehe Tabelle 4.7) Informationen für das Identifier Mapping oder den Status (siehe Abschnitt 4.3.1) gespeichert. Der folgende Abschnitt gibt eine genauere Betrachtung der gespeicherten Attribute der KIT-Mitarbeiter.

4.2.6.2 Attribute

Tabelle 4.7 zeigt die in der KISS-Repository Datenbank gespeicherten Attribute gemeinsam mit deren Beschreibung und Beispielen.

4.2.6.3 Prozesse

Das KISS-Repository ist eine Hilfsdatenbank des Identitätsmanagementsystem und ist ebenfalls in vielen Prozessen notwendig. Für die Provisionierung ist das sowohl die Provisionierung der Campus Süd Mitarbeiter (siehe Abschnitt 4.3.1), als auch die Provisionierung der Campus Nord Mitarbeiter (siehe Abschnitt 4.3.2). Außerdem sind die Aktivierung des *kit.edu*-Kontos (siehe Abschnitt 5.2.2), die Einrichtung der E-Mail-Weiterleitung (siehe Abschnitt 5.2.3).

4.2.6.4 Konfiguration im Sun Identity Manager

Das KISS-Repository ist eine reine Datensenke für den Sun Identity Manager. Abbildung 4.19 zeigt die Datenbankeinstellungen. Der Sun Identity Manager greift hierbei per JDBC auf den SQL Server zu. Neben dem Host und Port, wird hier auch der Benutzer für den Datenbank-Connect angegeben.

Abbildung 4.20 stellt die zu verwaltenden Datenbankspalten dar. Abbildung 4.21 gibt eine Übersicht der Account Attributes der KISS-Repository Datenbank. Auf der rechten Seite stehen die Attributbezeichner der Datenbank. Auf der linken Seite sind die entsprechenden Attribute innerhalb des Sun IdM aufgeführt.

Abbildung 4.22 zeigt die Einstellungen für das Identity Template. Dieses wird benötigt, falls auf einer Ressource Benutzer angelegt werden sollen. In diesem Fall genügt es den Identifier anzugeben.

Die Konfiguration der Identity System Parameter, siehe Abbildung 4.23, spezifiziert die Passwort Richtlinie, Retry Mechanismen usw. der Ressource *kit.edu*-AD.

Attribut	Beschreibung	Beispiel
firstname	Vorname eines Mitarbeiters	Thorsten
lastname	Nachname eines Mitarbeiters	von Rechtenfeld
email	*kit.edu*-E-Mail-Adresse eines Mitarbeiters. Diese hat die Form [vorname.nachname]@kit.edu.	thorsten.hoellrigl@kit.edu
username	Durch das Identitätsmanagement generierter Identifier der Form `[a-zA-Z]^{2}[0-9]^{4}`. Dieser ist AD-weit eindeutig und dient dem Benutzer als zusätzliche Kennung.	ab1234
localidentifier1	Für Mitarbeiter des Campus Süd die Kennung von HIS-SVA. Für Mitarbeiter des Campus Nord ist dieses Feld leer.	123456
localidentifier2	SCC-Benutzername eines Campus Süd Mitarbeiters	rz02
localidentifier3	FZK-E-Mail-Adresse eines Campus Nord Mitarbeiters	frank.schell@extern.fzk.de
localidentifier4 - 10	Noch nicht verwendet; sind für weitere lokale Benutzerkennungen eines Mitarbeiters reserviert.	
rzemail	SCC-E-Mail-Adresse eines Campus Süd Mitarbeiters	kj72@rz.uni-karlsruhe.de
userid	Durch das Identitätsmanagement generierte GUID	0003477D-0D29-11CF-74FE-EFD6FF8ED168
status	Status eines Mitarbeiters (siehe Abschnitt 4.3.1)	activated
rzoldpassword	Initiales Passwort für einen neuen SCC-Account	
dateofcreation	Datum des initialen Anlegens	01.04.2008 14:30
dateofdeletion	Datum der Deprovisionierung	01.09.2008 14:30

Tabelle 4.7: Attribute der Ressource KISS-Repository

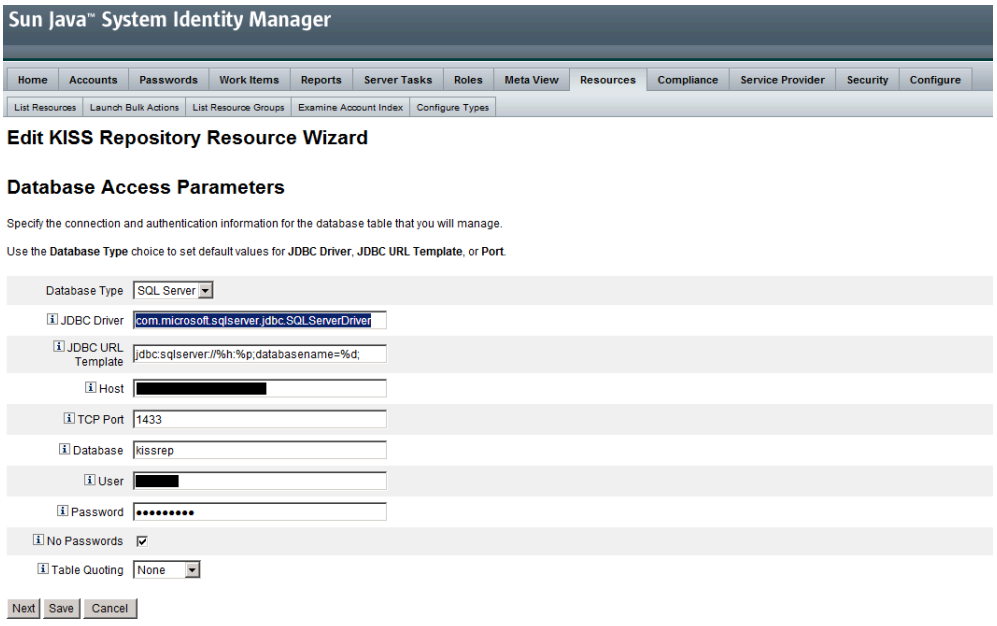


Abbildung 4.19: Database Access Parameters der Datenquelle KISS-Repository

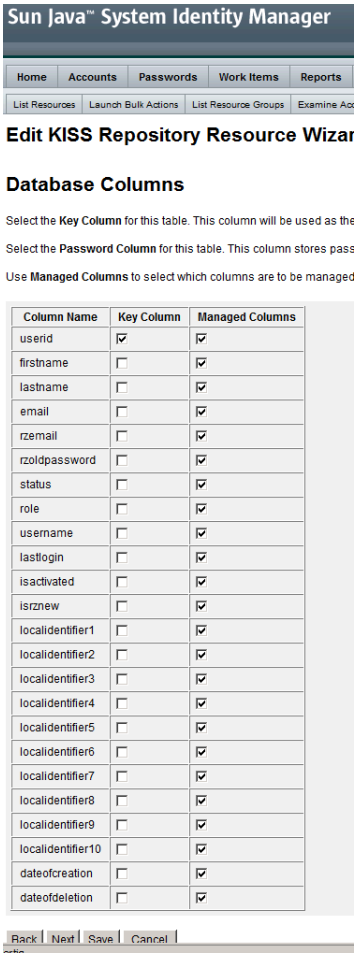


Abbildung 4.20: Datenbankspalten der Datenquelle KISS-Repository

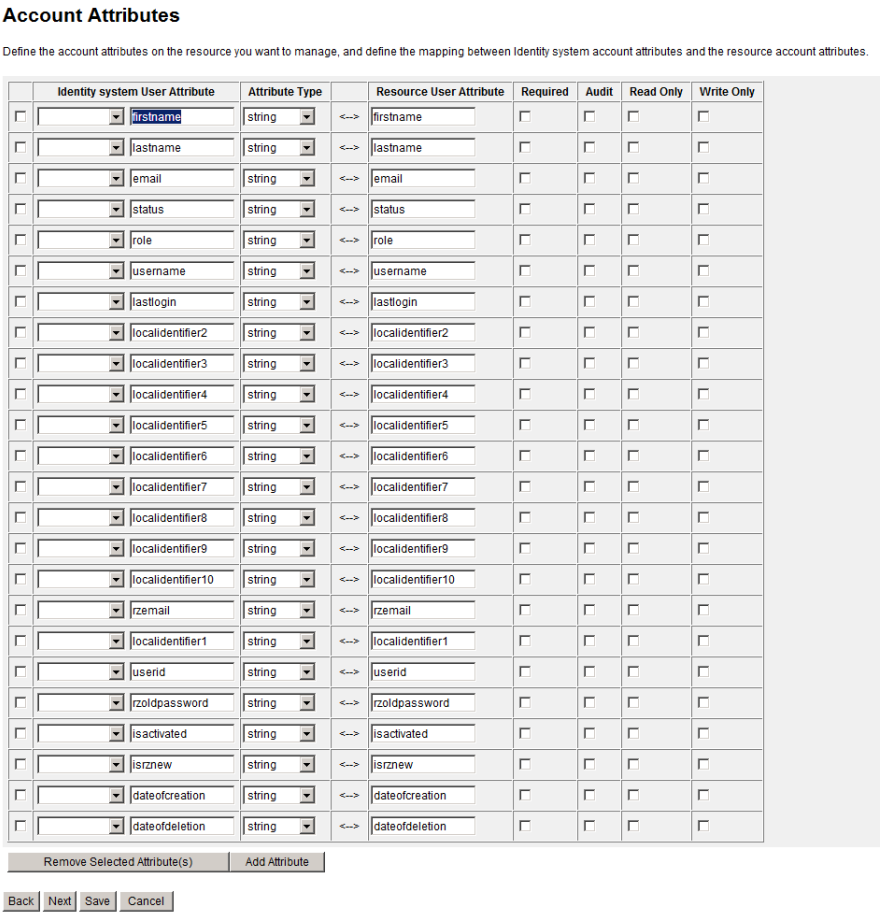


Abbildung 4.21: Account Attribute der Datenquelle KISS-Repository

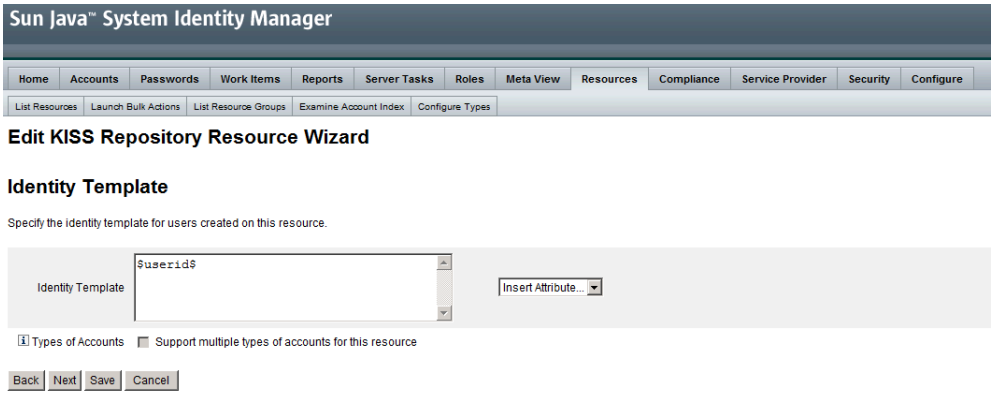


Abbildung 4.22: Identity Template der Datenquelle KISS-Repository

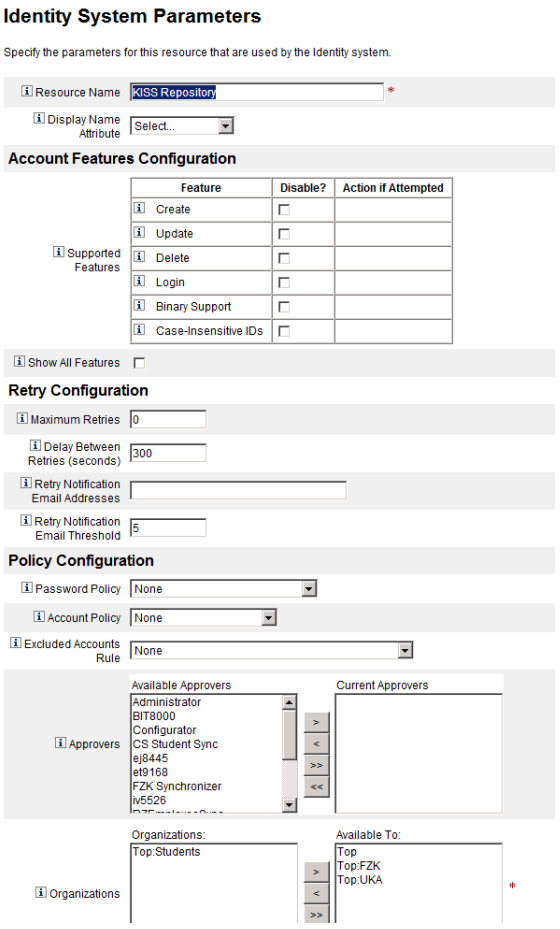


Abbildung 4.23: Identity System Parameters der Datenquelle KISS-Repository

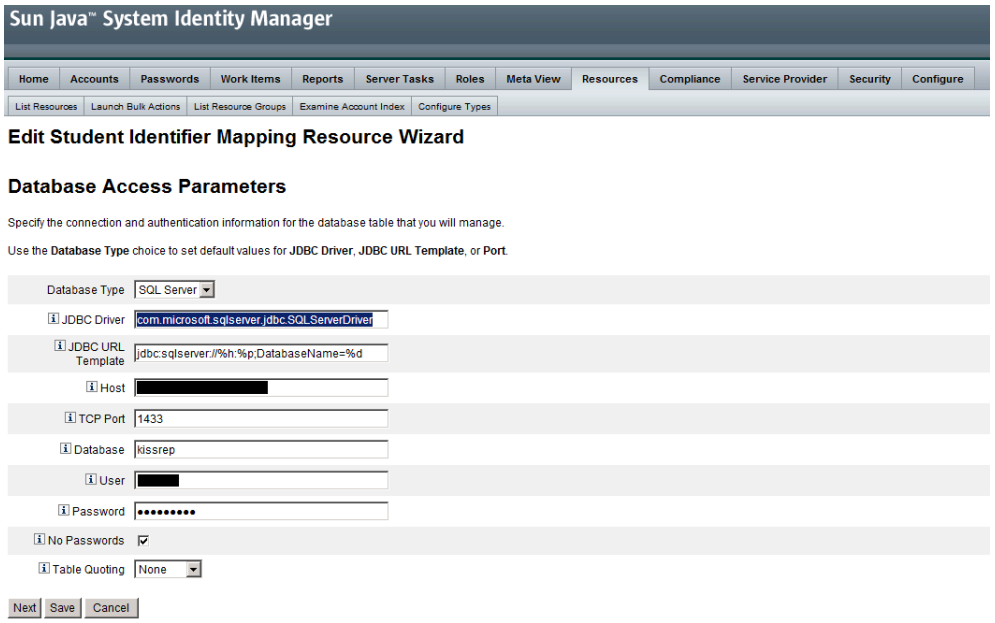


Abbildung 4.24: Database Access Parameters der Datenquelle Student Identifier Mapping

4.2.7 Student Identifier Mapping

4.2.7.1 Beschreibung

Die Datenquelle Student Identifier Mapping dient KIM-IDM Informationen über Studierende zu speichern. Z.B. werden hier auch die unterschiedlichen Identifier, die ein Studierender in den einzelnen Institutionen der Universität besitzt gespeichert.

4.2.7.2 Attribute

Die Tabelle 4.8 zeigt die Attribute der Student Identifier Mapping Datenquelle gemeinsam mit deren Beschreibung und Beispielen.

4.2.7.3 Prozesse

Die Student Identifier Mapping Datenquelle wird für die Provisionierung der Studierenden (siehe Abschnitt 4.3.3) benötigt.

4.2.7.4 Konfiguration im Sun Identity Manager

Die Student Identifier Mapping Datenbanktabelle ist eine reine Datensenke für den Sun Identity Manager. Abbildung 4.24 zeigt die Datenbankeinstellungen. Der Sun Identity Manager greift hierbei per JDBC auf den SQL Server zu. Neben dem Host und Port, wird hier auch der Benutzer für den Datenbank-Connect angegeben.

Abbildung 4.25 stellt die zu verwaltenden Datenbankspalten dar. Abbildung 4.26 gibt eine Übersicht der Account Attributes der Student Identifier Mapping Datenbanktabelle. Auf der rechten Seite stehen die Attributbezeichner der Datenbank. Auf der linken Seite sind die entsprechenden Attribute innerhalb des Sun IdM aufgeführt.

Attribut	Beschreibung	Beispiel
firstname	Vorname eines Studierenden	Thorsten
lastname	Nachname eines Studierenden	von Rechtenfeld
id_kit	*kit.edu*-E-Mail-Adresse eines Studierenden. Diese hat die Form [vorname.nachname] @student.kit.edu	hans.mai@student.kit.edu
id_zuv	Matrikelnummer eines Studierenden	1028129
id_fricard	Intern auf der FriCard gespeicherte Nummer.	3310930008
id_fricardchip	Aufgedruckte FriCard-Nummer	158001177538
id_rz	Benutzerkennung des SCC-Benutzerkontos eines Studierenden	uum0
sex	Geschlecht eines Studierenden	männlich
coursofstudiesnumber	Nummer des Studiengangs	11
status	Status eines Studierenden (siehe Abschnitt 4.3.3	Rückmeldung
street	Straße der in der ZUV geführten Adresse eines Studierenden	Daimlerstr.3
postcode	Postleitzahl	76689
city	Wohnort	Karlsruhe
addresssupplement	Zusätzliche Angabe zur Adresse	App. 13
guid	Eine durch das Identitätsmanagement generierte GUID	000079C7-681C-9B17-F5F0-36E3D68BC2D6

Tabelle 4.8: Attribute der Ressource Student Identifier Mapping

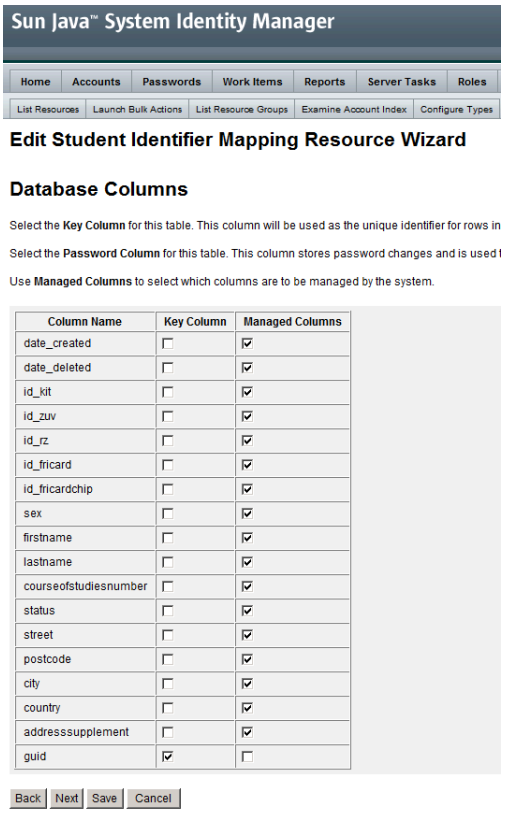


Abbildung 4.25: Datenbankspalten der Datenquelle Student Identifier Mapping

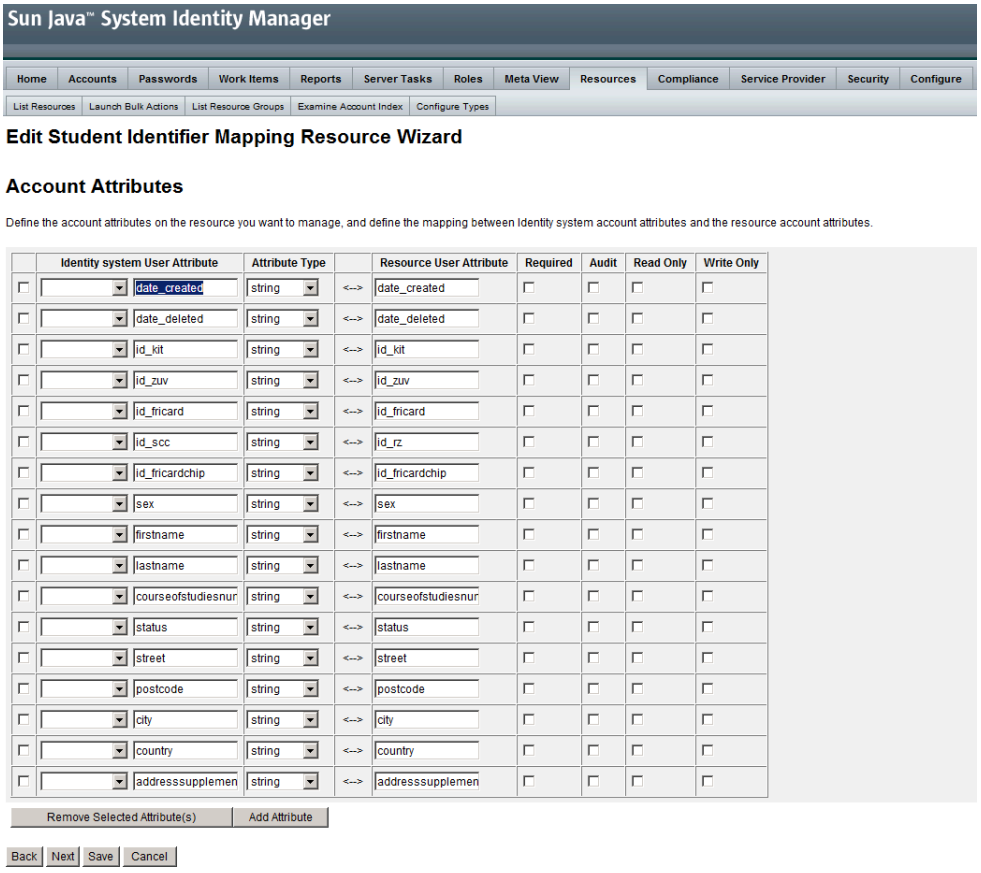


Abbildung 4.26: Account Attribute der Datenquelle Student Identifier Mapping

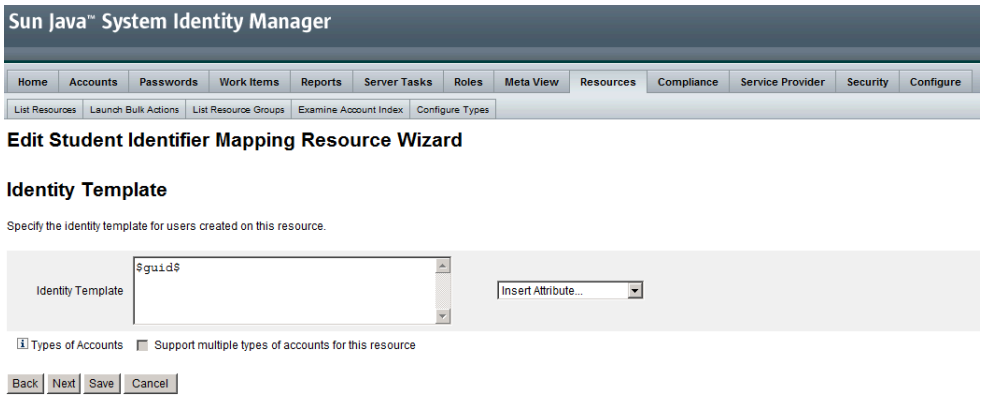


Abbildung 4.27: Identity Template der Datenquelle Student Identifier Mapping

Abbildung 4.27 zeigt die Einstellungen für das Identity Template. Dieses wird benötigt, falls auf einer Ressource Benutzer angelegt werden sollen. In diesem Fall genügt es den Identifier anzugeben.

Die Konfiguration der Identity System Parameter in Abbildung 4.28 spezifiziert die Passwort Richtlinie, Retry Mechanismen usw. der Ressource *kit.edu*-AD.

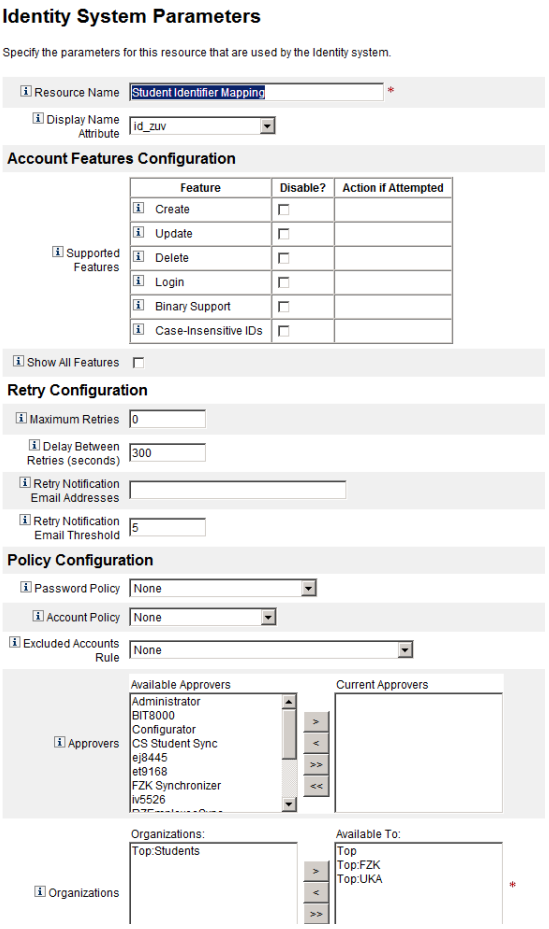


Abbildung 4.28: Identity System Parameters der Datenquelle Student Identifier Mapping

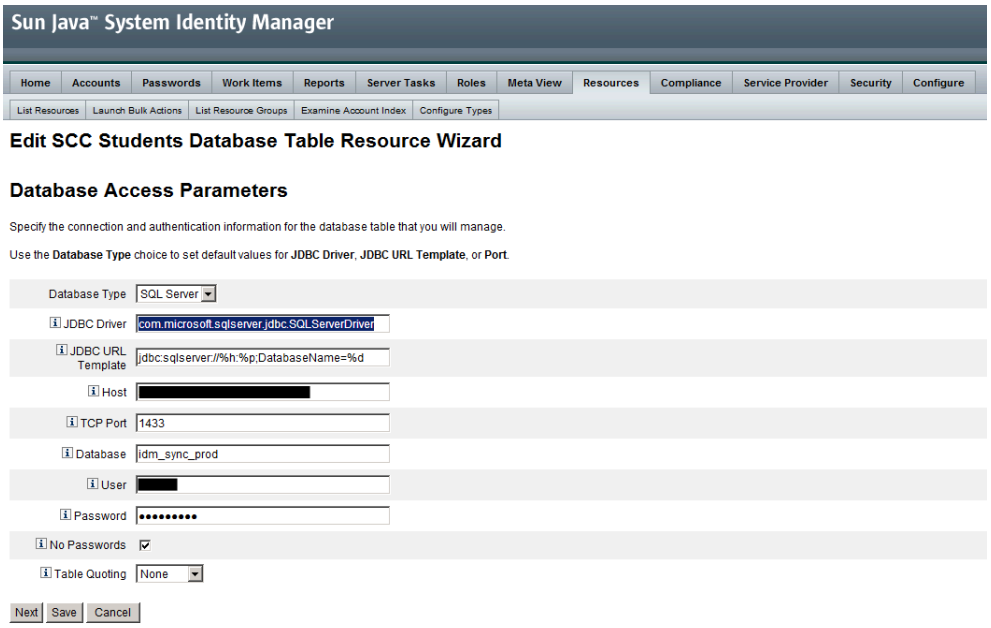


Abbildung 4.29: Database Access Parameters der Datenquelle SCC Students Database Table

4.2.8 SCC Students Database Table

4.2.8.1 Beschreibung

Die SCC Students Database Table dient als Schnittstelle zum SCC für den Austausch von Studierendeninformationen. Im Prinzip werden hier bis auf die lokalen Identifikatoren der Studierenden, dieselben Attribute gespeichert wie im Student Identifier Mapping (siehe Abschnitt 4.2.7).

4.2.8.2 Attribute

Die Tabelle 4.9 zeigt die Attribute der Student Identifier Mapping Datenquelle gemeinsam mit deren Beschreibung und Beispielen.

4.2.8.3 Prozesse

Die SCC Students Database Table ist für die Provisionierung der Studierenden (siehe Abschnitt 4.3.3) notwendig.

4.2.8.4 Konfiguration im Sun Identity Manager

Die SCC Students Database Table Datenbanktabelle ist für den Sun Identity Manager sowohl eine Datenquelle als auch eine Datensenke. Abbildung 4.29 zeigt die Datenbankeinstellungen. Der Sun Identity Manager greift hierbei per JDBC auf den SQL Server zu. Neben dem Host und Port, wird hier auch der Benutzer für den Datenbank-Connect angegeben.

Abbildung 4.30 stellt die zu verwaltenden Datenbankspalten dar. Abbildung 4.31 gibt eine Übersicht der Account Attributes der SCC Students Database Table Datenbanktabelle. Auf der rechten Seite stehen die Attributbezeichner der Datenbank.

Attribut	Beschreibung	Beispiel
firstname	Vorname eines Studierenden	Thorsten
lastname	Nachname eines Studierenden	von Rechtenfeld
id_kit	*kit.edu*-E-Mail-Adresse eines Studierenden. Diese hat die Form [vorname.nachname] @student.kit.edu.	hans.mai@student.kit.edu
id_zuv	Matrikelnummer eines Studierenden	1028129
id_fricard	Intern auf der FriCard gespeicherte Nummer.	3310930008
id_fricardchip	Aufgedruckte FriCard-Nummer	158001177538
id_rz	Benutzerkennung des SCC-Benutzerkontos eines Studierenden	uum0
sex	Geschlecht eines Studierenden	männlich
coursofstudiesnumber	Nummer des Studiengangs	11
status	Status eines Studierenden (siehe Abschnitt 4.3.3	Rückmeldung
street	Straße der in der ZUV geführten Adresse eines Studierenden	Daimlerstr.3
postcode	Postleitzahl	76689
city	Wohnort	Karlsruhe
addresssupplement	Zusätzliche Angabe zur Adresse	App. 13
guid	Eine durch das Identitätsmanagement generierte GUID	000079C7-681C-9B17-F5F0-36E3D68BC2D6

Tabelle 4.9: Attribute der Ressource SCC Students Database Table

Database Columns

Select the **Key Column** for this table. This column will be used as the un

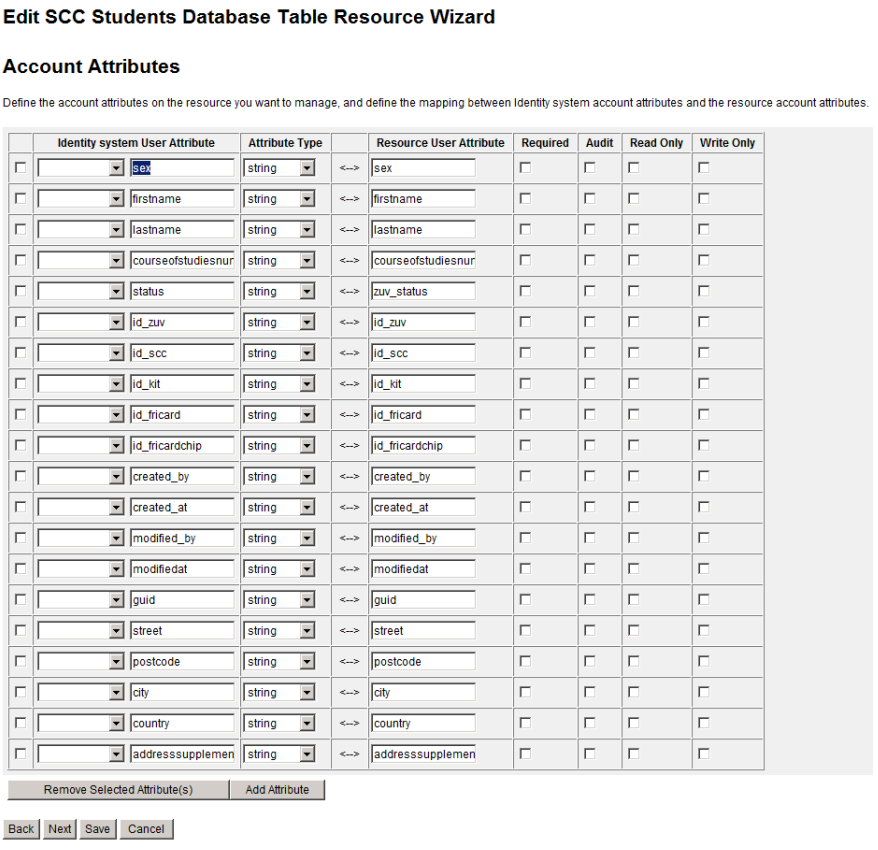
Select the **Password Column** for this table. This column stores passwo

Use **Managed Columns** to select which columns are to be managed by

Column Name	Key Column	Managed Columns
guid	☑	☑
sex	☐	☑
firstname	☐	☑
lastname	☐	☑
courseofstudiesnumber	☐	☑
zuv_status	☐	☑
id_zuv	☐	☑
id_scc	☐	☑
id_kit	☐	☑
id_fricard	☐	☑
id_fricardchip	☐	☑
created_by	☐	☑
created_at	☐	☑
modified_by	☐	☑
modifiedat	☐	☑
street	☐	☑
postcode	☐	☑
city	☐	☑
country	☐	☑
addresssupplement	☐	☑
acceptedtermsofuse	☐	☐
briefgesendet	☐	☐

Back Next Save Cancel

Abbildung 4.30: Datenbankspalten der Datenquelle SCC Students Database Table



Edit SCC Students Database Table Resource Wizard

Account Attributes

Define the account attributes on the resource you want to manage, and define the mapping between Identity system account attributes and the resource account attributes.

	Identity system User Attribute	Attribute Type		Resource User Attribute	Required	Audit	Read Only	Write Only
☐	sex	string	<-->	sex	☐	☐	☐	☐
☐	firstname	string	<-->	firstname	☐	☐	☐	☐
☐	lastname	string	<-->	lastname	☐	☐	☐	☐
☐	courseofstudiesnur	string	<-->	courseofstudiesnur	☐	☐	☐	☐
☐	status	string	<-->	zuv_status	☐	☐	☐	☐
☐	id_zuv	string	<-->	id_zuv	☐	☐	☐	☐
☐	id_scc	string	<-->	id_scc	☐	☐	☐	☐
☐	id_kit	string	<-->	id_kit	☐	☐	☐	☐
☐	id_fricard	string	<-->	id_fricard	☐	☐	☐	☐
☐	id_fricardchip	string	<-->	id_fricardchip	☐	☐	☐	☐
☐	created_by	string	<-->	created_by	☐	☐	☐	☐
☐	created_at	string	<-->	created_at	☐	☐	☐	☐
☐	modified_by	string	<-->	modified_by	☐	☐	☐	☐
☐	modifiedat	string	<-->	modifiedat	☐	☐	☐	☐
☐	guid	string	<-->	guid	☐	☐	☐	☐
☐	street	string	<-->	street	☐	☐	☐	☐
☐	postcode	string	<-->	postcode	☐	☐	☐	☐
☐	city	string	<-->	city	☐	☐	☐	☐
☐	country	string	<-->	country	☐	☐	☐	☐
☐	addresssupplemen	string	<-->	addresssupplemen	☐	☐	☐	☐

Remove Selected Attribute(s) Add Attribute

Back Next Save Cancel

Abbildung 4.31: Account Attribute der Datenquelle SCC Students Database Table

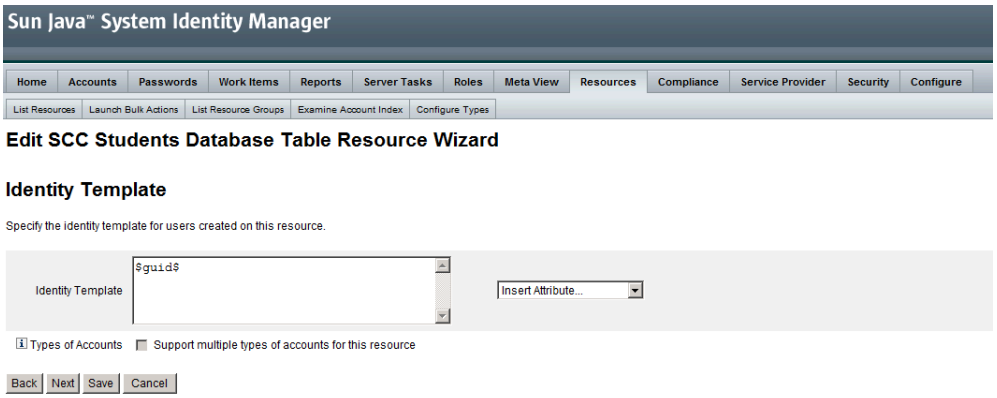


Abbildung 4.32: Identity Template der Datenquelle SCC Students Database Table

Auf der linken Seite sind die entsprechenden Attribute innerhalb des Sun IdM aufgeführt.

Abbildung 4.32 zeigt die Einstellungen für das Identity Template. Dieses wird benötigt, falls auf einer Ressource Benutzer angelegt werden sollen. In diesem Fall genügt es den Identifier anzugeben.

Die Konfiguration der Identity System Parameter (siehe Abbildung 4.33) spezifiziert die Passwort Richtlinie, Retry Mechanismen usw. der Ressource SCC Student Database Table.

Die Abbildung 4.34 zeigt die Synchronisationseinstellungen für die SCC Students Database Table. Neben dem Zeitplan für die Synchronisation werden hier auch bspw. die Log-Einstellungen angegeben.

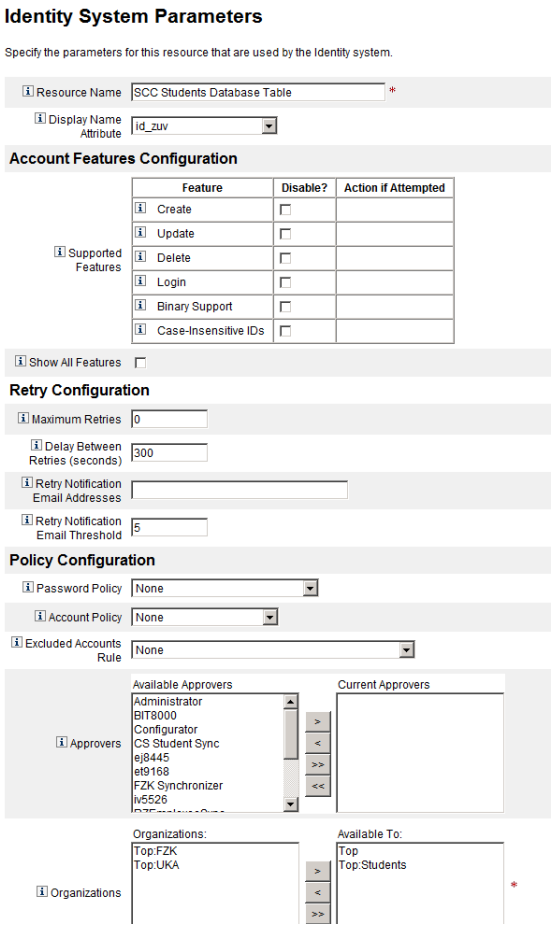


Abbildung 4.33: Identity System Parameters der Datenquelle SCC Students Database Table

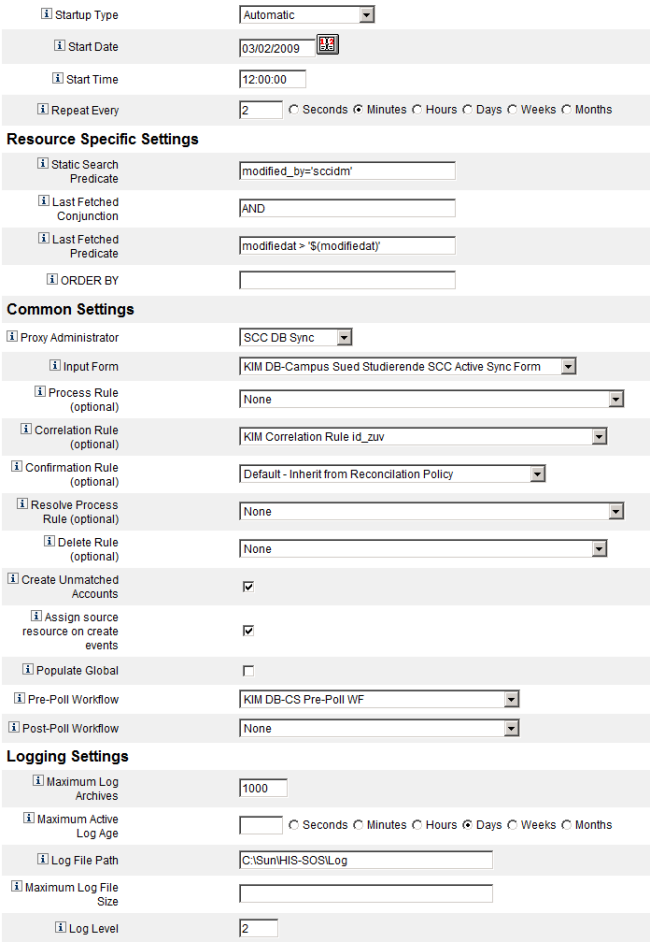


Abbildung 4.34: Synchronization Policy der Datenquelle SCC Students Database Table

4.2.9 RZ LDAP Employee

4.2.9.1 Beschreibung

Die Datenquelle RZ LDAP Employee ist für die Campus Süd Mitarbeiter die Schnittstelle zwischen dem SCC und dem KIT-weiten Identitätsmanagement. Diese spielt für die Mitarbeiterprovisionierung eine wichtige Rolle, da über diesen Verzeichnisdienst ein bidirektionaler Datenaustausch zwischen dem SCC und dem KIT-weiten Identitätsmanagement stattfindet. So wird bspw. durch das SCC die SCC-Benutzerkennung an das KIT-weite Identitätsmanagement geliefert. Diese wird bspw. für das Einrichten der E-Mail-Adress-Weiterleitung im Mitarbeiterportal benötigt.

4.2.9.2 Attribute

Tabelle 4.10 zeigt die relevanten Attribute eines Mitarbeiters im RZ-LDAP.

4.2.9.3 Prozesse

Der RZ-LDAP ist für die Provisionierung von Mitarbeitern des Campus Süd (vgl. Abschnitt 4.3.1) von Bedeutung. Außerdem bei der Aktivierung eines Mitarbeiter-Benutzerkontos (vgl. Abschnitt 5.2.2) und der Einrichtung der E-Mail-Weiterleitung (vgl. Abschnitt 5.2.3).

4.2.9.4 Konfiguration im Sun Identity Manager

Das RZ-LDAP ist für den Sun Identity Manager sowohl eine Datenquelle als auch eine Datensenke. Abbildung 4.35 zeigt die Resource Parameters des RZ-LDAP. Unter anderem wird hier der Host, der TCP Port, der Benutzer mit welchem auf das AD zugegriffen wird, dessen Credentials und weitere Angaben zu bspw. der Verschlüsselung der zu übertragenden Daten angegeben.

Abbildung 4.36 gibt eine Übersicht der Account Attributes des RZ-LDAP. Auf der rechten Seite stehen die Attributbezeichner des RZ-LDAP. Auf der linken Seite sind die entsprechenden Attribute innerhalb des Sun IdM aufgeführt.

Abbildung 4.37 zeigt die Einstellungen für das Identity Template. Dieses wird benötigt, falls auf einer Ressource Benutzer angelegt werden sollen. Da dies für die zuvor beschriebenen Ressourcen nicht der Fall war, war diese Einstellung dort nicht notwendig.

Die Konfiguration der Identity System Parameter in Abbildung 4.38 und 4.17 spezifiziert die Passwort Richtlinie, Retry Mechanismen usw. der Ressource *kit.edu*-AD.

Die Abbildungen 4.39 und 4.40 zeigen die Synchronisationseinstellungen für die SCC Students Database Table. Neben dem Zeitplan für die Synchronisation werden hier auch bspw. die Log-Einstellungen angegeben.

Attribut	Beschreibung	Beispiel
givenName	Vorname eines Mitarbeiters	Thorsten
sn	Nachname eines Mitarbeiters	von Rechtenfeld
kitEmailAlias	*kit.edu*-E-Mail-Adresse eines Mitarbeiters. Diese hat die Form [vorname.nachname]@.kit.edu	hans.mai@kit.edu
kimKostenstelle	Kostenstelle eines Mitarbeiters.	50102011
kimOrgEinheit	Organisationseinheit eines Mitarbeiters	50100000
rzbvInitialPwd	Falls ein Mitarbeiter bei einer Neueinstellung ein neues SCC-Benutzerkonto benötigt, wird hier das Initialpasswort gespeichert. Dieses wird im Hintergrund bei der Aktivierung des Benutzerkontos mitgeschickt.	
rzbvuid	Benutzerkennung des SCC-Benutzerkontos eines Mitarbeiters	rz01
rzbvolduid	Benutzerkennung des SCC-Benutzerkontos eines Mitarbeiters, welches im initial zugeordnet wurde. Da es sein kann, dass ein Mitarbeiter mehrere RZ-Konten hat, können das rzbvuid und rzbvolduid unterschiedlich sein.	rz02
kimGeschlecht	Geschlecht eines Mitarbeiters	männlich
syncStatus	Status eines Mitarbeiters	KIM-IDM:neu

Tabelle 4.10: Attribute der Ressource RZ LDAP

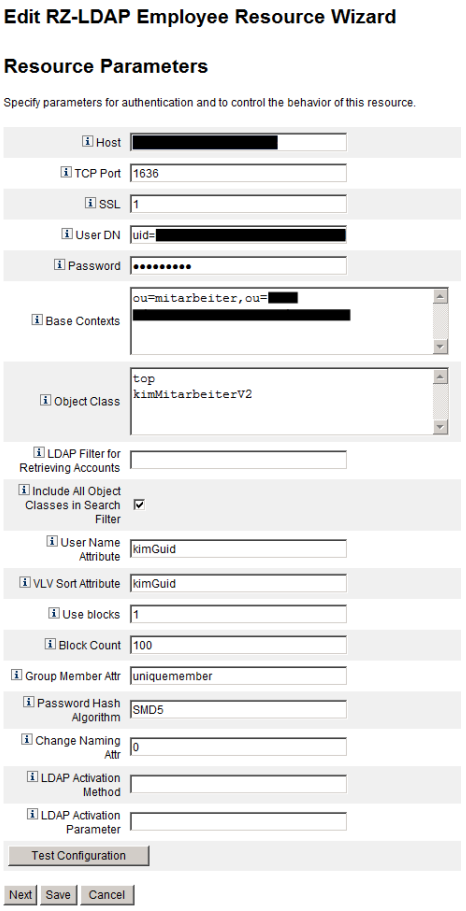


Abbildung 4.35: Resource Parameters der Datenquelle RZ-LDAP

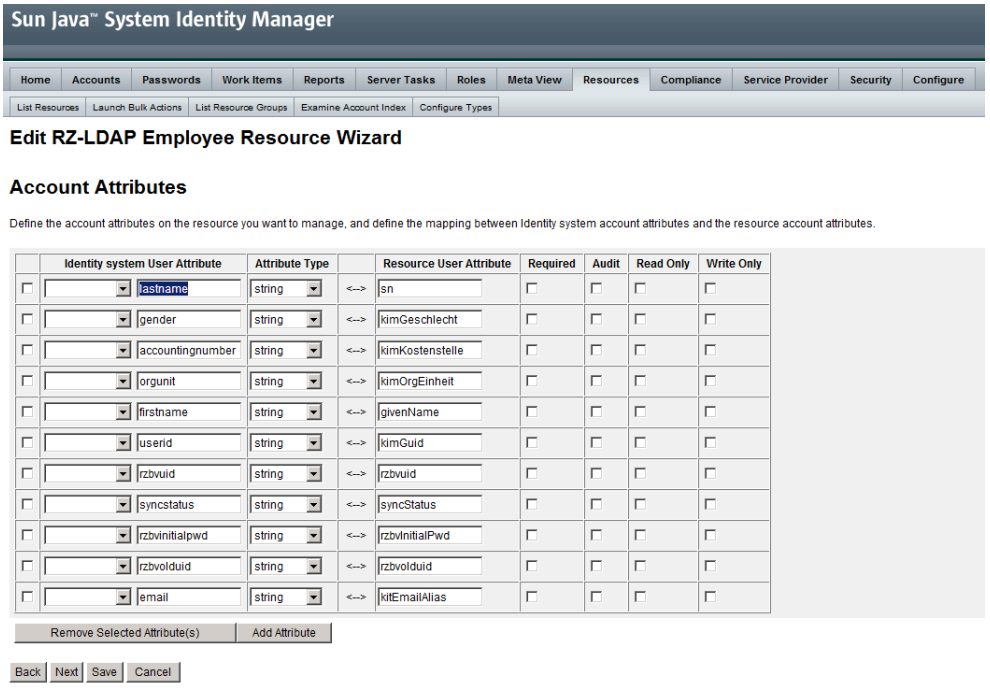


Abbildung 4.36: Account Attribute der Datenquelle RZ-LDAP

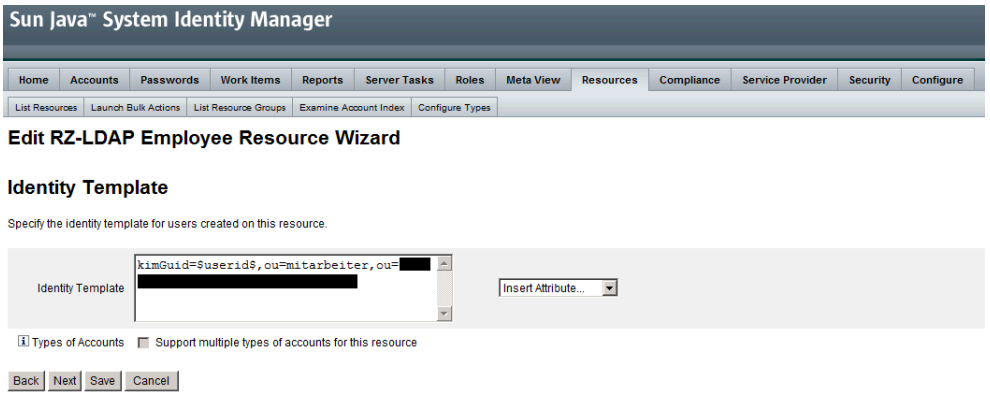


Abbildung 4.37: Identity Template der Datenquelle RZ-LDAP

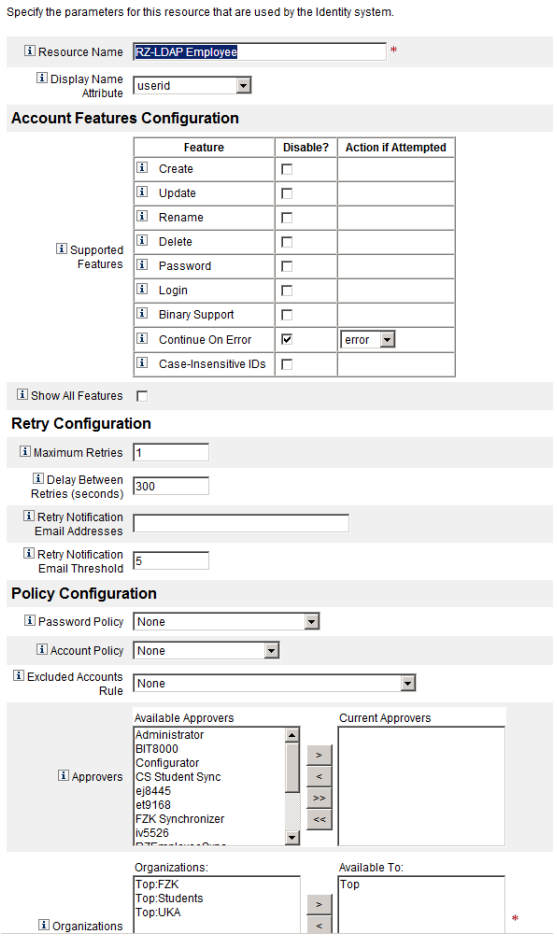


Abbildung 4.38: Identity System Parameters der Datenquelle RZ-LDAP

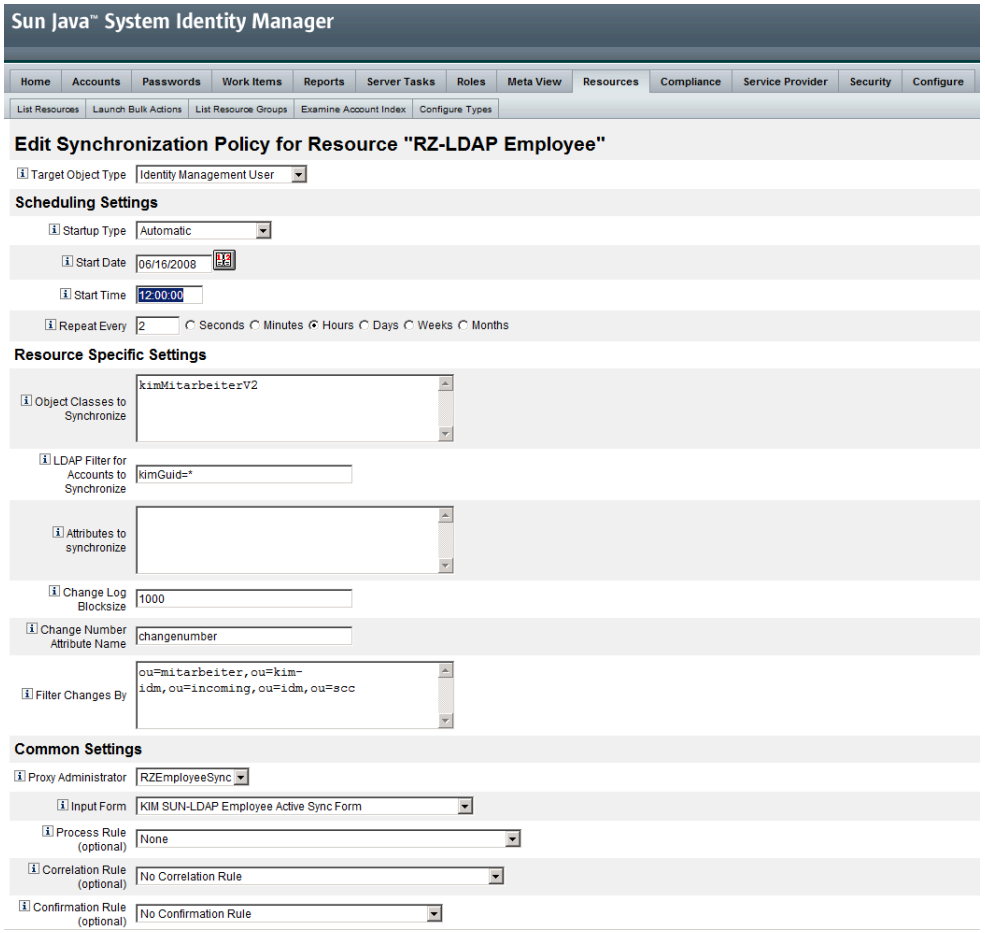


Abbildung 4.39: Synchronization Policy (1) der Datenquelle RZ-LDAP

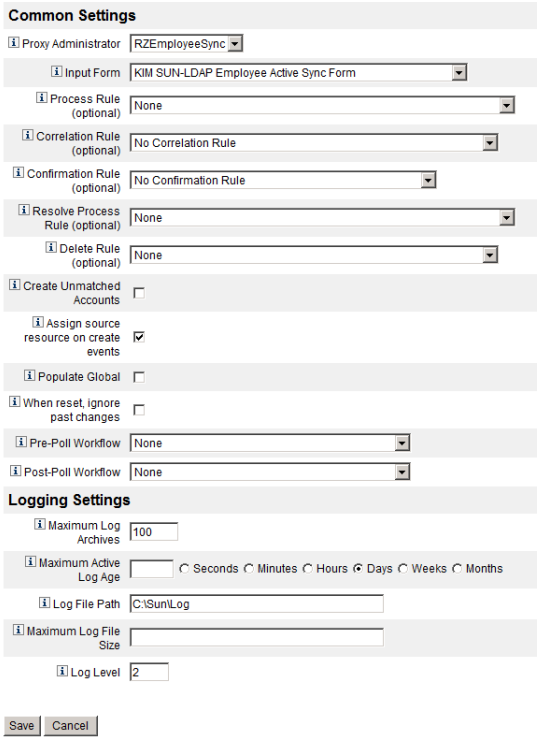


Abbildung 4.40: Synchronization Policy (2) der Datenquelle RZ-LDAP

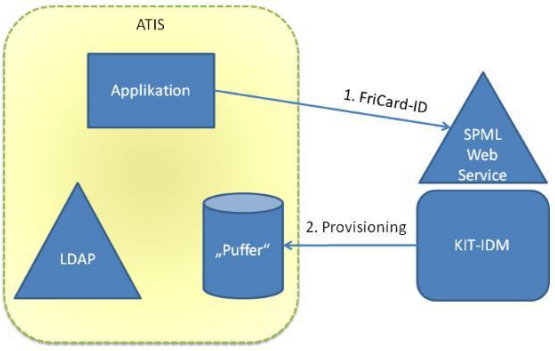


Abbildung 4.41: Anbindung der ATIS

4.2.10 ATIS

4.2.10.1 Beschreibung

Die Provisionierung der Abteilung Technische Infrastruktur (ATIS) durch das KIT-weite Identitätsmanagement wird aktuell nur testweise durchgeführt. Das Szenario umfasst das proaktive Anstoßen der Provisionierung von Studierendendaten in eine definierte Schnittstelle, wie in Abbildung 4.41 dargestellt. Die Studierendendaten sollen nach der initialen Provisionierung durch das KIT-IDM aktuell gehalten werden und die Exmatrikulation eines Studierenden mitgeteilt werden.

4.2.10.2 Attribute

Tabelle 4.11 beschreibt die entsprechenden Attribute.

4.2.10.3 Prozesse

Da die ATIS bislang nur auf einem Testsystem angebunden ist, sind von dieser auch keine Prozesse abhängig.

4.2.10.4 Konfiguration im Sun Identity Manager

Die ATIS ist für den Sun Identity Manager eine reine Datensenke. Abbildung 4.42 zeigt die Resource Parameters. Unter anderem wird hier der Host, der TCP Port, der Benutzer mit welchem auf das AD zugegriffen wird, dessen Credentials und weitere Angaben zu bspw. der Verschlüsselung der zu übertragenden Daten angegeben.

Abbildung 4.43 gibt eine Übersicht der Account Attributes des ATIS. Auf der rechten Seite stehen die Attributbezeichner des ATIS-LDAP. Auf der linken Seite sind die entsprechenden Attribute innerhalb des Sun IdM aufgeführt.

Abbildung 4.44 zeigt die Einstellungen für das Identity Template. Dieses wird benötigt, falls auf einer Ressource Benutzer angelegt werden sollen. Da dies für die

Attribut	Beschreibung	Beispiel
givenname	Vorname eines Studierenden	Thorsten
sn	Nachname eines Studierenden	von Rechtenfeld
atisMatriculationNumber	Matrikelnummer eines Studierenden	1023434
atisFricardNumber	Fricardnummer eines Studierenden	1500034343
cn	Muss noch spezifiziert werden.	
atisId	Muss noch spezifiziert werden.	
kimGuid	Eine durch das Identitätsmanagement generierte GUID	0003477D-0D29-11CF-74FE-EFD6FF8ED168
atisItStudent	Muss noch spezifiziert werden.	
atisEnrolledStudent	Muss noch spezifiziert werden.	

Tabelle 4.11: Attribute der Ressource ATIS

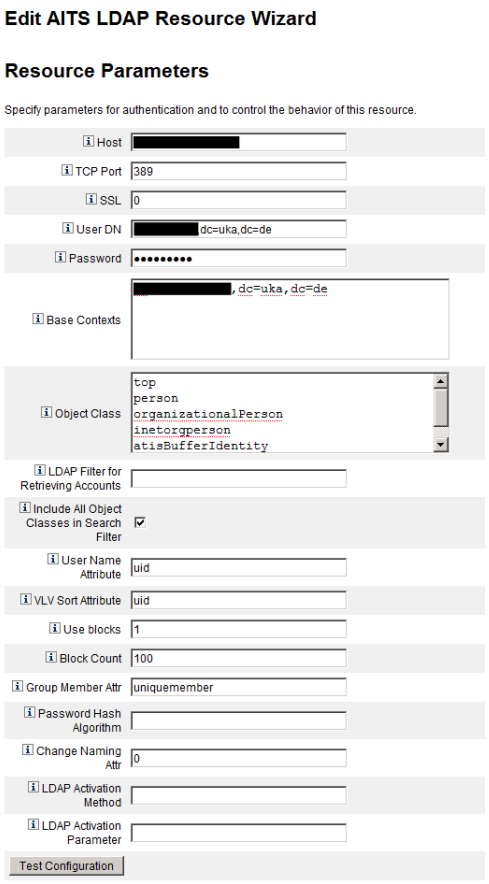


Abbildung 4.42: Resource Parameters der Datenquelle ATIS

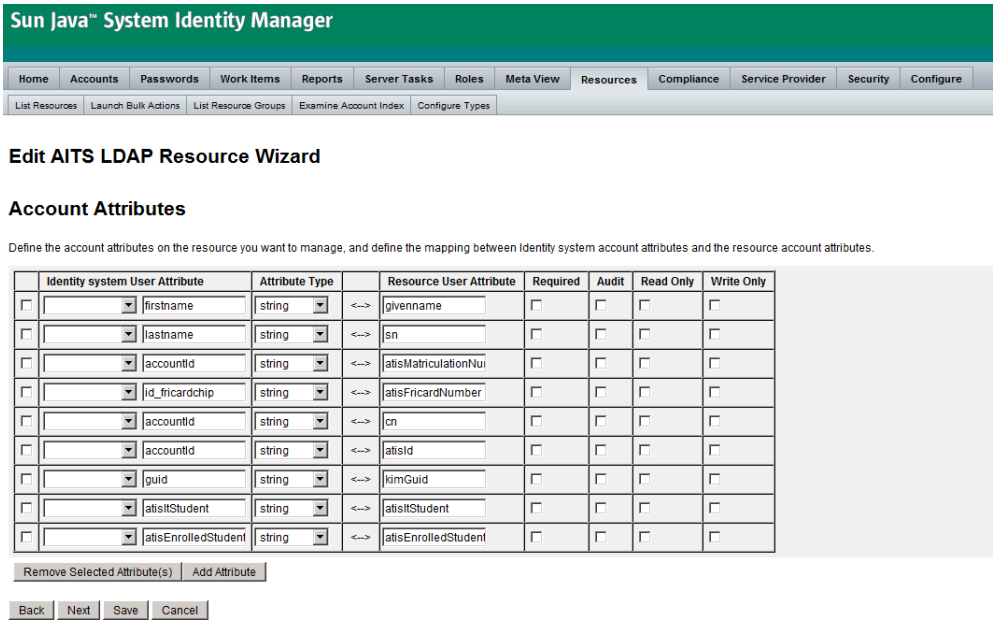


Abbildung 4.43: Account Attribute der Datenquelle ATIS

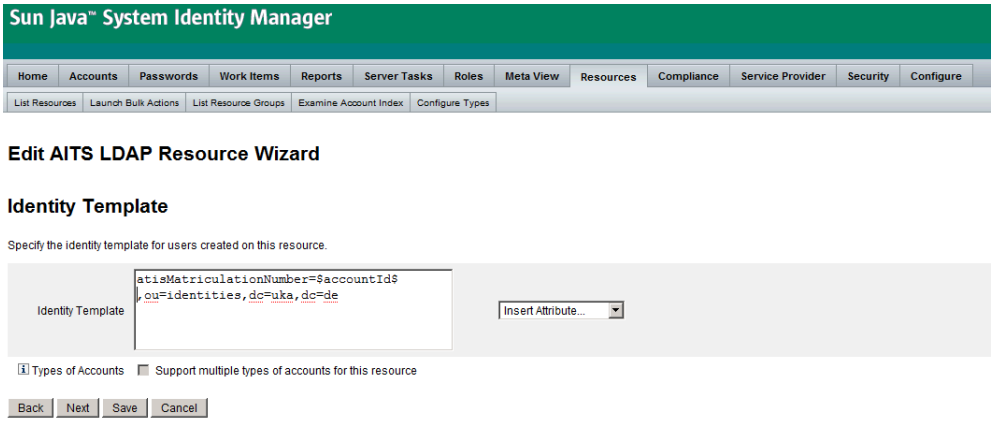


Abbildung 4.44: Identity Template der Datenquelle ATIS

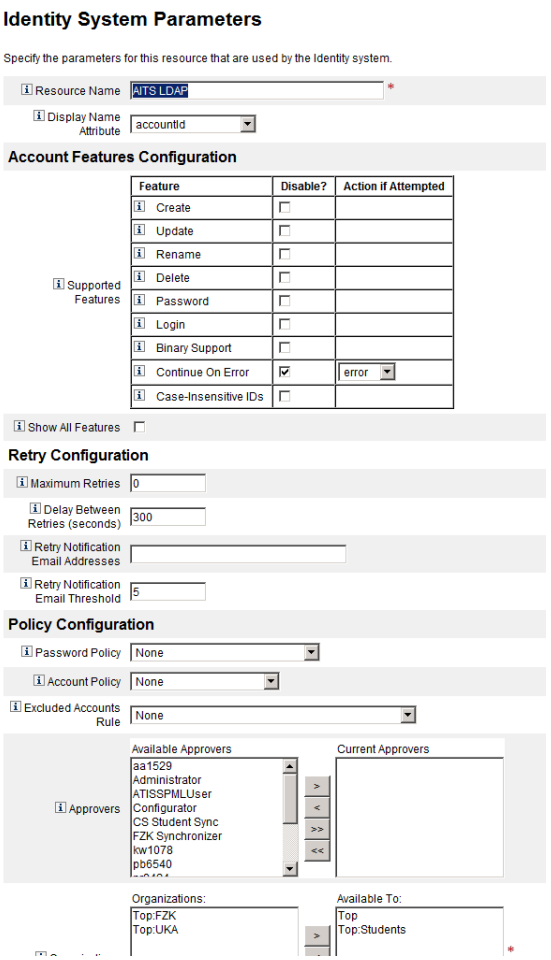


Abbildung 4.45: Identity System Parameters der Datenquelle ATIS

zuvor beschriebenen Ressourcen nicht der Fall war, war diese Einstellung dort nicht notwendig.

Die Konfiguration der Identity System Parameter, siehe Abbildung 4.45 und 4.17, spezifiziert die Passwort Richtlinie, Retry Mechanismen usw. der Ressource ATIS.

4.3 Provisionierungs-Prozesse

Die Provisionierung der Mitarbeiter des KIT ist eines der Hauptziele des Identitätsmanagements. Hierbei geht es darum folgende Ziele zu erreichen:

- Zeitnahe Zurverfügungstellung einer eindeutigen KIT-weiten Identität und ein hiermit verbundenes Benutzerkonto für jeden Mitarbeiter
- Automatisierung der Provisionierungs-Prozesse
- Automatischer Synchronisation bei aufkommenden Änderungen
- Verbesserung der Sicherheit und des Datenschutzes
- Zeitnaher Entzug aller Berechtigungen bei Ausscheiden von Mitarbeitern
- Verbesserung der Erreichbarkeit von Mitarbeitern
- Verbesserung der Benutzerfreundlichkeit

4.3.1 Mitarbeiter am Campus Süd

Abbildung 4.46 gibt eine Übersicht des Provisionierungsprozesses für Mitarbeiter am Campus Süd. Hierbei werden regelmäßig (einmal am Tag) aktuelle Daten aus der Zentralen Universitätsverwaltung verschlüsselt per ssh an das Identitätsmanagement übertragen. Dieser Vorgang ist automatisiert und muss nur initial eingerichtet werden. Daraufhin werden durch das Identitätsmanagement eine Vielzahl von Konsistenzchecks durchgeführt. Hierbei wird unter anderem die Anzahl der Mitarbeiterdatensätze, die korrekte Anzahl der Attribute und das Vorhandensein verpflichtender Attributwerte geprüft. Falls diese Überprüfung erfolgreich abläuft, werden durch das Identitätsmanagementsystem Änderungen an den Mitarbeiterdaten ermittelt. Hierbei wird typischerweise zwischen den Events *Create*, *Update*, *Disable*, *Enable*, *Delete* unterschieden. Je nach Änderung wird ein entsprechender Prozess des Identitätsmanagementsystems angestoßen. Folgende Unterprozesse werden unterschieden:

- *Create* – Der initiale Prozess legt ein neues Benutzerkonto für einen Mitarbeiter an. Die zusätzlich generierten Attribute werden gemeinsam mit den Attributen der Zentralen Universitätsverwaltung in die angeschlossenen System provisioniert. Die angeschlossenen Systeme und die dazugehörigen Attribute sind in Abschnitt 4.2 zu finden. Nach dem erfolgreichen Anlegen der lokalen Benutzerkonten werden dem Benutzer in Form eines Briefes die Benutzerkennung und das dazugehörige Passwort übermittelt.
- *Update* – Anpassungen bei Änderungen sind bspw. bei Heirat oder Umzug notwendig. Hierbei werden die Änderungen an die entsprechenden Ressourcen weitergeleitet.

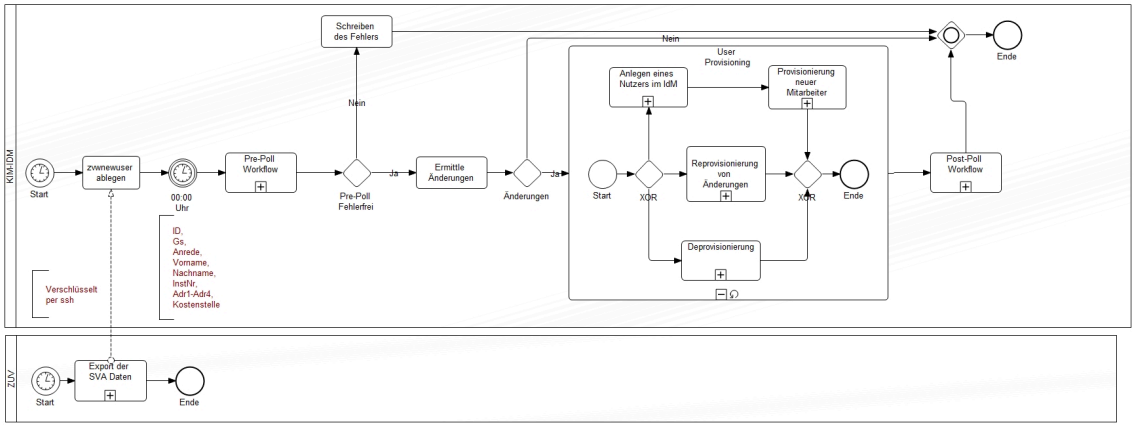


Abbildung 4.46: Hauptprozess Provisionierung Mitarbeiter Campus Süd

- *Disable* – Eine der wichtigsten Schritte ist die Deprovisionierung. Nachdem ein Benutzer das KIT verlässt, werden unmittelbar alle an das KIT-weite Identitätsmanagement angeschlossenen und für diesen Mitarbeiter relevante Systeme hierüber informiert. Dies ermöglicht ein schnelles Entziehen aller Berechtigungen eines Benutzers und verhindert darüber hinaus „Account-Leichen“. Aktuell werden Mitarbeiter, die das KIT verlassen haben, unmittelbar im KIT AD (siehe Abschnitt 4.2.4) deaktiviert. Diese Deaktivierung bewirkt, dass sich der Mitarbeiter nicht mehr gegen das AD authentifizieren kann, gleichzeitig jedoch noch seine E-Mail-Weiterleitung nutzen kann.

- *Enable* – Da die Möglichkeit besteht, dass Mitarbeiter bspw. urlaubsbedingt kurzzeitig nicht als Mitarbeiter geführt werden, ist es notwendig, diese nach der Wiedereinstellung zu „reaktivieren“. Hierbei wird für einen Mitarbeiter kein neuer Account angelegt, sondern es wird der bereits bestehende Account, welcher sich aktuell im Status „deactivated“ befindet wieder aktiviert.

- *Delete* – Die Frage wann ein Benutzer und das dazugehörige Benutzerkonto vollständig gelöscht oder alternativ anonymisiert wird, muss noch genauer geklärt werden. Demnach werden Benutzer aktuell noch mit dem Status „deactivated“ im System geführt.

Alle grafischen Darstellungen der Unterprozesse dieses Prozesses sind im Anhang B.1 zu finden.

Ein Mitarbeiter wird im Identitätsmanagement während seines Lebenszyklus in unterschiedlichen Status geführt. Folgende Status werden hierbei unterschieden:

- *Created* – Ein Benutzer im Status „Created“ wurde angelegt und hat keinerlei Aktivitäten mit seinem Account durchgeführt.

- *Activated* – Nachdem ein Benutzer über das Mitarbeiterportal die Aktivierung durchgeführt hat, wird er unter dem Status „Activated“ im Identitätsmanagement geführt.

- *Forwarding completed* – Durch das Einrichten der E-Mail-Weiterleitung wechselt ein Benutzer in den Status „forwarding completed“.

- *Deactivated* – Ein Benutzer, welcher in der Zentralen Universitätsverwaltung als nicht aktiver Mitarbeiter geführt wird, wird im Identitätsmanagement als „deactived“ geführt. In diesem Zustand kann ein Mitarbeiter seine E-Mail-Adresse weiterhin verwenden. Weitere Funktionalitäten, wie der Zugang auf das Mitarbeiterportal stehen nicht zur Verfügung.

4.3.2 Mitarbeiter am Campus Nord

Die Provisionierung der Mitarbeiter am Campus Nord findet ebenfalls einmal täglich statt. Hierbei werden aus dem Active Directory des Forschungszentrums (siehe Abschnitt 4.2.3), welches als autoritatives System für die Mitarbeiter des Forschungszentrums dient, alle als KIT-Mitarbeiter gekennzeichneten Benutzer exportiert und per ssh auf einen Server des Identitätsmanagementsystems geschoben. Das Identitätsmanagementsystem provisioniert die Benutzer in die angeschlossenen Systeme. Auftretende Änderungen und Deaktivierungen nach dem Ausscheiden von Mitarbeitern werden synchronisiert. Alle Prozessschritte erfolgen analog zu der Provisionierung von Campus Süd Mitarbeiter. Da es aktuell keine gemeinsame autoritative Quelle für alle KIT-Mitarbeiter gibt, ist diese Trennung notwendig.

Die Status eines Forschungszentrumsmitarbeiter unterscheiden sich von denen eines Universitätsmitarbeiters dadurch, dass die E-Mail-Weiterleitungsadresse eines Forschungszentrumsmitarbeiter bereits in der autoritativen Quelle vorhanden sind. Demnach muss ein FZK-Mitarbeiter die Weiterleitung nicht einrichten. Um dennoch die gleichen Bezeichnungen für die Status zu verwenden, wurde der erste Status übersprungen. Ein FZK-Mitarbeiter nimmt demnach folgende Status während eines Lebenszyklus ein:

- *Activated* - Ein neu angelegter FZK-Mitarbeiter befindet sich unmittelbar nach der Erzeugung im Status „activated“.

- *Forwarding completed* - Durch die Aktivierung des Accounts über das Mitarbeiterportal wechselt ein Benutzer in den Status „forwarding completed“. Die Aktivierung findet hierbei durch das Setzen eines Passwortes statt.

- *Deactivated* - Ein Benutzer, welcher im FZK Active Directory nicht mehr als aktiver Mitarbeiter geführt wird, wird im Identitätsmanagement als „deactived“ geführt.

4.3.3 Studierende

Neben der Provisionierung von Mitarbeitern, werden auch Studierende provisioniert (siehe Abschnitt 4.47). Hierbei wird jedem Studierenden unmittelbar nach deren Immatrikulation ein KIT-Account und eine hiermit verbundene *student.kit.edu*-E-Mail-

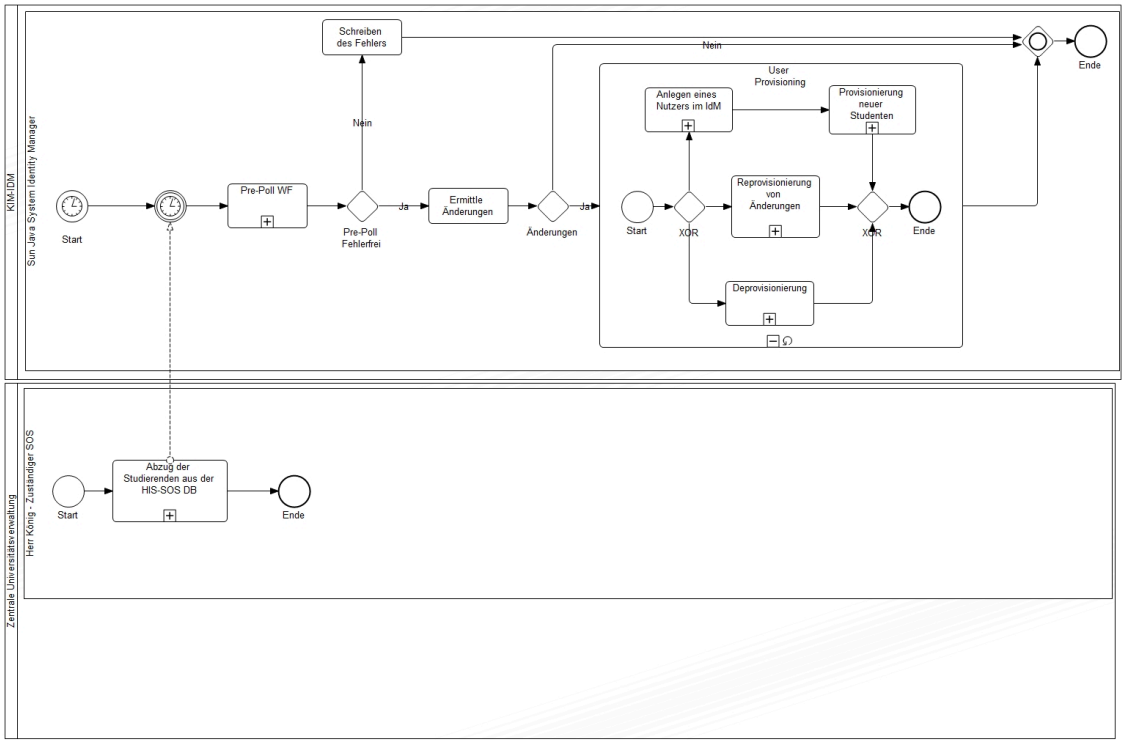


Abbildung 4.47: Hauptprozess Provisionierung Studierende

Adresse der Form `vorname.nachname@student.kit.edu` angelegt. Dieser Vorgang wird im Gegensatz zu der Mitarbeiterprovisionierung jedoch nicht durch das KIT-weite Identitätsmanagement durchgeführt, sondern durch das SCC. Das KIT-weite Identitätsmanagement ist hierbei für die Stammdaten (siehe Abschnitt 4.2) und zusätzliche Attribute wie die *student.kit.edu*-E-Mail-Adresse, das Passwort für das Benutzerkonto und ein SCC-Benutzerkonto.

Es werden einmal täglich aktuelle Daten aus der Zentralen Universitätsverwaltung verschlüsselt per ssh an das KIT-weite Identitätsmanagement übertragen. Dieser Vorgang ist automatisiert und muss nur initial eingerichtet werden. Daraufhin werden durch das Identitätsmanagement eine Vielzahl von Konsistenzchecks auf den gelieferten Daten durchgeführt. Unter anderem wird die Gesamtanzahl der Studierenden, die korrekte Anzahl der Attribute und das Vorhandensein verpflichtender Attributwerte geprüft. Falls dieser Check erfolgreich abläuft, werden durch das Identitätsmanagementsystem Änderungen an den Studierendendaten ermittelt. Es wird typischerweise zwischen den Events *Create*, *Update*, *Disable*, *Enable*, *Delete* unterschieden. Je nach Änderung wird ein entsprechender Prozess innerhalb des Identitätsmanagementsystems durchgeführt. Folgende Unterprozesse sind hierbei möglich:

- *Create* – Der initiale Prozess legt ein neues Benutzerkonto für einen Studierenden an. Die zusätzlich generierten Attribute werden gemeinsam mit den Attributen der Zentralen Universitätsverwaltung in die angeschlossenen System provisioniert. Die angeschlossenen Systeme und die dazugehörigen Attribute sind in Abschnitt 4.2 zu finden.

- *Update* – Anpassungen bei Änderungen sind bspw. bei Heirat oder Umzug notwendig. Hierbei werden die Änderungen an die entsprechenden Ressourcen weitergeleitet.

- *Disable* – Eine der wichtigsten Schritte ist die Deprovisionierung. Nachdem ein Studierender das KIT verlässt, werden unmittelbar alle an das KIT-weite Identitätsmanagement angeschlossenen und für diesen Studierenden relevante Systeme hierüber informiert. Dies ermöglicht ein schnelles Entziehen aller Berechtigungen eines Benutzers und verhindert darüber hinaus „Account-Leichen“.

- *Enable* – Da die Möglichkeit besteht, dass Studierende bspw. aufgrund einer zu späten Rückmeldung kurzzeitig als exmatrikuliert geführt werden, ist es notwendig, diese nach der erfolgreichen Rückmeldung zu „reaktivieren“. Hierbei wird für einen Studierenden kein neuer Account angelegt, sondern es wird der bereits bestehende Account, welcher sich aktuell im Status „deactivated“ befindet wieder aktiviert.

- *Delete* – Die Frage wann ein Studierender und das dazugehörige Benutzerkonto vollständig gelöscht oder alternativ anonymisiert wird, muss noch genauer geklärt werden. Demnach werden Benutzer aktuell noch mit dem Status „deactivated“ im System geführt.

Ein Studierender wird im Identitätsmanagement während seines Lebenszyklus in unterschiedlichen Status geführt. Folgende Status werden unterschieden. Die Status sind hierbei unabhängig von der Anzahl der Semester und beziehen sich immer auf das aktuelle Semester:

- *Ersteinschreibung* – Ein Studierender mit dem Status „Ersteinschreibung“ war zuvor an keiner weiteren Hochschule und auch zu keinem anderen Studiengang immatrikuliert.

- *Rückmeldung* – Ein Studierender, der sich nach Beendigung des ersten Semesters rückmeldet, wird unter dem Status „Rückmeldung“ geführt.

- *Neueinschreibung* – Bei einem Wechsel von der eigenen Hochschule in einen anderen Studiengang oder von einer fremden Hochschule an das KIT erhält den Status „Neueinschreibung“.

- *Beurlaubung* – Ein Studierender, der sich in einem Urlaubssemester befindet, wird unter dem Status „Beurlaubung“ geführt.

- *Exmatrikuliert* – Ein Studierender, der nicht mehr am KIT eingeschrieben ist, wird unter dem Status „Exmatrikuliert“ geführt.

5. Portaldienste

Es wurden sowohl Dienste für das Mitarbeiterportal *intra.kit.edu* als auch für das Studierendenportal *studium.kit.edu* entwickelt, die in diesem Kapitel näher erläutert werden. Zunächst erfolgt in eine Beschreibung der grundlegenden Architektur des Portals. In den folgenden beiden Abschnitten werden die einzelnen entwickelten Dienste im Detail beschrieben.

5.1 Grundlegende Portal-Architektur

Abbildung 5.1 gibt eine Übersicht der integrierten Systemkomponenten. Der auf oberster Ebene angesiedelte Microsoft Office Sharepoint Servers 2007 (MOSS 2007) realisiert die Portale. An diesem authentifiziert sich der Nutzer durch die so genannte Forms Authentication, welche im Gegensatz zur integrierten Windows Authentication den Einsatz eigener integrierter Login Formulare ermöglichen. Die Authentifikation durch Forms Authentication wird mit Hilfe eines Active Directory realisiert. Dieses Active Directory wird zuvor mit Hilfe des Sun Identity Manager aus dem Datenbestand der HIS-Software mit einer minimalen Menge an personenbezogenen Attributen provisioniert (siehe Kapitel 4). Ferner wird die Windows Workflow und Communication Foundation für die Umsetzung der Workflows genutzt, die im folgenden Abschnitt näher beschrieben werden.

5.1.1 Integrierte Workflows

Der in Abbildung 5.2 dargestellte Zustandsautomat wurde mittels Visual Studio 2005 modelliert und darauf basierend das Rahmengerüst des Zustandsautomaten in C# automatisiert generiert. Die Kästen in der Abbildung 5.2 symbolisieren die Zustände (States), in den Zuständen befinden sich auf Ereignisse (Events) gebundene Aktivitäten (Activities), und die gerichteten Pfeile symbolisieren Zustandsübergänge (State Transitions). Bei der Erstellung des Zustandsautomaten wurde jede interaktive Webseite als genau ein Zustand modelliert, der in der Regel auf der ersten Ebene aus zwei Activities besteht. Zum einen die Entry Activity, die beim Betreten

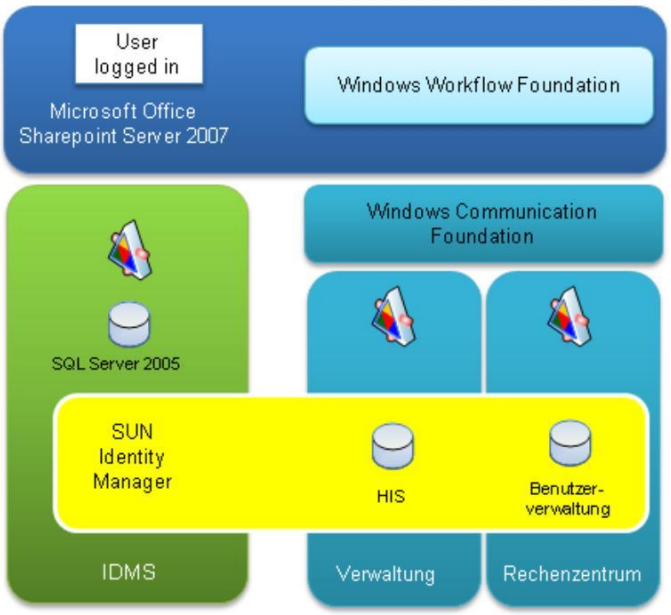


Abbildung 5.1: Systemkomponenten der Portal-Architektur

des Zustandes ausgeführt wird und in der Regel Web Service Anfragen, Logikauswertungen und Aktionen wie z.B. die dynamische Generierung einer Portalwebseite ausführt und zum anderen die User Interaction Activity, die durch eine Benutzeraktion ausgelöstes Event aufgerufen wird und in der Regel Aktionen, Logikauswertungen und Zustandsübergänge ausführt. Eine Activity beim Windows Workflow Foundation Framework muss nicht atomar sein, sondern kann sich wieder aus mehreren anderen Activities zusammensetzen.

Der Ablauf der Zustandsautomaten beinhalten Web Service Aufrufe, über die notwendige nutzerspezifische Attribute aus den datenführenden Systemen geliefert und wieder geschrieben werden. Dabei kommen sowohl .Net-Web Services auf Basis der Windows Communication Foundation (WCF) als auch Java-basierte Web Services zum Einsatz, um die verschiedenen vorhandenen Systeme zu integrieren. Die Provisionierung von Identitätsattributen wird über die SPML-Schnittstelle des Sun Identity Manager vorgenommen.

5.1.2 CRUDS+F*

Mit dem CRUDS+F*-Muster wird ein Entwurfsmuster zur Realisierung feingranularer, geschäftsprozessorientierter Dienste vorgestellt, ohne auf die Wiederverwendbarkeit eines grobgranularen Ansatzes zu verzichten.

Das CRUDS+F*-Muster definiert hierfür so genannte Informationszugangspunkte, die den Zugriff auf Fachklassen – Obermengen semantisch stark kohäsiver Geschäftsobjekte – kapseln. Beispiele für Fachklassen sind Personen, Veranstaltungen oder Prüfungen. In der Fachklasse Personen werden alle personenbezogenen Geschäftsobjekte wie Mitarbeiter und Studierende subsumiert. Die Menge aller Fachklassen bildet den Informationsraum der verfügbaren Informationen. Um auf die Anforderun-

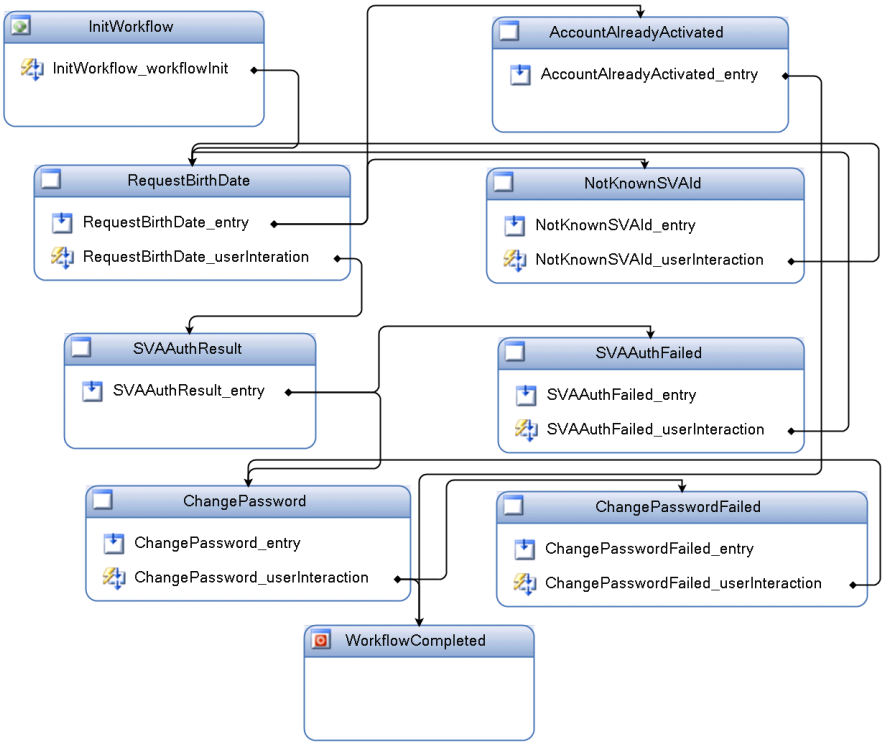


Abbildung 5.2: Beispielhafter Workflow

gen einzelner Nutzungsszenarien eingehen zu können, ermöglicht das CRUDS+F*-Entwurfsmuster die Ausprägung von Informationszugangspunkten in zwei Dimensionen. Zum einen kann eine Fachklasse um zusätzliche und neue Versionen bestehender Geschäftsobjekte ergänzt werden. Zum anderen kann ein Informationszugangspunkt um verfeinerte und spezialisierte Schnittstellen erweitert werden.

Mit dem CRUDS+F*-Muster wird ein einheitlicher Zugriff auf die durch Fachklassen gebildeten Segmente des Informationsraums durch eine generische CRUDS-Schnittstelle ermöglicht. Aufbauend auf dieser einheitlichen Schnittstelle kann der Informationszugangspunkt um spezifisch zugeschnittene Schnittstellen erweitert werden. Die für Informationszugangspunkte verwendete CRUDS-Schnittstelle umfasst hierbei die Methoden Create, Read, Update, Delete und Search. Jede der CRUDS-Methoden wird mit einem sogenannten Kontext in Form eines XML-Dokuments parametrisiert aufgerufen und liefert wiederum ein XML-Dokument zurück. Sowohl die Kontexte als auch die Rückgabewerte der einzelnen Methoden können in verschiedenen Ausprägungen existieren und werden jeweils durch ein XML-Schema definiert.

Eine abrufbare ServiceCard informiert über die von einem Dienst unterstützten Ausprägungen der Kontexte und Rückgabewerte. Hierdurch können verschiedene Geschäftsobjekte unterschiedlicher Granularitäten und Versionen durch einen einzelnen Informationszugangspunkt behandelt werden. Mit dem vorgestellten Schnittstellenentwurf wird eine Mischung aus Methoden-orientierter Schnittstelle einerseits (CRUDS-Methoden) und Nachrichten-orientierter Schnittstelle andererseits (Kontexte und Rückgabewerte) verfolgt. Diese Realisierung einer parametrischen Polymorphie führt zu einem hohen Grad an Abstraktion und ermöglicht damit eine kon-

tinuierliche Evolution auf Kontext- und Geschäftsobjektebene. In Listing 5.1 ist ein exemplarischer SearchContext abgebildet, der eine Suche nach Personen anhand der im Query-Element definierten Suchkriterien ermöglicht. Das Element OutputSchema gibt den erwarteten Typ des Rückgabewertes an.

Listing 5.1: Beispielhafter SearchContext

```
<SearchContextXmlQuery xmlns="...">
  <QueryXml>
    <PersonQuery xmlns="...">
      <LastName>Meier</LastName>
      <PlaceOfBirth>Karlsruhe</PlaceOfBirth>
    </PersonQuery >
  </QueryXml>
  <Sort>LastName ASC</Sort>
    <From>0</From>
    <To>-1</To>
    <QueryMode>Tile</QueryMode>
  <OutputSchema>http://../types/PersonType.xsd:Student</OutputSchema>
</SearchContextXmlQuery>
```

In vielen Anwendungsszenarien sind die durch Informationszugangspunkte behandelten Geschäftsobjekte zu grobgranular oder es existieren rechtliche Beschränkungen für die Nutzung der bereitgestellten Daten. Um Dienstnehmern einen auf ihre Anforderungen zugeschnittenen Zugriff auf Informationen zu ermöglichen, wird im CRUDS+F*-Entwurfsmuster die Erweiterung von Informationszugangspunkten durch Fassadierung der CRUDS-Schnittstelle erreicht. Fassaden verbergen die Komplexität der CRUDS-Schnittstelle und stellen Kunden genau die von ihnen gewünschte Funktionalität zur Verfügung, wobei Fassadenmethoden auf die CRUDS Methoden und entsprechende Kontexte abgebildet werden.

Durch die Fassadierung der CRUDS-Schnittstelle können spezifische Schnittstellen für bestimmte Nutzungsszenarien, wodurch den Anforderungen des Datenschutzes entsprochen werden kann, bereitgestellt werden. Durch die generische CRUDS-Schnittstelle bieten Informationszugangspunkte einen einheitlichen und stabilen Zugriff auf Informationen einer Fachklasse und ermöglichen eine Datenintegration und -konsolidierung über verschiedene datenführende Systeme hinweg. Durch die Entwicklung von Fassaden kann die Ausprägung eines Informationszugangspunkt spezialisiert werden und so auf neue Anforderungen reagiert werden.

5.1.3 WS*-Spezifikationen

Der Erfolg von Web Services ist nicht zuletzt auf die Einigung von Herstellern auf Standards zurückzuführen. In der WS*-Architektur werden Standards zu Sicherheit, Zuverlässigkeit und Transaktionen von Web Services zusammengefasst.

WS-Security ist die Grundlage aller weiteren WS*-Spezifikationen und definiert Erweiterungen von SOAP. Zum einen dient WS-Security zur Gewährleistung von Vertraulichkeit und Integrität von SOAP Nachrichten. Hierfür werden keine neuen Sicherheitsmechanismen definiert, sondern beschrieben, wie existierende Sicherheitsstandards wie XML Encryption und XML Signature angewendet werden können. Zum anderen spezifiziert WS-Security die Verwendung von Security Tokens, die zum

sicheren Austausch von Identitäts-, Authentifikations- und Autorisationsinformationen in SOAP Nachrichten verwendet werden.

WS-SecurityPolicy beschreibt Anforderungen, Fähigkeiten und Zusicherungen, die die Sicherheit von Web Services betreffen. Ein Web Service, der WS-Security benutzt, muss viele Details klären, bspw. die akzeptierten Verschlüsselungs- und Signierungsalgorithmen oder die zur Authentifikation erlaubten Credentials wie Kerberos Tickets, X.509 Zertifikate oder SAML Token. Wie allerdings Policies verteilt oder bezogen werden können, liegt außerhalb dieser Spezifikation. Hierfür spezifiziert WS-PolicyAttachment die Verbindung von Policies mit der WSDL eines Web Service. Aufbauend hierauf definiert WS-MetaDataExchange, wie diese Informationen von einem Web Service erhalten werden können.

WS-Trust definiert ein Nachrichtenprotokoll, das zur Ausstellung, Erneuerung und Bestätigung von Security Tokens zwischen einem Web Service und einem Security Token Service (STS) dient. Der Ablauf und das Zusammenspiel zwischen WS-Security und WS-Trust ist wie folgt: Der Web Service Client verwendet WS-Trust, um mit dem STS zu kommunizieren und sich ein vertrauenswürdiges Security Token, welches die Identität des Benutzers repräsentiert, ausstellen zu lassen. Der Web Service Client fügt beim Aufruf des Web Service Providers das vom STS ausgestellte Token in den WS-Security Header der SOAP Nachricht ein. Um das Security Token zu validieren, muss eine Vertrauensbeziehung zwischen dem Web Service Provider und dem STS, der den Token ausgestellt hat, bestehen. Auf Grundlage dieser Vertrauensbeziehung kann der Web Service Provider selbst das Security Token validieren. Alternativ kann der Web Service Provider den STS auch dazu benutzen, die Überprüfung des Security Tokens vorzunehmen. Mittels der Informationen, welche in dem Token enthalten sind, kann der Web Service Provider eine Authentifikations- und Autorisationsentscheidung treffen.

Leider traten bei der Umsetzung der WS-Security-geschützten Kommunikation mehrere Inkompatibilitäten zwischen den Web Services der Windows Communication Foundation (WCF) und derer der Java Web Services Interoperability Technologies (WSIT) auf. Bei der Nutzung eines WCF-Clients als Konsument für einen WSIT-Service entstanden größere Herausforderungen. Dies liegt vor allem darin begründet, dass die mit WSIT-internen Tools erstellte WSDL-Beschreibung des Web Services nicht mit WCF interoperabel ist. Die größten Probleme sind hierbei zum einen das Setzen der Transportkonfiguration (SOAP 1.2) auf einen mit WCF nicht kompatiblen Wert und zum anderen die nicht standardgemäße Nutzung von WS-Addressing, was für die Kommunikation mit WCF (wsHttpBinding) notwendig ist. Weitere Inkompatibilitätsprobleme wurden durch Einstellungen der Richtlinien, wie etwa Policies für WS-Security oder WS-SecureConversation verursacht. Als weniger problematisch stellte sich die Verwendung eines WSIT-Clients als Konsument eines WCF-Service heraus. In diesem Fall sind keine Microsoft-eigenen Erweiterungen erlaubt. Eine weitere Herausforderung im Umgang mit Web Services ist die architekturelle Betrachtung der Authentifikation und Autorisationsdurchführung [Höllrigl et al. 2008].

5.2 Mitarbeiterportal

5.2.1 Login-Prozedur

Die Login-Prozedur des Mitarbeiterportals umfasst im Wesentlichen zwei Teilbereiche. Zum Einen sorgt ein sogenannter Membershipprovider für die Authentifikation der Nutzer, ein Roleprovider weist den einzelnen Nutzern anschließend anhand einer Datenkollektion aus unterschiedlichen Quellen diverse Rollen zu. Diese Rollen sind notwendig, um Autorisationsentscheidungen auf den einzelnen Diensten im Portal treffen zu können.

Abbildung 5.3 zeigt den modellierten Login-Prozess des Mitarbeiterportals. Ein Nutzer möchte das Portal nutzen und gelangt über *https://intra.kit.edu* zunächst auf die Login-Seite. Hier wird ihm die Aufforderung zur Eingabe der Login-Daten visualisiert. Die eingegebenen Werte werden nach einem Klick auf den Button „Anmelden“ an den Membershipprovider übergeben. Dieser überprüft zunächst, ob der Nutzer sich mit dem UPN (User Principal Name) oder dem sAMAccountName anmelden möchte. Wurde der sAMAccountName verwendet, wird der dazu gehörige UPN aus dem Active Directory gelesen. UPN und Passwort werden anschließend gegen die hinterlegten Werte im Active Directory geprüft. Das Resultat der Authentifikation wird zurück gegeben. Ist die Authentifikation fehlgeschlagen, wird dies dem Nutzer entsprechend visualisiert. Bei erfolgreicher Überprüfung der Eingabedaten wird der Roleprovider angesprochen. Dieser zieht verschiedene Attribute aus unterschiedlichen Datenquellen zusammen, um dem Nutzer Rollen zuzuordnen. Hinter dem Task „Assign Roles“ steckt dabei ein komplexer Prozess der Attributauswertung und Zuweisung der diversen Rollen. Die Auswertungslogik wird im folgenden Abschnitt beschrieben. Die abgefragten Datenquellen sind das KISS-Repository, eine Hilfsrelation im KISS-Repository, die den Wissenschaftler-Status der Mitarbeiter des Campus Süd enthält und die WissMA Datenbank, die diesen Status für die Mitarbeiter des Campus Nord beinhaltet. Das Befüllen der Hilfsrelation erfolgt über das DB-Interface und ist im unteren Abschnitt der Abbildung 5.3 modelliert.

Die Logik des Roleprovider ist im Entscheidungsbaum in Abbildung 5.4 und in Tabelle 5.1 verzeichnet. Abbildung 5.4 zeigt die Abbildung der Attribute auf den Account-Status eines Mitarbeiters. Tabelle 5.1 bildet den Status auf die Rollen für das Mitarbeiterportal ab. Die den unterschiedlichen Rollen zugewiesenen Mitarbeiterportal-Features sind in Tabelle 5.2 aufgetragen. Hierbei zeigt ein ’x’, dass ein Träger der angegebenen Rolle das Feature explizit nutzen darf, ein ’-’ bedeutet, dass der Zugriff auf das Feature dem Träger der angegebenen Rolle verwehrt wird. Eine leere Zelle weißt darauf hin, dass das Feature implizit genutzt werden kann, da ein Träger der angegebenen Rolle auch im Besitz der für den Zugriff notwendigen Rolle ist.

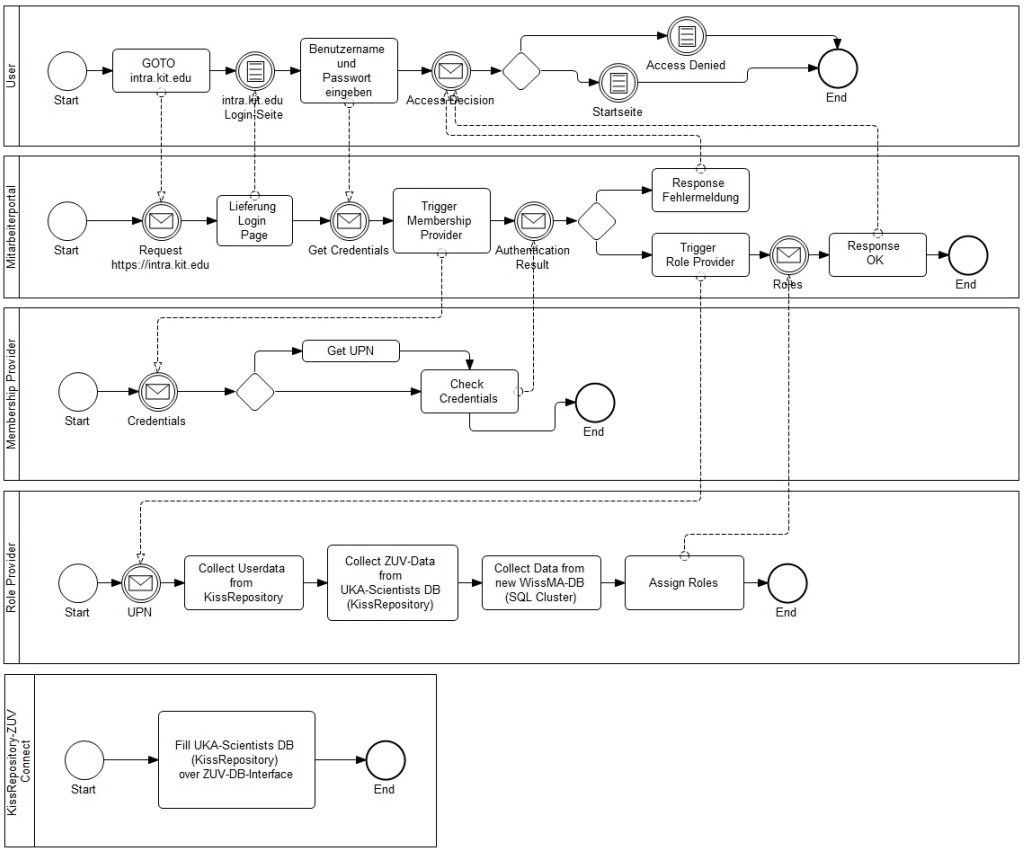


Abbildung 5.3: Prozessmodell des Membership- und Roleprovider

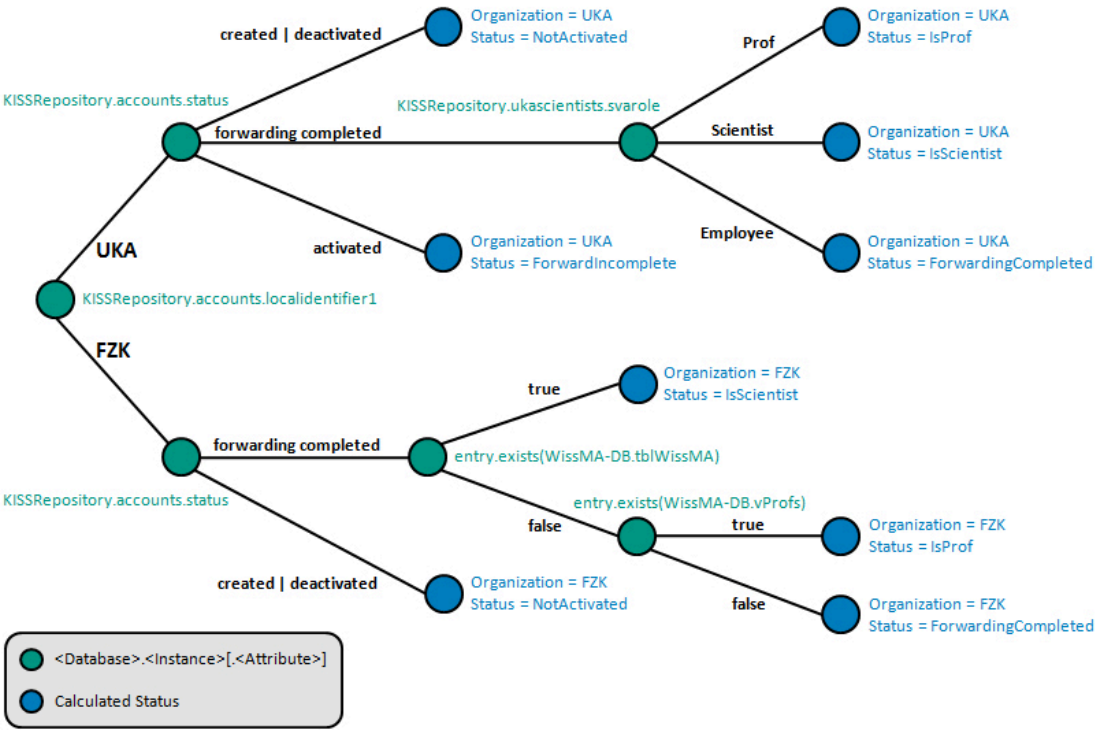


Abbildung 5.4: Auswertung der Attributwerte und Mapping auf jeweiligen Status

Organization	Status	NotActivatedUKA	NOTActivatedFZK	NotForwardedUKA	KIT-Mitarbeiter	KIT-Mitarbeiter (wiss.)	UKA-Mitarbeiter	UKA-Professoren	FZK-Mitarbeiter	FZK-Professoren
UKA	NotActivated	x	-	-	-	-	-	-	-	-
	ForwardIncomplete	x	-	x	-	-	-	-	-	-
	ForwardingCompleted	-	-	-	x	-	x	-	-	-
	IsProf	-	-	-	x	x	x	x	-	-
	IsScientist	-	-	-	x	x	x	-	-	-
FZK	NotActivated	-	x	-	-	-	-	-	-	-
	ForwardingCompleted	-	-	-	x	-	-	-	x	-
	IsProf	-	-	-	x	x	-	-	x	x
	IsScientist	-	-	-	x	x	-	-	x	-

Tabelle 5.1: Mapping vom ermittelten Status auf Rollen für das Mitarbeiterportal

Portal-Feature	NotActivatedUKA	NOTActivatedFZK	NotForwardedUKA	KIT-Mitarbeiter	KIT-Mitarbeiter (wiss.)	UKA-Mitarbeiter	UKA-Professoren	FZK-Mitarbeiter	FZK-Professoren
Benutzerkonto aktivieren UKA	x	-	x	-	-	-	-	-	-
Benutzerkonto aktivieren FZK	-	x	-	-	-	-	-	-	-
KIT-Kommunikation	-	-	-	x					
Kompetenzportfolio	-	-	-	-	x	-		-	
Vodafone-Mobilität	personenbezogen								
Passwort ändern	-	-	-	x		x		x	
KIT-E-Mail-Adresse	-	-	x	x		x		x	
E-Mail-Alias	-	-	-	x					

Tabelle 5.2: Mapping von zugewiesenen Rollen auf Features im Mitarbeiterportal

5.2.2 Aktivierungsdienst

Ein Nutzer des Mitarbeiterportals kann zunächst auf keinerlei weiterführenden Features zugreifen. Er muss sich für diese Dienste zunächst freischalten, indem ein Aktivierungsprozess durchlaufen wird. Während dieses Prozesses wird in Interaktion mit dem Benutzer eine Abbildung von dem Portalaccount und den Verwaltungsaccount vorgenommen.

Nach der erfolgreichen Authentifikation des Nutzers am Portal, wird eingebettet in einen MOSS Webpart über die Workflow Engine der Windows Workflow Foundation der Account Linking Prozess ausgeführt. In Abbildung 5.5 ist das Prozessmodell des Aktivierungsprozess für Mitarbeiter dargestellt. Während dieses Ablaufs authentifiziert sich der Benutzer gegenüber der Zentralen Universitätsverwaltung (ZUV) mit der zusätzlichen Angabe seines Geburtsdatums und setzt nach erfolgreicher Authentifikation sein Portalpasswort neu. Nach dem erfolgreichen Setzen des neuen Portalpasswortes wird der Benutzeraccount endgültig aktiviert, und der Benutzer kann weitere Funktionen des Portals nutzen. Das Passwort wird über einen SPML Service (siehe Abschnitt 6.1.8) bereitgestellt, welcher über das Identitätsmanagementsystem zur Verfügung gestellt wird. Dies ermöglicht die Nutzung der Logging Funktionalität des Sun Identity Managers. Es ist demnach auch zu einem späteren Zeitpunkt noch möglich die Aktivierung durch einen Audit-Log-Report nachzuvollziehen.

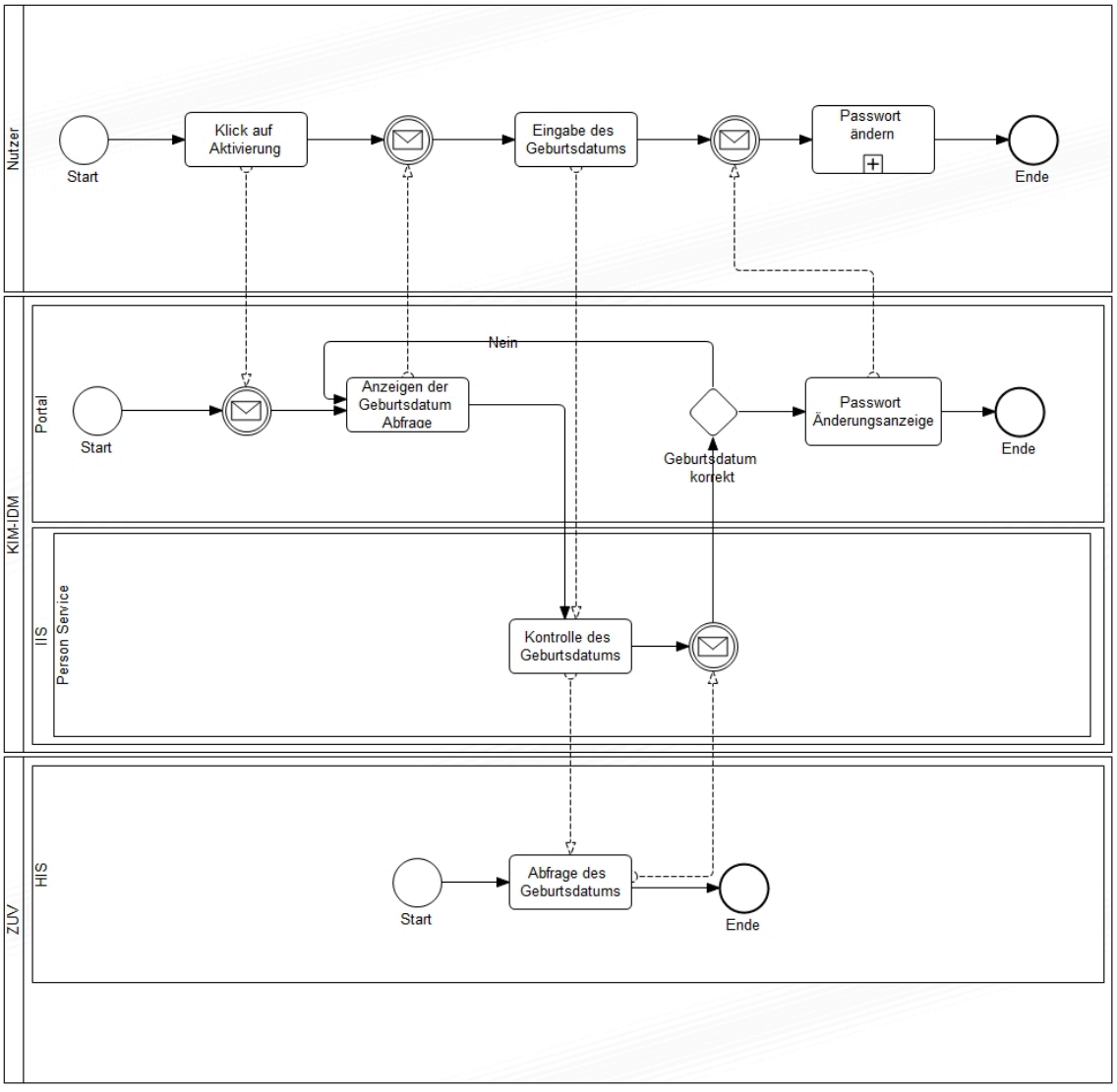


Abbildung 5.5: Aktivierungsprozess für Mitarbeiter des Campus Süd

5.2.3 Weiterleitungeinrichtungsdienst

Über den Weiterleitungeinrichtungsdienst kann ein Mitarbeiter des Campus Süd die E-Mail-Adresse, auf welche er seine *kit.edu*-E-Mail-Adresse weiterleiten möchte, einrichten. Dies ist hauptsächlich auch deshalb notwendig, da der *kit.edu*-E-Mail-Account kein eigenes Postfach beinhaltet. Als Weiterleitungsadresse kann hierbei allerdings ausschließlich ein Account des Steinbuch Centre for Computing verwendet werden. Um bei der ersten Anmeldung am Portal die Weiterleitung einzurichten, muss der Mitarbeiter am Portal auf den Menüpunkt E-Mail-Weiterleitung navigieren. Über einen Web Service wird nun geprüft, ob der Mitarbeiter seinen Account bereits aktiviert hat. Falls dies nicht der Fall ist, bekommt der Mitarbeiter einen Hinweis auf den Aktivierungsprozess (vgl. Abschnitt 5.2.2). Die erfolgreiche Durchführung der Aktivierung ist zwingend notwendig, um eine Weiterleitung einzurichten.

Falls der Mitarbeiter die Aktivierung erfolgreich durchgeführt hat, bekommt er auf der E-Mail-Weiterleitungsseite die Möglichkeit seinen SCC-Account und das dazugehörige Passwort anzugeben. Sollte ein Mitarbeiter noch keinen SCC-Account besitzen, kann er an dieser Stelle einen ihm vorgeschlagenen Account durch das setzen eines neuen Passworts anlegen. Es ist demnach nicht mehr notwendig sich am SCC über das alte Papier-basierte Verfahren einen Account zu beantragen, sondern der Account kann direkt im Mitarbeiterportal angelegt werden.

Im Falle der Einrichtung der Weiterleitung auf einen bestehenden Account, wird das Passwort gegen einen LDAP am SCC geprüft. Sollte diese Überprüfung positiv verlaufen, wird über einen SPML Service (siehe Abschnitt 6.1.8) die Weiterleitung auf dem Exchange Server eingerichtet. Dieser Service wird durch das Identitätsmanagementsystem zur Verfügung gestellt und nutzt die Logging Funktionalität. Es ist demnach auch zu einem späteren Zeitpunkt möglich, die Einrichtung der Weiterleitung durch einen Audit-Log-Report nachzuvollziehen.

Die erfolgreiche Einrichtung der Weiterleitung wird durch das Versenden einer Bestätigungsemail an die neue *kit.edu*-E-Mail-Adresse abgeschlossen. Der Benutzer bekommt zusätzlich das Resultat der Einrichtung auf einer Bestätigungsseite im Mitarbeiterportal angezeigt.

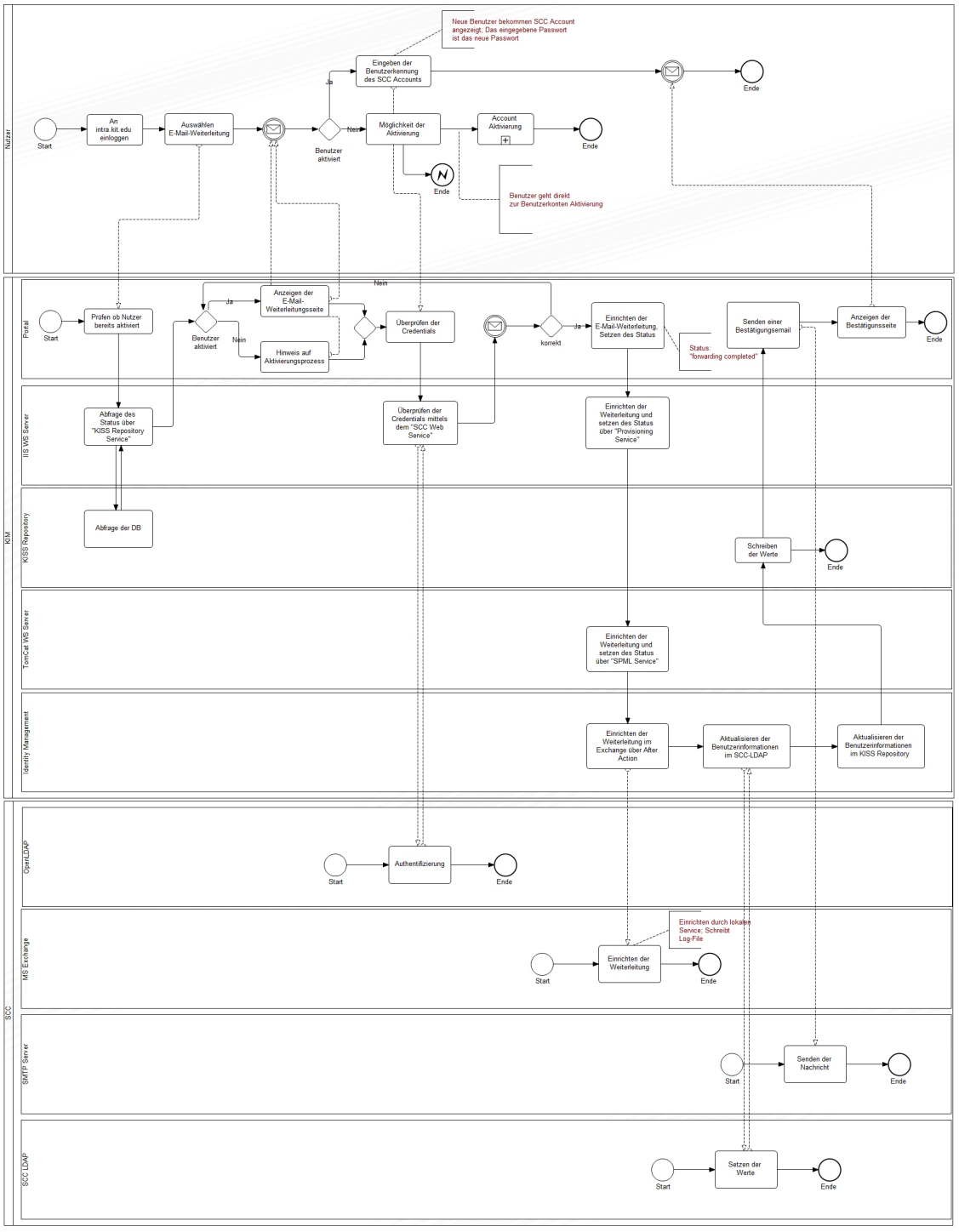


Abbildung 5.6: Prozess zur Einrichtung der einer *kit.edu*-E-Mail-Weiterleitung am Campus Süd

5.2.4 KIT-E-Mail-Alias-Dienst

Der E-Mail-Alias-Dienst gibt den KIT-Mitarbeitern die Möglichkeit, für die bereits eingerichtete *kit.edu*-E-Mail-Adresse einen E-Mail-Alias zu generieren und automatisiert freischalten zu lassen. Ein E-Mail-Alias ist eine weitere E-Mail-Adresse, die alle eingehenden E-Mails an die Haupt-E-Mail-Adresse weiterleitet. Als Namenskonvention für *kit.edu*-E-Mail-Aliase wurde eine Richtlinie zum Anlegen eingesetzt. Diese wird im folgenden Abschnitt vorgestellt. Im weiteren Verlauf sollen die Prozesse im Zusammenhang mit dem E-Mail-Alias-Dienst beschrieben werden und dem interessierten Leser eine technische Betrachtung gegeben werden.

5.2.4.1 KIT E-Mail-Alias-Richtlinie

Um das Auftreten des KIT möglichst einheitlich zu gestalten, wurden durch das SCC für E-Mail-Adressen, demnach auch für E-Mail-Aliase, Richtlinien aufgestellt und in der Sitzung des KIT-Senatsausschusses IV-A vom 16. April 2008 beschlossen. Die für den hier beschriebenen Dienst gültige E-Mail-Alias-Richtlinie ist aus oben genanntem Beschluss abgeleitet. Der Dienst unterstützt dabei nicht das Anlegen eines E-Mail-Alias mit den Initialen eines Nutzers als Präfix, wie es der Beschluss erlauben würde. Mit dieser Einschränkung möchte das SCC Namenskollisionen mit den Institutsabkürzungen des KIT vermeiden. Im Folgenden sind die Regeln aufgelistet, wie sie über das Mitarbeiterportal *intra.kit.edu* kommuniziert und geprüft werden.

- Jeder Nutzer kann genau einen E-Mail-Alias selbst erstellen.
- Zulässig sind E-Mail-Aliase der Form *nachname*@kit.edu, wobei *nachname* aus einer beliebigen Menge der Nachnamen eines Nutzers bestehen kann. Das bedeutet, eine Person mit einem Nachnamen hat maximal eine Option. Eine Person mit Doppel-Nachnamen bekommt mehrere Optionen zur Auswahl, sofern diese noch verfügbar und nicht reserviert sind.
- Eine zweite Regel lässt E-Mail-Aliase der Form *Teil1.Teil2*@kit.edu zu, wobei *Teil1* mindestens einen Buchstaben oder eine Zahl enthalten muss und die Sonderzeichen '.' (Punkt) und '-' (Bindestrich) enthalten darf. Es dürfen jedoch nicht zwei Sonderzeichen aufeinander folgen. *Teil2* muss mindestens mit den ersten drei Buchstaben eines Nachnamens anfangen und kann nur eine Teilmenge des Nachnamens oder einer Nachnamenkombination sein.
- Neben den oben genannten Regeln, gelten die allgemeinen Bedingungen für E-Mail-Adressen gemäß RFC 2822.

Mit dieser E-Mail-Alias-Richtlinie kann ein Nutzer mit dem Namen „Max Muster“ den E-Mail-Alias muster@kit.edu nach dem Muster nachname@kit.edu (nachfolgenden Typ 1 genannt) einrichten. Ebenfalls möglich ist ein E-Mail-Alias der Form Teil1.Teil2@kit.edu (nachfolgend Typ 2 genannt), wobei Teil2 mit den Buchstaben „mus“ beginnen muss und nur ein Fragment des Nachnamens sein kann. Das bedeutet Teil2 könnte „mus“, „must“, „muste“ oder „muster“ lauten. Teil1 hingegen

Julia Muster	j.muster@kit.edu julia.mus@kit.edu
Sandra Muster-Koenig	s.mus@kit.edu sandra.koe@kit.edu beliebig-text.muster@kit.edu s.koenig@kit.edu s.muster-koenig@kit.edu
Tobias König Müller	t.koe@kit.edu tobias.mue@kit.edu beliebig-text.koenig@kit.edu t.mueller@kit.edu tobias.koenigmueller@kit.edu t.muellerkoenig@kit.edu

Tabelle 5.3: Beispiele für *kit.edu*-E-Mail-Aliase

kann aus beliebigen Buchstaben und Ziffern, sowie aus den Sonderzeichen „.“ und „-“ bestehen. Es dürfen jedoch keine zwei Sonderzeichen aufeinander folgen. Weitere Beispiele für die freie E-Mail-Aliaswahl finden sich in Tabelle 5.3. Es ist hier lediglich eine Auswahl an Kombinationsmöglichkeiten pro Name dargestellt.

5.2.4.2 Prozessbeschreibung „E-Mail-Alias anlegen“

Die E-Mail-Aliase können von den Mitarbeitern des KIT über das Mitarbeiterportal *intra.kit.edu* für den eigenen Benutzeraccount nach den im vorigen Abschnitt beschriebenen Regeln eingerichtet werden.

Hierzu müssen sich die Mitarbeiter im Portal einloggen und den entsprechenden Link zum E-Mail-Alias-Dienst wählen. Die sich daraufhin öffnende Sicht zeigt einen Status-Überblick mit folgenden Informationen: Aktuell aktive E-Mail-Adressen des angemeldeten Mitarbeiters, ggf. bereits eingerichtete E-Mail-Aliase und eine Information, ob sich der Einrichtungsprozess des selbsterstellten E-Mail-Alias im Bearbeitungsfenster (s.u.) befindet oder ob diese Zeitspanne abgelaufen ist. Wurde der E-Mail-Alias bereits aktiviert, wird er zusätzlich in der Liste der aktiven *kit.edu*-E-Mail-Adressen aufgeführt.

Um diese Informationen anzeigen zu können wird der Alias Web Service aufgerufen. Dieser liefert über die Methode `getMailAddresses(string:upn)` die aktuell aktiven *kit.edu*-E-Mail-Adressen, inklusive gegebenenfalls bereits aktivierter E-Mail-Aliase. Den vom Benutzer eingerichteten E-Mail-Alias, sowie einen Zeitstempel mit Datum und Uhrzeit der Ersteinrichtung und die aktuelle Server Uhrzeit liefert die Alias Web Service Methode `getAlias(string:upn)`. Sofern bereits eine Ersteinrichtung erfolgte, wird aus diesen Daten die noch zur Verfügung stehende Bearbeitungszeit für den reservierten E-Mail-Alias berechnet. Hier werden dem Nutzer 24 Stunden gewährt, bevor der gewählte E-Mail-Alias fixiert und aktiviert wird. Innerhalb dieses Zeitfensters steht die Option „E-Mail-Alias bearbeiten...“ zur Verfügung. Wurde noch kein

E-Mail-Alias eingerichtet wird dem Benutzer ein Button mit der Aufschrift „E-Mail-Alias einrichten...“ angezeigt. Nach der Ersteinrichtung beginnt die Bearbeitungszeit, die bei Bearbeitung eines eingerichteten E-Mail-Alias nicht verlängert wird.

Richtet der Nutzer seinen E-Mail-Alias erstmalig ein, gelangt er nach der Statusseite zur Einrichtungsseite. Neben der Darstellung der Richtlinie für *kit.edu*-E-Mail-Adressen, werden hier alle zum Nutzer passenden und zum Ladezeitpunkt der Seite noch verfügbaren E-Mail-Aliase des Typs 1 angezeigt. Beim Aufbau dieser Seite wurden dafür alle möglichen E-Mail-Aliase des Typs 1 zur Verfügbarkeitsprüfung an den Alias Web Service geschickt. Die als nicht mehr verfügbar zurück gemeldeten E-Mail-Aliase werden ausgeblendet. Eine zusätzliche Zeile mit Freitextboxen dient zur Einrichtung eines E-Mail-Alias nach Typ 2. Sollte kein E-Mail-Alias des Typ 1 verfügbar sein, wird nur diese letzte Zeile dargestellt. Nach Auswahl oder Eingabe eines Vorschlags, muss der Benutzer auf den Button „Verfügbarkeit prüfen...“ klicken. Sollte diese Anfrage, ebenfalls gegen den Alias Web Service, positiv beantwortet werden, wird dem Nutzer noch einmal der ausgewählte E-Mail-Alias und ein Button „E-Mail-Alias reservieren“, sowie ein weiterer Button zur Änderung der Auswahl präsentiert. Mit dem Klick auf den Reservierungsbutton, wird die Auswahl des Nutzers an die Alias Web Service Methode `setAlias(string:upn, string:alias)` gesendet. Erst hier entscheidet sich, ob der Nutzer den Wunsch-E-Mail-Alias erhält. Hat bis zum Klick auf den Reservierungsbutton ein anderer Nutzer die gleiche Auswahl getroffen und die Reservierung bestätigt, kann die Auswahl nicht mehr angenommen werden. In diesem Fall kann der Nutzer erneut wählen.

Innerhalb von 24 Stunden kann der Nutzer seine Auswahl nach Belieben ändern. Die Bearbeitung eines E-Mail-Alias innerhalb des gewährten 24 Stunden Bearbeitungsfensters verläuft analog der Ersteinrichtung. Zusätzlich wird hier noch der zuvor angelegte E-Mail-Alias, sowie die verbleibende Zeit visualisiert. Erst nach Ablauf von 24 Stunden, wird der vom Nutzer ausgewählte E-Mail-Alias aktiviert und in die entsprechenden E-Mail-Systeme des SCC provisioniert.

Der Nutzer wird über die Ersteinrichtung, eventuell getätigte Änderungen innerhalb des Bearbeitungsfensters und nach Aktivierung des E-Mail-Alias per E-Mail benachrichtigt. Dabei dient die letzte Informations-E-Mail gleichzeitig als Test für die neue Adresse.

5.2.4.3 Prozessbeschreibung „Änderungsantrag“

Nach Ablauf der 24 Stunden Bearbeitungszeit hat der Benutzer keine direkte Möglichkeit mehr, seinen selbst eingerichteten E-Mail-Alias über das Mitarbeiterportal zu ändern. Hierzu liegt auf der E-Mail-Alias-Portal-Seite ein Link zu einem Online-Änderungsantrag bereit. Dieser wird nach dem Ausfüllen, durch Klicken des Buttons „Absenden“ an den ServiceDesk übersendet und dort weiter bearbeitet.

Inhalt des Antrags

1. Haupt-E-Mail-Adresse (wird vom System ausgefüllt)

Begründungen / Antragsoptionen	**Antrag genehmigen**	**E-Mail-Alias beibehalten**
Namensänderung	Ja	ist möglich
Schreibfehler	Ja	nicht möglich
Keine eindeutige Personenzuordnung	Ja	nicht möglich
Sonstige Begründung	Nein	nicht möglich

Tabelle 5.4: Zulässige Begründungen zur Genehmigung eines E-Mail-Alias-Änderungsantrags

2. Aktueller E-Mail-Alias (wird vom System ausgefüllt)
3. Begründungsfeld
4. Checkbox zum Erhalt des unter 2. genannten E-Mail-Alias

Genehmigung einer E-Mail-Alias-Änderung

Wie im weiteren Verlauf beschrieben wird, prüft der ServiceDesk des SCC die angegebene Begründung des Benutzers. Einem Änderungsantrag wird stattgegeben, wenn beim Benutzer eine Namensänderung vorliegt, der bestehende E-Mail-Alias Schreibfehler enthält oder eine eindeutig Personenzuordnung vom E-Mail-Alias zum Benutzer nicht möglich ist. Der alte E-Mail-Alias darf nur dann behalten werden, wenn sich der Name des Benutzers geändert hat (vgl. Tabelle 5.4). Nach erfolgreicher Genehmigung, bewirkt ein Änderungsantrag eine Löschung des bestehenden E-Mail-Alias, sofern der Nutzer diesen nicht behalten möchte oder darf. Weiterhin wird der Benutzer für die E-Mail-Alias Einrichtung erneut freigeschaltet, so dass er sich über das Mitarbeiterportal wieder einen neuen E-Mail-Alias anlegen kann.

Post Prozess Support

Nach Entgegennahme eines E-Mail-Alias-Änderungsantrags und Prüfung der Gültigkeit der angegeben Begründung durch den ServiceDesk, soll das Anliegen in das Ticketsystem Remedy überführt werden. Hierfür stehen zwei Ticketvorlagen zur Verfügung. Die eine Vorlage dient der Information des Benutzers, sofern dessen Antrag nicht stattgegeben werden konnte. Mit der zweiten Vorlage kann eine Änderung beauftragt werden. Hierbei wird je ein Vertreter der SCC Abteilungen DMK und ISM informiert, die beide ihre Teilaufgaben auf „gelöst“ setzen müssen, bevor das Gesamtticket in den Status „gelöst“ übergeht. Nach Abschluss der Bearbeitung des Tickets durch ISM und DMK sendet das Ticketsystem dem Benutzer eine Rückmeldung.

Post Prozess ISM

Bei Eingang eines E-Mail-Alias-Tickets in der SCC-Abteilung ISM muss zunächst überprüft werden, ob für den zu invalidierenden E-Mail-Alias ein Zertifikat ausgestellt wurde. Ist dies der Fall, muss das zugehörige Zertifikat auf die Revocation Liste für Zertifikate gesetzt werden. Diese Aktion wird jedoch nur dann durchgeführt, wenn der bestehende E-Mail-Alias nicht beibehalten werden soll/darf. Anschließend ist das Teilticket als „gelöst“ zu markieren.

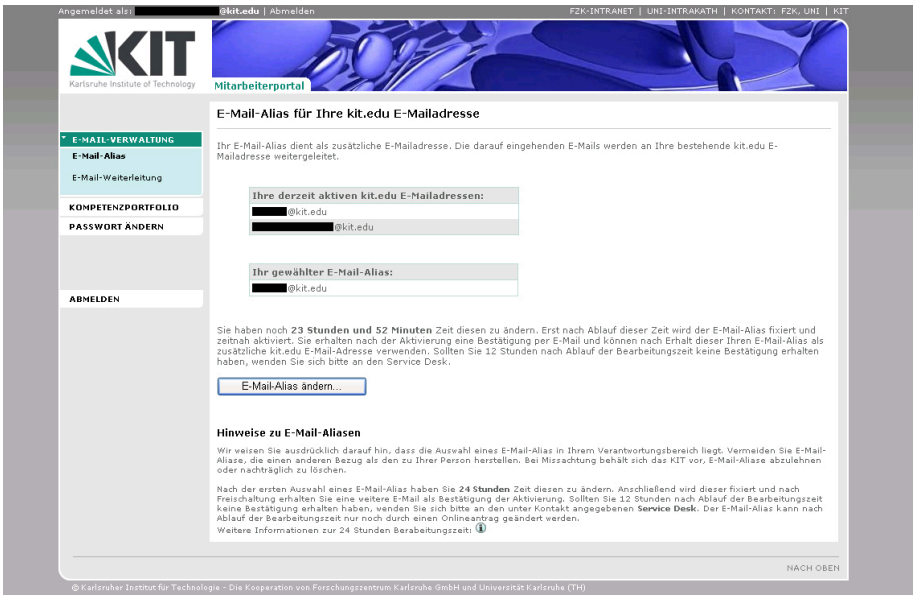


Abbildung 5.7: E-Mail-Alias-Dienst WebPart integriert in Mitarbeiterportal

Post Prozess DMK

Bei Eingang eines E-Mail-Alias-Tickets in der SCC-Abteilung DMK soll nach dem folgenden Muster verfahren werden:

Fall 1: Der Nutzer möchte/kann den alten E-Mail-Alias nicht behalten
Zunächst muss der abzugebende E-Mail-Alias auf die 15-Monate-Sperrliste (E-Mail Revocation List) gesetzt werden. Danach ist der E-Mail-Alias aus dem Exchange Konto des Benutzers zu löschen. Mit der dritten Aktion, dem Löschen des E-Mail-Alias aus der Datenbank 24-Stunden-Zwischenspeicher, wird dem Nutzer automatisch die Möglichkeit gegeben, sich einen neuen E-Mail-Alias über das Mitarbeiterportal anlegen zu können. Hierbei wird aus Datenschutzgründen die gesamten Informationen (gesamte Zeile in der Datenbankrelation) gelöscht.

Fall 2: Der Nutzer möchte und darf den alten E-Mail-Alias behalten
Der E-Mail-Alias wird lediglich aus dem 24-Stunden-Zwischenspeicher gelöscht. Hier wird ebenfalls die gesamte Zeile aus der Datenbankrelation gelöscht. Damit ist der Nutzer automatisch berechtigt, sich einen neuen E-Mail-Alias über das Mitarbeiterportal einzurichten.

Eine modellierte Ansicht der oben beschriebenen Prozesse findet sich in Anhang B.2

5.2.4.4 Technische Dokumentation

Der E-Mail-Alias-Dienst wurde als Sharepoint WebPart realisiert und in das Mitarbeiterportal *intra.kit.edu* eingebunden. Die C# .NET Entwicklung bietet dem Nutzer eine Oberfläche zur Einrichtung eines E-Mail-Alias. Die getätigten Angaben werden in einer Datenbank zwischengespeichert, die durch einen Web Service gekapselt wurde. Die Übernahme der E-Mail-Aliase in den Exchange-Server des KIT und damit die Freischaltung dieser, übernimmt ein Service Daemon. Abbildung 5.7 zeigt das E-Mail-Alias-Dienst WebPart, welches in das Mitarbeiterportal integriert wurde.

E-Mail-Alias Dienst-WebPart

Das E-Mail-Alias-Dienst WebPart wird von der Klasse `AliasWP.cs` dominiert. Diese instanziiert die nachgelagerten Web Services und implementiert den Pageflow, die Benutzerführung durch die einzelnen Webseiten des Dienstes. Weiterhin ist in dieser Klasse die Berechnung der verbleibenden Bearbeitungszeit, das Versenden von E-Mails zur Bestätigung an den Nutzer, sowie das Senden von Änderungsanträgen an den ServiceDesk des SCC integriert. Auch ein Sicherheitscheck zur Vermeidung von SQL-Injections und der Umgehung von clientseitig durch Javascript abgefangenem ungewollten Verhaltens findet sich innerhalb dieser WebPart-Klasse. Beispielhaft ist hier der reguläre Ausdruck für die Überprüfung des Teil1 eines E-Mail-Alias nach Typ 2 dargestellt:

```
(^([a-z]|[A-Z]|[0-9])([a-z]|[A-Z]|[0-9]|
(\.([a-z]|[A-Z]|[0-9]))|(-([a-z]|[A-Z]|[0-9])))*$)
```

Vorgaben, die mit diesem Regulären Ausdruck überprüft werden (vgl. 5.2.4.1):

- Mindestens ein Zeichen
- Buchstaben
- Zahlen
- Sonderzeichen „.“
- Sonderzeichen „-“
- Keine zwei Sonderzeichen aufeinander folgend
- Kein Sonderzeichen am Anfang der Eingabe
- Kein Sonderzeichen am Ende der Eingabe

Ein Bundle an diversen String-Operationen, die speziell auf die Bedürfnisse des E-Mail-Alias Dienstes angepasst wurden, findet sich in der Klasse `AliasUtilities.cs`. Hier werden spezielle Buchstaben, wie Umlaute oder Ähnliches aus dem Namen entfernt und gegen entsprechende Synonyme aus dem Grundalphabet ersetzt. Weiterhin berechnet diese Klasse Permutationen von Nachnamen und bereinigt Arraylisten von Duplikaten und null-Einträgen. Die Klasse `AliasExcWSCaller.cs` übernimmt die Kommunikation zwischen dem WebPart und den Web Service Methoden des Alias Web Services. Hier werden Daten abgerufen und für das WebPart entsprechend aufbereitet. Abbildung 5.8 zeigt den Informationsfluss innerhalb des E-Mail-Alias-Dienstes. Dabei ist ersichtlich, dass neben der Kommunikation zwischen Benutzer und WebPart der größte Teil der Interaktion im Backend des Systems abläuft. Über den Web Service des KISS-Repository, in dem diverse Nutzerdaten gespeichert sind, hat das WebPart Zugriff auf die notwendigen Daten zur Identifikation des Nutzers und dessen E-Mail-Alias-Einrichtung. Der Alias Web Service speichert in der

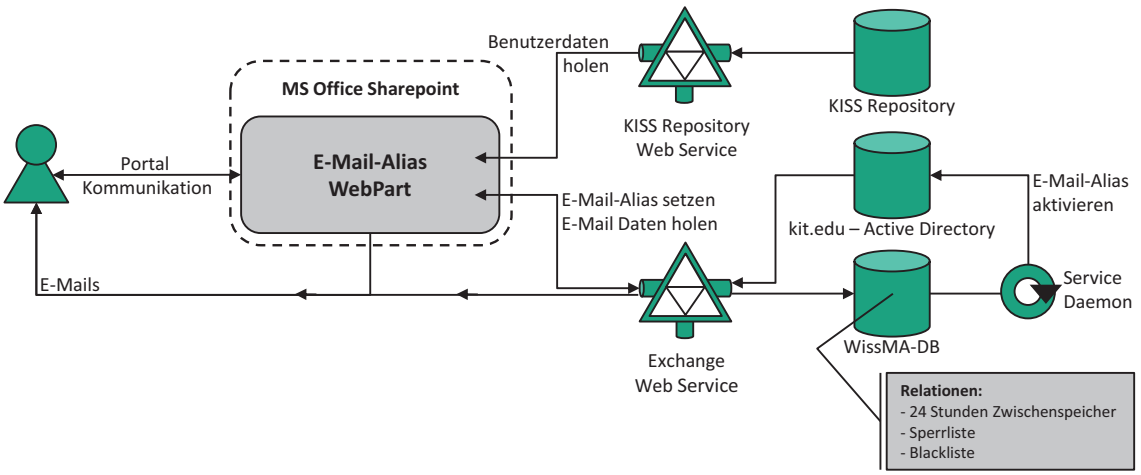


Abbildung 5.8: Informationsfluss des E-Mail-Alias-Dienstes

SQL4Alias-Datenbank des SCC SQL-Clusters alle eingerichteten E-Mail-Aliase. Außerdem sind hier die später im Datenbank-Teil dieses Abschnitts erwähnten Relationen „Blackliste“ und „Sperrliste“ hinterlegt. Ein Service Daemon überprüft im Hintergrund der Datenbank die gespeicherten Einrichtungszeitpunkte. Ist bei einem Benutzer die eingeräumte Bearbeitungszeit von 24 Stunden abgelaufen, wird der E-Mail-Alias über den im Bild nicht verzeichneten Exchange Server in das Active Directory provisioniert. Damit wird die neue Adresse gleichzeitig für den Empfang von E-Mails aktiviert.

E-Mail-Alias Support-WebPart

Das E-Mail-Alias Support-WebPart ist eine zusätzliche Entwicklung für den E-Mail-Alias-Dienst, die es dem ServiceDesk des SCC erlaubt, für jeden Kunden den aktuellen Status der E-Mail-Alias-Einrichtung einzusehen. Zusätzlich können die Mitarbeiter des Supports auch den E-Mail-Alias für einen Kunden einrichten. Zur Bedienung des E-Mail-Alias-Dienst WebParts und des Support-WebParts wurde der ServiceDesk in mehreren Veranstaltungen entsprechend geschult.

Das Support WebPart gleicht dem E-Mail-Alias-Dienst WebPart. Der Unterschied der beiden Entwicklungen liegt darin, dass dem Support-WebPart eine Eingabeaufforderung vorangestellt wurde, die es erlaubt das E-Mail-Alias Datenblatt eines bestimmten Kunden aufzurufen. Hierzu wird der UPN, bzw. die Haupt-E-Mail-Adresse des Kunden eingegeben und anschließend auf einen Button „Support starten...“ geklickt. Der sich anschließende Programmablauf ist analog dem E-Mail-Alias-Dienst WebPart, dabei wird jedoch nicht der eigene E-Mail-Alias bearbeitet, sondern der des zuvor spezifizierten Kunden.

Service Daemon

Der Service Daemon ist eine Entwicklung der SCC-Abteilung SYS und wird von der Abteilung ISM betreut und gewartet. Der Service Daemon übernimmt reservierte E-Mail-Aliase nach Ablauf der Bearbeitungszeit über den SCC-Exchange-Server in das

Active Directory. Dazu liest der Daemon alle 10 Minuten die Einrichtungszeitpunkte der im 24 Stunden Zwischenspeicher der SQL4Alias-Datenbank hinterlegten E-Mail-Aliase aus.

Fehlerdokumentation

Nachfolgend werden die möglichen Fehlermeldungen des Dienstes annotiert und Lösungsvorschläge gegeben. Die Fehlermeldungen sieht der Benutzer im Mitarbeiterportal und kann sie dementsprechend an den ServiceDesk weitergeben.

Fehlermeldung: Es ist ein Fehler aufgetreten (Code:0):
Fehlerquelle: WebPart fehlerhaft
2nd Level Support: T. Z****
Fehlerbeschreibung: Das E-Mail-Alias WebPart reagiert fehlerhaft
3rd Level Support: S. L*******

Fehlermeldung: Es ist ein Fehler aufgetreten (404-0):
Fehlerquelle: Alias Web Service (Host: scc-kim-06.scc.kit.edu)
2nd Level Support: F. S*****, S. L*******
Fehlerbeschreibung: Web Service nicht erreichbar (Host: scc-kim-06.scc.kit.edu)
Mögliche Gründe: Server down, Web Service nicht gestartet, ...

Fehlermeldung: Es ist ein Fehler aufgetreten (404-1):
Fehlerquelle: Alias Web Service (Host: scc-kim-06.scc.kit.edu)
2nd Level Support: F. S*****, S. L*******
Fehlerbeschreibung: Fehler in Web Service - Methode isAvailable(alias)
Mögliche Gründe: AD, 24h-Zwischenspeicher, Blacklist oder Sperrliste nicht verfügbar

Fehlermeldung: Es ist ein Fehler aufgetreten (404-2):
Fehlerquelle: Alias Web Service (Host: scc-kim-06.scc.kit.edu)
2nd Level Support: F. S*****, S. L*******
Fehlerbeschreibung: Fehler in Web Service - Methode getAlias(upn)
Mögliche Gründe: 24h-Zwischenspeicher nicht verfügbar

Fehlermeldung: Es ist ein Fehler aufgetreten (404-3):
Fehlerquelle: Alias Web Service (Host: scc-kim-06.scc.kit.edu) oder Active Directory
2nd Level Support: SCC-Abteilung DMK und F. S*****, S. L*******
Fehlerbeschreibung: Fehler in Web Service - Methode getMailAddresses(upn)
Mögliche Gründe: AD/Exchange nicht verfügbar

Fehlermeldung: Es ist ein Fehler aufgetreten (404-4):
Fehlerquelle: Alias Web Service (Host: scc-kim-06.scc.kit.edu)
2nd Level Support: F. S*****, S. L*******
Fehlerbeschreibung: Fehler in Web Service - Methode setAlias(upn, alias)
Mögliche Gründe: 24h-Zwischenspeicher nicht verfügbar

Fehlermeldung: Es ist ein Fehler aufgetreten (404-5):
Fehlerquelle: KISS-Repository Web Service (Host: scc-kim-06.scc.kit.edu)

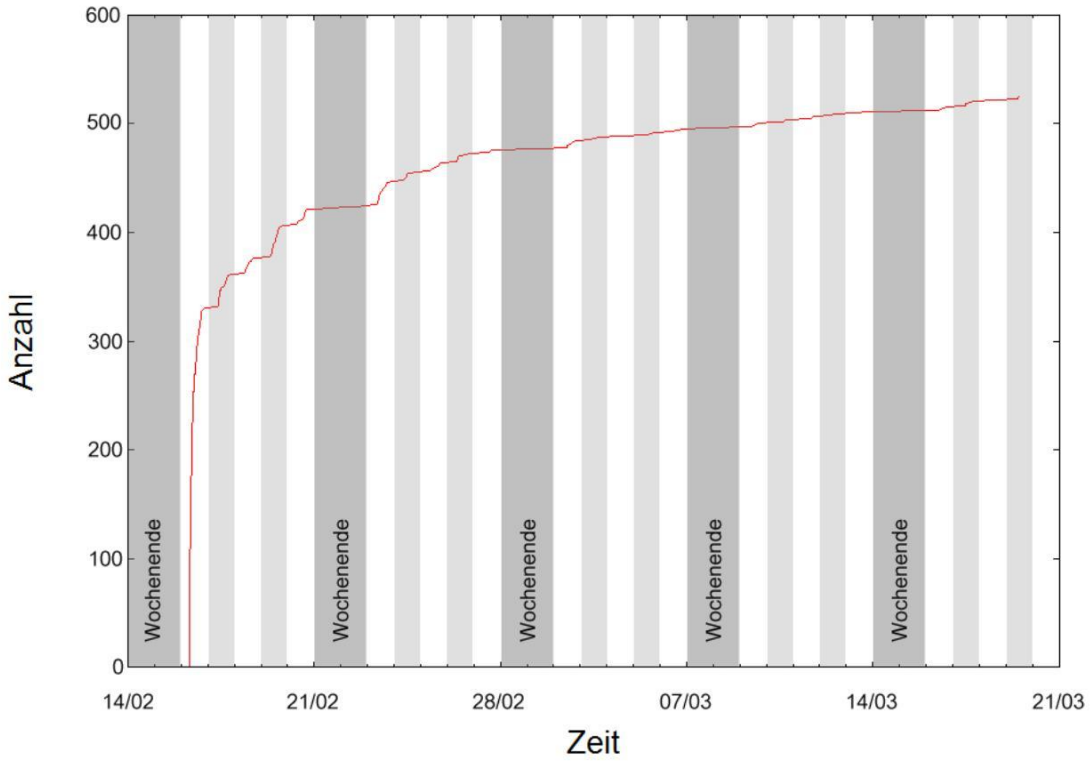


Abbildung 5.9: Anzahl eingerichteter E-Mail-Aliase

2nd Level Support: F. S*****
Fehlerbeschreibung: KISS-Repository Service nicht verfügbar
Mögliche Gründe: Web Service nicht gestartet

Fehlermeldung: Es ist ein Fehler aufgetreten (404-6):
Fehlerquelle: WebPart fehlerhaft - Änderungsantrags-E-Mail konnte nicht versendet werden
2nd Level Support: S. L*******
Fehlerbeschreibung: Das E-Mail-Alias WebPart reagiert fehlerhaft
Mögliche Gründe: SMTP-Server down

Fehlermeldung: Es ist ein Fehler aufgetreten (404-7):
Fehlerquelle: WebPart fehlerhaft - Reservierungs-E-Mail konnte nicht versendet werden
2nd Level Support: S. L*******
Fehlerbeschreibung: Das E-Mail-Alias WebPart reagiert fehlerhaft
Mögliche Gründe: SMTP-Server down

5.2.4.5 Status

Die Entwicklung des KIT E-Mail-Alias-Dienst ist abgeschlossen und produktiv installiert. Die endgültige Freischaltung erfolgte am 16. Februar 2009. In einigen Fällen war zunächst eine Portfolio-Erweiterung einzelner Abteilungen des SCC notwendig. Das vom SCC eingerichtete Stakeholder-Forum nahm sich dieser Aufgabe an und schloss die organisatorische Verankerung des Dienstes zum 04. Dezember 2008 ab.

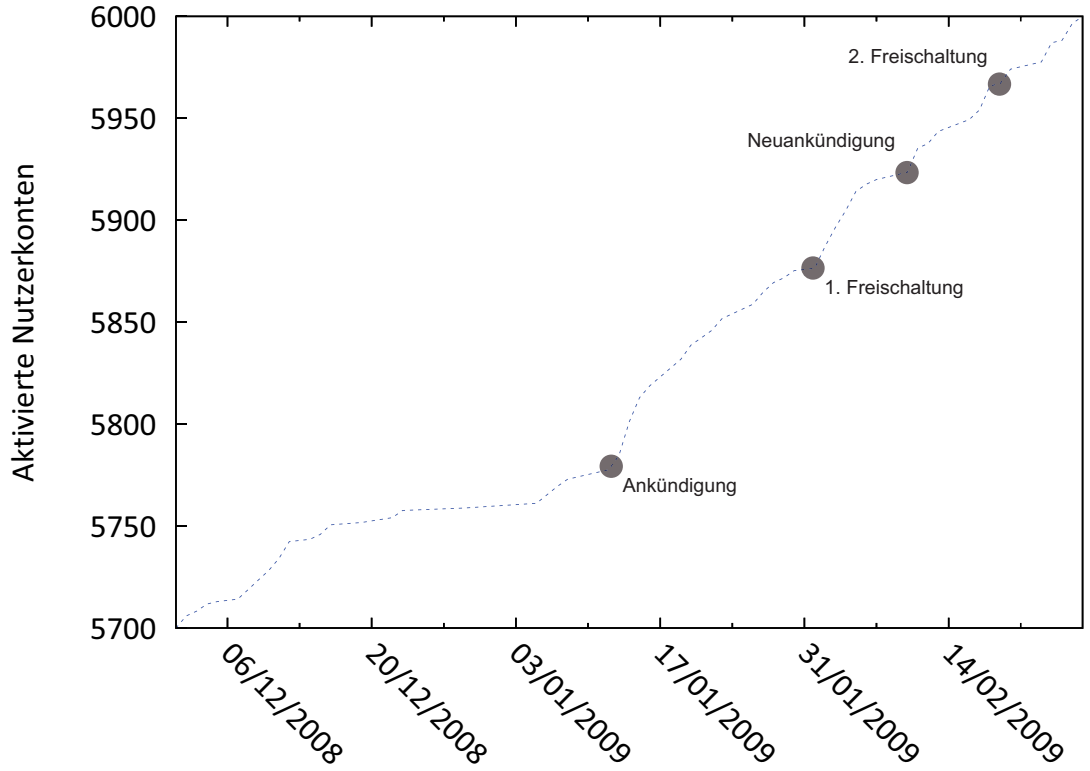


Abbildung 5.10: Anzahl der aktivierten Mitarbeiter im Mitarbeiterportal im Rahmen der Ankündigungs- und Anlaufphase des E-Mail-Alias-Dienstes

Die Ankündigung des E-Mail-Alias-Dienstes wurde per Hauspost und E-Mail an alle KIT-Mitarbeiter versendet. Die Anzahl der Einrichtungen werden in Abbildung 5.9 visualisiert. Abbildung 5.10 zeigt ferner die Anzahl der für das Mitarbeiterportal aktivierten Mitarbeiter im Zeitraum der Ankündigung und des Dienstanlaufs (vgl. auch Abbildung 1.2).

5.2.5 Passwortänderungsdienst

Abbildung 5.11 gibt eine Übersicht des Passwortänderungs-Prozesses. Es ist hierbei allen Mitarbeitern des KIT möglich das Passwort für den KIT-Benutzerkonto anzupassen. Dieses Benutzerkonto ist nicht das E-Mail-Benutzerkonto des SCC sondern ein eigenes KIT-Benutzerkonto, welches bspw. zur Anmeldung am Mitarbeiterportal oder im Falle eines Dozenten auch am Studierendenportal verwendet wird.

Über die Auswahl „Passwort ändern“ im Mitarbeiterportal gelangt der Benutzer zur Passwortänderungs-Seite. Über einen Web Service wird nun geprüft, ob der Mitarbeiter seinen Account bereits aktiviert hat. Falls dies nicht der Fall ist, bekommt der Mitarbeiter einen Hinweis auf den Aktivierungsprozess, siehe Abschnitt 5.2.2. Die erfolgreiche Durchführung der Aktivierung ist zwingend notwendig um das Passwort zu ändern. Falls der Benutzer die Aktivierung erfolgreich durchgeführt hat, kann er durch Eingeben seines bisherigen Passwortes und dem zweimaligen Eingeben des neuen Passwortes die Änderung durchführen. Es wird hierbei Client-seitig geprüft, ob die Passwort-Richtlinien erfüllt werden und dem Benutzer durch das Anzeigen grüner „Smilies“ verdeutlicht. Erst wenn alle Anforderungen der Passwortrichtlinie erfüllt sind, kann der Benutzer den „Weiter“ Button klicken. Nachdem eine Überprüfung des bisherigen Passwortes gegen das KIT.AD erfolgreich war, wird das Passwort über einen Web Service (siehe Abschnitt 6.1.8) geändert, welcher über das Identitätsmanagementsystem zur Verfügung gestellt wird. Dies ermöglicht die Nutzung der Logging Funktionalität des Sun Java System Identity Managers. Es ist demnach auch zu einem späteren Zeitpunkt noch möglich die Aktivierung durch einen Audit-Log-Report nachzuvollziehen. Das Ergebnis der Änderung wird dem Benutzer wiederum über eine Bestätigungsseite angezeigt.

Die Richtlinien für das Passwort lauten folgendermaßen:

- Mindestens 8, maximal 16 Zeichen lang
- Enthält mindestens 1 Buchstaben und mindestens 1 Zahl
- Enthält zwei verschiedene Sonderzeichen:

  `| ! & ’ ( ) * + , - . / : ; < = > ? [ \ ] ^ _ ‘ { } ~`

- Enthält keines dieser Zeichen: `@ \# % " $`
- Enthält keine alphanumerische Zeichenfolge in Klammern: `() [] {} <>`
- Enthält keine Leerzeichen bzw. Whitespaces am Anfang oder am Ende
- Enthält keine Umlaute
- Enthält weder Ihren Vor- noch Nachnamen
- Passwort und seine Wiederholung müssen gleich sein

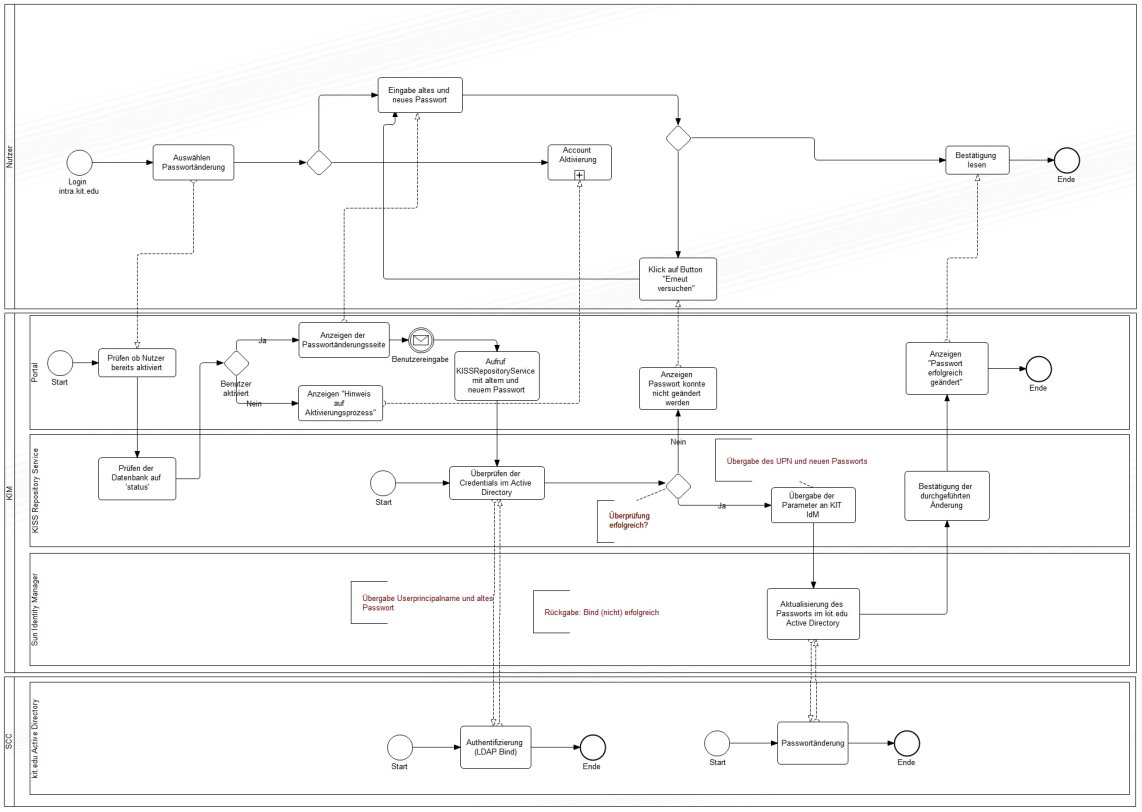


Abbildung 5.11: Änderung des Passwortes für KIT-Mitarbeiter

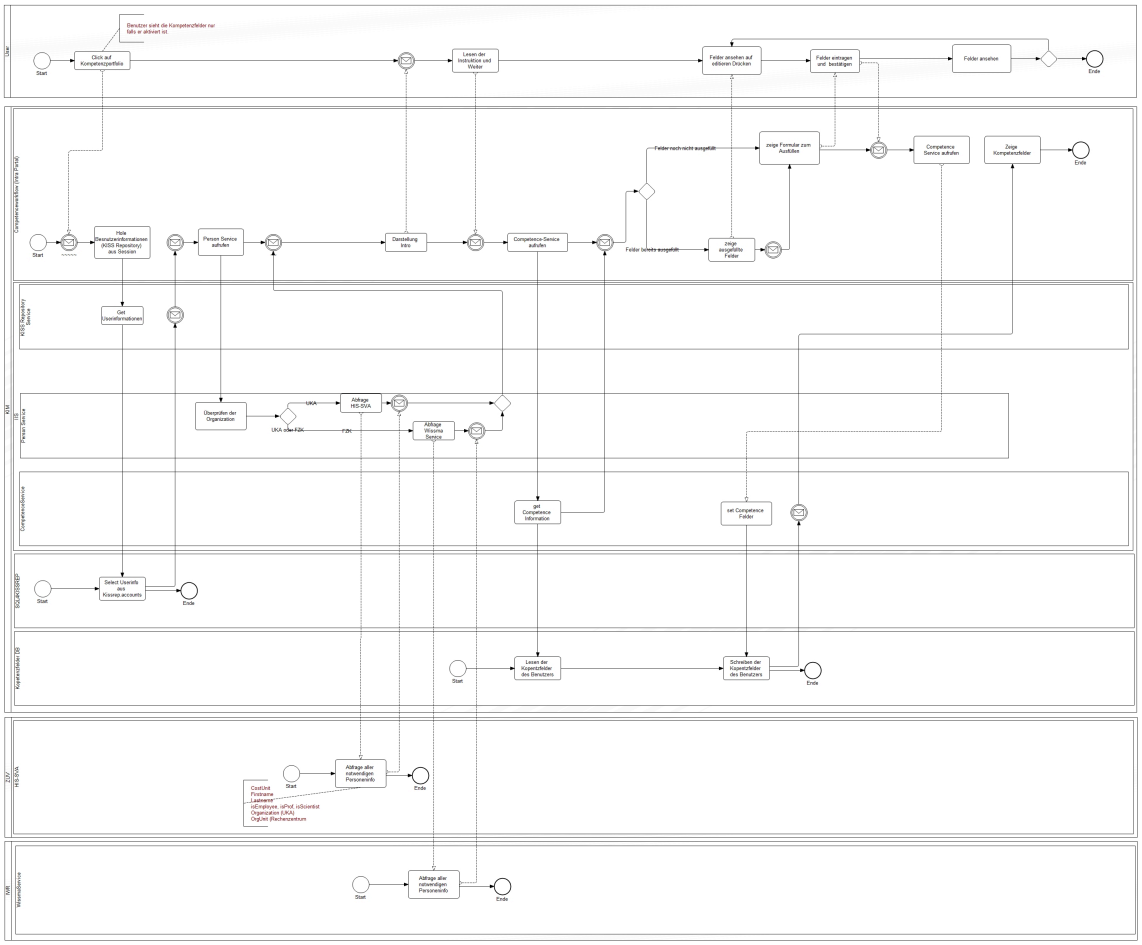


Abbildung 5.12: Prozessdiagramm des Kompetenzfeldzuordnungsdiensts

5.2.6 Kompetenzfeldzuordnungsdienst

Der Kompetenzfelderzuordnungsdienst dient Mitarbeitern des KIT sich entsprechenden Kompetenzfeldern zuzuordnen. Das Portfolio der Kompetenzfelder des KIT, welches sich dynamisch weiterentwickeln kann und soll, umfasst aktuell 30 Kompetenzfelder, die in 6 Kompetenzbereichen gebündelt sind. Durch die Implementierung des Kompetenzfelderzuordnungsdienstes hat das Projekt KIM-IDM die Grundlage dafür geliefert, dass sich Mitarbeiter den unterschiedlichen Kompetenzfeldern zuordnen können.

Nähere Informationen zu der Bedeutung von Kompetenzfeldern und deren Rolle innerhalb des KIT finden sich im Mitarbeiterportal *intra.kit.edu*.

Abbildung 5.12 gibt einen Überblick des Kompetenzfelderzuordnungs-Prozesses und der beteiligten Systeme. Hierbei kommt ein Mitarbeiter über das Mitarbeiterportal durch den Navigationspunkt „Kompetenzportfolio“ auf die Seite der Kompetenzfelder. Da es nur wissenschaftlichen Mitarbeitern erlaubt ist Kompetenzfelder zu wählen, wird dies zunächst durch unterschiedliche Systeme überprüft. Hierbei ist für die Mitarbeiter des Campus Süd das System HIS SVA die entscheidende Datenquelle und für die Mitarbeiter des Campus Nord der „Wissma Service“. Falls der Mitarbeiter in diesen Systemen als Wissenschaftler geführt wird gelangt er auf die

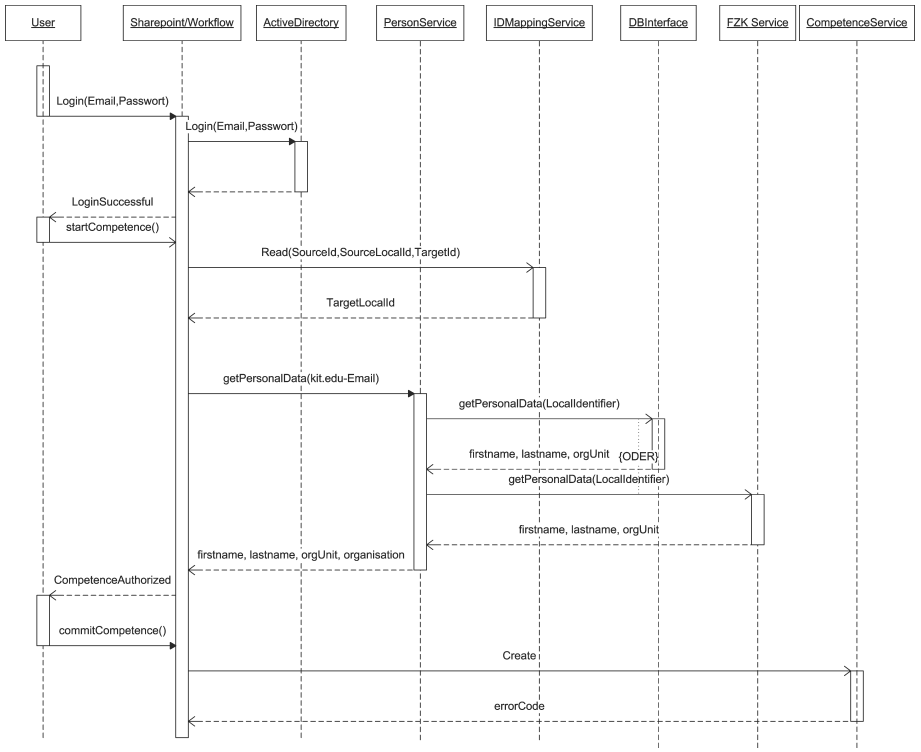


Abbildung 5.13: Sequenzdiagramm des Kompetenzfeldzuordnungsdiensts

erste Seite der Kompetenzfelder, eine Seite mit Instruktionen und Erklärungen zu den Kompetenzfeldern. Nach der Lektüre dieses Textes gelangt der Benutzer durch das Klicken auf den „Weiter“ Button auf die Seite mit den Kompetenzfeldern. Sollte der Benutzer den Kompetenzfelder-Prozess bereits einmal erfolgreich durchgeführt haben, bekommt er zunächst seine letzte Auswahl angezeigt. Auf der nächsten Seite kann er dann Änderungen durchführen und das Formular abschicken. Die Auswahl wird anschließend gespeichert und dem Benutzer bei einem erneuten Besuch zur Ansicht und Änderung dargestellt.

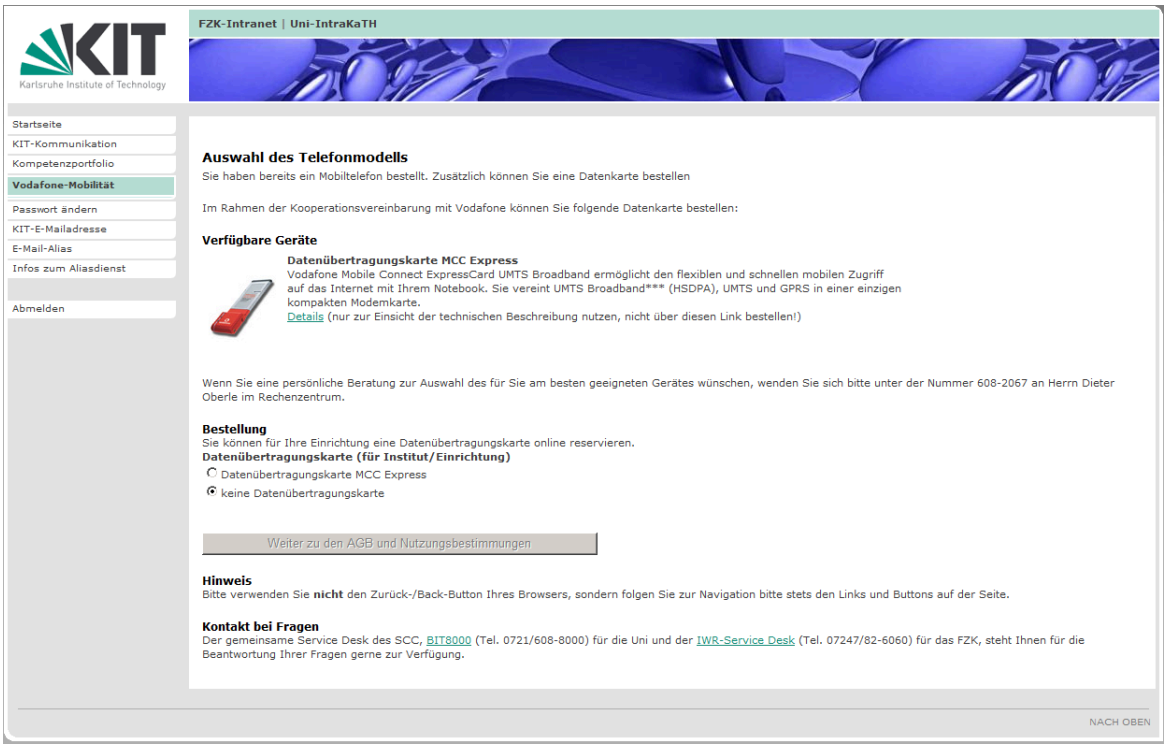


Abbildung 5.14: Modellauswahl beim Vodafone-Beantragungsdienst

5.2.7 Vodafone-Beantragungsdienst

Im Mitarbeiterportal wurde im Rahmen der Kooperationsvereinbarung zwischen Vodafone und der Universität Karlsruhe (TH) unter Einbezug des FZK eine Anwendung entwickelt, die es Universitätsprofessoren (Besoldungsgruppe C4, W3, C3, W2) ermöglicht ein Gerät mit freigeschalteter SIM Karte für den dienstlichen Gebrauch zu beantragen. Ferner besteht auch die Möglichkeit für private Zwecke von einem Vodafone Angebot Gebrauch zu machen.

Abbildung 5.14 zeigt einen Screenshot des Mitarbeiterportals. Die Seite zeigt die Auswahlmöglichkeit eines Gerätes. Nachdem ein Gerät ausgewählt wurde, muss der Antragsteller noch den Nutzungsbedingungen zustimmen (siehe Abbildung 5.15).

Das in Abbildung 5.16 gezeigte Sequenzdiagramm veranschaulicht den genauen Ablauf und die am Prozess beteiligten Komponenten.

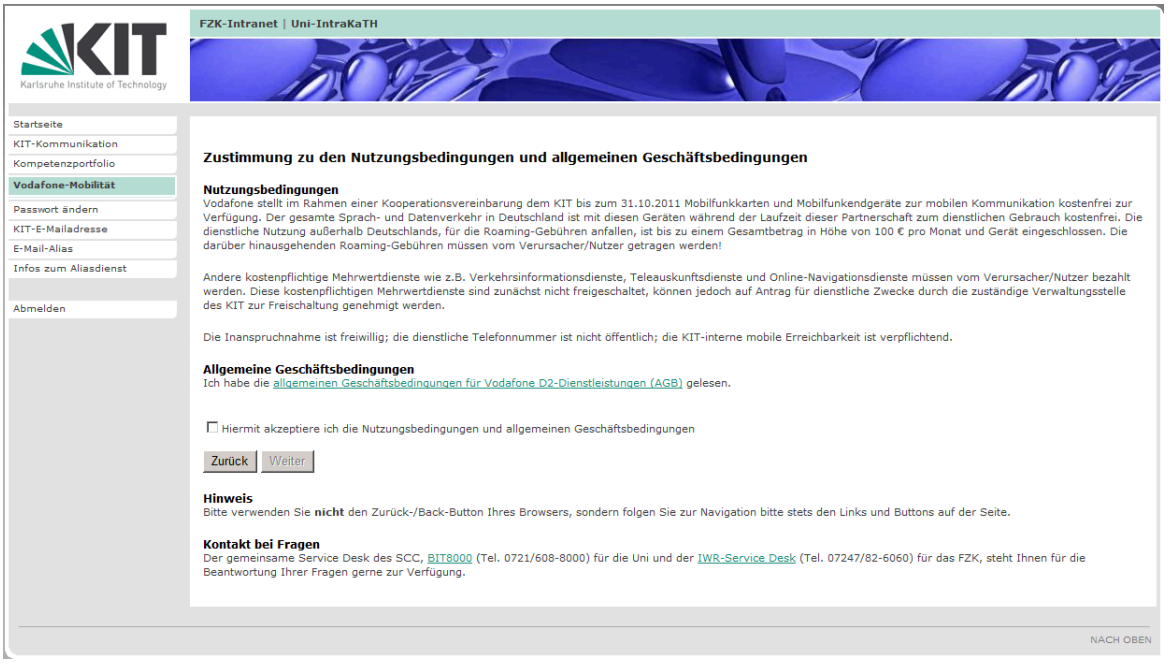


Abbildung 5.15: Darstellung der Nutzungsbedingungen des Vodafone-Beantragungsdienstes

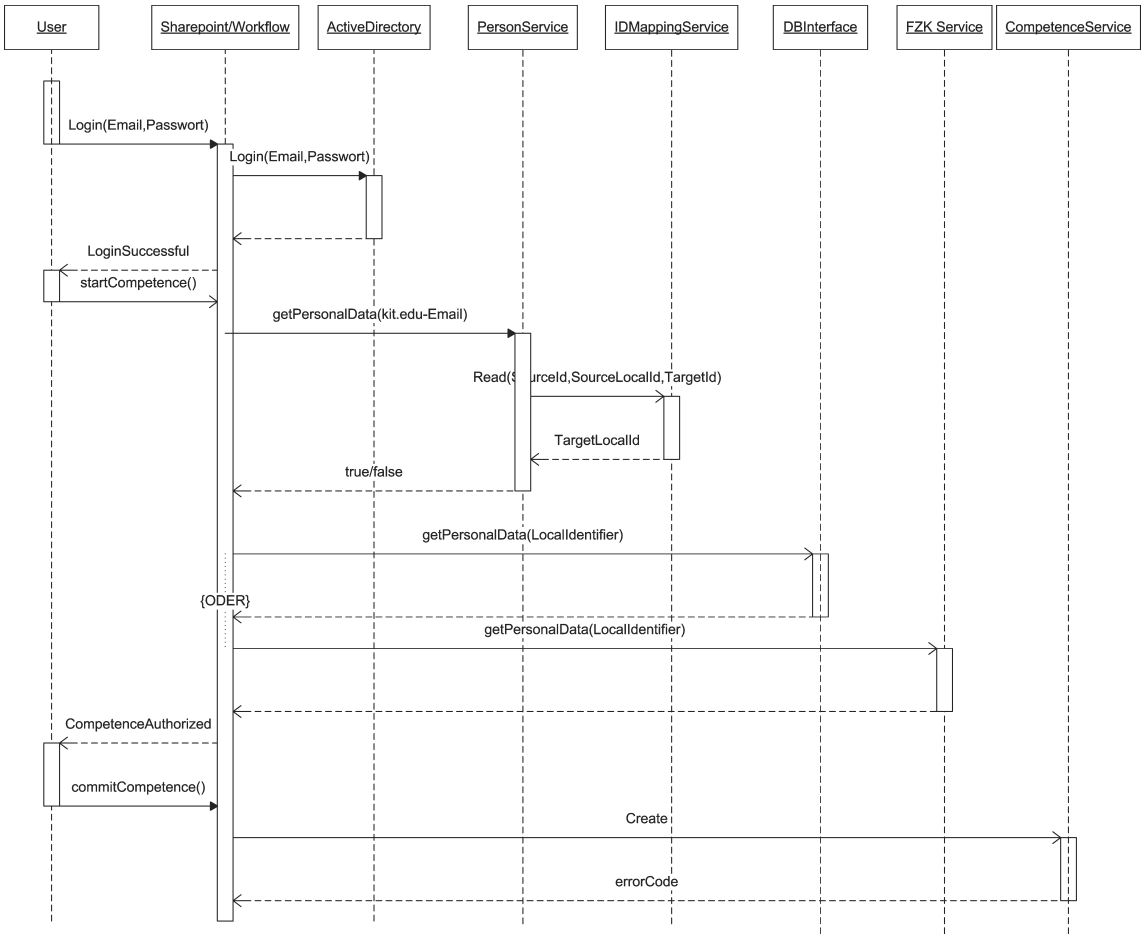


Abbildung 5.16: Sequenzdiagramm des Vodafone-Beantragungsdiensts

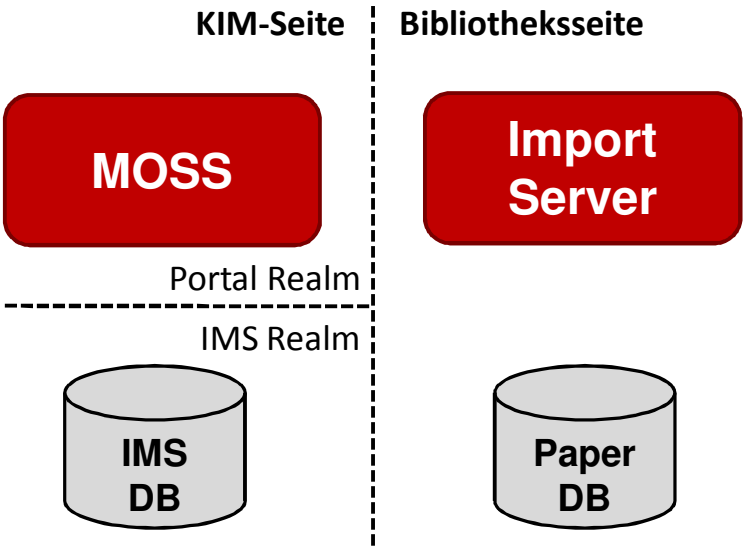


Abbildung 5.17: Überblick über die Systeme des Paper-Veröffentlichungsdienstes

5.2.8 Paper-Veröffentlichungsdienst

Der Paper-Veröffentlichungsdienst (Paper Publisher) ist eine Entwicklung zum transferieren wissenschaftlicher Arbeiten zur dafür vorgesehenen Datenbank der Universitätsbibliothek Karlsruhe. Abbildung 5.17 zeigt einen Überblick der Systemlandschaft mit den beteiligten Komponenten. Dabei teilt sich die KIM-Seite in den Mitarbeiterportal-Bereich und den Identifier Mapping Service (vgl. auch Abschnitt 6.1.2) auf. Die verzeichnete Datenbank ist SQL-basiert und soll die Benutzeridentifikatoren des SCC und der Bibliothek verknüpfen. Auf Bibliotheksseite ist der Import-Server skizziert. Dieser ist die Zieldatenbank für wissenschaftliche Arbeiten nachgelagert. Da der letzte Schritt einer Transaktion des Paper Publishers, das Importieren der hochgeladenen Papiere, von Seiten der Universitätsbibliothek implementiert wird, bleibt die Art der Paper Datenbank hier unerheblich.

In diesem Dokumentationsteil werden die wesentlichen Entwurfsentscheidungen der Paper Publisher Entwicklung skizziert. Während die ersten Abschnitte einen Überblick über die Anforderungen und die konzipierte Gesamtarchitektur des Dienstes geben, wird in den darauf folgenden Teilen näher auf die verschiedenen Komponenten eingegangen. Der Pageflow-Abschnitt soll die dem Benutzer offensichtliche Schnittstelle zeigen und den entwurfsspezifischen Hintergrund erklären. Weiterhin werden alle entwickelten Web Services vorgestellt und in Zusammenhang mit der Gesamtarchitektur gebracht. Die folgenden Abschnitte befassen sich mit der Implementierung des Anwendungsfalls. Der Teil schließt mit einer Darstellung interessanter Implementierungsdetails, einer Übersicht über die beteiligten Klassen und einem Statusbericht.

Der Paper-Veröffentlichungsdienst befindet sich bislang noch in der Testphase. Für eine Überführung in den produktiven Betrieb müssen insbesondere Dienste auf Seiten der Universitätsbibliothek angepasst bzw. entwickelt werden. Der Dienst steht insofern noch nicht über das Mitarbeiterportal zur Verfügung.

5.2.8.1 Anforderungsanalyse

Es war zunächst notwendig, dass für das Portal ein Pageflow entwickelt wurde, der den Anwender durch die notwendigen Schritte leitet, um dem System benötigte Daten einzugeben, notwendige Dateien hochzuladen und den Austauschprozess zwischen Mitarbeiterportal und der Universitätsbibliothek anzustoßen. Der Pageflow sollte die Variante ein einzelnes wissenschaftliches Papier zur Bibliothek zu übertragen bieten und weiterhin eine Upload-Funktion für Arbeiten, die mit Hilfe einer BibTEX-Datei katalogisiert und deren Volltexte optional in ein ZIP-Archiv zusammengefasst sind.

Neben den primären Aufgaben, müssen während des Prozesses weitere sekundäre Systeme etabliert und integriert werden. Zwei Organisationen nutzen im Allgemeinen voneinander verschiedene Identifikatoren, um Benutzer im System zu lokalisieren, beziehungsweise zu authentifizieren. Es musste eine Instanz geschaffen werden, die Identifikatoren der Benutzer des einen Systems mit denen des anderen verknüpft. Der Service, der diese Aufgabe erfüllt, wird hier als Identifier Mapping Service (IMS) bezeichnet (vgl. auch Abschnitt 6.1.2).

Bei einem erstmaligen Benutzen des Systems muss dem IMS zum bekannten KIM-Identifikator der benutzereigene Bibliotheks-Identifikator übermittelt werden. Hierzu muss mittels Bibliotheksbenutzername und Bibliothekspasswort zunächst überprüft werden, ob der Anwender im Bibliothekssystem vorhanden ist und authentifiziert werden kann. Aus Sicherheitsgründen, darf das eingegebene Passwort nach erfolgreicher Überprüfung nicht weiter gespeichert werden. Die Bibliothekskennung ist in den IMS zu übernehmen. Sollte die Authentifikation fehlschlagen, wird dem Benutzer eine Rückmeldung über die Fehlerursache gegeben und eine neue Aufforderung zur Dateneingabe präsentiert.

Für den Anwendungsfall „Massenupload“, dem Upload von mehreren wissenschaftlichen Arbeiten in einem Arbeitsgang, ist es notwendig, dass die entgegennehmenden Systeme auf Bibliotheksseite mit sehr großen Dateien umgehen können. Für den Uploadvorgang großer Datenmengen soll der Benutzer nicht zwingend angemeldet bleiben, sondern nach Beendigung der Transaktion über Erfolg oder Misserfolg seiner Transaktion informiert werden.

5.2.8.2 Architektur

Wie aus Abbildung 5.18 deutlich wird, gliedert sich das System in drei große Teilbereiche, die verschiedenen Realms. Der Bereich, der mit *KIM-Realm* umschrieben ist, spiegelt die KIM-Seite und damit das Portal, die Benutzerdatenbank und weitere Komponenten innerhalb dieser Domäne wieder. Der Bereich *Bibliothek* beschreibt die wesentlichen Komponenten des Realms Universitätsbibliothek, demnach alle Komponenten die innerhalb der IT-Systeme der Bibliothek gehostet sind. Als dritter Realm ist der *IMS* aufgeführt. Dieser wurde als zusätzliche logische Domäne konstruiert. Gehostet ist sie innerhalb des KIM-Systems, wird jedoch in diesem und folgenden Abschnitten als eigenständiger Realm betrachtet.

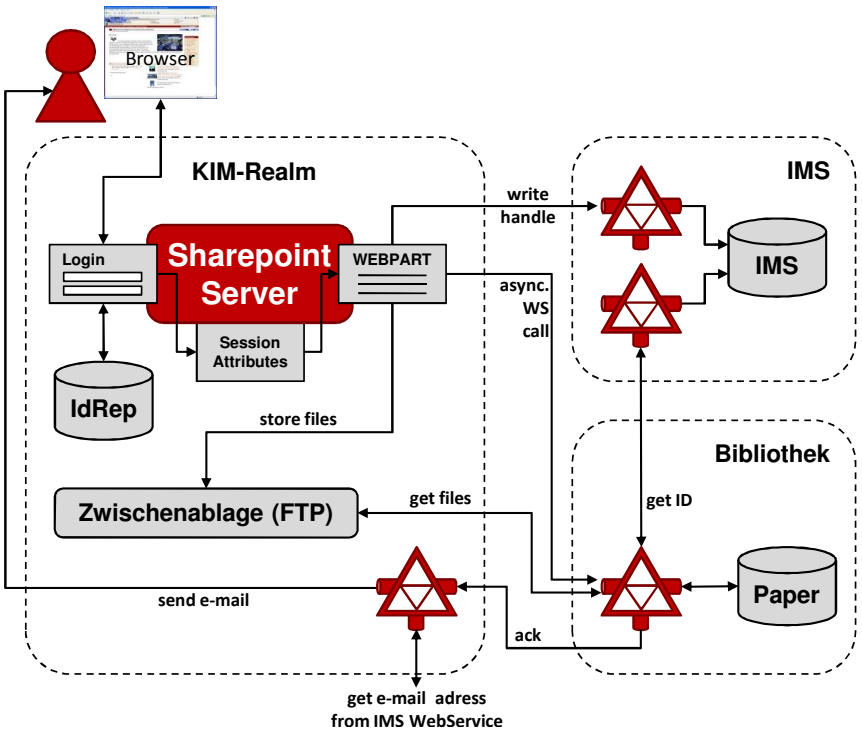


Abbildung 5.18: Architektur des Paper-Veröffentlichungsdienstes

KIM-Realm

Auf der Portal-Seite sind in Abbildung 5.18 mehrere Systeme verzeichnet. An zentrale Stelle tritt hier der MOSS, der das Mitarbeiterportal über eine Webschnittstelle anbietet und in den die Funktionalität in Form eines Sharepoint WebParts integriert wurde. An dieser Stelle sei vermerkt, dass das WebPart Zugriff auf Web Services der Realms *IMS* und *Bibliothek* hat und damit eine Randkomponente des *KIM-Realms* darstellt.

Innerhalb des *KIM-Realms* befinden sich zwei weitere wesentliche Komponenten. Ein FTP-Server dient hier als Zwischenablage der vom Benutzer hochgeladenen Dateien. Auf diese kann nun ein Dienst der Bibliothek zugreifen, um die den Eintragungen des Anwenders entsprechenden Dateien herunter zu laden. Durch den Einsatz eines FTP-Servers, können die Bibliothekssysteme bei sehr großer Netzlast das Herunterladen der Dateien vom FTP-Server auf eine Zeit mit minderer Netzlast verschieben und haben somit die Möglichkeit des Schedulings[1]. Der FTP-Server dient somit zur Datenablage für den Zeitraum des Hochladens der Dateien über den Sharepoint Server durch den Anwender, bis zum erfolgreichen Download durch die Bibliotheksapplikation.

Der am unteren Bildrand dargestellte Web Service wird seinerseits von Bibliotheksseite aufgerufen und ist eingerichtet, um dem Benutzer eine Rückmeldung in Form einer E-Mail zukommen zu lassen. Diese soll ihn über Erfolg oder Misserfolg seiner angestoßenen Transaktion informieren.

[1]Scheduling: (dt. Planung) Als Scheduling wird die Erstellung eines Ablaufplanes (Schedule) bezeichnet.

IMS

Im *IMS-Realm* ist im Wesentlichen eine Datenbank enthalten, an die mittels vorgelagerten Web Services Anfragen getätigt werden können. Die Web Services setzen die an sie gelieferten Übergabeparameter in SQL-Anweisungen um und geben die Ergebnisse der Datenbankzugriffe an den aufrufenden Dienst zurück.

Der *KIM-Realm* hat dabei Zugriff auf eine Web Methode, die überprüft, ob bereits ein Mapping zum aktuell aktiven Benutzer existiert. Eine weitere Methode verknüpft die Identifikatoren mit einem sogenannten Handle, einem organisationsübergreifend eindeutigen Identifikator, der auf die organisationsspezifischen Identifikatoren verweist. Eine dritte Methode gibt zu einem gegebenem Handle den KIM Benutzer-Identifikator zurück.

Dem Realm *Bibliothek* steht dabei lediglich ein Web Service zur Verfügung, der zu einem übergebenen Handle den Bibliotheks-Identifikator ausliest und an den Aufrufenden Dienst der Bibliothek zurück liefert und umgekehrt.

Bibliothek

Innerhalb der Bibliotheksdomäne wird ein Web Service etabliert, der den Transaktionsauftrag aus dem KIM-Portal entgegen nimmt und eine so genannte Queue-Datei (Warteschlangen-Datei) der Import Applikation mit diesen Daten nährt. Die Anwendung soll ihrerseits anhand der übermittelten Daten die entsprechenden Dateien vom KIM-FTP-Server herunterladen. Nur diese Applikation ist dann berechtigt, die bereits existierende Import-Schnittstelle der Datenbank für wissenschaftliche Publikationen der Bibliothek zu benutzen, um die Arbeiten in das Datenbank-System der Bibliothek zu importieren.

5.2.8.3 Paper Publishing WebPart

Das Paper Publishing-WebPart beinhaltet einen Großteil der Funktionalität und ist in C#.NET implementiert. Neben dem Pageflow, der im nächsten Abschnitt näher erläutert wird, werden hieraus die zuvor angesprochenen Web Services mit den jeweils nötigen Sitzungs- und Eingabedaten des Nutzers aufgerufen. Das WebPart ist somit verantwortlich für die Orchestrierung, dem Zusammenspiel der Web Services, und bildet den Zugangspunkt für den Benutzer, sowie die Schnittstelle zur serviceorientierten Architektur im Backend.

Pageflow

In Abbildung 5.19 ist der Ablauf der Benutzerführung skizziert. Im initialen Bereich hat der Anwender die Möglichkeit, zwischen einem Massenupload oder einem Einzelupload von wissenschaftlichen Papieren zu wählen. Die Entscheidung wird gespeichert und der Identifier Mapping Service befragt, ob ein Mapping zum aktuell angemeldeten Benutzer existiert. Ist dies nicht der Fall, wird eine Seite eingeblendet, in die Benutzername und Passwort der Bibliothek einzugeben sind. Dabei sei angemerkt, dass das Passwort lediglich zur initialen Authentifikation dient und innerhalb

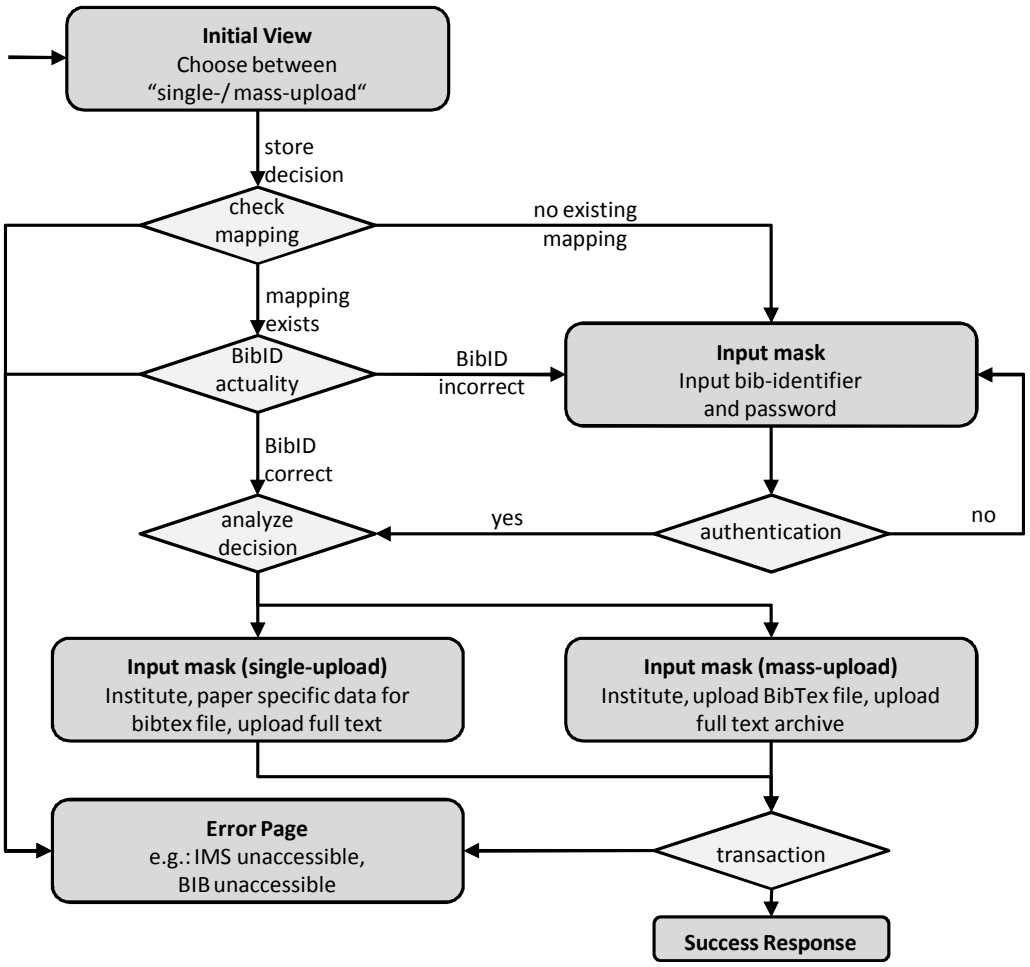


Abbildung 5.19: Pageflow des Paper Publisher mit wesentlichen Entscheidungspunkten

des KIM-Systems nicht gespeichert wird. Ist bereits ein Mapping vorhanden, wird überprüft, ob der im IMS vorhandene Bibliotheks-Identifikator noch aktuell ist. Sollte dies nicht der Fall sein, wird auch hier der Benutzer aufgefordert seine Kenndaten einzugeben. Dies könnte z.B. passieren, wenn ein Bibliotheksausweis verloren gegangen ist und der Benutzer im Zuge der Ausstellung eines neuen Ausweises einen neuen Identifikator zugewiesen bekam. Hat der Benutzer die Kenndaten eingegeben, werden diese durch einen Authentifikationsdienst der Bibliothek validiert. Ist Benutzername oder Passwort falsch eingegeben worden, wird eine entsprechende Fehlermeldung generiert und zusammen mit einer erneuten Eingabeaufforderung angezeigt.

Sind die Bibliotheksdaten korrekt eingegeben oder war bereits ein Mapping vorhanden, wird die anfängliche Entscheidung über Einzel- oder Massenupload ausgewertet und eine entsprechende Eingabemaske präsentiert. Im Bereich Einzelupload hat der Anwender die Möglichkeit, die Metadaten des Papiers direkt einzugeben und optional einen Volltext hochzuladen. Im Bereich Massenupload ist es erforderlich, dass bereits eine BibTEX-Datei existiert, welche die wissenschaftlichen Papiere katalogisiert. Weiterhin müssen die Volltexte als ZIP-Archiv zu einer Datei zusammengefasst sein, sofern diese mit übertragen werden sollen. Unter Angabe des Institutsnamen und der Quellen der zwei Dateien, kann eine Transaktion zur Universitäts-Bibliothek

gestartet werden. Tritt während der Verarbeitung ein Fehler auf, bspw. die Nicht-Erreichbarkeit eines Services, erhält der Benutzer eine entsprechende Fehlerseite mit detaillierten Angaben zum aufgetretenen Problem angezeigt. Bei erfolgreich versendetem Transaktionsauftrag an den Bibliotheksdienst, erhält der Benutzer eine Bestätigung über den Eingang seines Auftrags.

Im Folgenden wird die bibliotheksseitige Anwendung die Dateien vom KIM-FTP-Server herunterladen und nach erfolgreicher Autorisation des Benutzers für den Vorgang die Daten importieren. Im Anschluss erhält der Benutzer über den KIM-seitig eingesetzten Response Web Service eine E-Mail über Erfolg oder Misserfolg der Transaktion.

Initiale Authentifikationsüberprüfung

Wie im vorigen Abschnitt bereits angesprochen, muss bei erstmaliger Eingabe der Benutzerdaten der Universitätsbibliothek die Authentizität des Anwenders durch das Bibliothekssystem überprüft werden. Zusätzlich wird in der hier aufgebauten Architektur bei jedem Benutzen des *Paper Publishing*-Systems überprüft, ob die Daten des IMS noch Gültigkeit haben. Für diese Überprüfung ist im Bibliothekssystem ein Common Gateway Interface-Skript (CGI-Skript) integriert, das via SSL verschlüsseltem HTTP POST abgefragt werden kann. Auch hier ist es denkbar, einen neuen Web Service zur Authentifikationsüberprüfung zu entwickeln. Es ist jedoch eine der Hauptanliegen des KIM-Projekts, vorhandene Systeme und Funktionalitäten zu integrieren und nicht zu reimplementieren.

Dem CGI-Skript auf Seiten der Bibliothek werden die HTTP Post Parameter *funktion*, *user* und *password* übergeben. Als Antwort sendet der Dienst unten stehende Nachrichten zurück (persönliche Informationen sind mit * ausgeblendet). Diese werden vom WebPart ausgewertet und nach einer erfolgreichen Authentifikation wird der Identifikator an den IMS übergeben. Denkbar ist hier eine Lösung, die den CGI-Skript-Aufruf aus dem IMS-Realm heraus initiiert. Dagegen spricht jedoch, dass der Benutzer seine Bibliotheks-Kenndaten zur initialen Authentifikationsüberprüfung auf der Portalseite eingibt und bei einer Implementierung innerhalb des Web-Parts diese Daten nur einmal an das Skript und nicht vorher zusätzlich an einen Web Service innerhalb des IMS-Realms über das Netz geschickt werden müssen.

Authentifikation OK:

```
OK
Benutzergruppe,Standard
Benutzer_ID,20******.**
STATISTIKGRUPPE_NAME,Student/in
STATISTIKGRUPPE_KUERZEL,1100
EMAIL,sebastian.********@gmail.com
```

Authentifikation gescheitert, Passwort falsch:

```
ERROR
Passwort ist falsch
```

Authentifikation gescheitert, Benutzername unbekannt:

```
ERROR
Benutzernummer ist falsch
```

5.2.8.4 Weitere Web Services

Neben den Web Services innerhalb des Identifier Mapping Services, die es dem Bibliotheks-Realm und der KIM-Domäne erlauben, organisationsspezifische Daten aus der Datenbank zu lesen oder Handles zu setzen, sind innerhalb der Architektur um das Paper Publishing System weitere Services implementiert. Innerhalb des Bereiches der Bibliothek soll im produktiven Betrieb ein weiterer Service für den Zugriff auf das Bibliothekssystem sorgen und hierzu entsprechend übermittelte Daten aus dem Paper Publishing-Modul des KIM-Portals entgegen nehmen. Ein weiterer Web Service im Bereich der KIM-Systeme dient dem endgültigen Abschluss der Transaktion. Dieser wird durch die Import-Applikation von Bibliotheksseite angestoßen. Er informiert den IMS über das, ab diesem Zeitpunkt nicht mehr gültige Handle, erfragt zuvor die E-Mail Adresse des Nutzers und sendet per E-Mail eine Benachrichtigung über Erfolg oder Misserfolg der Transaktion an den Anwender.

Bibliotheksschnittstelle

Die Bibliotheksschnittstelle wurde KIM-seitig prototypisch entwickelt. Für einen produktiven Betrieb ist eine Anpassung bzw. Erweiterung der entsprechenden Dienste der Universitätsbibliothek nötig. Im Folgenden wird die prototypische Schnittstelle beschrieben:

Der Web Service auf Bibliotheksseite nimmt aus dem WebPart des Sharepoint-Server Portals auf KIM-Seite die folgenden Daten entgegen: Den Institutsnamen, den der Benutzer eingegeben hat, das Handle, welches das WebPart gesetzt hat, ein Einmal-Passwort[2] für den KIM-FTP-Server, die Dateinamen der relevanten Dateien und die IP-Adresse des FTP-Servers.

Der Bibliotheks-Web Service ruft mit diesen Daten die nachgelagerte Import Applikation auf. Diese soll im Produktiveinsatz die Autorisation des Benutzers vornehmen und die Transaktion bibliotheksseitig abschließen. Mit dem übermittelten Handle kann die Applikation dafür beim IMS den organisationsspezifischen Identifikator erfragen. Anschließend soll diese Applikation den E-Mail Response Web Service des KIM-Realms aufrufen, um auch auf KIM-Seite die Transaktion abzuschließen.

[2]Einmal Passwort = Dieses Passwort ist nur für einen Login beim KIM-FTP gültig und kann nur zum herunterladen der mit der aktuellen Transaktion verknüpften Dateien benutzt werden.

E-Mail Responder

Der Web Service auf KIM-Seite wird durch die Import-Applikation der Universitätsbibliothek angestoßen. Diesem wird das Handle mitgeteilt, mit dem wiederum über den IMS der KIM-Identifikator ermittelt werden kann. Dem so identifizierten Benutzer wird an die systembekannte E-Mail-Adresse eine Nachricht gesendet. Weiterhin wird der IMS über den Abschluss der Transaktion informiert, um das entsprechende Feld zu editieren, welches die aktuell laufenden Transaktionen anzeigt.

Funktionalität aus Benutzersicht

Dieser und die folgenden Abschnitte sind der Implementierung des vorgestellten Paper Publishers gewidmet. Nachdem in diesem Abschnitt eine Einführung in die entwickelte Benutzerschnittstelle gegeben wird, werden in den darauffolgenden Abschnitten einige Implementierungsdetails vorgestellt.

Abbildung 5.20 zeigt die Ansicht der Upload-Variante *Einzelupload*. Hier müssen die entsprechend angezeigten Felder ausgefüllt werden und die Lokation des Volltextes der Veröffentlichung angegeben werden. Bei der Massenupload-Variante ist eine zuvor erstellte BibTEX-Datei notwendig, die sämtliche Meta-Informationen zu den in einem Archiv zusammengefassten wissenschaftlichen Arbeiten enthält. Diese Dateien können mit Hilfe der in Abbildung 5.21 dargestellten Maske auf den Server geladen werden, so dass der Transaktionsauftrag zur Bibliothek gesendet werden kann.

Zwischen dem initialen Bereich mit der Auswahl der Uploadart und den einzelnen Uploadmasken, wird überprüft, ob bereits ein Mapping zum Bibliotheks-Identifikator des angemeldeten Benutzers existiert. Sollte dies nicht der Fall sein, wird dem Anwender vor der Präsentation der Uploadmaske noch eine Eingabemaske für die Benutzerkennung der Bibliothek zur Authentifikationsüberprüfung dargestellt.

5.2.8.5 Klassenübersicht

In Abbildung 5.22 ist eine Übersicht der Klassen der föderativen Dienstanbindung dargestellt. Abgegrenzt werden diese durch die drei in vorangegangenen Abschnitten vorgestellten Realms *KIM-Seite*, *Bibliothek* und *IMS*. Neben den Klassennamen und zugehörigen Methoden ist für jede Klasse die Funktion, beziehungsweise der Typ notiert. Alle Klassen, die mit *WP* etikettiert sind, gehören unmittelbar zur Implementierung des Sharepoint WebParts. Eine *WS*-Bezeichnung steht für den Einsatz dieser Klasse als Web Service. Die mit *Appl.* gekennzeichnete Klasse ist eine eigenständige Applikation. Die *WP*-Klassen sind in C#.NET implementiert, alle anderen Klassen in JAVA.

Als zentrale Klasse innerhalb des KIM-Realms ist die Klasse `WebPart` erkennbar. In dieser Klasse sind der bereits angesprochene Pageflow und die entsprechenden Eventhandler für die verschiedenen Buttons, die dem Benutzer präsentiert werden, integriert. Die Klassen `AuthCheck`, `IMSCaller` und `Publish2Bib` stellen einen Zugangspunkt zu dritten Systemen für das WebPart dar. In diesen werden Web Services instanziiert und aufgerufen oder Verbindungen zu weiteren Diensten geschaffen.

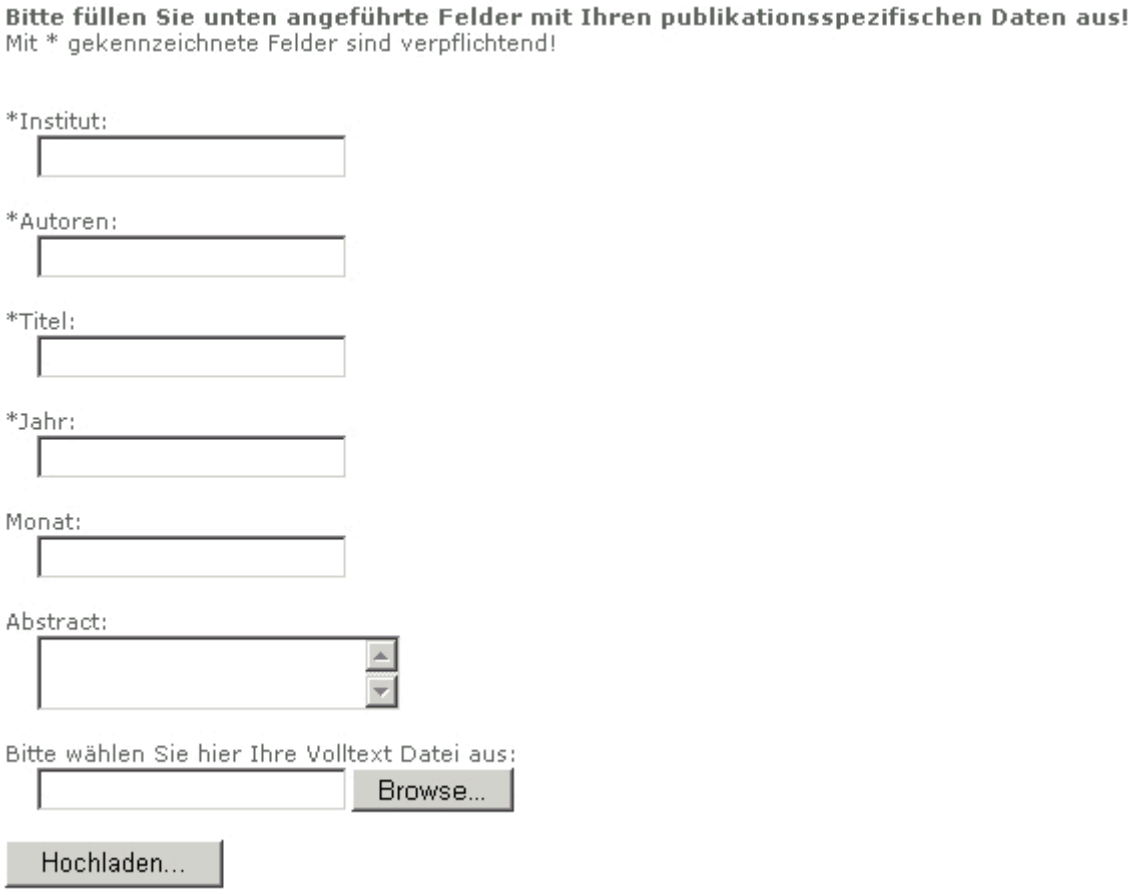


Abbildung 5.20: Eingabemaske für einen Einzelupload im Paper-Veröffentlichungsdienst

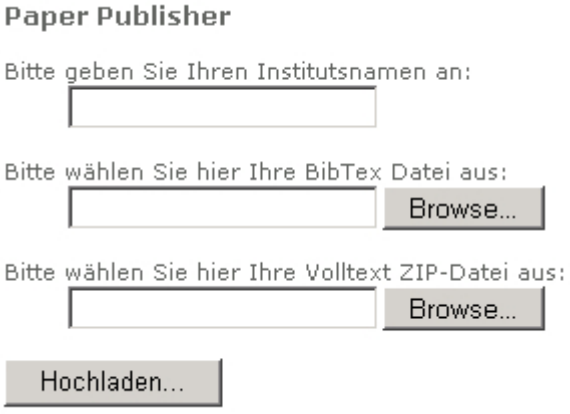


Abbildung 5.21: Eingabemaske für einen Massenupload im Paper-Veröffentlichungsdienst

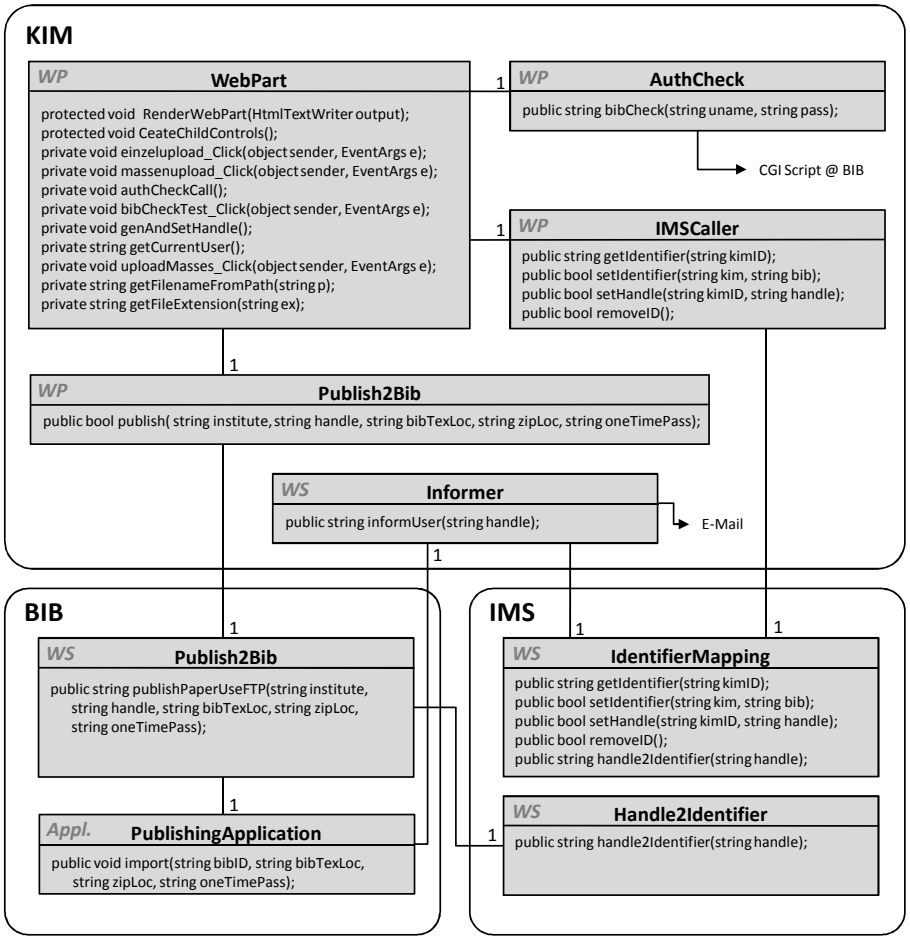


Abbildung 5.22: Klassen des Paper Publisher

Im Folgenden werden die einzelnen Klassen vorgestellt, wobei an interessanten Stellen näher auf die Implementierung eingegangen werden soll. Dabei liegt der Fokus innerhalb dieses Abschnitts auf der rein funktionalen Sicht der entstandenen Architektur.

WebPart.cs

Die Klasse `WebPart` ist, wie oben beschrieben, die Klasse, die den Pageflow und damit die Benutzerführung implementiert. Ihr liegen im Wesentlichen die Methoden `RenderContents(writer HtmlTextWriter)` und `CreateChildControl()` zugrunde. Die Methode `CreateChildControl()` definiert die einzelnen Seitenobjekte mit ihren Eventhandlern. In der Methode `RenderContents` wird eine globale Statusvariable ausgewertet. Je nach Wert dieser Variable werden die entsprechenden Elemente in die anzuzeigende Seite integriert. Globale Variablen bleiben so lange erhalten, bis der Anwender die Seite manuell neu lädt, die Seite verlässt, auf der das WebPart integriert ist oder seine Webanwendungs-Session beendet. Somit besteht über eine globale Variable eine einfache und schnell zu implementierende Möglichkeit, mit einem WebPart verschiedene Seiten oder Ansichten sequentiell anzuzeigen.

Die Eventhandler `einzelupload_Click` und `massenupload_Click` speichern jeweils lediglich die Entscheidung zwischen den zwei Uploadvarianten in einer Statusvariable

und rufen die Methode `authCheckCall()` auf. Diese Methode überprüft, ob bereits ein Mapping im Identifier Mapping Service vorhanden ist. Hierzu wird die Klasse `IMSCaller` instanziiert und die Methode `getIdentifier` aufgerufen. Ist dies nicht der Fall, setzt die Methode die Pageflow steuernde Statusvariable auf den Wert *mapping* und ist damit durchlaufen. Die im Browser gezeigte Seite wird neu geladen und der Anwender bekommt eine Eingabeaufforderung für seine Bibliothekskennung präsentiert. Besteht bereits ein Mapping im IMS, wird der dort zu findende Bibliotheks-Identifikator mit dem Passwort an den Authentifikations-Prüfdienst (CGI-Skript) der Bibliothek gesendet. Dies geschieht über die Klasse `AuthCheck` mit der Methode `bibCheck(string, string)`. Ist der Benutzername dem System unbekannt, so wird der gespeicherte Bibliotheks-Identifikator aus dem Identifier Mapping Service entfernt und die Statusvariable auf den Wert *mapping* gesetzt. Dies veranlasst, wie oben bereits beschrieben, einen Seitenaufbau mit Eingabefeldern für die Bibliothekskennung. Sollte der IMS nun unvorhergesehen nicht mehr erreichbar sein, wird dem Benutzer eine Fehlermeldung präsentiert. Ohne den IMS ist eine Transaktion zur Bibliothek derzeit nicht vorgesehen.

Zwei weitere Methodenaufrufe sind Bestandteil der Authentifikationsüberprüfung. Die Methode `getCurrentUser()` liefert einen String mit dem Benutzernamen des angemeldeten Anwenders zurück. Dieser Name ist als Session-Variable gespeichert und kann über die Methode ausgelesen werden. Die Methode `genAndSetHandle` generiert ein Handle, sofern im IMS kein gültiges eingetragen ist. Dieses wird mittels eines sogenannten *Global Unique Identifiers (GUID)* erzeugt und an den IMS-Web Service übergeben, beziehungsweise an dessen Methode `setHandle`.

Die Methode `uploadMasses_Click` schließt die Transaktion zunächst KIM-seitig ab und ruft die Methode `publish` auf, wie es in Abbildung 5.22 verdeutlicht ist. Diese wiederum spricht den Web Service `Publish2Bib` des Realms Bibliothek an und übergibt die relevanten Parameter. Dies sind neben dem vom Benutzer angegebenen Institutsnamen, das Handle, die Lokation der zur Transaktion gehörigen BibTEX-Datei, sowie des ZIP-Archivs und ein Einmal-Passwort für den FTP-Server.

IMSCaller.cs und Publish2Bib.cs

Die beiden Klassen `IMSCaller` und `Publish2Bib` werden benutzt, um die unterschiedlichen Web Services anzusprechen. Die `IMSCaller`-Klasse ist für die Kommunikation zum Web Service des Identifier Mapping Services zuständig, während die `Publish2Bib`-Klasse die Übergabe der Transaktionsparameter an den entgegennehmenden prototypisch entwickelten Web Service der Bibliothek handhabt. Um die Strukturen einfach und überschaubar zu halten, sowie eine bestätigte Übertragung zu gewährleisten, werden die Web-Methoden innerhalb dieses Dienstes stets synchron aufgerufen. Die Asynchronität bezüglich des Ansprechens des FTP-Servers gelangt erst auf Bibliotheksseite in die Implementierung.

AuthCheck.cs

Die Klasse `AuthCheck` bewältigt die initiale Authentifikationsüberprüfung beim ersten Nutzen des Paper Publisher und die bei jeder Nutzung durchgeführte Überprüfung nach der Existenz des Benutzeridentifikators auf den Systemen der Bibliothek.

Für diese Kontrolle wird ein CGI-Skript auf Bibliotheksseite über eine SSL gesicherte Verbindung vom WebPart genutzt, wie es im vorigen Abschnitt vorgestellt wurde.

Publish2Bib.java und Publishing-Application

Der prototypische Bibliotheks-Web Service `Publish2Bib` nimmt die Transaktionsdaten entgegen und schreibt sie in eine Datei, welche die Publishing-Applikation der Bibliothek periodisch auslesen soll. Diese kann die Daten überprüfen und eine autoritative Entscheidung treffen, um dann die Daten der Datenbank-Schnittstelle zu übergeben. Zuvor erfragt der Web Service `Publish2Bib` beim IMS den bibliothekseigenen Identifikator und übergibt diesen ebenfalls als Grundlage für die Autorisation an die Import-Applikation.

IdentifierMapping.java und Handle2Identifier.java

Dem IMS liegt eine SQL-Datenbank zugrunde, die von den einzelnen abfragenden Web Services ausgelesen und editiert werden kann. Der KIM-Seite steht ein Web Service zur Verfügung, der Methoden implementiert, die zum Übersetzen des Handles in einen KIM-Identifikator und umgekehrt dienen. Weiterhin können durch eine Methode für einen KIM-Identifikator ein neues Handle gesetzt und ein Mapping angelegt werden. Für den Fall, dass ein Bibliotheks-Identifikator nicht mehr existent ist, kann das zugehörige Mapping durch eine weitere Methode annulliert werden.

Der Bibliothek stehen lediglich zwei Methoden zum Übersetzen des Handles in einen Bibliotheks-Identifikator und umgekehrt zur Verfügung. Die Bibliothek kann weder Handles setzen, noch Mappings aus der IMS-Datenbank löschen. Diese Datenbank enthält in der ersten Spalte den Identifikator des KIM-Realms, die zweite Spalte repräsentiert das Handle und die Bibliothekskennung wird in Spalte drei abgelegt. Ein ActivityCount zeigt aktuell laufende Transaktionen an.

Informer.java

Die Klasse Informer ist ebenfalls ein Web Service, der von der Bibliotheksseite aufgerufen werden kann. Zu diesem wird der finale Status einer Transaktion und das aktuell gültige Handle gesendet. Dieses wird in den KIM-Identifikator übersetzt. Weiterhin wird, da dieser Identifikator eine E-Mail-Adresse darstellt, an diesen und damit den entsprechenden Anwender eine Information über den positiven oder auch negativen Verlauf der Transaktion gesendet.

5.2.8.6 Zusammenfassung und Status

Zum Paper Publishing Dienst wurden zunächst die Gesamtarchitektur und detaillierte Designentscheidungen zu den einzelnen Komponenten vorgestellt. Es wurden interessante Implementierungsdetails der Dienstentwicklung und -anbindung aufgezeigt. Neben den einzelnen Klassen wurden deren Integration und ihre Aufgaben innerhalb der SOA dargestellt. Es wurde gezeigt, dass das Sharepoint WebPart als zentrale Komponente der Architektur implementiert ist und dieses dem Anwender die Schnittstelle zur nachgelagerten Funktionalität bietet. Besonders sei in diesem Zusammenhang noch einmal darauf hingewiesen, dass dem Benutzer lediglich die WebPart-Schnittstelle visualisiert wird. Die Nutzung weiterer Dienste bleibt bei der Verwendung des integrierten Dienstes der Paperveröffentlichung weitestgehend verborgen. Dabei wird dem Anwender jedoch nicht vorenthalten, dass er mit der Verwendung des Systems Dienste der Universitätsbibliothek aus dem Mitarbeiterportal heraus benutzt. Der Paper Publisher ist KIM-seitig fertig entwickelt und befindet sich zurzeit im Testbetrieb. Zur endgültigen Freischaltung bedarf es einer Anpassung bzw. Erweiterung der Dienste der Universitätsbibliothek.

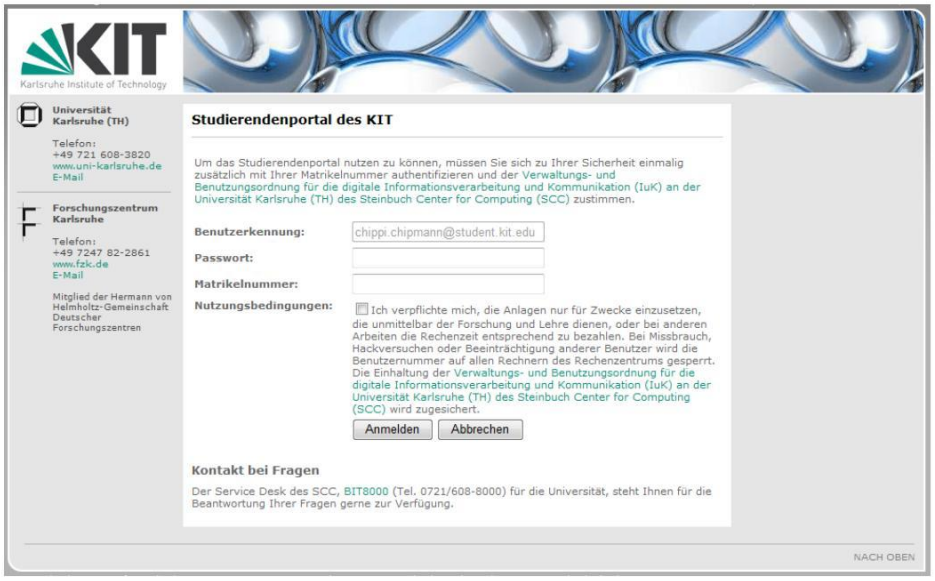


Abbildung 5.23: Nutzungsbedingungen-Überprüfung im Studierendenportal

5.3 Studierendenportal

Neben den entwickelten Diensten für das Mitarbeiterportal wurden im Rahmen des Projektes KIM-IDM folgende Funktionalitäten für das Studierendenportal entworfen und implementiert.

5.3.1 Nutzungsbedingungen-Überprüfung

Studierende, die sich immatrikulieren, erhalten ihre Zugangsdaten für das Studierendenportal per Brief, sollen aber trotzdem den Nutzungsbedingungen des SCC zustimmen. Hierfür wurde die Nutzungsbedingungen-Überprüfung umgesetzt. Studierende, die sich zum ersten Mal am Studierendenportal anmelden, müssen der Verwaltungs- und Benutzungsordnung für die digitale Informationsverarbeitung und Kommunikation des SCC zustimmen. Der Nutzer wird verpflichtet, die Anlagen nur für Zwecke einzusetzen, die unmittelbar der Forschung und Lehre dienen, oder bei anderen Arbeiten die Rechenzeit entsprechend zu bezahlen. Bei Missbrauch, Hackversuchen oder Beeinträchtigung anderer Benutzer wird die Benutzernummer auf allen Rechnern gesperrt.

Hierfür wurde das Login-Control auf der Startseite des Studierendenportals modifiziert. Das Control fragt nach erfolgreicher Überprüfung der Nutzer-Credentials über den SCCIDM-Service (siehe Abschnitt 6.1.7) ab, ob der Studierende bereits die Nutzungsbedingungen des SCC akzeptiert hat. Falls er bereits zugestimmt hat, wird der Nutzer auf die interne Startseite des Studierendenportals weitergeleitet. Andernfalls erhält der Nutzer die in Abbildung 5.23 gezeigte Mitteilung mit der Bitte um Akzeptanz der Nutzungsbedingungen des SCC. Falls er durch Setzen der entsprechenden Checkbox zustimmt, wird der SCCIDM-Service erneut aufgerufen, der dann das aktuelle Datum in einer Datenbank hinterlegt. Dies löst eine Reaktion des SCC-IDM-Systems aus, das das Nutzerkonto auf Seiten des SCC über die Benutzerverwaltungssoftware anlegt.

5.3.2 Selbstbedienungsdienst

Eine wesentliche Funktionalität des Studierendenportals ist die Selbstbedienungsfunktionalität der Universitätsverwaltung. Neben den Informationen zu Vorlesung und Studienablauf sind die von den Studierenden am stärksten nachgefragten Funktionen die Studienbescheinigungen, der Notenauszug, die Anmeldung zur Prüfung und Änderung der eigenen Adresse. Während es sich bei den beiden ersten Diensten um rein lesende Zugriffe handelt, für die bereits sichere Web Service Schnittstellen realisiert wurden, müssen für die Anmeldung zur Prüfung und zur Änderung der Adresse schreibende Zugriffe erfolgen. Ein direkter schreibender Zugriff auf die zu Grunde liegenden Datenbanken scheidet grundsätzlich aus, da die Anwendungslogik unbekannt ist und somit die Konsistenz der Daten nicht gewährleistet werden kann. Aufgrund der fehlenden Web Service-Schnittstelle des HIS-Systems wurde für diese Dienste auf die Selbstbedienungs-Webanwendung des HIS-Systems zurückgegriffen werden, da dort die notwendigen Konsistenzprüfungen im Rahmen der Anwendungslogik vorhanden sind. Da alle Dienste personenbezogene Daten von Studierenden verarbeiten, muss besonderen Wert auf den Schutz der Daten gelegt werden. Dieser Schutz umfasst neben der Sicherung vor nicht autorisiertem Zugriff auch die Garantie von Konsistenz und Aktualität.

Der hier verfolgte Ansatz sieht eine direkte Integration der Web-Anwendung in die Sharepoint-Umgebung des Studierendenportals vor. Dazu wurden folgende Integrationsszenarien realisiert.

- Aus dem Sharepoint wird ein neu neues Browserfenster mit der HIS Anwendung geöffnet. Dabei findet zuvor über den Sharepoint eine automatische Authentifikation (engl. Single Sign-On, SSO) an der Web-Anwendung des HIS-Systems statt und die gewünschte Seite wird direkt geöffnet (direct-call).
- Auf der Seite im Sharepoint wird ein WebPart angezeigt, das die Inhalte der Web-Anwendung des HIS direkt darstellt und dem Nutzer Änderungen ermöglicht. Dazu werden neben der Realisierung der SSO und der direct-call Funktionen noch die HTML-Filterfunktion des Sharepoint genutzt.

Das HIS-Portal unterstützt SSO sowohl gegen HIS als auch gegen andere Systeme. Im ersten Fall agiert die HIS-Infrastruktur als Authentifikationsserver für andere Systeme; im zweiten Fall verwendet die HIS-Infrastruktur ein externes System zur Authentifikation. In unserem Szenario findet die Authentifikation in einem externen System, dem Sharepoint, statt. Das HIS-Portal verwendet die Authentifikationsinformation aus dem externen System, vor allem die Benutzeridentität, um dem Benutzer entsprechende Portalfunktionalitäten anzubieten. Für die Verwendung eines externen Systems zur Authentifikation ist eine entsprechende Konfiguration auf dem HIS-Server erforderlich.

Das HIS-Portal speichert alle zur Authentifikation relevanten Informationen in der Konfigurationsdatei *LoginConf.xml*. Diese Konfigurationsdatei befindet sich im Verzeichnis *[TOMCAT]/webapps/qisserver/WEB-INF/conf* des Tomcat-Servers. Hierfür müssen folgende Konfigurationen müssen in dieser Datei vorgenommen werden:

- *Single Sign-On Authentifikations-Plugin.* Diese Einstellung gibt an, welche Java Klasse als Plugin für die Authentifikation des SSO-Token verwendet werden soll.
- *Single Sign-On Einstellungen.*
  - *Service* gibt den Identifikator des Single Sign-On Dienstes an.
  - *Shared Secret* dient als gemeinsames Geheimnis zwischen HIS-System und Studierendenportal. Es enthält eine beliebige Kombination aus Buchstaben und Zahlen. Diese Kombination wird während des SSO-Vorgangs dazu verwendet, den SSO-Token mit Hilfe des Hash-Verfahrens zu generieren. Diese Information begründet die Vertrauensstellung der Systeme zueinander und darf Dritten nicht zugänglich gemacht werden.
  - *Timeout* gibt den Gültigkeitszeitraum eines SSO-Token an. Wenn dieser Zeitraum überschritten ist, wird das empfangene SSO-Token vom HIS-Portal ignoriert. Dies verlangt nach einer Zeit-Synchronisation der beteiligten Server.
- *Auswertung von Benutzerinformationen.* Nachdem das HIS-Portal die Überprüfung des SSO-Token erfolgreich durchgeführt hat, braucht das HIS-Portal Anweisungen zum Auslesen der Benutzer- und Rolleninformationen aus der HIS-Datenbank.

Das Single Sign-On basiert auf einem symmetrischen Verfahren mit einem für beiden Seiten bekannten Geheimnis, dem *Shared Secret*. Zuerst wird eine Zeichenkette bestehend aus Zielsystem, Benutzerinformation, Zeitstempel erstellt und mit dem Shared Secret verknüpft. Eine solche Zeichenkette sieht folgendermaßen aus: *1.0/1115814654/qis/Schmidt/kahC1oo3pieg6FaekEhou1aipEivae4fe*.

Anschließend wird aus dieser Zeichenkette der Hashwert mittels der Hashfunktion MD5 berechnet. Daraus wird das SSO-Token erstellt, welches neben dem Hashwert auch Informationen über das Zielsystem, den aktuellen Benutzer sowie einen neu generierten Zeitstempel enthält. Anschließend wird das SSO-Token über HTTP-Post an das HIS-Portal übermittelt. Nachdem das HIS-Portal das übermittelten SSO-Token erhalten hat, extrahiert das Authentifikations-Plugin die benutzerbezogene Information aus dem Token wie etwa *1.0/1115814654/qis/Schmidt* und verknüpft diese wieder mit dem Shared Secret. Die daraus resultierende Zeichenkette wird verwendet, um einen lokalen Hashwert zu berechnen. Dieser lokale Hashwert wird mit dem übermittelten Hashwert verglichen. Wenn beide Werte übereinstimmen, dann ist der entsprechende Benutzer für das HIS-Portal authentifiziert. In diesem Fall wird der Benutzer zum personalisierten Benutzerbereich weitergeleitet. Ansonsten ist die Authentifikation fehlgeschlagen und eine entsprechende Fehlerseite wird angezeigt.

Um zu einer bestimmten Seite in der HIS-Web-Anwendung zu springen, kann zusätzlich die Ziel-URL (beginnend hinter dem Fragezeichen) als Parameter *re* übergeben werden. Dabei müssen Gleichheitszeichen als %3D und ein kaufmännisches Und-Zeichen als %26 maskiert werden.

Die Konfiguration des WebParts erfolgt direkt im Sharepoint, über die Application Settings in der *web.config* des Sharepoints und über eine Datei, die eine Beschreibung der Ziel-URLs in Form von XML enthält. Zunächst müssen in der web.config das gemeinsame Shared Secret und das Zielsystem angegeben werden. Diese Informationen werden wie oben beschrieben zusammen mit einem Identifier, wie der Matrikelnummer, die aus dem Session Context geholt wird, und mit einem jeweils neu generierten Zeitstempel zur Generierung des Tokens verwendet. Die Konfiguration des WebParts im Sharepoint lässt nun mehrere Einstellungen zu. Zunächst kann durch Setzen einer CheckBox bestimmt werden, welches der beiden Nutzungsszenarien gewünscht wird. Zum einen kann das HIS-Portal direkt im WebPart gerendert werden, zum anderen kann das WebPart einen Hyperlink darstellen, über den ein neues Browserfenster geöffnet wird. Als weiterer Konfigurationsparameter muss eine URL angegeben werden, unter der das HIS-Portal erreichbar ist, wie bspw. *http://his-portal:8080/qisserver9/rds*. Zusätzlich können eine oder mehrere Zielseiten des HIS-Portals, zu denen direkt navigiert werden soll, spezifiziert werden. Damit kann der direct-call zu spezifischen Seiten durchgeführt werden, etwa zur Adressänderung oder zum Ausdruck der Studienbescheinigung.

Die Konfiguration der möglichen Zielparameter erfolgt über eine externe Datei. Der Pfad zu dieser Datei kann über die Edit-Funktion des WebParts über den Parameter *Target Map Location* konfiguriert werden. Jedes Ziel wird über drei Attribute spezifiziert: *DisplayName* ist der Text der als Hyperlink dargestellt wird, über *Key* wird das Ziel referenziert und *URL* bestimmt die Zielseite im HIS-Portal. Im WebPart können in der *Target Key List* nun beliebig viele Schlüssel zu Zielseiten mit Semikolon getrennt eingegeben werden, wie etwa *TelNo*, *Tan* oder *StdPln*.

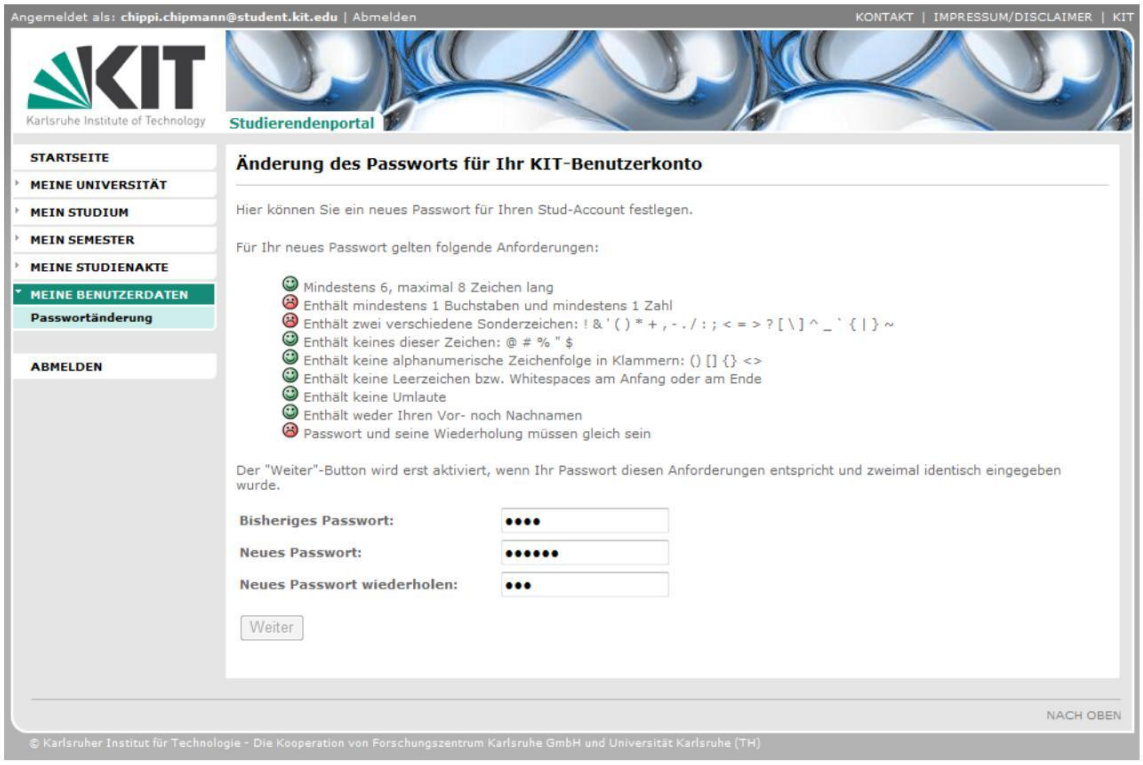


Abbildung 5.24: Website zur Passwortänderung im Studierendenportal

5.3.3 Passwortänderungsdienst

Analog zu dem bereits in Abschnitt 5.2.5 erläuterten Passwortänderungsdienst steht im Studierendenportal der gleiche Dienst zur Verfügung. Dieser Dienst ist im Studierendenportal unter „Meine Benutzerdaten\Passwortänderung“ momentan allein der Nutzergruppe der Studierenden zugänglich. Dozenten und Mitarbeiter können diese Funktion im Mitarbeiterportal nutzen. Die Richtlinien an ein Passwort sind dabei analog zu denen in Abschnitt 5.2.5.

Diese Richtlinie wird sowohl Client-seitig über Javascript als auch Server-seitig mittels regulärer Ausdrücke überprüft. Nach erfolgreicher Eingabe des alten und neuen Passworts werden diese Daten an den SCCIDM-Service (siehe Abschnitt 6.1.7) übermittelt. Anschließend werden diese Berechtigungsnachweise gegen das OpenLDAP-Verzeichnis des SCC geprüft. Nach einer erfolgreichen Authentifikation des Nutzers werden diese Daten an ein CGI-Skript unter der URL *https://www.rz.uni-karlsruhe.de/cgi-bin/bvpasswd* gesendet, das die Benutzerverwaltung des SCC anstößt, eine Provisionierung des Passworts vorzunehmen. Nach erfolgreichem Setzen des Passworts wird dem Nutzer eine entsprechende Meldung angezeigt. Abbildung 5.24 zeigt die Website zur Passwortänderung.

6. Infrastrukturdienste

In diesem Kapitel werden die grundlegenden Infrastrukturdienste näher erläutert. Zunächst werden die implementierten Web Services aufgezeigt, worauf eine Beschreibung des Single Sign-On-Dienstes Shibboleth folgt.

6.1 Web Services

Die grundlegenden Komponenten der im Projekt KIM-IDM aufgebauten integrierten Service-orientierten Architektur stellen die hier aufgeführten Web Services dar.

6.1.1 KISS-Repository Service

Der KISS-Repository Service kapselt den Zugriff auf die KISS-Repository-Datenbank mit der Relation KISS-Repository, die die Mitarbeiter-Daten des KIT vorhält.

6.1.1.1 Datenbankrelationen

In der Relation KISS-Repository (siehe Tabelle 6.1) werden die wichtigsten Daten sowohl über Mitarbeiter des Campus Süd als auch Mitarbeiter des Campus Nord gehalten. Dies umfasst personenbezogene Daten wie Vornamen, Nachnamen und *kit.edu*-E-Mail-Adresse, aber auch für das Identitätsmanagement notwendige Daten wie Erstellungs- und Löschdatum, lokale Identifikatoren der angebundenen Einrichtungen sowie Statusinformationen.

6.1.1.2 Read

Beim lesenden Zugriff auf den KISS-Repository Service können die entsprechenden Daten für Mitarbeiter aus der Relation KISS-Repository ausgelesen werden. Hierfür wird der KIT-Identifier, der dem UserPrincipalName oder auch der *kit.edu*-E-Mail-Adresse entspricht, übergeben. Der Service ruft die Daten aus der Datenbank über die Stored Procedure *kim_select_KISS-Repositoryservice* (siehe Anhang A.1) ab. Das zurückgegebene Objekt entspricht folgendem Schema.

Attribut	Beschreibung
firstname	Vorname eines Mitarbeiters
lastname	Nachname eines Mitarbeiters
email	*kit.edu*-E-Mail-Adresse eines Mitarbeiters
username	Generierter Identifikator
localidentifier1	Identifikator der Universitätsverwaltung
localidentifier2	Identifikator des SCC
localidentifier3	Identifikator des FZK
localidentifier4 - 10	Reserviert für weitere Identifikatoren
rzemail	SCC-E-Mail-Adresse eines Mitarbeiters
userid	Durch das Identitätsmanagement generierte global eindeutige GUID
status	Status eines Mitarbeiters (siehe Abschnitt 4.3.1)
rzoldpassword	Initiales Passwort für einen neuen SCC-Account
dateofcreation	Erstellungsdatum
dateofdeletion	Löschdatum

Tabelle 6.1: Attribute der Relation KISS-Repository

Listing 6.1: KISSRepositoryUserType

```
<xs:schema id="KISSRepositoryUserType"
    targetNamespace=...>
  <xs:element name="KISSRepositoryUserType">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="cont:ContentObject">
          <xs:sequence>
          <xs:element name="Useridentifier" type="xs:string"/>
          <xs:element name="Firstname" type="xs:string" />
          <xs:element name="Lastname" type="xs:string" />
          <xs:element name="Email" type="xs:string" />
          <xs:element name="Status" type="xs:string" />
          <xs:element name="Role" type="xs:string" />
          <xs:element name="Rzemail" type="xs:string" />
          <xs:element name="Rzoldpassword" type="xs:string" />
          <xs:element name="Username" type="xs:string" />
          <xs:element name="Password" type="xs:string" />
          <xs:element name="Lastlogin" type="xs:string" />
          <xs:element name="Isactivated" type="xs:string" />
          <xs:element name="Isrznew" type="xs:string" />
          <xs:element name="Emailalias"
                type="xs:string"
                minOccurs="0"
                maxOccurs="unbounded" />
          <xs:element name="Localidentifier"
                type="LocalidentifierType"
                minOccurs="0"
                maxOccurs="unbounded" />
          </xs:sequence>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Die Werte werden entsprechend dem Datenbank-Eintrag gefüllt. Über das XML-Element *Localidentifier* können die Identifikatoren der einzelnen Einrichtungen übergeben werden. Dies beinhaltet aktuell die Identifikatoren für die Universitätsverwaltung, das SCC und das Forschungszentrum. Der komplexe Typ LocalidentifierType beinhaltet den Identifikator der Einrichtung und den Namen der Einrichtung.

Listing 6.2: LocalidentifierType

```
<xs:complexType name="LocalidentifierType" >
  <xs:sequence>
    <xs:element name="Localid" type="xs:string" />
    <xs:element name="Resource">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="RZ" />
          <xs:enumeration value="UB" />
          <xs:enumeration value="FZK" />
          <xs:enumeration value="ZUV" />
          <xs:enumeration value="ATIS" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Falls für den Nutzer kein Eintrag in der Datenbank vorhanden ist, wird eine Exception mit der Meldung „400:User not found.“ geworfen und ein Eintrag in der Log-Datei „REP-READ.log“ vorgenommen.

6.1.1.3 Update - Password

Der KISS-Repository Service kann zwei unterschiedliche Änderungen an den Mitarbeiterdaten durchführen. Zum einen kann das Passwort eines Mitarbeiters neu gesetzt werden und zum anderen kann die E-Mail-Weiterleitung angepasst werden.

Der UpdateContext für die Passwortänderung eines Mitarbeiters enthält als Parameter *KISSRepositoryUpdateParamsPassword* wie in Listing 6.3 dargestellt. Es wird der Identifikator des Nutzers, sein altes Passwort und sein neues Passwort übergeben. Das alte Passwort wird gegen das *kit.edu*-Active Directory geprüft und bei erfolgreicher Überprüfung wird anschließend das neue Passwort über den SPML Service (siehe Abschnitt 6.1.8) and das Identitätsmanagementsystem weitergeleitet und unter anderem in das *kit.edu*-Active Directory provisioniert. Der zum Aufruf der updatePassword-Methode des SPML Service notwendige sAMAccountName wird zuvor aus der KISS-Repository-Datenbank geholt.

Listing 6.3: UpdateContext mit KISSRepositoryUpdateParamsPassword

```
<UpdateContextWithParams xmlns="http://schemas.../cruds">
  <ISID>
    <Identifier xmlns="...">vorname.nachname@kit.edu</Identifier>
    <ServiceId xmlns="...">D6510679-...-6C02</ServiceId>
  </ISID>
  <Params>
    <KISSRepositoryUpdateParamsPassword
        xmlns="..../KISSRepositoryService_Password_UpdateParamTypes.xsd">
      <OldPassword>meinaltespasswort</OldPassword>
      <NewPassword>meinneuespasswort</NewPassword>
```

```
    </KISSRepositoryUpdateParamsPassword>
  </Params>
</UpdateContextWithParams>
```

6.1.1.4 Update - Weiterleitung

Der UpdateContext für die Aktualisierung der E-Mail-Weiterleitung ist an dem Parameter *KISSRepositoryUpdateParamsRZ* erkennbar. Es wird dabei der lokale Identifikator des SCC übergeben sowie die Information ob es sich um einen neuen Mitarbeiter handelt. Zunächst wird der sAMAccountname des Nutzers aus der KISS-Repository-Datenbank ausgelesen. Anschließend erfolgt der Aufruf des SPML Service (siehe Abschnitt 6.1.8), der den Sun Identity Manager veranlasst, die Weiterleitung in die entsprechenden Stellen zu schreiben. In der Log-Datei „REP-UPDATE.log“ werden die durchgeführten Update-Operationen festgehalten.

Listing 6.4: UpdateContext mit KISSRepositoryUpdateParamsRZ

```
<UpdateContextWithParams xmlns="http://schemas.../cruds">
  <ISID>
    <Identifier xmlns="...">vorname.nachname@kit.edu</Identifier>
    <ServiceId xmlns="...">D6510679-...-6C02</ServiceId>
  </ISID>
  <Params>
    <KISSRepositoryUpdateParamsRZ
        xmlns=".../KISSRepositoryService_RZ_UpdateParamTypes.xsd">
      <RZIdentifier>kj73</RZIdentifier>
      <Isrznew>true</Isrznew>
    </KISSRepositoryUpdateParamsRZ>
  </Params>
</UpdateContextWithParams>
```

6.1.2 Identifier Mapping Service

Der Identifier Mapping Service liefert für eine Identität die Identifikatoren unterschiedlicher Domänen. Der Service implementiert die *Read*-Operation nach dem CRUDS+F*-Muster (siehe Abschnitt 5.1.2) und dient ausschließlich dem lesenden Zugriff auf die Identifikatoren. Der Service wird sowohl von Mitarbeiterportal-Diensten als auch Studierendenportal-Diensten genutzt.

6.1.2.1 Read

Nachfolgend ist im Listing 6.5 ein beispielhafter *ReadContext* dargestellt, der zu einem Mitarbeiter, dessen KIT-Identifikator bekannt ist, den Identifikator der Domäne Forschungszentrum liefert.

Listing 6.5: ReadContext mit IdentifierMappingReadParam

```
<ReadContextWithParams>
  <OutputSchema>.../IdentifierMappingType.xsd</OutputSchema>
  <Params>
    <IdentifierMappingReadParam>
          <Source>
      <SourceIdentifier>frank.schell@kit.edu</SourceIdentifier>
      <Resource>KIT</Resource>
    </Source>
    <Targets>
```

```
        <TargetResource>FZK</TargetResource>
      </Targets>
      <Status>Employee</Status>
    </IdentifierMappingReadParam>
  </Params>
</ReadContextWithParams>
```

Die Informationen sind in der KISS-Repository-Datenbank für Mitarbeiter in der Relation *accounts* (siehe Abschnitt 4.2.6) und für Studierende in der Relation *studentidentifiermapping* (siehe Abschnitt 4.2.7) hinterlegt.

6.1.3 Person Service

Der Person Service kapselt den Zugriff auf die Datenbanken bzw. die HIS-Systeme der Universitätsverwaltung, wobei der Zugriff auf HIS SOS (Studierendendaten) vom Projekt KIM-LPS und der Zugriff auf die HIS SVA (Mitarbeiterdaten) vom Projekt KIM-IDM implementiert wurde.

6.1.3.1 Read - EmployeeType

Das OutputSchema *EmployeeType* dient zur Abfrage zusätzlicher Informationen über einen Nutzer, die für die Zuweisung von Zugriffsrechten im Mitarbeiter- oder Studierendenportal genutzt werden. Im *ReadContext* muss der jeweilige lokale Identifikator und die Einrichtung des Mitarbeiters (Universitätsverwaltung (ZUV) oder FZK) spezifiziert werden. Nachfolgend ist ein beispielhafter *ReadContext* dargestellt.

Listing 6.6: ReadContext mit KISSPersonServiceReadParam_Employee

```
<ReadContextWithParams xmlns="http://.../cruds">
    <rc:ISID>
      <rc:Identifier>svaid_des_nutzers</rc:Identifier>
      <rc:ServiceId>1234</rc:ServiceId>
    </rc:ISID>
    <OutputSchema>http://.../KISSPersonType_200708.xsd:EmployeeType</OutputSchema>
    <Params>
      <KISSPersonServiceReadParam_Employee xmlns="http://.../ReadParamTypes.xsd">
        <Resource>ZUV</Resource>
      </KISSPersonServiceReadParam_Employee>
    </Params>
  </ReadContextWithParams>
```

Für Forschungszentrumsmitarbeiter folgt der Aufruf zweier Web Services, die Informationen zum Status und den Attributen des Mitarbeiters liefern. Hierfür wird zum einen der FZK-Professoren-Service aufgerufen, der bestimmt, ob ein Nutzer Professor ist und zum anderen wird der WissMa-Service aufgerufen, der die Attribute der Mitarbeiter liefert. Auf Seiten der Universität erfolge eine dreimalige Abfrage des so genannten DBInterface, das die HIS-Datenbanken über eine SOAP-Schnittstelle kapselt. Zunächst wird über das DBInterface die View *SVAProfessorView* angesprochen, die alle notwendigen Informationen über Professoren enthält. Falls der Nutzer kein Professor ist, wird eine weitere View *SVAScientistView* abgefragt, die alle Daten für wissenschaftlichen Mitarbeiter enthält. Nur falls der Nutzer hier auch nicht zu finden ist, wird die *SVAEmployeeView* aufgerufen.

Letztlich wird das Outputschema mit dem Element *EmployeeType* erstellt und zurückgegeben. Die drei Boolean-Werte *isProf*, *isScientist* und *isEmployee* werden je nach Datenlage auf *true* gesetzt.

Listing 6.7: EmployeeType

```
<xs:complexType name="EmployeeType">
  <xs:complexContent>
    <xs:extension base="cont:ContentObject">
      <xs:sequence>
        <xs:element name="Lastname" type="xs:string"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="Firstname" type="xs:string"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="OrgUnit" type="xs:string"
                minOccurs="1" maxOccurs="1" />
        <xs:element name="Organization" type="xs:string"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="CostUnit" type="xs:string"
                minOccurs="1" maxOccurs="1" />
      </xs:sequence>
      <xs:attribute name="isProf" type="xs:boolean" use="required"/>
      <xs:attribute name="isScientist" type="xs:boolean" use="required"/>
      <xs:attribute name="isEmployee" type="xs:boolean" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

6.1.3.2 Read - IDMEmployeeType

Ein weiteres OutputSchema wird durch *IDMEmployeeType* spezifiziert. Dieses dient ausschließlich zur Abfrage von Daten aus HIS SVA. Über das DBInterface wird die View *IDMEmployeeView* bezüglich der Attribute eines Mitarbeiters abgefragt und mit folgendem Element zurückgegeben.

Listing 6.8: IDMEmployeeType

```
<xs:complexType name="IDMEmployeeType">
  <xs:complexContent>
    <xs:extension base="cont:ContentObject">
      <xs:sequence>
        <xs:element name="SvaId" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="Lastname" type="xs:string"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="Firstname" type="xs:string"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="OrgUnit" type="xs:string"
                minOccurs="1" maxOccurs="1" />
        <xs:element name="CostUnit" type="xs:string"
                    minOccurs="1" maxOccurs="1" />
        <xs:element name="Gender" type="xs:string" minOccurs="1" maxOccurs="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Diese Abfrage wird bspw. im Vodafone-Beantragungsdienst des Mitarbeiterportals (siehe Abschnitt 5.2.7) benötigt, um die entsprechenden Daten eines Nutzers auszulesen.

Name der Relation	Beschreibung
users	Nutzerdaten
simcardnumbers	SIM-Kartendaten
phonenumbers	Vergebene Telefonnummern
devicestophonenumbers	Zuordnung Gerät zu Telefonnummer
devices	Verfügbare Gerätetypen
delegatees	Delegierte Nutzer
deactivatedusers	Deaktivierte Nutzer

Tabelle 6.2: Datenbankrelationen für den Vodafone Service

6.1.3.3 Read - IsSVAEmployeeCountType

Das OutputSchema IsSVAEmployeeCountType wird während der Aktivierung eines Mitarbeiters zur Überprüfung des Geburtstags verwendet. Hierfür werden der lokale Identifikator und das Geburtsdatum des zu überprüfenden Mitarbeiters übergeben und über das DBInterface gegen den entsprechenden Eintrag in dem View SVAEmployeeCount der HIS-Datenbank geprüft.

Listing 6.9: ReadContextWithParams mit IsSVAEmployeeCountType

```
<ReadContextWithParams>
  <rc:ISID>
    <rc:Identifier>svaiddesnutzers</rc:Identifier>
    <rc:ServiceId>1234</rc:ServiceId>
  </rc:ISID>
  <OutputSchema>http://.../KISSPersonType.xsd:IsSVAEmployeeCountType</OutputSchema>
  <Params>
    <KISSPersonServiceReadParam
        xmlns="http://.../KISSPersonService/ReadParamTypes.xsd">
      <DateOfBirth>1973-01-25</DateOfBirth>
    </KISSPersonServiceReadParam>
  </Params>
</ReadContextWithParams>
```

6.1.4 Vodafone Service

Über den Vodafone Service werden die Daten aller Mitarbeiter bezüglich der Vodafone Handys verwaltet. Hierzu greift er auf Relationen der SQL4PUB-Datenbank zurück.

6.1.4.1 Datenbankrelationen

Tabelle 6.2 listet die dem Dienst zugrunde liegenden Relationen auf. Die einzelnen Attribute werden in Tabelle 6.3 näher erläutert.

6.1.4.2 Create

Das Anlegen eines neuen Datenbank-Eintrags geschieht nach der erfolgreichen Auswahl eines oder mehrer Endgeräte im Vodafone-Beantragungsdienst des Mitarbeiterportals (siehe Abschnitt 5.2.7). Hierfür wird der VIPPersonType (siehe Listing 6.10) mit den entsprechenden Werten übergeben. Im XML-Element *Devices* kann es mehrere Einträge geben, je nach Anzahl der georderten Geräte. Das Anlegen geschieht über die Stored Procedure *kim_insert_vipperson* (siehe Anhang A.3)

Relation	Attribute	Attributbeschreibung
users	email	Eindeutige Nutzer-ID des AD
	firstname	Vorname
	lastname	Nachname
	costunit	Kostenstellennummer
	costunitdescription	Kostenstelle
	vodafoneidentifier	Identifikator gegenüber Vodafone
	twincard	Eintrag, falls Nutzer twincard beantragt hat
	duobill	Eintrag, falls Nutzer duobill beantragt hat
simcardnumbers	simcardnumber	SIM-Kartennummer
	phonenumber	zugehörige Telefonnummer
	status	Status der SIM-Karte
	note	Kommentar
phonenumbers	prefix	Vorwahl
	phonenumber	Telefonnummer
	simcardnumber	SIM-Kartennummer
	emailref	Fremdschlüssel users.email
devices	modelid	Identifikator eines Gerätetyps
	description	Gerätebeschreibung
devicestophonenumbers	modelidref	Fremdschlüssel devices.modelid
	phonenumberref	Fremdschlüssel phonenumbers.phonenumber
	dateordered	Zeitpunkt der Bestellung
	dateordered	Zeitpunkt der Geräteübergabe
	deviceserialnumber	Seriennummer des Geräts
	note	Kommentare zum Status des Geräts
delegatees	delegator	Eindeutige Nutzer-ID aus dem AD des Delegierenden
	delegatOrgUnit	Organisatorische Einheit des Delegierenden
	delegatorCostUnit	Kostenstelle des Delegierenden
	delegatee	Delegierter
	hasordered	Status der Bestellung
	twincard	Eintrag, falls Nutzer twincard beantragt hat
	duobill	Eintrag, falls Nutzer duobill beantragt hat
deactivatedusers	email	Eindeutige Nutzer-ID des AD
	dateofdeletion	Zeitpunkt der Löschung

Tabelle 6.3: Attribute der Relationen des Vodafone Service

Listing 6.10: CreateContextWithParams mit VIPPersonType

```
<CreateContextWithParams xmlns="http://.../cruds">
  <ISID>
    <Identifier xmlns="http://www.wsls.net/2004/03/gts/isid">4711</Identifier>
    <ServiceId xmlns="http://www.wsls.net/2004/03/gts/isid">D6...4B</ServiceId>
  </ISID>
  <Params />
  <Protoype>
    <VIPPersonType
        xmlns="http://schemas.kim.uni-karlsruhe.de/.../VIPPersonType.xsd">
      <Email>jan.buck@kit.edu</Email>
      <Firstname>Jan</Firstname>
      <Lastname>Buck</Lastname>
      <CostUnit>123456</CostUnit>
      <CostUnitDescription>RZ</CostUnitDescription>
      <Devices>
        <Device>
          <Model>SonyK800i</Model>
        </Device>
      </Devices>
    </VIPPersonType>
  </Protoype>
</CreateContextWithParams>
```

6.1.4.3 Read - VIPersonType

Es gibt die zwei OutputSchemas *VIPPersonType* und *VIPDelegateeType* für den Vodafone Service. Falls das OutputSchema *VIPPersonType* angefragt wird, nutzt der Vodafone Service die Stored Procedure *sp_select_vipperson* (siehe Anhang A.3), um Daten über diese Person abzufragen. Falls kein Eintrag existiert, wird *null* zurückgegeben.

Listing 6.11: VIPPersonType

```
<xs:schema targetNamespace="http://.../VIPPersonType.xsd" ...>

 <xs:element name="VIPPersonType" >
  <xs:complexType>
   <xs:complexContent>
    <xs:extension base="cont:ContentObject">
     <xs:sequence>
      <xs:element name="Email" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="Firstname" type="xs:string"  maxOccurs="1"/>
      <xs:element name="Lastname" type="xs:string" maxOccurs="1"/>
      <xs:element name="CostUnit" type="xs:string" maxOccurs="1"/>
      <xs:element name="CostUnitDescription" type="xs:string" maxOccurs="1"/>
      <xs:element name="VodafoneIdentifier" type="xs:string" maxOccurs="1"/>
      <xs:element name="Devices" maxOccurs="1">
      <xs:complexType>
       <xs:sequence>
        <xs:element name="Device" minOccurs="1" maxOccurs="unbounded">
         <xs:complexType>
          <xs:sequence>
           <xs:element name="Model" type="ModelType" minOccurs="1" maxOccurs="1"/>
           <xs:element name="Phonenumber" type="xs:string" maxOccurs="1"/>
          </xs:sequence>
         </xs:complexType>
        </xs:element>
       </xs:sequence>
      </xs:complexType>
     </xs:element>
    </xs:sequence>
   </xs:extension>
```

```
  </xs:complexContent>
 </xs:complexType>
 </xs:element>

  <xs:simpleType name="ModelType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="SonyK800i"/>
      <xs:enumeration value="VPAcompactIII"/>
      <xs:enumeration value="VPAcompactV"/>
      <xs:enumeration value="VPAcompactGPS" />
      <xs:enumeration value="MCC" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

6.1.4.4 Read - VIPDelegateeType

Bei der Anfrage nach dem OutputSchema *VIPDelegateeType* werden über die Stored Procedure *kim_select_delegatee* (siehe Anhang A.3) die entsprechenden Daten geholt und eine Instanz von VIPDelegateeType (siehe Listing 6.12) zurückgegeben.

Listing 6.12: VIPDelegateeType

```
<xs:schema targetNamespace="...">

  <xs:import
    namespace="http://www.wsls.net/2002/03/gts/contentObject"
    schemaLocation="../Metadata/ContentObject.xsd"/>

<xs:element name="VIPDelegateeType" >
 <xs:complexType>
  <xs:complexContent>
   <xs:extension base="cont:ContentObject">
    <xs:sequence>
     <xs:element name="IsDelegatee" type="xs:boolean"
                 minOccurs="1" maxOccurs="1"/>
     <xs:element name="canOrder" type="xs:boolean"  maxOccurs="1"/>
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
</xs:element>

</xs:schema>
```

6.1.5 Alias Service

Der Alias Service implementiert die Logik zum Zugriff auf die E-Mail-Alias-Datenbank und wird im KIT-E-Mail-Alias-Dienst des Mitarbeiterportals verwendet (vgl. Abschnitt 5.2.4). Nachfolgend werden die einzelnen Web Service Methoden und ihre Funktionen vorgestellt.

6.1.5.1 Datenbankrelationen

Mit dem E-Mail-Alias-Dienst wurden drei neue Datenbankrelationen eingerichtet. Eine Relation, der 24 Stunden Zwischenspeicher, hält alle selbst erstellen E-Mail-Aliase mit den jeweiligen Einrichtungszeitpunkten vor. Anhand dieser Informationen wird entschieden, ob der Benutzer noch Zeit zur Bearbeitung seines E-Mail-Alias erhält oder ob diese bereits abgelaufen ist. Eine weitere Relation, die Sperrliste, ist eine

Relation/View	Beschreibung
tblAlias	Alias-Relation, 24 Stunden Zwischenspeicher
tblAliasBlackList	Relation mit nicht erwünschten E-Mail-Präfixen
tblAliasRevocationList	Sperrliste: 15 Monate gesperrte Adressen
vAliasBlacklist	View: Zusammenfassung Blacklist und Sperrliste

Tabelle 6.4: Datenbankrelationen des Alias Service

Relation/View	Attribute	Attributbeschreibung
tblAlias	SID	Eindeutige Nutzer-ID des AD
	alias	Aktuell gewählter Alias
	timestamp	Zeitpunkt der Ersteinrichtung
	activated	Aktivierungsstatus (0:not avtivated)
	upn	UserPrincipalName des AD
	email-send	Versandstatus der Aktivierungsmail
tblAliasBlacklist	blacklistPart	Nicht zulässige E-Mail-Präfixe
	timestamp	Anlegezeitpunkt
tblAliasRevocationList	email	Für 15 Monate gesperrte Adressen
	timestamp	Sperrzeitpunkt
vAliasBlacklist	blacklistPart	Nicht zulässige E-Mail-Präfixe

Tabelle 6.5: Attribute der Datenbankrelationen des Alias Service

Einrichtung die sämtliche *kit.edu*-E-Mail-Adressen betrifft. Hier werden nach der Deaktivierung einer E-Mail-Adresse oder eines E-Mail-Alias diese nicht mehr aktiven Adressen eingetragen und 15 Monate vorgehalten. Mit dem Anschluss aller E-Mail-Adressen und -Alias vergebenden Systeme erreicht man, dass eine E-Mail-Adresse innerhalb von 15 Monaten nach der Deaktivierung nicht neu vergeben werden kann. Dies soll Fehlzustellungen verhindern. Eine Blacklist als dritte neue Relation unterbindet das Anlegen von E-Mail-Adressen und -Aliasen, die in Zukunft genutzt werden sollen, bzw. die durch Institutionen des KIT bereits reserviert wurden. Die Relationen und eine View auf eine Zusammenfassung der Eintragungen der Blacklist und Sperrliste sind in der Datenbank `SQL4Alias` abgelegt. Die Datenbanklokation ist `sqlc22.ka.fzk.de\sqlcin22`. Eine Übersicht der Relationen findet sich in Tabelle 6.4. Die in den Relationen abgelegten Attribute zeigt Tabelle 6.5.

Zur Aktivierung der E-Mail-Aliase werden diese in den Exchange Server und damit das Active Directory des KIT übernommen. Hier wird die bereits bestehende Infrastruktur genutzt und das Active Directory Attribut `proxyAddresses` um den E-Mail-Alias erweitert.

6.1.5.2 Methoden

`bool isAvailable(String alias)`
Input: Alias-String
Output: *true* wenn dieser Alias noch frei ist, sonst *false*

`bool setAlias(String upn, String alias)`
Input: UserPrincipalName (UPN), gewählter Alias
Output: *true* wenn Alias im 24 Stunden Zwischenspeicher angelegt wurde, sonst *false*

Beschreibung: Wenn der Benutzer bereits einen Alias hat, wird eine Exception ausgelöst. Falls nicht: Wenn bereits ein Alias im 24 Stunden Zwischenspeicher liegt, wird der Alias dort aktualisiert und *true* zurückgeben. (Zuvor erfolgt eine zusätzliche Überprüfung ob 24 Stunden abgelaufen sind!) Falls nicht: Alias wird in dem 24 Stunden Zwischenspeicher anlegt sowie die aktuelle Zeit in dem timestamp Feld gespeichert und *true* zurückgegeben.

`String[] getMailAddresses(String upn)`
Input: UPN
Output: Alle aktiven E-Mail Adressen des AD Felds `proxyAddresses`

`AliasObject getAlias(String upn)`
Input: UPN
Output: Alias Objekt in welchem der Alias enthalten, der im 24 Stunden Zwischenspeicher (tblAlias) abgelegt ist (Falls dort kein Eintrag vorhanden ist, wird *null* zurück gegeben.). Zusätzlich beinhaltet dieses Objekt den Zeitpunkt der Ersteinrichtung als Timestamp und den UserPrincipalName.

6.1.6 SCC Service

Der SCC Service kapselt Identitätsmanagement-spezifische Aufgaben unter Nutzung des CRUDS-Schnittstelle (vgl. Abschnitt 5.1.2).

6.1.6.1 Read

Um zu überprüfen ob ein Nutzer dem SCC bekannt ist, kann der *ReadContext* verwendet werden, der im Listing 6.13 dargestellt ist. Es wird gegen das OpenLDAP-Verzeichnis des SCC geprüft, indem ein *LDAP-Bind* gegen das Nutzerkonto *uid="Identifikator",ou=people,dc=rz,dc=uni-karlsruhe,dc=de* mit dem übergebenen Passwort durchgeführt wird.

Listing 6.13: ReadContext mit SCCReadParams

```
<ReadContextWithParams>
 <rc:ISID>
  <rc:Identifier>xy0815</rc:Identifier>
  <rc:ServiceId></rc:ServiceId>
 </rc:ISID>
 <OutputSchema xmlns="http://.../SSCUserType.xsd</OutputSchema>
 <Params>
  <SCCReadParams>
   <Password>userPassword</Password>
  </SCCReadParams>
 </Params>
</ReadContextWithParams>
```

Als Ergebnis der Abfrage wird folgendes BusinessObject zurückgeliefert. Das XML-Element *IsSCCUser* hat den Wert *true*, falls der Nutzer im OpenLDAP-Verzeichnis

gefunden wurde. Das Element *SCCEmail* enthält die im SCC gültige E-Mail-Adresse des Nutzers, bspw. xy0815@rz.uni-karlsruhe.de

Listing 6.14: SCCUserType

```
<xs:element name="SCCUserType">
 <xs:complexType>
  <xs:complexContent>
   <xs:extension base="cont:ContentObject">
    <xs:sequence>
     <xs:element name="IsSCCUser" type="xs:boolean" />
     <xs:element name="SCCEmail" type="xs:string" />
    </xs:sequence>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
</xs:element>
```

6.1.6.2 Update

Über den UpdateContext mit dem Parameter SCCServiceUpdateParams (siehe Listing 6.15) lässt sich das im SCC gültige Passwort eines Nutzers ändern. Als Parameter werden das alte und das neue Passwort eines Nutzers übergeben. Das neue Passwort wird mit Hilfe von regulären Ausdrücken auf die geltenden Richtlinien (siehe Abschnitt 5.2.5) geprüft. Nach erfolgreicher Überprüfung wird ein Web Request über das unter *https://www.rz.uni-karlsruhe.de/cgi-bin/bvpasswd* verfügbare CGI-Skript abgesetzt.

Listing 6.15: UpdateContext mit SCCServiceUpdateParams

```
<UpdateContextWithParams xmlns="http://.../cruds" xmlns:rc="http://.../cruds">
 <rc:ISID>
  <rc:Identifier>kj73</rc:Identifier>
  <rc:ServiceId>1234</rc:ServiceId>
 </rc:ISID>
 <Params>
  <SCCServiceUpdateParams xmlns="http://.../SCCServiceUpdateParams.xsd">
   <OldPassword>oldPwd</OldPassword>
   <NewPassword>newPwd</NewPassword>
  </SCCServiceUpdateParams>
 </Params>
</UpdateContextWithParams>
```

6.1.7 SCCIDM Service

Der SCCIDM Service kapselt Identitätsmanagement-spezifische Aufgaben des SCC ohne CRUDS-Schnittstelle. Die beiden implementierten Methoden dienen zur Verwaltung der Nutzungsbedingungen des SCC (vgl. auch Abschnitt 5.3.1).

6.1.7.1 Datenbankrelationen

Die dem SCCIDM Service zugrunde liegende Datenbank ist SQL4PUB mit der Relation SCC Students Database Table (siehe Tabelle 6.6). Das Attribut *acceptedTermsOfUse* ist für den SCCIDM Service wesentlich. Hier wird nach Akzeptieren der Nutzungsbedingungen das aktuelle Datum abgelegt. Solange die Nutzungsbedingungen nicht akzeptiert sind, ist der Wert für dieses Attribut *null*.

Attribut	Beschreibung
firstname	Vorname eines Studierenden
lastname	Nachname eines Studierenden
id_kit	*kit.edu*-E-Mail-Adresse eines Studierenden
id_zuv	Matrikelnummer eines Studierenden
id_fricard	Intern auf der FriCard gespeicherte Nummer
id_fricardchip	Aufgedruckte FriCard-Nummer
id_rz	Benutzerkennung des SCC-Benutzerkontos
sex	Geschlecht eines Studierenden
coursofstudiesnumber	Nummer des Studiengangs
status	Status eines Studierenden (vgl. Abschnitt 4.3.3)
street	Straße der in HIS SOS geführten Adresse
postcode	Postleitzahl der in HIS SOS geführten Adresse
city	Wohnort der in HIS SOS geführten Adresse
addresssupplement	Zusätzliche Angabe zur Adresse
guid	Durch das Identitätsmanagement generierte GUID
acceptedTermsOfUse	Zeitstempel, wann die Nutzungsbedingungen akzeptiert wurden

Tabelle 6.6: Attribute der Relation SCC Students Database Table

6.1.7.2 Methoden

`bool setAcceptedTermsOfUse(string kitIdentifier)`
Input: *kit.edu*-Identifikator
Output: *true*, falls Nutzungsbedingungen erfolgreich gesetzt werden konnten

Beschreibung: Ein Studierender, der im Studierendenportal den Nutzungsbedingungen des SCC zustimmt, erhält über diese Methode einen Eintrag in der Datenbank SCC Students Database, wodurch beim nächsten Synchronisationslauf mit dem SCC-IDM-Systems ein Update des Nutzers durchgeführt wird und ein vollständiges Konto über die Benutzerverwaltungssoftware des SCC angelegt wird. Die Methode gibt *true* zurück, falls der Eintrag erfolgreich vorgenommen werden konnte und *false*, falls ein Fehler aufgetreten ist und kein Eintrag vorgenommen wurde. Die Fehlermeldungen werden in der Datei „sccidmservice.log“ protokolliert. Zum Setzen des entsprechenden Werts in der SCC Students Database wird die Stored Procedure *kim_update_acceptedtermsofuse* aufgerufen. Es wird eine Exception geworfen, falls in der *web.config* nicht die notwendige Konfiguration gefunden wird.

`bool hasAcceptedTermsOfUse(string kitIdentifier)`
Input: *kit.edu*-Identifikator
Output: *true*, falls Nutzungsbedingungen bereits akzeptiert waren

Beschreibung: Über diese Methode kann abgefragt werden, ob ein Studierender den Nutzungsbedingungen des SCC über das Studierendenportal zugestimmt hat. Die

Methode gibt *true* zurück, falls der Eintrag erfolgreich gefunden werden konnte und *false*, falls ein Fehler aufgetreten ist oder kein Eintrag gefunden wurde. Die Fehlermeldungen werden in der Datei „sccidmservice.log“ protokolliert. Zum Auslesen des entsprechenden Werts aus der SCC Students Database Table wird die Stored Procedure *kim_select_acceptedtermsofuse* aufgerufen. Es wird eine Exception geworfen, falls in der *web.config* nicht die notwendige Konfiguration gefunden wird.

6.1.8 SPML Service

Der in Java umgesetzte Web Service dient zur Kapselung der SPML-Schnittstelle des Sun Identity Manager, da bisher noch keine Implementierung von SPML für .Net vorliegt. Der Service implementiert hierfür die beiden Methoden `updateRZIDentifier` und `updatePassword`.

6.1.8.1 Methoden

`String updatePassword(String user, String newPassword)`
Input: Identifikator des Nutzers entspricht dem sAMAccountName, neues Passwort
Output: Es wird ein ErrorCode zurückgegeben

Beschreibung: Die Methode validiert zunächst das Passwort mittels regulärer Ausdrücke auf die geltenden Richtlinien (siehe Abschnitt 5.2.5). Anschließend wird das neue Passwort mittels eines *ExtendedRequest* mit dem Operator *changeUserPassword* gesetzt. Das Attribut isactivated wird auf *true* gesetzt, wodurch dieser Eintrag im KISS-Repository aktualisiert wird und der Nutzer den ersten Schritt der Aktivierung durchgeführt hat (siehe Abschnitt 5.2.2). Das Attribut *expirePassword* wird auf *false* gesetzt, ansonsten wäre das Passwort im Active Directory nicht aktiviert.

```
String updateRzid(String rzid, String kitEmail,
                  String user, bool isRZnew)
```
Input: SCC-Identifikator, *kit.edu*-E-Mail-Adresse, Identifikator des Nutzers ist der sAMAccountName, neuer SCC-Account
Output: Es wird ein ErrorCode zurückgegeben

Beschreibung: Für den Nutzer werden die Werte localidentifier2, rzemail, kitemail und syncstatus, isrznew entsprechend den Eingabeparametern über eine Modification-Operation geändert. Vorher wird die *kit.edu*-E-Mail-Adresse auf einen leeren Wert gesetzt, damit in der UpdateAfterAction diese Änderung sichtbar wird. In der dem SPML-Nutzer zugewiesenen *KIM SPML User Form* wird zusätzlich noch für die Ressource *kit.edu*-Active Directory UKA Node (siehe Abschnitt 4.2.4) die UpdateAfterAction *KIM AD Update After Action* aufgerufen. Diese löst über ein Powershell-Skript (siehe Anhang A.4) das Anlegen eines Exchange-Kontos mit einer Weiterleitung auf den SCC-Account aus.

6.2 Single Sign-On mit Shibboleth

Als zentrales Authentifikationssystem für browserbasierte Webdienste hat sich, vor Allem im Bereich der Forschung und Lehre, der Standard Shibboleth und das zugehörige Softwareprodukt der „Internet2 Middleware Initiative“ etabliert. Das IDM-Team hat prototypisch einen zum Einsatz von Shibboleth erforderlichen *Identity Provider* aufgesetzt. Dieser soll im Laufe des Jahres 2009 weiter auf die KIT-spezifischen Anforderungen angepasst und in den Betrieb des SCC eingebunden werden. Mit diesem Identity Provider können Organisationseinheiten KIT-Mitarbeiter und -Studierende für Webdienste authentifizieren und zusätzliche Informationen über diese, in Form von Attributen, geliefert bekommen.

Durch eine Integration der Shibboleth-Infrastruktur des KIT in die Föderation des Deutschen Forschungsnetzes (DFN-AAI), kann die Authentifikation für einzelne Webdienste auch auf Mitglieder anderer nationaler Bildungseinrichtungen ausgeweitet werden. Ferner ermöglicht es die Integration Mitglieder des KIT für Dienste dritter Einrichtungen zu authentifizieren. Hier sei als Beispiel der Dienst „Regionale Datenbank-Information Baden-Württemberg“ (ReDI) angeführt, der zukünftig von Mitgliedern des KIT genutzt werden kann. Die Möglichkeit einer Benutzerauthentifikation bedarf einer Freischaltung des Webdienstes innerhalb des KIT Identity Providers. Eine optionale Lieferung von Benutzerattributen an angeschlossene Webdienste erfordert jeweils eine vorangestellte datenschutzrechtliche Überprüfung und eine entsprechende Konfiguration des KIT Identity Provider auf Seiten des SCC.

6.2.1 Kurzbeschreibung von Shibboleth

Shibboleth ist ein auf Standards basierendes Softwarepaket zur Realisierung von Web Single Sign-On innerhalb von Organisationen sowie über deren Grenzen hinweg. Es erlaubt, dass Webseiten über Authentifikationsentscheidungen informiert werden, so dass individueller Zugang zu geschützten Online-Ressourcen erteilt werden kann [WWW Shibboleth 2009]. Damit muss ein Dienstanbieter lediglich die Autorisation eines Anwenders auf Basis der ausgelagerten Authentifikation und der Attribute durchführen, die vom Identity Provider zur Verfügung gestellt wurden. Shibboleth basiert auf der Security Assertion Markup Language (SAML) [SAML V2.0 2005]. Das SAML-Protokoll ist ein XML-basiertes Request-Response-Protokoll zum Austausch von Authentifikations- und Autorisationsdaten zwischen Sicherheitsdomänen [Djordjevic & Dimitrakos 2005]. Die Spezifikation weißt im Wesentlichen die drei Teilbereiche Assertions, Protocols und Bindings auf.

Die SAML-Assertions bilden den Kern der Spezifikation und liefern Informationen zur Authentifikation, Autorisation, sowie weiterer Session-Attribute. Nur autorisierte Komponenten dürfen Assertions ausstellen, sodass den Informationen eines SAML-Nachrichtenteils vertraut werden kann. Assertions tragen die SAML Versionsnummer (Version), eine „AssertionID“ (eindeutiger Identifikator der Assertion), den Zeitpunkt der Ausstellung (IssueInstant), den Identifikator des Ausstellers (Issuer), ein Subject und optional Bedingungen zur Gültigkeit (Conditions), weitere Hinweise (Advice) und eine Signatur.

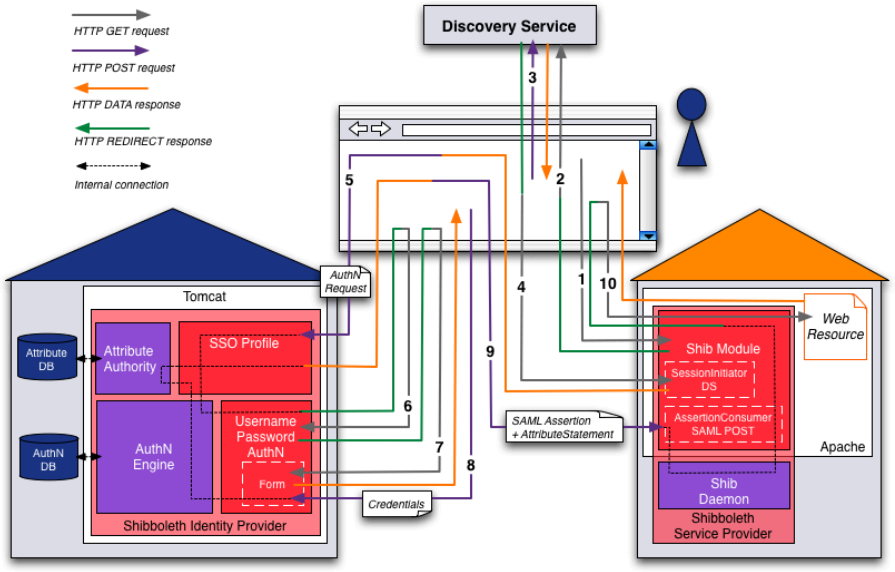


Abbildung 6.1: Typischer Ablauf Authentifikation mittels einer Shibboleth-Infrastruktur (Quelle: http://www.switch.ch/)

Das SAML-Protocol definiert über ein XML-Schema die Kommunikation zwischen SAML-Requestor und -Responder. Auf eine Anfrage wird eine Antwort geschickt, die je nach Anfrage eine oder mehrere Assertions enthält.

Die SAML-Bindings und -Profiles legen fest, wie Teile von SAML in andere Dokumente eingebunden werden können und entsprechende Rahmen aus anderen Spezifikationen zu erweitern sind [Dostal et al. 2005, S. 189 ff]. In SAML 2.0 sind, als Ergänzung der SAML-Profiles hauptsächlich Shibboleth 1.3 und ID-FF der Liberty Alliance eingeflossen.

Abbildung 6.1 zeigt den Ablauf einer Nutzerauthentifikation an einem Shibboleth-Identity Provider und die zugehörige Attributlieferung an einen Service Provider nach der Shibboleth Protokollversion 2.x. Im ersten Schritt versucht der Benutzer auf die geschützte Web Ressource zuzugreifen. Der Ressource ist das Shibboleth Service Provider Modul vorgeschaltet. Dieses überprüft zunächst, ob der Benutzer ein gültiges *Cookie* besitzt, dass eine vorangegangene Authentifikation bei einem für diesen Service Provider zugelassenen Identity Provider belegt. Ist dies nicht der Fall, wird der Benutzer im Standardfall zu einem Discovery Service weitergeleitet. Dieser Service bietet eine Auswahl an Identity Providern diverser Einrichtungen. Der Benutzer wählt hier die Einrichtung aus, bei der er mit einem Account erfasst ist und bestätigt. Als Nächstes bekommt der Benutzer die Anmeldeseite seines ausgewählten Identity Providers angezeigt. Hier gibt er seine Zugangsdaten ein, der Identity Provider verifiziert diese und leitet den Benutzer nach erfolgreicher Authentifikation zurück zum Service Provider. Dieser entscheidet anhand der vom Identity Provider gelieferten Informationen abschließend über den Zugriff auf die Web Ressource (Autorisation). Die Informationen enthalten die Bestätigung der gelungenen Authentifikation und optional vom Service Provider angeforderte Attribute. Versucht der

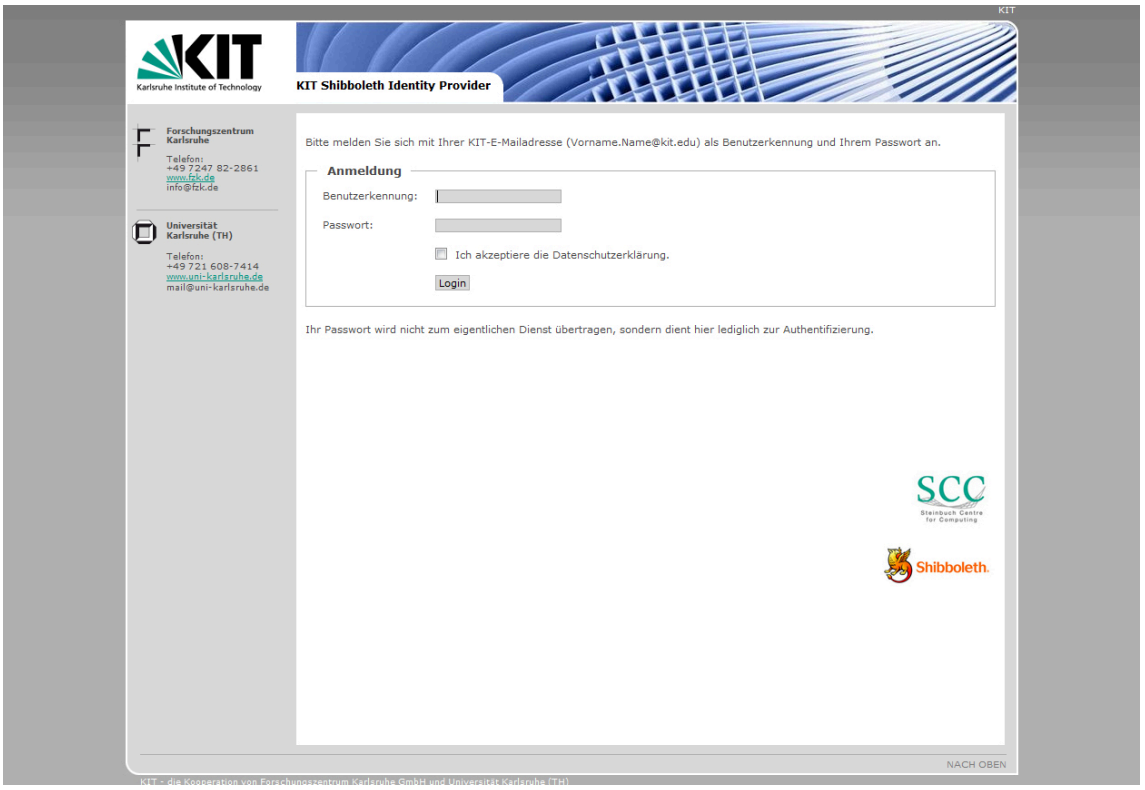


Abbildung 6.2: Login-Maske des Shibboleth Identity Provider am KIT

Benutzer auf eine weitere Web Ressource zuzugreifen, der ebenfalls ein Shibboleth Service Provider vorgeschaltet ist und die Authentifikationen des Identity Providers zulässt, bei dem sich der Benutzer bereits eingeloggt hat, kann die Web Ressource ohne zusätzlichen Login-Vorgang genutzt werden (Single Sign-On).

6.2.2 KIT Identity Provider

Abbildung 6.2 zeigt die Login-Maske des prototypischen KIT Identity Providers. Dieser prüft bei einem Login-Versuch den Benutzernamen und das Passwort gegen das KIT-Active Directory und falls dort der Benutzer nicht zu finden ist in weiteren angeschlossenen Identitätsdatenbanken, wie in Abbildung 6.3 ersichtlich ist. Als Benutzername und Passwort dienen analog zum Mitarbeiterportal bzw. Studierendenportal die *kit.edu*-E-Mail-Adresse und das zugehörige Passwort. Sollte ein Nutzer keine *kit.edu*-E-Mail-Adresse haben, soll z.B. auch ein Login mit den Benutzerdaten der Universitätsbibliothek ermöglicht werden. Loggt sich der Benutzer zum ersten Mal ein, muss er eine Einverständniserklärung zur Datenprovisionierung bestätigen. Daraufhin werden über den SUN Identity Manager die notwendigen Attribute in das *Shibboleth Attribute Repository* provisioniert. Aus diesem können dann Shibboleth Service Provider mit Attributen versorgt werden.

Abbildung 6.3 zeigt die geplante Shibboleth-Infrastruktur am KIT und die Integration in die DFN-Föderation. Links im Bild ist der KIT Identity Provider mit dessen Attributdatenbank skizziert. Dem Identity Provider soll ein Authentifikationsmodul integriert werden, das individuell für die KIT-spezifischen Anforderungen entwickelt

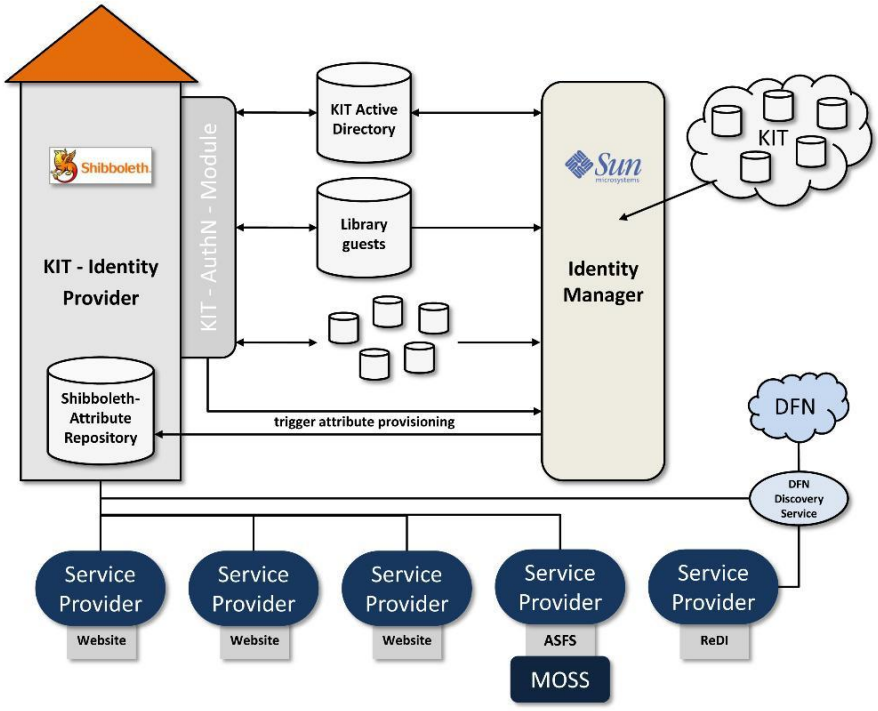


Abbildung 6.3: Architekturkonzept - Shibboleth Identity Provider am KIT

wurde. Shibboleth kann zur Authentifikation mit Benutzername und Passwort durch die mitgelieferten Module lediglich eine einzelne Accountdatenbank anbinden. Das *KIT AuthN Module* hingegen bietet die Möglichkeit mehrere Datenquellen an einen Identity Provider anzubinden. Durch dieses Konzept haben alle Shibboleth-fähigen Dienste am KIT und gegenüber dem DFN einen dedizierten Zugangspunkt. Die DFN-AAI spannt eine Föderation auf, indem Nutzer von DFN Diensten eine Auswahl aller integrierten Identity Provider der einzelnen Hochschulen zur Auswahl gegeben wird. Shibboleth selbst sieht keine Hierarchie von Föderationen vor. Müsste demnach für jede Datenquelle ein Identity Provider aufgesetzt werden, wären auch in der Identity Provider Auswahl der DFN-AAI mehrere KIT Identity Provider zu verzeichnen. Die DFN-AAI lässt jedoch pro Einrichtung nur einen Identity Provider zu, um die Auswahlliste möglichst klein zu halten. Aus diesem Grund wurde entschieden ein eigenes Authentifikationsmodul zu implementieren, mit dem alle vorhandenen Accountdatenbanken an einen Shibboleth Identity Provider angebunden werden können. Solche Module existieren bereits für Shibboleth-Infrastrukturen anderer Einrichtungen. Da die KIT-spezifischen Anforderungen von den betrachteten Modulen anderer Einrichtungen jedoch nicht abgedeckt werden, soll hier auf dem Wissen der Shibboleth Entwicklergemeinschaft aufgebaut und ein eigenes Modul für das KIT entwickelt werden.

Eine weitere Kernkomponente der Shibboleth Infrastruktur ist der SUN Identity Manager (rechts im Bild). Diese Komponente provisioniert unter anderem die einzelnen Accountdatenbanken des KIT oder liest aus diesen die notwendigen Informationen zur Diensterbringung. Beim ersten Login eines Nutzers spricht das Shibboleth Authentifikationsmodul des KIT den SUN Identity Manager über einen „Trigger“ an,

um die für die Erbringung der Shibboleth-Dienste notwendigen Attribute in die Attribut-Datenbank des KIT Identity Providers schreiben zu lassen. Nach erfolgreicher Attributprovisionierung können den Shibboleth-Diensten die angeforderten Attribute der Nutzer vom Identity Provider zur Verfügung gestellt werden.

Im unteren Teil der Abbildung 6.3 sind unterschiedliche Shibboleth-Service Provider verzeichnet, die verschiedene Web Dienste kapseln, wie z.B. einen Microsoft Office Sharepoint Server oder sonstige browserbasierte Webdienste. Dienste die innerhalb des KIT und nur für Personen, die am KIT einen Nutzeraccount haben, angeboten werden sollen, können direkt den KIT Identity Provider zur Authentifikation nutzen. Externe Dienste oder solche die auch externen Nutzern zur Verfügung gestellt werden sollen, können in die DFN-Föderation integriert werden. Im Bild ist hier exemplarisch der dort bereits integrierte ReDI Dienst verzeichnet, dessen Nutzer zunächst zum Discovery Service der DFN-AAI, der Identity Provider Auswahl, geleitet werden. Von hier aus kann dann der KIT Identity Provider ausgewählt werden.

6.2.3 Anbindung von Service Providern

Shibboleth lässt die Anbindung verschiedenster Web Ressourcen zu. So können Internet-Foren, Webseiten oder Portale geschützt werden. Um die Nutzer des Mitarbeiter- und Studierendenportal mit Shibboleth zu authentifizieren, wurde ein Kontakt zum Softwareanbieter 9StarResearch aufgebaut. Die texanische Firma bietet ein Produkt zur Anbindung von Microsoft Office Sharepoint-Portalen an Shibboleth Infrastrukturen an. Die Software ActiveShareFS (ASFS) der Firma 9StarResearch wurde evaluiert und für einsetzbar eingestuft. Die Integration der Sharepoint Server in die Shibboleth-Infrastruktur ist nach Abschluss des Infrastruktur-Aufbaus geplant.

Um Institutionen außerhalb des KIT die Möglichkeit zu geben, Studierenden und Mitarbeitern des KIT Zugang zu Web Ressourcen zu geben, soll der KIT Identity Provider an der Föderation des Deutschen Forschungsnetzes (DFN) teilnehmen. Damit wird es zusätzlich möglich sein, dass Mitglieder des KIT auf Webinhalte diverser externer Institution zugreifen können, wie etwa auf das Bibliothekssystem „Regionale Datenbank-Information Baden-Württemberg“ (ReDI) der Universität Freiburg. Als erster Schritt wurde bereits eine Integration in das Testsystem der DFN-AAI arrangiert und diverse Tests durchgeführt.

In Abschnitt 6.2.1 wurde kurz auf die Rolle und Funktion eines Discovery Service eingegangen. Bei der internen Verwendung des Shibboleth Identity Provider (IdP) ist zunächst kein Discovery Service notwendig, da das Architekturkonzept lediglich einen IdP für das KIT vorsieht. Dieser soll aus Gründen der Verfügbarkeit im produktiven Betrieb redundant ausgelegt werden und alle Mitglieder des KIT in einer Authentifikationskomponente vereinen. Möchten KIT-Mitarbeiter oder -Studierende Dienste des DFN oder daran angeschlossene Dienste nutzen, werden sie an den DFN Discovery Service geleitet. Hier ist die Auswahl des KIT Identity Provider möglich und kann zur Authentifikation und gegebenenfalls zur Attributlieferung genutzt werden.

6.2.4 Shibboleth Datenquellen

Shibboleth trennt auf Ebene der Konfiguration die Authentifikation und die Lieferung von Attributen. Für beide Module ist die Angabe mindestens einer Datenquelle notwendig. Als Referenzquelle für die Authentifikation wurde das KIT Active Directory eingesetzt, das über das KIT-Identitätsmanagement provisioniert wird. Durch das „KIT-AuthN Module“ wird ebenfalls die Anbindung weiterer Accountdatenbanken möglich sein. Das Modul entscheidet anhand des eingegebenen Nutzernamens gegen welche Datenquelle ein Anwender authentifiziert werden muss.

Der Attributlieferung kann ebenfalls das Active Directory oder eine gesonderte Datenquelle zugrunde gelegt werden. Nutzt man das bestehende Active Directory, ist ein Mapping des verwendeten Schemas auf das von Shibboleth genutzte Schema *eduPerson*notwendig. Vorteil dabei ist, dass man kein neues Schema einführen, sondern lediglich das Mapping im Identity Provider pflegen muss. Generell ist die Nutzung vorhandener Datenquellen zu bevorzugen und eine Schemakonvertierung zu vermeiden. Gründe dafür liegen auf Seiten der Service Provider, die ihre Services auf die Attribute, beziehungsweise auf deren unterstütztes Schema konfigurieren müssen, um Autorisationsentscheidungen auf Basis der Attribute treffen zu können. Möchte ein Service Provider die Art der Authentifikation und Attributlieferung ändern, so ist auch eine Anpassung auf das sich gegebenenfalls ändernde Schema notwendig. Wird organisationsintern nur ein Schema verwendet, ist ein zusätzlicher Aufwand beim Wechsel des Zugriffssystems obsolet.

Verschiedene Service Provider, sowie übergeordnete Föderationen, wie die DFN AAI, verlangen in der Kombination mit einer Shibboleth-Infrastruktur das Schema *eduPerson*. Da innerhalb der für Shibboleth autoritativen Quelle, dem KIT Active Directory, dieses Schema keine Verwendung findet, ist zur Nutzung dieser Service Provider eine Schema-Anpassung oder ein Attribut-Mapping unabdingbar.

Shibboleth unterstützt in der Standardimplementierung die Anbindung einer Datenquelle zur Bereitstellung von Attributen. Kommen im Laufe der Zeit Service Provider hinzu, die durch die Shibboleth Infrastruktur des KIT versorgt werden sollen, aber Attribute benötigen, die nicht in der Attributquelle (z.B. im Active Directory) vorhanden sind, stehen mehrere Möglichkeiten zur Auswahl. Zum einen könnte man diese Attribute durch das Identitätsmanagement in die Datenquelle provisionieren lassen. Eine weitere Möglichkeit wäre die Erweiterung des KIT Identity Provider, um mehrere Datenquellen als Attributquellen zu aggregieren. Aufgrund des zu erwartenden Mehraufwands für letztere Möglichkeit, wurde für den prototypisch aufgesetzten Identity Provider eine eigens angelegte Attributquelle definiert, die mit allen für die angeschlossenen Service Provider notwendigen Attribute durch das Identitätsmanagement versorgt wird. Diese Attributquelle ist in der Abbildung 6.3 als „Shibboleth Attribute Repository“ verzeichnet. Für die Authentifikation dagegen kommt diese Lösung nicht zum Einsatz, da vermieden werden soll, dass Passwörter außer im Active Directory oder in den dafür vorgesehenen Datenbanken in zusätzlichen Datenquellen abgelegt werden.

A. Skripte und Stored Procedures

A.1 KISS-Repository

Listing A.1: kim_select_studenttarget.sql

```
USE [kissrep]
GO


SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[kim_select_studenttarget]
(
        @source_localidentifier varchar(255),
        @sourceid varchar(255),
        @targetid varchar(255)
)

AS

DECLARE @targetcolumnname VarChar(255)
DECLARE @sourcecolumnname VarChar(255)
DECLARE @sql nvarchar(4000)

BEGIN

SET NOCOUNT ON;

SET @sql = 'SELECT '+ CASE @targetid
WHEN 'KIT' THEN 'id_kit AS KIT'
WHEN 'RZ' THEN 'id_rz AS RZ '
WHEN 'ATIS' THEN 'id_atis AS ATIS'
WHEN 'UB' THEN 'id_fricard AS UB'
WHEN 'ZUV' THEN 'id_zuv AS ZUV'
END+' FROM studentidentifiermapping WHERE '+CASE @sourceid
WHEN 'KIT' THEN 'id_kit'
WHEN 'RZ' THEN 'id_rz'
WHEN 'ATIS' THEN 'id_atis'
WHEN 'UB' THEN 'id_fricard'
WHEN 'ZUV' THEN 'id_zuv'
END+' = '+quotename(@source_localidentifier,'''')
```

```
EXEC sp_executesql @sql

END
```

Listing A.2: kim_select_employeetarget.sql

```
USE [kissrep]
GO


SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[kim_select_employeetarget]
(
        @source_localidentifier varchar(255),
        @sourceid varchar(255),
        @targetid varchar(255)
)

AS


DECLARE @targetcolumnname VarChar(255)
DECLARE @sourcecolumnname VarChar(255)
DECLARE @sql nvarchar(4000)

BEGIN

SET NOCOUNT ON;

SET @sql = 'SELECT '+ CASE @targetid
WHEN 'KIT' THEN 'email AS KIT'
WHEN 'RZ' THEN 'localidentifier2 AS RZ '
WHEN 'KISS' THEN 'username AS KISS'
WHEN 'FZK' THEN 'localidentifier3 AS FZK'
WHEN 'ZUV' THEN 'localidentifier1 AS ZUV'
END+' FROM accounts WHERE '+CASE @sourceid
WHEN 'KIT' THEN 'email'
WHEN 'KISS' THEN 'username'
WHEN 'RZ' THEN 'localidentifier2'
WHEN 'FZK' THEN 'localidentifier3'
WHEN 'ZUV' THEN 'localidentifier1'
END+' = '+quotename(@source_localidentifier,'''')

EXEC sp_executesql @sql


END
GO
```

Listing A.3: kim_select_samaccountname.sql

```
USE [kissrep]
GO


SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE PROCEDURE [dbo].[kim_select_samaccountname]
(
        @email varchar(255)
)

AS

DECLARE @sql nvarchar(4000)

BEGIN

SET NOCOUNT ON;

SET @sql = 'SELECT username FROM accounts WHERE email = '+quotename(@email,'''')

EXEC sp_executesql @sql


END
```

Listing A.4: kim_select_roleprovider.sql

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[kim_select_roleprovider]
(
        @email varchar(255)
)

AS

DECLARE @sql nvarchar(4000)

BEGIN

SET NOCOUNT ON;

SET @sql = 'SELECT email, firstname, lastname, isactivated, status, rzemail,
rzoldpassword, username, localidentifier1, localidentifier2, localidentifier3,
svarole
FROM ukascientists
RIGHT OUTER JOIN accounts
ON ukascientists.svaid = accounts.localidentifier1
COLLATE Latin1_General_CI_AS WHERE email = '+quotename(@email,'''')

EXEC sp_executesql @sql


END
GO
```

Listing A.5: kim_select_kissrepositoryservice.sql

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE PROCEDURE [dbo].[kim_select_kissrepositoryservice]
(
        @email varchar(255)
)

AS

DECLARE @sql nvarchar(4000)

BEGIN

SET NOCOUNT ON;

SET @sql = 'SELECT email, firstname, lastname, isactivated, status, rzemail,
rzoldpassword, username, localidentifier1, localidentifier2, localidentifier3
FROM accounts WHERE email = '+quotename(@email,'''')

EXEC sp_executesql @sql


END
GO
```

A.2 SCC Students Database Table

Listing A.6: kim_select_acceptedtermsofuse.sql

```
USE [idm_sync_prod]
GO



SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[kim_select_acceptedtermsofuse]
(
        @identifier varchar(255)
)
AS

DECLARE @sql nvarchar(400)
DECLARE @count int

BEGIN

        SET NOCOUNT ON;

        SET @sql = 'SELECT @count=count(1) FROM studierende
        WHERE id_kit = '+quotename(@identifier,'''')
        EXEC sp_executesql @sql, N'@count int OUT', @count OUT

        IF @count <> 1
        BEGIN
        RAISERROR('0: user not found. specify the valid identifier of a student.'
        ,16,1)
        RETURN
        END

        SET @sql = 'SELECT @count=count(1) FROM studierende
        WHERE (NOT (acceptedtermsofuse IS NULL))
        AND id_kit = '+quotename(@identifier,'''')
        EXEC sp_executesql @sql, N'@count int OUT', @count OUT
```

```
        IF @count <> 1
        BEGIN
        RETURN '0'
        END

        RETURN '1'

END
```

Listing A.7: kim_update_acceptedtermsofuse.sql

```
USE [idm_sync_prod]
GO



SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[kim_update_acceptedtermsofuse]
(
        @identifier varchar(255),
        @matriculationnumber varchar(255)
)
AS

DECLARE @sql nvarchar(400)
DECLARE @countkit int
DECLARE @countmatriculationnumber int

BEGIN

        SET NOCOUNT ON;

        BEGIN TRANSACTION

        SET @sql = 'SELECT @countkit=count(1) FROM studierende
  WHERE id_kit = '+quotename(@identifier,'''')
        EXEC sp_executesql @sql, N'@countkit int OUT', @countkit OUT

        IF @countkit = 0
        BEGIN
        ROLLBACK
        RAISERROR('0: user not found. Wrong kit.edu-E-Mail.
  Specify the valid identifier of a student.',16,1)
        RETURN
        END

        IF @countkit <> 1
        BEGIN
        ROLLBACK
        RAISERROR('1: Too many users with same email found.
  Check data in databasetable studierende.',16,1)
        RETURN
        END

        SET @sql = 'SELECT @countmatriculationnumber=count(1)
        FROM studierende WHERE id_kit = '+quotename(@identifier,'''')+'
        AND id_zuv = '+quotename(@matriculationnumber,'''')
        EXEC sp_executesql @sql,
  N'@countmatriculationnumber int OUT', @countmatriculationnumber OUT

        IF @countmatriculationnumber <> 1
        BEGIN
```

```
        ROLLBACK
        RAISERROR('2: user not found. Wrong Matriculation Number.
  Specify the valid identifier of a student.',16,1)
        RETURN
        END

        SET @sql = 'UPDATE studierende SET acceptedtermsofuse = GetDate()
        WHERE id_kit = '+quotename(@identifier,'''')
        EXEC sp_executesql @sql

        IF @@ERROR <> 0
        BEGIN
        ROLLBACK
        RAISERROR('3: update error. user was found but update was not successful.'
        ,16,1)
        RETURN
        END

        COMMIT

END
```

A.3 Vodafone

Listing A.8: kim_insert_vipperson.sql

```
USE [Vodafone]
GO


SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[kim_insert_vipperson]

@email VarChar(255),
@firstname VarChar(255),
@lastname VarChar(255),
@costunit VarChar(255),
@costunitdescription VarChar(255),
@modelid VarChar(255),

@prefix VarChar(255),
@startrangemcc VarChar(255),

@dateordered VarChar(255)

AS

DECLARE @counter int
DECLARE @maxvodafoneidentifier int
DECLARE @isdelegate int
DECLARE @phonenumber VarChar(255)
DECLARE @simcardnumber VarChar(255)
DECLARE @alreadycreated int

BEGIN


        SET NOCOUNT ON;

BEGIN TRANSACTION
```

```
        SELECT @alreadycreated=count(1) FROM users WHERE email = @email

        IF (SELECT count(1) FROM delegatees WHERE delegatee = @email) > 0
        BEGIN
                UPDATE delegatees
                SET hasordered = '1'
                WHERE delegatee = @email
        END

        SELECT @maxvodafoneidentifier=max(vodafoneidentifier) FROM users
        SET @maxvodafoneidentifier = @maxvodafoneidentifier+1

        IF (SELECT count(1) FROM devicestophonenumber
  JOIN phonenumbers
  ON devicestophonenumber.phonenumberref = phonenumbers.phonenumber
  AND emailref=@email) = 0

  BEGIN
                IF @maxvodafoneidentifier IS NULL
                BEGIN
                        SET @maxvodafoneidentifier = 1
                END
                INSERT users(firstname,lastname,costunit,
    costunitdescription,vodafoneidentifier,email)
                VALUES (@firstname,@lastname,@costunit,
    @costunitdescription,@maxvodafoneidentifier,@email)
        END

IF @@ERROR <> 0
BEGIN
ROLLBACK
RAISERROR('10: User already in Database',16,1)
RETURN
END

        IF @modelid LIKE 'MCC%'
        BEGIN

                IF (SELECT count(1) FROM devicestophonenumber
    JOIN phonenumbers
    ON devicestophonenumber.phonenumberref = phonenumbers.phonenumber
    AND modelidref
    LIKE 'MCC%' AND emailref=@email) > 0
                BEGIN
                        ROLLBACK
                        RAISERROR('11: User has already MCC',16,1)
                        RETURN
                END

        END

        IF NOT @modelid LIKE 'MCC%'
        BEGIN
                IF (SELECT count(1) FROM devicestophonenumber
    JOIN phonenumbers
    ON devicestophonenumber.phonenumberref = phonenumbers.phonenumber
    AND NOT modelidref LIKE 'MCC%' AND emailref=@email) > 0
                BEGIN
                        ROLLBACK
                        RAISERROR('12: User has already cell phone',16,1)
                        RETURN
                END

        END

        SELECT      @phonenumber=MIN(simcardnumbers.phonenumber)
        FROM          simcardnumbers LEFT OUTER JOIN
```

```
                                                    phonenumbers
  ON simcardnumbers.phonenumber = phonenumbers.phonenumber
        WHERE       (phonenumbers.phonenumber IS NULL)

IF @phonenumber = 0
BEGIN
ROLLBACK
RAISERROR('20: No phonenumber available',16,1)
RETURN
END

        SELECT @simcardnumber=simcardnumbers.simcardnumber
  FROM simcardnumbers
  WHERE simcardnumbers.phonenumber = @phonenumber

        INSERT phonenumbers(phonenumber,prefix,simcardnumber,emailref)
        VALUES (@phonenumber,@prefix,@simcardnumber,@email)

IF @@ERROR <> 0
BEGIN
ROLLBACK
RAISERROR('30: Phonenumber already in Database',16,1)
RETURN
END

        INSERT devicestophonenumber(modelidref,phonenumberref,dateordered)
        VALUES (@modelid,@phonenumber,@dateordered)

IF @@ERROR <> 0
BEGIN
ROLLBACK
RAISERROR('40: DeviceMapping already in Database: ',16,1)
RETURN
END

COMMIT

END
```

Listing A.9: sp_select_vipperson.sql

```
USE [Vodafone]
GO



SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[sp_select_vipperson]
        (
        @email varchar(255)
        )

AS
        SELECT users.email, users.firstname, users.lastname, users.costunit,
        users.costunitdescription,users.vodafoneidentifier,
        phonenumbers.phonenumber,phonenumbers.prefix, phonenumbers.simcardnumber,
        devicestophonenumber.modelidref
        FROM devicestophonenumber, phonenumbers,users
        WHERE users.email = @email
        AND devicestophonenumber.phonenumberref = phonenumbers.phonenumber
                AND phonenumbers.emailref = users.email
RETURN
```

Listing A.10: kim_select_delegatee.sql

```
USE [Vodafone]
GO



SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO



ALTER PROCEDURE [dbo].[kim_select_delegatee]
        (
        @email varchar(255)
        )

AS
        SELECT hasordered
        FROM delegatees
        WHERE delegatee = @email
RETURN
```

A.4 Exchange

Listing A.11: Powershell zum Anlegen einer Exchange-Mailbox und Weiterleitung

```
@echo off
rem     Create Mailbox

rem     Parameter: Internal_Email, External_Email


set Email=%1
set Email_localPart=%Email:@kit.edu=%

set Database="KIT-MSX-01"
set OU="Adatum.com/Contacts"
set DeliverAndForward=$False

set Identity=%1
set Alias=%Email_localPart%
set ExternalEmailAddress=SMTP:%2
set Contact_DisplayName=%2
set Contact_Name=%2
set Contact_Alias=Contact-%Email_localPart%
set ForwardingAddress=%2


"C:\Program Files\Microsoft Command Shell\v1.0\Msh.exe"
-mshconsolefile
"C:\Program Files\Microsoft\Exchange Server\bin\exshell.mcf1"
-command
"Enable-Mailbox -Identity:%Identity%
-Alias:%Alias% -Database:%Database%"


"C:\Program Files\Microsoft Command Shell\v1.0\Msh.exe"
-mshconsolefile
"C:\Program Files\Microsoft\Exchange Server\bin\exshell.mcf1"
-command
"New-MailContact -ExternalEmailAddress:%ExternalEmailAddress%
-Name:%Contact_Name% -Alias:%Contact_Alias%
-OrganizationalUnit:%OU% -DisplayName:%Contact_DisplayName%"
```

```
"C:\Program Files\Microsoft Command Shell\v1.0\Msh.exe"
-mshconsolefile
"C:\Program Files\Microsoft\Exchange Server\bin\exshell.mcf1"
-command
"Set-Mailbox -Identity:%Identity%
-DeliverToMailboxAndForward:%DeliverAndForward%
-ForwardingAddress:%ForwardingAddress%"
```

B. Prozesse

B.1 Provisionierung der Mitarbeiter am Campus Süd

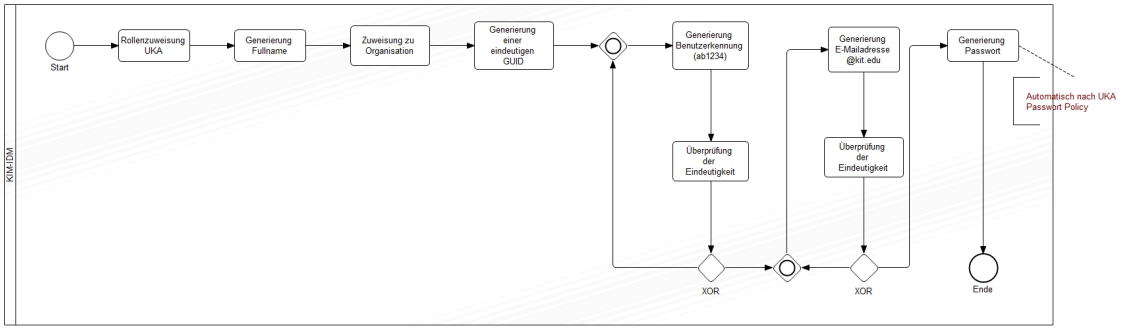


Abbildung B.1: Anlegen eines Mitarbeiters am Campus Süd

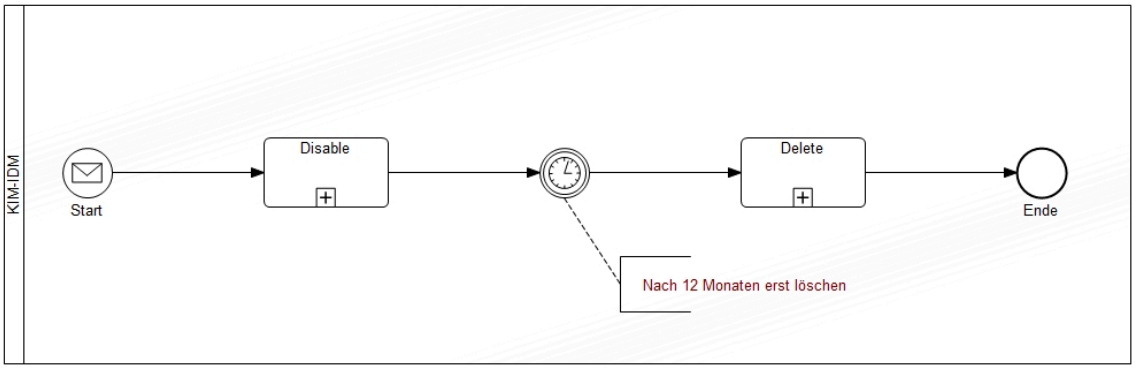


Abbildung B.2: Deprovisionierung eines Mitarbeiters am Campus Süd

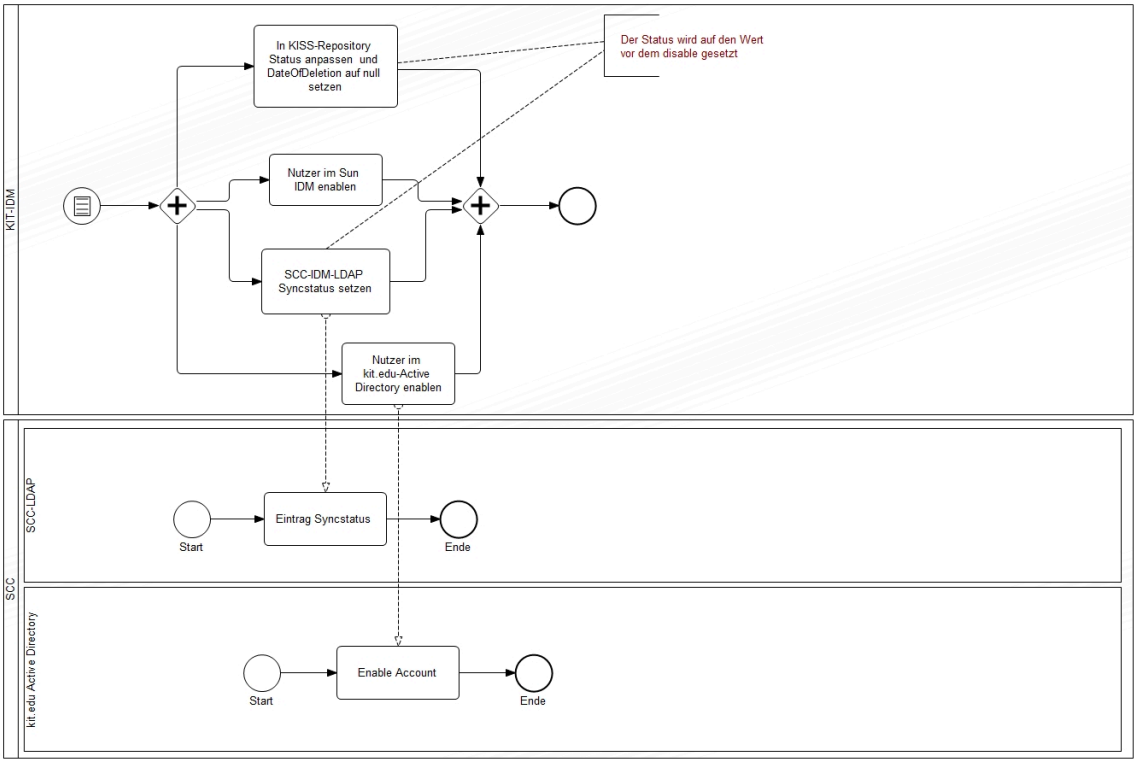


Abbildung B.3: Aktivierung eines Mitarbeiterbenutzerkontos am Campus Süd

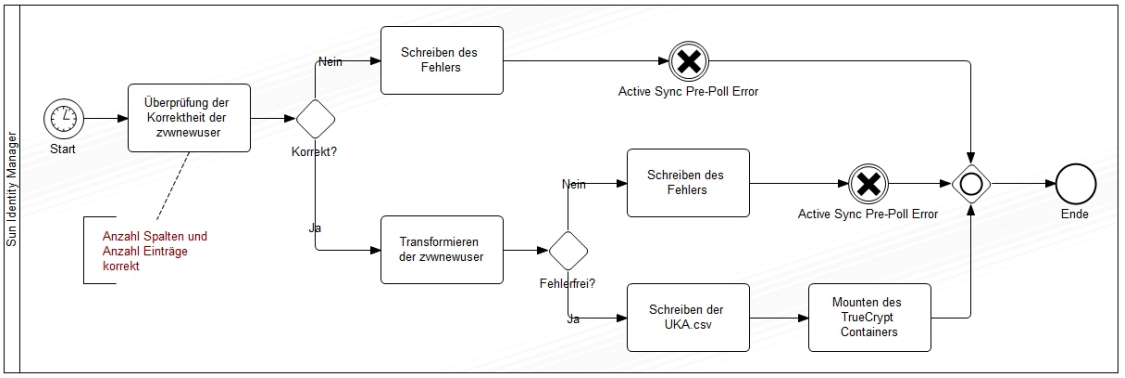


Abbildung B.4: „Pre-Poll“ am Campus Süd

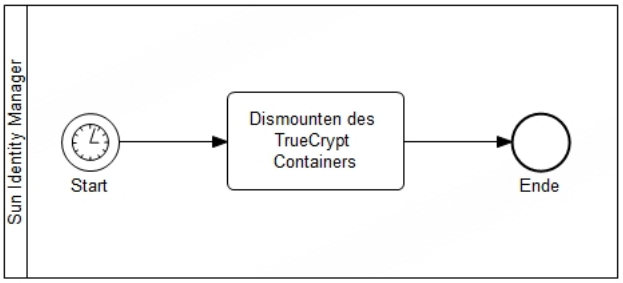


Abbildung B.5: „Post-Poll“ am Campus Süd

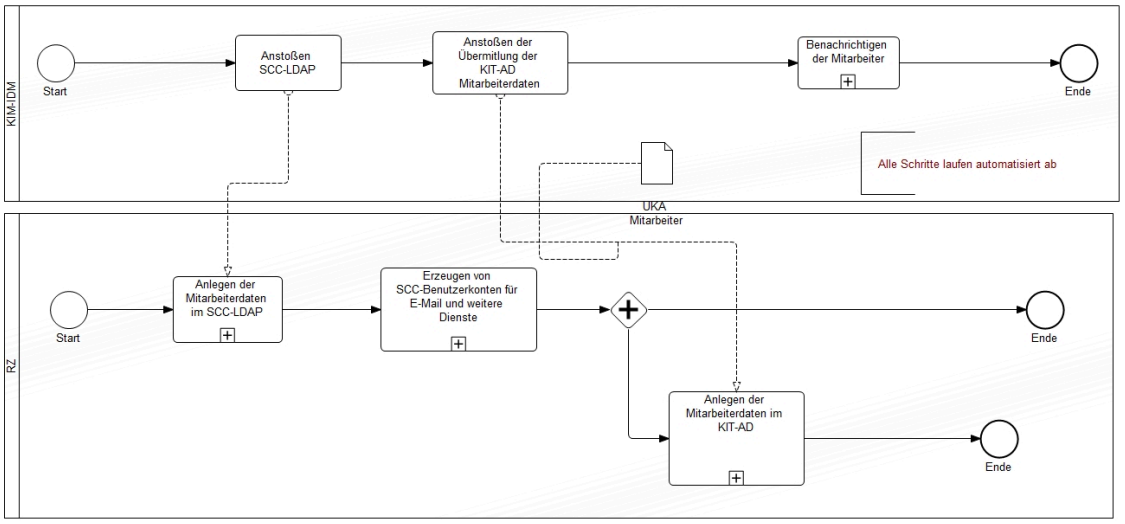


Abbildung B.6: Provisionierung in Ressourcen des Campus Süd

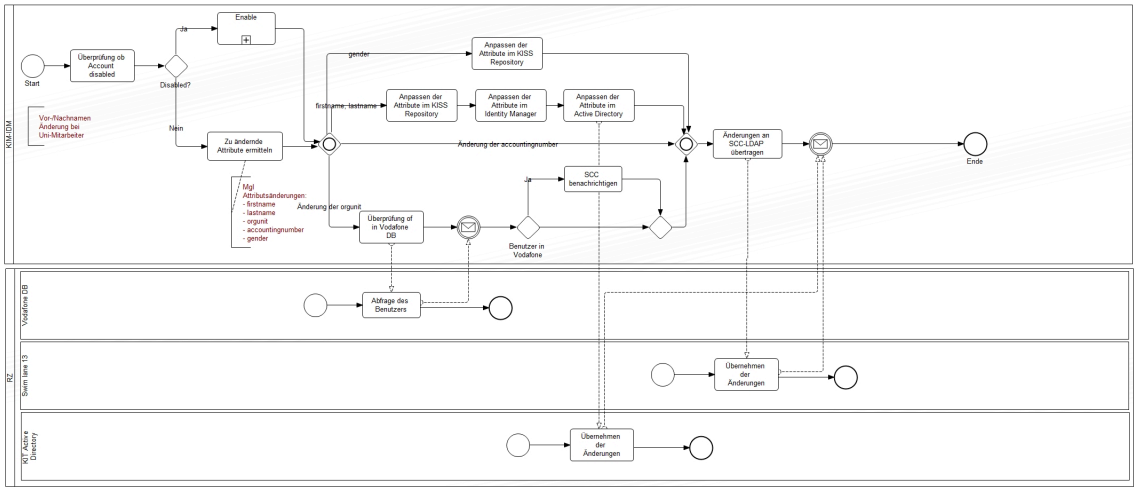


Abbildung B.7: Reprovisionierung in Ressourcen des Campus Süd

B.2 Prozesse des E-Mail-Alias-Dienstes

In Abbildung B.9 ist der Einrichtungsprozess eines E-Mail-Alias dargestellt. Abbildung B.8 zeigt den Prozessablauf, wenn ein Kunde einen Änderungsantrag für seinen bereits aktivierten E-Mail-Alias einreicht, sowie der dadurch angestoßenen Subprozesse.

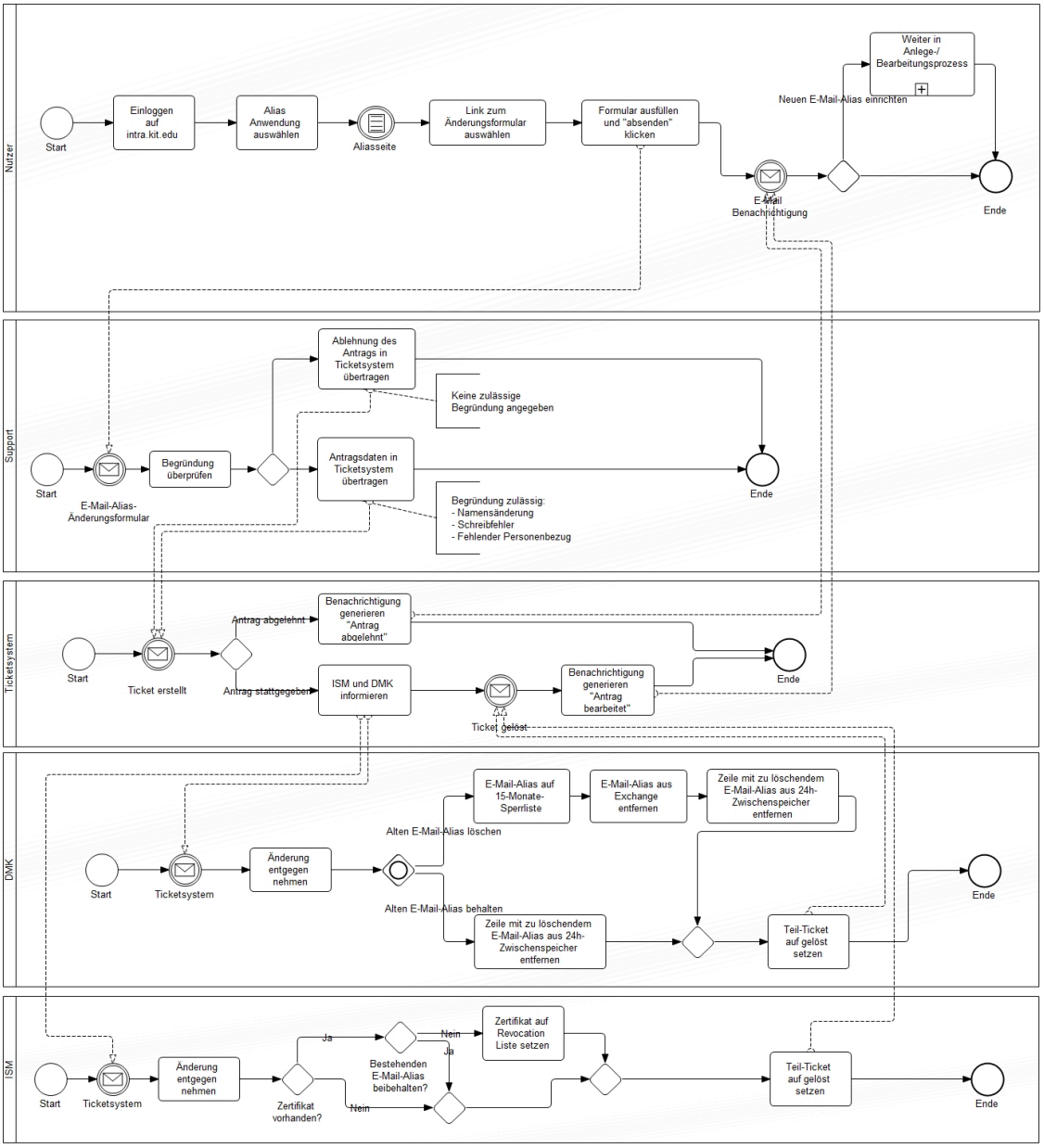


Abbildung B.8: Prozessmodell zur Bearbeitung eines E-Mail-Alias-Änderungsantrags

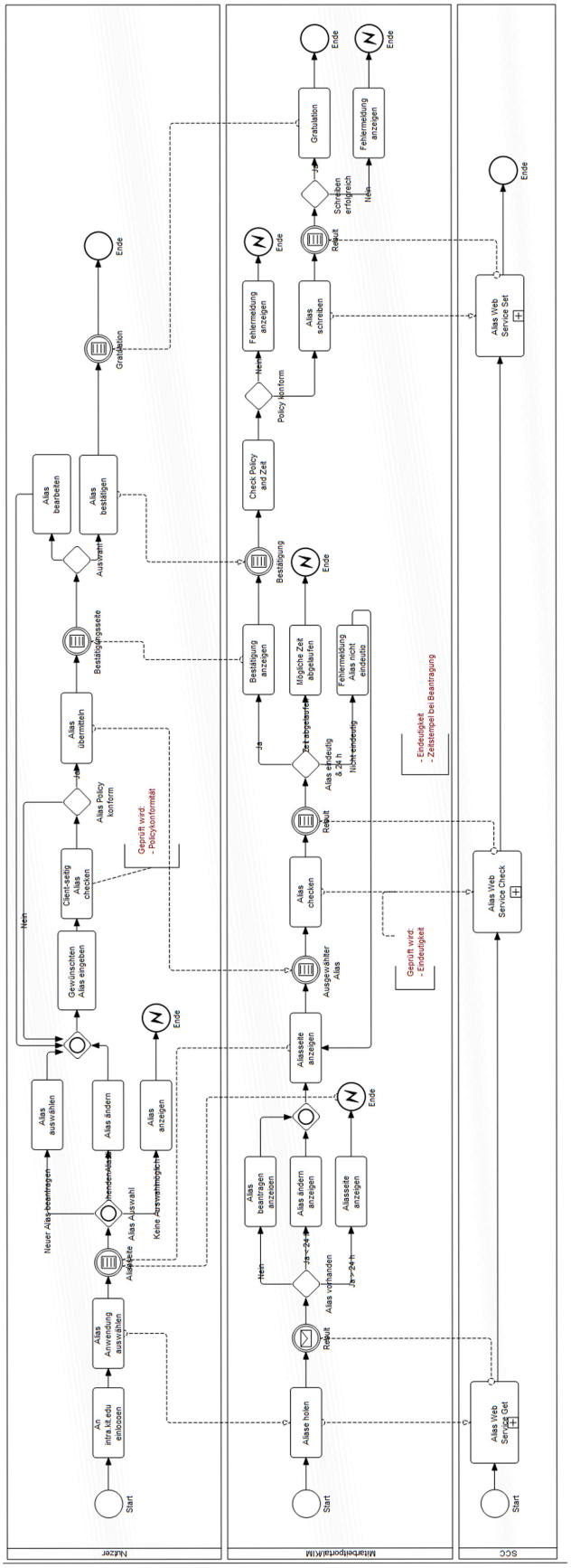


Abbildung B.9: Prozessmodell zur Einrichtung eines E-Mail-Alias

C. Impressum

Projektleitung	Prof. Dr. Hannes Hartenstein
Projektmanagement	Dipl.-Inform. Axel Maurer
Projekt	KIM - Karlsruher Integriertes InformationsManagement http://www.kim.uni-karlsruhe.de/
Teilprojekt	KIM - Identitätsmanagement (KIM-IDM)
Autoren	Dipl.-Inform. Thorsten Höllrigl Dipl.-Inform. Sebastian Labitzke Dipl.-Inform. Frank Schell Dr.-Ing. Jochen Dinger Dipl.-Inform. Axel Maurer Prof. Dr. Hannes Hartenstein
Veröffentlichung	August 2009
Fotobasis Titelgrafik	© 2008 stormpic / aboutpixel.de / Motiv: „e-mail“


```
@TECHREPORT{KIM-IDM09,
  author = {Thorsten Höllrigl and Sebastian Labitzke and Frank Schell
            and Jochen Dinger and Axel Maurer and Hannes Hartenstein},
  title = {KIM-Identitätsmanagement - Projektdokumentation},
  institution = {Steinbuch Centre for Computing (SCC)},
  year = {2009},
  month = {August}
}
```

Literaturverzeichnis

Bazijanec et al. 2007

Bazijanec, B. ; Gausmann, O. ; Klöckner, S. ; Turowski, K. ; Beran, O.: Analyse von Risikofaktoren bei der Einführung, Integration und Migration von integrierten Informationssystemen an mittelgroßen deutschen Hochschulen. In: Gaedke, M. (Hrsg.) ; Borgeest, R. (Hrsg.): *Integriertes Informationsmanagement an Hochschulen (Tagungsband zum Workshop IIM 2007)*, Universitätsverlag Karlsruhe, 2007, S. 38–56

Djordjevic & Dimitrakos 2005

Djordjevic, I. ; Dimitrakos, T.: A note on the anatomy of federation. In: *BT Technology Journal* 23 (2005), Nr. 4, S. 89–106. – DOI 10.1007/s10550–006–0011–3. – ISSN 1358–3948

Dostal et al. 2005

Dostal, W. ; Jeckle, M. ; Melzer, I. ; Zengler, B.: *Service-orientierte Architekturen mit Web Services.* Elsevier Spektrum Akademischer Verlag, 2005

Höllrigl et al. 2006

Höllrigl, T. ; Maurer, A. ; Schell, F. ; Wenske, H. ; Hartenstein, H.: Dienstorientiertes Identitätsmanagement für eine Pervasive University. In: *Jahreskonferenz der GI - Lecture Notes in Informatics*, 2006, S. 70–74

Höllrigl et al. 2008

Höllrigl, T. ; Schell, F. ; Hartenstein, H.: Towards Systematic Engineering of Service-Oriented Access Control in Federated Environments. In: *Proceedings of the IEEE Congress on Services Part II (SERVICES-2)*, 2008. – DOI 10.1109/SERVICES–2.2008.24, S. 104–111

Höllrigl et al. 2007

Höllrigl, T. ; Schell, F. ; Wenske, H. ; Hartenstein, H.: Föderatives und dienstorientiertes Identitätsmanagement: Konzept und Erfahrungen. In: *Praxis der Informationsverarbeitung und Kommunikation (PIK)* Jg. 30, Nr. 3 (2007), S. 156–162

Jahn & Stamms 2004

Jahn, G. ; Stamms, R.: *Identity Management und Zentraler Verzeichnisdienst.* Workshop - Campus Web, Portale für Forschung und Lehre, 2004

Klingenstein 2007

Klingenstein, N.: Attribute Aggregation and Federated Identity. In: *Proceedings of the 2007 IEEE International Symposium on Applications and the Internet Workshops (SAINTW '07)*, 2007, S. 26–29

SAML V2.0 2005

Security Assertion Markup Language (SAML) V2.0 Specification and Schema Set. 2005 http://docs.oasis-open.org/security/saml/v2.0/saml-2.0-os.zip

Schell et al. 2009

Schell, F. ; Höllrigl, T. ; Hartenstein, H.: Federated Identity Management as a Basis for Integrated Information Management. In: *it - Information Technology* 51 (2009), Nr. 1, S. 14–23. – DOI 10.1524/itit.2009.0518. – ISSN 1611–2776

WWW Liberty Alliance 2009

Liberty Alliance Project Homepage. 2009 http://www.projectliberty.org/

WWW Shibboleth 2009

Shibboleth - A project of the Internet2 Middleware Initiative. 2009 http://shibboleth.internet2.edu/

WWW SPML 2009

Service Provisioning Markup Language. 2009 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=provision

Steinbuch Centre for Computing (SCC)
76128 Karlsruhe
Tel: 0721/608-3754 oder 07247/82-5601

E-Mail scc@kit.edu

SCC-TB-2009-1