

Identitätsmanagement im akademischen Umfeld

Entwicklung eines Pilotsystems mit OpenID

Bachelor-Report von Dennis Blöte (Matrikelnummer 2083225)
Universität Bremen, Digitale Medien, Sommersemester 2008

Erstgutachter: Prof. Dr. Carsten Bormann

Zweitgutachter: Prof. Dr. Andreas Breiter

Fertiggestellt am 10. August 2008

Für Veröffentlichung überarbeitet am 25. August 2008

Inhaltsverzeichnis

1	Einführung.....	1
1.1	Motivation.....	1
1.2	Gliederung und Aufbau.....	3
2	Grundlagen und aktueller Forschungsstand.....	4
2.1	Begriffsdefinitionen.....	4
2.1.1	Digitale Identität und Attribute.....	4
2.1.2	Zentralisierte und föderierte Identität.....	5
2.1.3	Identity Provider.....	6
2.1.4	Service Provider / Relying Party.....	6
2.1.5	Benutzerzentriertes Identitätsmanagement.....	7
2.1.6	Situated Software.....	8
2.2	Standards und Technologien.....	9
2.2.1	LDAP und Verzeichnisdienste.....	9
2.2.2	Shibboleth.....	10
2.3	OpenID.....	14
2.3.1	Das Protokoll und seine Erweiterungen.....	16
2.3.2	Ablauf einer OpenID-Authentifizierung.....	19
2.3.3	Potentielle Gefahren und Probleme.....	22
2.4	Identitätsmanagement.....	24
2.4.1	Anwendungsfälle im akademischen Umfeld.....	25
2.4.2	Identitätsmanagement an der Universität Bremen.....	28
2.4.3	Identitätsmanagement an anderen Hochschulen.....	29
3	Entwicklung des Pilotsystems.....	31
3.1	Beschreibung des Pilotsystems.....	31
3.1.1	Anforderungen.....	31
3.1.2	Vorgaben und Rahmenbedingungen.....	31
3.1.3	Motivation für den Einsatz von OpenID.....	32
3.1.4	Technische Anforderungen.....	33
3.1.5	Der Identity Provider.....	34
3.2	Integration in eine Relying Party.....	35
3.2.1	Login-Formular.....	35
3.2.2	Attribute Exchange.....	37

3.2.3	Simple Registration.....	38
3.2.4	Referenzimplementierung einer Relying Party.....	39
3.2.5	Einbindung in bestehende Anwendungen.....	39
3.3	Sicherheitsaspekte.....	40
3.3.1	Authentifizierung.....	40
3.3.2	Verifizierte Attribute.....	40
3.3.3	Autorisierung.....	41
3.3.4	Phishing.....	41
3.3.5	Haltbarkeit bzw. Verlust der OpenID.....	42
3.3.6	Erstellung eines Benutzerprofils.....	43
4	Zusammenfassung und Ausblick.....	44
5	Quellenverzeichnis.....	46
6	Abbildungsverzeichnis.....	50
7	Ehrenwörtliche Erklärung.....	51

1 Einführung

1.1 Motivation

Im Fachbereich 3 der Universität Bremen gibt es ca. 4.000 Studenten und Mitarbeiter¹. Jede dieser Personen besitzt am zentralen Server des Fachbereichs ein Benutzerkonto, durch das der Zugang zu wichtigen Diensten wie WLAN und E-Mail gewährt wird. Darüber hinaus besitzen sowohl Studenten als auch Mitarbeiter mehrere zusätzliche Benutzerkonten für kleinere Anwendungen, welche im Laufe der Zeit im Fachbereich entstanden sind. Zu diesen Anwendungen zählen unter anderem der interne Lehrveranstaltungsplaner², projektbezogene Wikis und Bugtracker, sowie Zugänge zu Materialien für bestimmte Lehrveranstaltungen. Diese im Regelfall sehr spezifischen Anwendungen implementieren jeweils eine eigenständige Benutzerverwaltung, da sie für kleine Benutzergruppen bestimmt sind und die Anbindung an das zentrale Benutzerkonto-System zu aufwändig wäre und aus Sicherheitsgründen nicht gewünscht ist.

Da jede dieser Anwendungen eine separate Benutzerverwaltung voraussetzt, wird die Pflege der persönlichen Daten und das Einrichten neuer Anwendungen unnötig erschwert: Wird eine weitere Anwendung entwickelt, muss diese nicht nur erneut eine Verwaltung der Zugänge implementieren, sondern auch die Benutzer müssen sich abermals registrieren und freigeschaltet werden. Diese Lösung ist nicht benutzerfreundlich und bedingt das Merken verschiedener Passwörter. Ebenfalls führt die Lösung zu unzuverlässigen Datensätzen, da die Informationen in der Regel nicht auf dem aktuellen Stand gehalten werden und somit veralten. Ein Problem in dieser Hinsicht sind beispielsweise Zugänge zu Ressourcen, welche ein Student auch noch nach der Beendigung des Studiums besitzt, weil entsprechende Berechtigungen nicht überall angepasst wurden.

Ziel dieser Arbeit ist die Konzeption und Integration eines leichtgewichtigen Identitätsmanagements für den Fachbereich 3, das die Datenpflege und den Zugang zu den Anwendungen auf Basis des zentralen Benutzerkontos ermöglicht. Das System soll es dem Benutzer zukünftig erlauben, seine persönlichen Daten in begrenztem Umfang selbst zu pflegen und diese gezielt an verschiedene Anwendungen zu übertragen. Dies kann

1 Der Fachbereich 3 umfasst die Fächer Informatik und Mathematik, vgl.

<http://www.informatik.uni-bremen.de/cms/detail.php?id=12>

2 Interner Lehrveranstaltungsplaner des Fachbereich 3, <http://ilvp.informatik.uni-bremen.de/>

beispielsweise von den Studenten genutzt werden, um sich in Lehrveranstaltungen einzutragen oder diese anonym zu evaluieren, den Nachweis von Berechtigungen zu erbringen oder um sich anderen Anwendungen gegenüber als Student der Universität Bremen auszuweisen. Mittels des Logins am Identitätsmanagementsystem wird es dem Benutzer möglich sein, alle unterstützten Anwendungen zu nutzen, ohne sich dort registrieren und erneut seine Benutzerdaten angeben zu müssen. Auf Seiten der Anwendungen entfällt damit das Einrichten einer eigenständigen Benutzerverwaltung, da diese in Zukunft auf dem zentralen Benutzerkonto aufbauen können.

Die Entwicklung von speziell auf einen Anwendungsfall optimierten Anwendungen spielt im Fachbereich 3 eine wichtige Rolle, da sie beispielsweise zur effizienteren Organisation oder für Verwaltungstätigkeiten genutzt werden können. Diese auch als Situated Software³ bezeichneten Anwendungen profitieren sehr davon, dass die Einbindung der bestehenden Benutzerkonten vereinfacht wird. Da es durch das Identitätsmanagementsystem zukünftig möglich sein wird die Benutzerverwaltung auszulagern, kann man sich bei der Entwicklung direkt auf den eigentlichen Anwendungsfall konzentrieren und somit schneller Ergebnisse erzielen.

Als technologische Basis des Identitätsmanagements wurde das Single Sign-On System OpenID⁴ gewählt, welches auf der bestehenden Infrastruktur aufgesetzt werden kann und im Vergleich zu anderen Lösungen wie Shibboleth⁵ einfach zu integrieren ist. Da OpenID ein offener Standard mit zunehmender Verbreitung ist, kann die bereitgestellte Identität von den Studenten und Mitarbeitern des Fachbereichs nicht nur im Kontext der Universität, sondern auch nach außen genutzt werden, um sich damit auf allen Websites, die OpenID unterstützen⁶, einzuloggen.

Wissenschaftlich betrachtet handelt es sich um ein interessantes Themenfeld, da OpenID sich zunehmend besserer Akzeptanz erfreut, es jedoch im Bereich der Verwendung für leichtgewichtiges Identitätsmanagement bisher kaum Erfahrungen gibt. Bisher wird OpenID an deutschen Hochschulen nicht genutzt, vereinzelt stattdessen jedoch amerikanische Universitäten, wie beispielsweise das Massachusetts Institute of Technology, ihre

3 Shirkey, C. (2004): Situated Software, http://www.shirky.com/writings/situated_software.html

4 OpenID.net (2008): What is OpenID? <http://openid.net/what/>

5 Shibboleth, <http://shibboleth.internet2.edu/>

6 The OpenID Directory, <http://openiddirectory.com/>

Studenten und Mitarbeiter bereits mit einer OpenID aus⁷, um von den vielfältigen Möglichkeiten zu profitieren.

Das akademische Umfeld des Fachbereich 3 dient als sehr gute Basis für die Erforschung des Themenfeldes, da es mit den Studenten und Mitarbeitern einen großen Kreis potentieller Anwender gibt. Die in dieser Arbeit gewonnenen Erkenntnisse sind nicht nur auf den Einsatz von OpenID im akademischen Umfeld beschränkt, sondern lassen sich auch auf andere Anwendungsgebiete abbilden. Beispielsweise kann die technische Komponente auch auf den Einsatz in einem Intranet übertragen werden, sofern infrastrukturelle Gemeinsamkeiten – wie die Verwendung eines LDAP-Servers als zentrale Datenquelle – zu Grunde liegen.

1.2 Gliederung und Aufbau

Der Bachelor-Report besteht aus einem Analyse- und einem Praxisteil. Die analytische Abhandlung gliedert sich in zwei Abschnitte: Der erste Teil umfasst die Definition grundlegender Begriffe, eine Einführung in die Thematik Identitätsmanagement und eine Beschreibung der Anwendungsfälle im akademischen Umfeld, sowie einen Überblick des aktuellen Forschungsstands und der zur Verfügung stehenden Technologien. Der zweite Abschnitt enthält Ausführungen, welche sich auf die Entwicklung des Pilotsystems beziehen: Die Beschreibungen der Rahmenbedingungen, die Motivation für den Einsatz von OpenID und Abgrenzung zu anderen Technologien, der Aufbau des Pilotsystems sowie Erläuterung zu Sicherheitsaspekten und die Dokumentation der Schnittstellen.

Praktischer Teil der Arbeit ist die Entwicklung und Integration eines OpenID-Servers für den Fachbereich 3. Dies umfasst die Planung der Infrastruktur, Installation des Dienstes, Gestaltung der Benutzeroberfläche, Dokumentation des Systems sowie eine Beispielimplementierung für die Einbindung in eine Anwendung.

⁷ AuthMitEdu: <http://auth.mit.edu/>

2 Grundlagen und aktueller Forschungsstand

2.1 Begriffsdefinitionen

2.1.1 Digitale Identität und Attribute

Die digitale Identität einer Person besteht aus all ihren verfügbaren, personenbezogenen Daten, den Attributen. Ein Attribut kann dabei jedes Merkmal sein, das eine Person identifiziert oder beschreibt – dazu gehören neben Eigenschaften wie Name oder Geburtsdatum auch Rechte und Privilegien. Die digitale Identität eines Studenten setzt sich beispielsweise aus seinen Stammdaten (Name, Matrikelnummer, E-Mail, etc.), zusätzlichen Eigenschaften, die seine studentische Laufbahn betreffen (ist im 5. Fachsemester, nimmt an Lehrveranstaltung XY teil, ist Tutor, etc.), und den damit verbundenen Privilegien zusammen. Privilegien können in diesem Fall beispielsweise Berechtigungen zum Zugriff auf bestimmte Lehrmaterialien oder Vergünstigungen sein, die er durch den Nachweis seines Studentenstatus erhält.⁸

Dies setzt voraus, dass digitale Identität immer auch mit dem Nachweis der Inhaberschaft verbunden ist. Dazu ist es notwendig, dass sich der jeweilige Benutzer der Vertrauensstelle gegenüber authentifiziert und somit beweist, dass er im Besitz der behaupteten Identität ist.⁹ Ein solcher Nachweis kann auf verschiedene Art und Weise erbracht werden: In der physischen Welt erfolgt dies unter anderem mittels Personalausweis, in der digitalen Welt ist die Authentifizierung durch ein Login mit Benutzername und Passwort üblich. Darüber ist jede Identität auch mit einem eindeutigen Identifier verbunden, der das Herstellen von Verbindungen ermöglicht. Dieser Bezeichner kann beispielsweise die Personalausweisnummer oder auch die OpenID einer Person sein.

Eine Person kann verschiedene Rollen ausprägen und somit mehrere, unterschiedliche digitale Identitäten haben. Ein Beispiel für diese Unterscheidung wäre die private und berufliche oder studentische Identität, in denen man als Person jeweils unterschiedlich auftreten kann: In den verschiedenen Rollen unterscheiden sich Eigenschaften wie Adresse und Telefonnummer, es sind zusätzliche weitere Merkmale vorhanden und

⁸ Vgl. Dinger, J.; Hartenstein, H. (2008): Netzwerk- und IT-Sicherheitsmanagement, Eine Einführung, <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/142064>

⁹ Vgl. Windley, P. (2005): Digital Identity, 1. Auflage, O'Reilly

andere fehlen. Durch diese Differenzierung ist es möglich, die digitalen Identitäten in verschiedenen Kontexten und gegebenenfalls mit unterschiedlichen Stufen der Vertraulichkeit einzusetzen, so dass man beispielsweise mit privaten Details vorsichtiger umgehen wird. Es empfiehlt sich jedoch generell möglichst wenig Attribute weiterzugeben – die anfragenden Dienste sind dabei gehalten, nur die tatsächlich benötigten Informationen anzufordern und es dem Benutzer zu überlassen, ob er die geforderten Attribute liefern möchte.¹⁰

Aus technischer Sicht wird ein Attribut immer durch einen Bezeichner und den dazugehörigen Wert beschrieben. Bezeichner und Format der Attribute werden durch den jeweils benutzten Standard festgelegt. Beispiele für Standards, auf denen aufbauend der Austausch von Attributen möglich ist, sind Shibboleths eduPerson oder das in OpenID verwendete Simple Registration Format und Attribute Exchange, welche im weiteren Verlauf der Arbeit näher beschrieben werden.

2.1.2 Zentralisierte und föderierte Identität

Für den Aufbau eines Identitätsmanagementsystems gibt es zwei grundlegende Ansätze, die sich auf die Zusammensetzung der Attribute einer Identität beziehen. Die Attribute können entweder aus einer zentralen Quelle stammen oder aus mehreren Quellen zusammengeführt werden.

2.1.2.1 Zentralisierte Identität

Eine zentralisierte Identität setzt sich aus Attributen zusammen, welche aus einer einzigen Quelle stammen. Dieser Ansatz eignet sich insbesondere dann, wenn alle erforderlichen Daten durch eine Quelle bereitgestellt werden können und es nur ein begrenztes Set an Attributen gibt, welche ausgetauscht werden müssen. Negativer Aspekt der zentralisierten Datenspeicherung ist, dass der Ansatz schlecht skaliert, da der zentrale Verwaltungsspeicher bei jeder Erweiterung angepasst werden muss.¹¹

2.1.2.2 Föderierte Identität

Der Begriff föderierte Identität beschreibt den Umstand, dass die Attribute, aus denen sich die Identität zusammensetzt, aus mehreren Quellen stammen. Die Attribute werden von

¹⁰ Vgl. Windley, P. (2005): Digital Identity, 1. Auflage, O'Reilly

¹¹ Vgl. Windley, P. (2005): Digital Identity, 1. Auflage, O'Reilly

verschiedenen Systemen verwaltet und bei Bedarf entsprechend zusammengeführt. Beispielsweise besitzt die Verwaltung einer Universität die Anschrift und Telefonnummer eines Studenten, nicht aber Informationen darüber, an welchen Kursen er teilnimmt. Diese Daten wiederum werden vom Fachbereich verwaltet, in dem der Student eingeschrieben ist. Muss ein Student diese Daten übermitteln, so können sie aus den beiden Quellen zusammengeführt und als föderierte Identität übertragen werden.¹²

Vorteil dieses Ansatzes ist, dass es eine Vielzahl von Datenquellen mit jeweils unterschiedlichen Informationen geben kann und keine große Datenbank mit allen Informationen benötigt wird. Dadurch ist der föderierte Ansatz flexibler und bietet sich insbesondere für organisationsübergreifendes Single Sign-On an¹³: So ist es mittels föderierter Identität realisierbar, dass Studenten der Universität Bremen ihr Benutzerkonto auch an der Hochschule in Bremen oder Bremerhaven nutzen können. Die jeweiligen Hochschulen verwalten in dem Fall nur die für sie relevanten Daten und übermitteln sie, wenn sich beispielsweise ein Student der Hochschule an der Universität für eine Lehrveranstaltung einschreibt.

2.1.3 Identity Provider

Ein Identity Provider dient der Bereitstellung und Verwaltung digitaler Identitäten. Bei so einem „Identitätsanbieter“ handelt es sich um Software, welche Benutzer authentifizieren kann und ihnen die Möglichkeit gibt, ihre Attribute zu verwalten und sie mit anderen Diensten auszutauschen.¹⁴ Der Identity Provider soll den Anwender im Umgang mit seiner digitalen Identität unterstützen und ihm ein möglichst hohes Maß an Sicherheit und Kontrollmöglichkeiten bieten.

2.1.4 Service Provider / Relying Party

Bei einem Service Provider handelt es sich um einen Dienst, welcher den Informationen eines Identity Providers vertraut und darauf basierend Zugriff auf Ressourcen erlaubt.¹⁵ Da der Service Provider durch dieses Vertrauensverhältnis vom Identity Provider abhängig ist, wird er auch als Relying Party (Vertrauende Partei) bezeichnet. Beispielsweise lagert ein

12 Vgl. Windley, P. (2005): Digital Identity, 1. Auflage, O'Reilly

13 Vgl. Dinger, J.; Hartenstein, H. (2008): Netzwerk- und IT-Sicherheitsmanagement, Eine Einführung, <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/142064>

14 Vgl. DFN-AAI (2008): Identity-Provider, <https://www.aai.dfn.de/dokumentation/identity-provider/>

15 Vgl. DFN-AAI (2008): Service-Provider, <https://www.aai.dfn.de/dokumentation/service-provider/>

Service Provider, welcher ein Login mittels OpenID erlaubt, die Authentifizierung zum jeweiligen Identity Provider des Benutzers aus und ist somit darauf angewiesen, dass der Identity Provider geeignete Sicherheitsanforderungen erfüllt.

Ein Service Provider stellt Ressourcen zur Verfügung und benötigt dafür gegebenenfalls Informationen über den Anwender. Er kann diese Attribute vom Identity Provider anfordern und, sofern der Benutzer zustimmt, werden die Attribute vom Identity Provider für den Service Provider freigegeben.¹⁶ Dies setzt voraus, dass Identity Provider und Service Provider den gleichen Standard unterstützen und somit die selbe Sprache sprechen.

Die Spezifikationen von Shibboleth und OpenID unterscheiden sich in der Bezeichnung dieser Komponente, daher wird im weiteren Verlauf bei Bezug auf Shibboleth der Begriff Service Provider benutzt, im Kontext von OpenID wird die Bezeichnung Relying Party verwendet.

2.1.5 Benutzerzentriertes Identitätsmanagement

Aus Sicht des Anwenders sollte das Identitätsmanagement einen transparenten und aktiven Prozess darstellen. Beim benutzerzentrierten Identitätsmanagement steht der Benutzer im Mittelpunkt der stattfindenden Transaktionen und hat die Kontrolle über Verwaltung und Schutz seiner Identitätsinformationen. „Er wird nicht bei jeder Transaktion eingeschaltet, aber die Identitätsdaten fließen immer durch seine Identitätsverwaltung. So kommt er in die Position, dem jeweiligen Gebrauch seiner Identitäten und Teilidentitäten zustimmen und die Verwendung kontrollieren zu können. Ein wichtiger Aspekt ist sicherlich auch, dass die Verwaltung von Identitätsdaten dezentralisiert wird und dort bleibt, wo sie am besten aufgehoben ist, nämlich beim Besitzer der Identitätsdaten selbst.“¹⁷

Kim Cameron¹⁸ beschreibt in seinen „Seven Laws of Identity“¹⁹ die Grundpfeiler eines Metasystems für Identitätsverwaltung, welches den Ansprüchen der Benutzerzentriertheit gerecht wird. Zu diesen Punkten zählen die Benutzerzustimmung, die minimale Preisgabe

16 Vgl. Dinger, J.; Hartenstein, H. (2008): Netzwerk- und IT-Sicherheitsmanagement, Eine Einführung, <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/142064>

17 Dr. Degenhardt, W.: Benutzerzentriertes Identitätsmanagement, <http://www.netigator.de/netigator/live/show.php3?id=47&aid=31227966>

18 Kim Cameron: Chief Architect of Identity bei Microsoft, <http://www.identityblog.com/>

19 Cameron, K. (2006): The laws of identity, <http://www.identityblog.com/?p=352>

von Informationen, ein dezentraler Aufbau mit einem Pluralismus von Betreibern und Technologien, sowie ein konsistentes Benutzungserlebnis über die verschiedenen Anwendungen hinweg.²⁰ Vereinfacht zusammengefasst lässt sich sagen, dass der Benutzer die Entscheidungsfreiheit darüber haben soll, wer sein Identity Provider ist, welcher Relying Party er vertraut und welche Informationen er austauschen möchte.

„Nebenbei hat dies zur Folge, dass der Service-Provider den Identitäts-Provider vor der Transaktion nicht kennen muss. So wird eine viel bessere Skalierbarkeit erreicht, als es mit servicezentrierter Identität jemals möglich wäre. Das Netzwerk der Sites kann sich bei benutzerzentriertem Identitätsmanagement ad hoc aufbauen, wie man es heute bereits von den Verfahren bei SMTP-Servern kennt. Die einzige Anforderung an den Service-Provider ist, dass er den Berechtigungsnachweisen des Benutzers vertraut.“²¹

2.1.6 Situated Software

Situated Software ist ein von Clay Shirky²² geprägter Begriff, der Software beschreibt, die für einen speziellen Anwendungsfall und im Regelfall für eine kleine Benutzergruppe entwickelt ist: „This is software designed in and for a particular social situation or context. [...] Situated software is in the small pieces category, with the following additional characteristic – it is designed for use by a specific social group, rather than for a generic set of 'users'.“²³

Durch die Ausrichtung auf eine bekannte Benutzergruppe ist es möglich, eine maßgeschneiderte Lösung vergleichsweise schnell und günstig zu entwickeln. Dabei spielt der Kontext, in dem die Software eingesetzt wird, die entscheidende Rolle: Die Software wird für eine bestimmte Aufgabe entwickelt und dient bewusst nicht der generellen Lösung einer Problemstellung. Ein Beispiel dafür wäre eine e-Learning Anwendung, die für einen Kurs und die daran teilnehmenden Studenten entwickelt wird und nicht den Anspruch hat, als hochschulweite Lösung eingesetzt zu werden. Durch diese Fokussierung kann die Software schneller entwickelt werden und zudem besser die Problemstellung des spezifischen Anwendungsfalls lösen.

20 Vgl. Dinger, J.; Hartenstein, H. (2008): Netzwerk- und IT-Sicherheitsmanagement, Eine Einführung, <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/142064>

21 Dr. Degenhardt, W.: Benutzerzentriertes Identitätsmanagement, <http://www.netigator.de/netigator/live/show.php3?id=47&aid=31227966>

22 Clay Shirky: Autor und Dozent für New Media an der New York University, <http://www.shirky.com/>

23 Shirkey, C. (2004): Situated Software, http://www.shirky.com/writings/situated_software.html

2.2 Standards und Technologien

2.2.1 LDAP und Verzeichnisdienste

LDAP (*Lightweight Directory Access Protocol*) ist ein Protokoll für die Abfrage von Verzeichnisdiensten. Bei einem Verzeichnisdienst handelt es sich um eine hierarchische Datenbank, in der Benutzerinformationen verwaltet und durchsucht werden können. Ein Beispiel für solch ein Verzeichnis sind Telefonbücher oder Melderegister. Der Unterschied gegenüber einer relationalen Datenbank ist, dass ein Verzeichnisdienst auf das Durchsuchen und Ausgeben der Informationen und somit auf Lese- anstatt Schreibzugriffe optimiert ist.²⁴

LDAP dient der Kommunikation von Clients mit dem Verzeichnisdienst (Server) und ermöglicht die Abfrage von Einträgen (*Entries*) des Verzeichnisses. Entries können jegliche Art von Objekt sein, beispielsweise Personen, Drucker oder Räume. Die grundlegenden Datenstrukturen eines LDAP-Verzeichnisses werden insgesamt als Schema bezeichnet. Das Schema gibt an, welche Objektklassen vom Verzeichnisdienst unterstützt werden und gibt somit implizit Auskunft über die im Verzeichnisdienst verfügbaren Attribute. Eine Objektklasse ist die Beschreibung einer Menge von Attributen: Sie definiert Bezeichner und Form der in ihr verfügbaren Attribute und legt fest, welche Attribute erforderlich beziehungsweise optional sind. Die Objektklasse *person* definiert beispielsweise die zu einer Person gehörenden Attribute: Der Wert *surname* (Nachname) ist zwingend erforderlich und darf nur aus Buchstaben bestehen, der Wert *telephoneNumber* hingegen ist optional und darf auch Bindestriche und Leerzeichen enthalten.²⁵

LDAP-Verzeichnisse können auch als Grundlage für die Authentifizierung von Benutzern eingesetzt werden. Dafür gibt es spezielle Objektklassen, die Benutzerkonten definieren – zum Beispiel die Klasse *posixAccount*, welche Unix-Benutzerdaten beschreibt. Clientanwendungen können somit ein LDAP-Verzeichnis als Basis nutzen, um Benutzer zu authentifizieren und die nötigen Benutzerdaten anzufragen.²⁶

24 Vgl. Windley, P. (2005): Digital Identity, 1. Auflage, O'Reilly

25 Vgl. Zeilenga, K. (2006): RFC 4512 - Lightweight Directory Access Protocol (LDAP), Directory Information Models, <http://tools.ietf.org/html/rfc4512>

26 Vgl. Windley, P. (2005): Digital Identity, 1. Auflage, O'Reilly

2.2.2 Shibboleth

Shibboleth ist eine vom Internet2-Konsortium definierte Open Source Lösung für verteilte, web-basierte Authentifizierung und Autorisierung. Auf Grund seines Aufbaus eignet sich Shibboleth insbesondere als organisationsübergreifendes Single-Sign On System, das von kooperierenden Organisationen innerhalb einer Föderation betrieben werden kann.²⁷ Die Architektur setzt sich aus den Komponenten Identity Provider und Service Provider zusammen – wird Shibboleth innerhalb einer Föderation eingesetzt, so ist ein zusätzlicher Lokalisierungsdienst (WAYF: Where Are You From) notwendig.²⁸

Der Ablauf eines Authentifizierungs- und Autorisierungsvorgangs ist folgendermaßen: Um auf Ressourcen eines Service Providers zugreifen zu können, muss ein Benutzer die nötige Autorisierung vorweisen können. Ist der Benutzer bislang nicht eingeloggt, so wird er vom Service Provider zum Login an seiner Heimatorganisation aufgefordert. Da der Service Provider nicht weiß, welcher Heimatorganisation der Benutzer angehört, findet eine Weiterleitung zum Lokalisierungsdienst der Föderation statt. Dort kann der Benutzer aus einer Liste der an der Föderation teilnehmenden Organisationen seine Heimatorganisation auswählen. Die Auswahl wird an den Service Provider übertragen, der den Benutzer zu dessen Heimatorganisation weiterleitet, wo der Benutzer schließlich zum Login aufgefordert wird. Nach erfolgreichem Login wird der Benutzer zusammen mit der Authentifizierungsbestätigung und Attributen, die dem Berechtigungsnachweis dienen (Assertions), zum Service Provider weitergeleitet. Der Service Provider hat nun die nötigen Informationen über die Berechtigungen des Benutzers und kann den Zugriff gewähren oder ablehnen.²⁹

27 Vgl. Shibboleth: About Shibboleth, <http://shibboleth.internet2.edu/about.html>

28 Vgl. SWITCH (2007): AAI Introductory Tutorial, <http://www.switch.ch/proxy/aai/support/presentations/infoday-2007/AAI-ID07-20-Intro.pdf>

29 SWITCH AAI (2008): Simple demo, <http://switch.ch/aai/demo/2/simple.html>

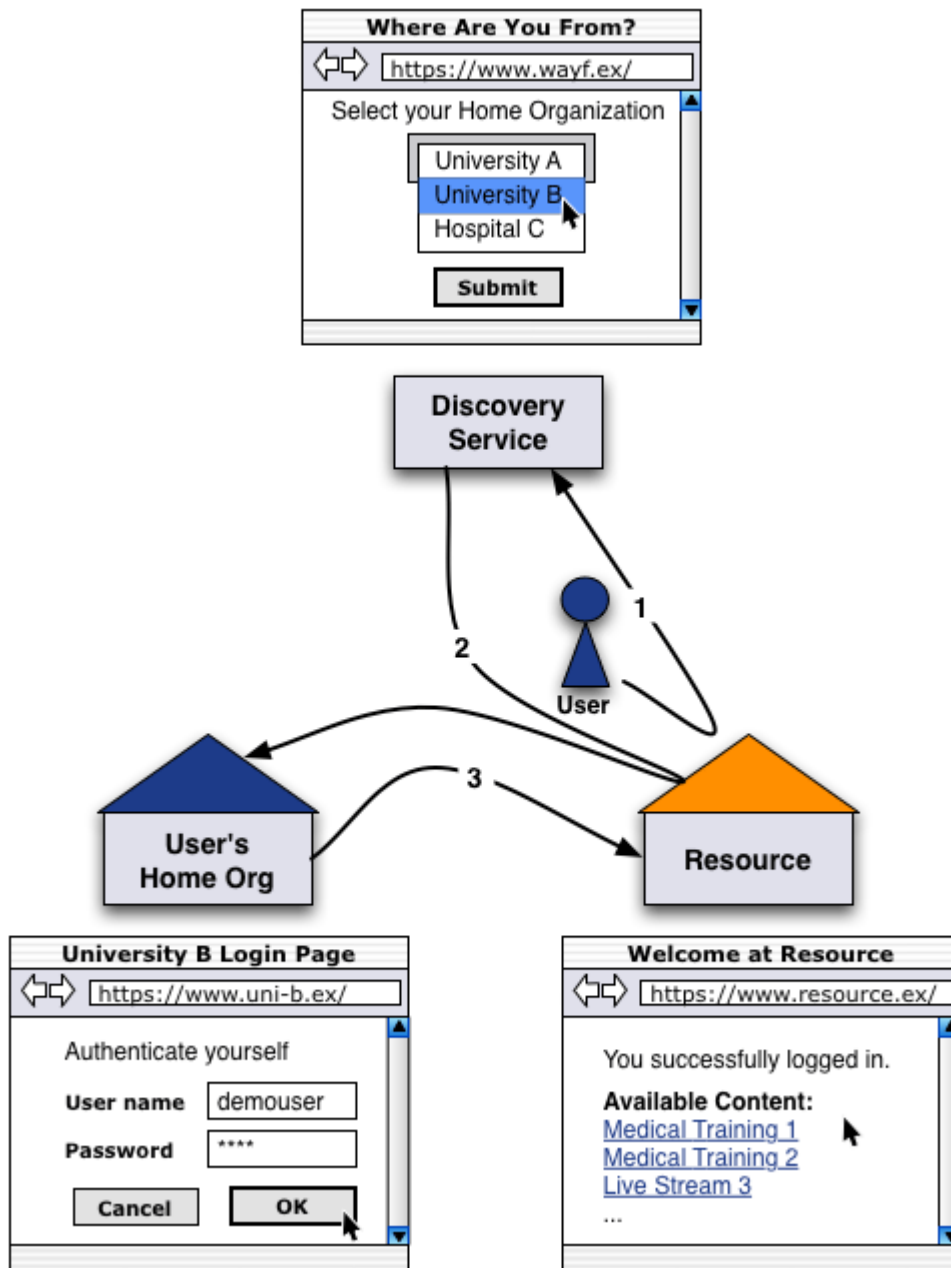


Abbildung 1: Ablauf eines Authentifizierungsvorgangs in Shibboleth²⁹

Finden innerhalb der Browser-Session weitere Zugriffe auf Ressourcen eines Service Providers der Föderation statt, so entfallen die Auswahl der Heimatorganisation und das Login dort. Die Abfolge der Weiterleitungen bleibt bestehen, jedoch ist eine Interaktion des Benutzers nicht mehr nötig, da die Heimatorganisation bereits bekannt und der Benutzer schon authentifiziert ist.³⁰

³⁰ SWITCH AAI (2008): Simple demo, <http://switch.ch/aaai/demo/2/simple.html>

Shibboleth setzt ein System mit verlässlichen Benutzerdaten voraus, auf dessen Basis die Authentifizierung und Autorisierung stattfinden kann: „[...] attributes are usually not stored within Shibboleth itself, but instead pulled from other sources. [Identity Provider] sites may choose to use a directory or database to store user data, while applications may augment the attributes they receive from an [Identity Provider].“³¹ Diese Datenbasis wird in der Regel durch ein LDAP-Verzeichnis bereitgestellt, in welchem die Benutzerdaten aggregiert und verwaltet werden. Damit die Benutzerdaten auch zur Autorisierung genutzt werden können, müssen bestimmte Attribute vorhanden sein und den Anforderungen bezüglich Datenqualität genügen. Das in Shibboleth verwendete Schema *eduPerson*, das eine Mindestmenge benötigter Attribute definiert, wird im folgenden Unterabschnitt vorgestellt.

2.2.2.1 *eduPerson*

eduPerson ist eine im Rahmen von Shibboleth definierte Objektklasse für universitäre Verzeichnisdienste. *eduPerson* besteht aus einem Satz von Attributen für Personen in Hochschulinstitutionen und definiert die Syntax und Semantik der Werte, die diesen Attributen zugewiesen werden sollen.³²

eduPerson wird als Erweiterung zu den LDAP-Objektklassen *person*, *inetOrgPerson* und *organizationalPerson* eingesetzt. Die folgende Tabelle zeigt die in *eduPerson* zusätzlich definierten Attribute.^{33 34}

Attributname	Beschreibung
<i>eduPersonAffiliation</i>	Spezifiziert verschiedene Kategorien für die Art der Zugehörigkeit einer Person zur Heimatorganisation, beispielsweise <i>student</i> , <i>employee</i> oder <i>alum</i>
<i>eduPersonEntitlement</i>	URI, der Rechte der Person an Ressourcen anzeigt
<i>eduPersonNickname</i>	Informeller Name oder Spitzname einer Person
<i>eduPersonOrgDN</i>	Der Bezeichner des Verzeichniseintrags, der die Organisation der Person repräsentiert

31 Shibboleth (2008): High Level Introduction to Shibboleth, <http://shibboleth.internet2.edu/HighLevelIntro.html>

32 EDUCAUSE (2008): *eduPerson* Object Class, <http://www.educause.edu/eduperson/>

33 Vgl. Internet2 Middleware (2007): *eduPerson* Object Class Specification, <http://www.nmi-edit.org/eduPerson/internet2-mace-dir-eduperson-200712.html>

34 Vgl. DFN-AAI (2006): Technische und organisatorische Voraussetzungen – Attribute, <https://www.aai.dfn.de/fileadmin/documents/vertraege/attribute.20061130.pdf>

eduPersonOrgUnitDN	Der Bezeichner des Verzeichniseintrags, der die Organisationseinheit der Person repräsentiert
eduPersonPrimaryAffiliation	Primärer Zugehörigkeitsstatus
eduPersonPrimaryOrgUnitDN	Der Bezeichner des Verzeichniseintrags, der die primäre Organisationseinheit der Person repräsentiert
eduPersonPrincipalName	Netz-ID, besteht aus einem linken und einem rechten Teil getrennt durch @, wobei der rechte Teil die Domain und der linke Teil eine innerhalb der Domain eindeutige ID beschreibt.
eduPersonScopedAffiliation	Art der Zugehörigkeit der Person zur eigenen Organisation, ergänzt um die zugehörige Domain
eduPersonTargetedID	Pseudonym einer Person

eduPerson dient als allgemeiner Standard für die Beschreibung von Attributen universitärer Personen, um die organisationsübergreifende Autorisierung zu ermöglichen. So wäre es beispielsweise möglich, auf Basis des Attributs *eduPersonAffiliation* nur den Studenten verschiedener Hochschulen Zugang zu einer zentralen Lernplattform zu gewähren.

Da eduPerson nur die zehn beschriebenen Attribute definiert, gibt es nationale Erweiterungen des Standards, um auf lokale Gegebenheiten einzugehen oder weitere Autorisierungsmerkmale einzuführen. Zum Beispiel definiert die SWITCH AAI zusätzliche Attribute wie *swissEduPersonMatriculationNumber* (Matrikelnummer) und *swissEduPersonUniqueID* (eindeutiger Bezeichner einer Person).³⁵

³⁵ Vgl. SWITCH (2007): Authentication and Authorization Infrastructure, http://www.switch.ch/aai/docs/AAI_Attr_Specs.pdf

2.3 OpenID

OpenID ist ein Single Sign-On System, welches es Benutzern ermöglicht, sich mit einer digitalen Identität auf verschiedenen Websites einzuloggen und dient dadurch als Ersatz für die übliche Authentifizierung mittels Benutzername und Passwort.³⁶ In einer OpenID-basierten Identitätsverwaltung gibt es – wie auch in anderen Single Sign-On Lösungen üblich – Identity Provider, welche den Benutzer authentifizieren und Identitätsinformationen bereitstellen, sowie Relying Parties, welche die nötigen Informationen über den jeweiligen Identity Provider beziehen.

OpenID ist ein offener Standard, dessen besonderes Merkmal die Auslegung auf Dezentralität ist. Zur Identitätsverifikation wird keine zentrale Autorität erfordert, daher kann theoretisch jeder Identity Provider sein. Um als Identity Provider aufzutreten ist keine Zertifizierung notwendig, es wird lediglich entsprechende Software, welche das OpenID-Protokoll unterstützt, benötigt. Dieser Aspekt ist einer der Gründe, warum OpenID als besonders benutzerzentriert gilt: Dem Benutzer steht die Wahl seines Identity Providers völlig frei und er kann diesen Teil gegebenenfalls sogar selbst übernehmen. Ebenso können Websitebetreiber durch die einfache Unterstützung von OpenID zur Relying Party werden und jedem Benutzer mit einer OpenID – ungeachtet seines Identity Providers – das Single Sign-On anbieten.

Die Entwicklung von OpenID wurde im Mai 2005 von Brad Fitzpatrick begonnen und hat im Dezember 2007 die Version 2.0 erreicht. Zusammen mit den Protokollerweiterungen Attribute Exchange, PAPE und weiteren an die Entwicklung angeschlossenen Standards (beispielsweise Yadis³⁷) bietet OpenID mittlerweile eine technologische Basis für leichtgewichtiges Identitätsmanagement: „We see OpenID as being an umbrella for the framework that encompasses the layers for identifiers, discovery, authentication, and a messaging services layer that sits atop and this entire thing has sort of been dubbed 'OpenID 2.0'.“³⁸

36 OpenID.net (2008): <http://openid.net/>

37 Yadis (2008): <http://yadis.org/>

38 Recordon, D. (2006): Moving OpenID Forward, <http://lists.danga.com/pipermail/yadis/2006-June/002631.html>

Seit Gründung der OpenID Foundation³⁹ im Juni 2007 gibt es eine formelle Basis für die Weiterentwicklung und Verbreitung des Standards. Mittlerweile haben sich Firmen wie Google, IBM, Microsoft und Yahoo! der OpenID Foundation angeschlossen, um sich an der zukünftigen Entwicklung zu beteiligen. Ebenso agieren Organisationen und Dienste wie AOL, VeriSign und Yahoo! ihren Nutzern gegenüber bereits als Identity Provider, was eine sehr hohe Anzahl potentiell nutzbarer OpenIDs bedeutet.

Die Akzeptanz von OpenID steigt zunehmend – insbesondere Betreiber von Webseiten möchten ihren Nutzern einen komfortablen Einstieg ohne Registrierung ermöglichen, was zu einer wachsenden Anzahl von Relying Parties führt. Die folgende Statistik⁴⁰ zeigt den Anstieg der von MyOpenID⁴¹ gezählten Relying Parties im Verlauf von September 2005 bis einschließlich Juli 2008.

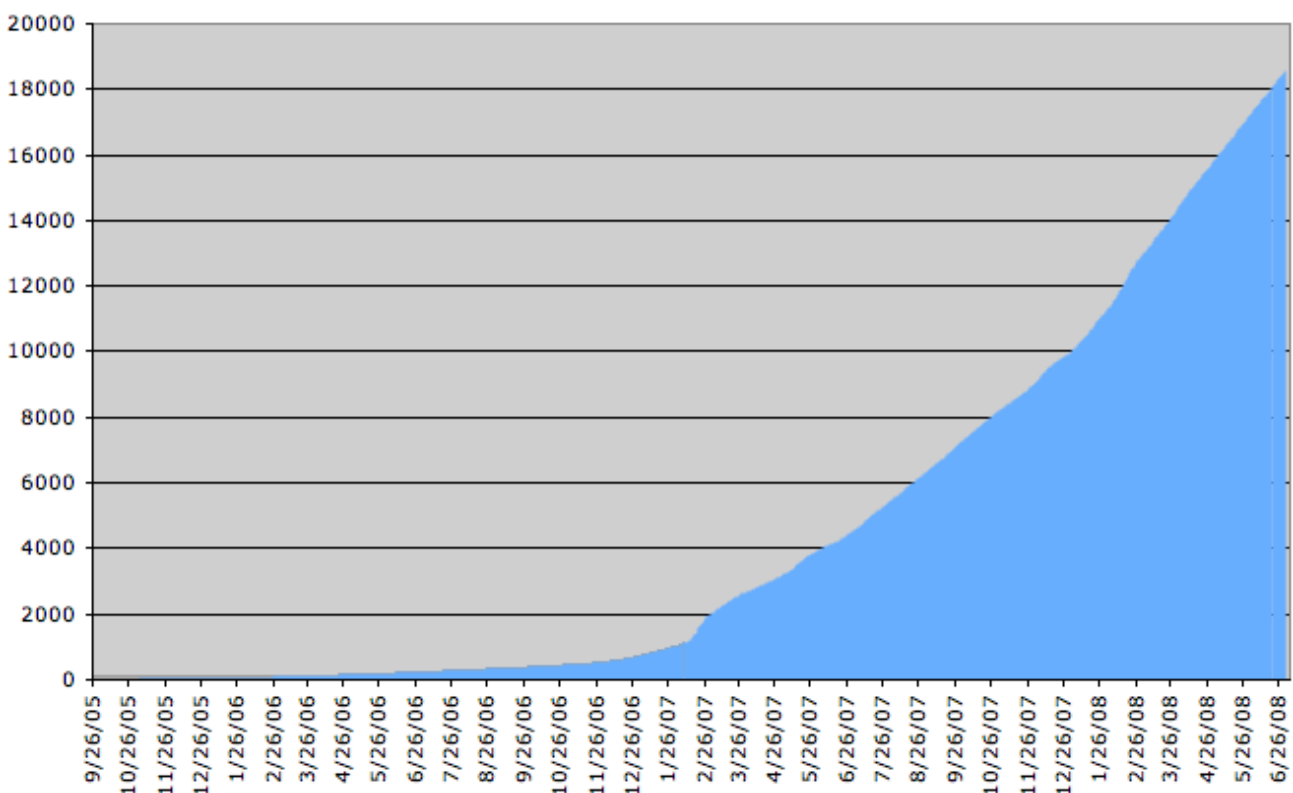


Abbildung 2: Von MyOpenID (Identity Provider) eindeutig gezählte Relying Parties⁴⁰

39 OpenID Foundation (2008): <http://openid.net/foundation>

40 Drebes, L.; JanRain (2008): Relying Party Stats as of July 1st 2008, <http://janrain.com/blog/2008/07/08/relying-party-stats-as-of-july-1st-2008/>

41 MyOpenID, <http://www.myopenid.com>

2.3.1 Das Protokoll und seine Erweiterungen

Aus technischer Sicht ist OpenID ein Protokoll, das auf HTTP-Requests und Redirects basiert und dazu dient, den Besitz einer URL zu verifizieren⁴². Bei dieser URL handelt es sich um den Identifier des Benutzers, welchen er Relying Parties gegenüber angibt – beispielsweise *https://openid.tzi.de/dbloete*. Da der Identifier eine URL ist, können die Relying Parties über die so genannte Discovery herausfinden, wer der Identity Provider des Benutzer ist und ihn zur Authentifizierung dorthin weiterleiten. Im Gegensatz zu Shibboleth wird daher kein separater WAYF-Dienst (*Where Are You From*) benötigt.

Da OpenID lediglich auf die Verifizierung des Besitzes einer URL ausgelegt ist, gibt es Erweiterungen des Protokolls, welche dem Austausch von Identitätsinformationen dienen. Die Protokollerweiterungen *Simple Registration* und *Attribute Exchange* werden im weiteren Verlauf detaillierter beschrieben, da sie zur Nutzung von OpenID für leichtgewichtiges Identitätsmanagement unabdingbar sind. Voraussetzung dafür ist, dass der Identity Provider die jeweilige Erweiterung und die auszutauschenden Attribute unterstützt.

2.3.1.1 Discovery

Für den Discovery-Vorgang gibt es mit Yadis⁴³ ein eigenständiges Protokoll, das dem Auffinden des Identity Providers und der von ihm unterstützten Erweiterungen dient. Der Ablauf dabei ist folgendermaßen⁴⁴: Die vom Benutzer als Identifier angegebene URL wird von der Relying Party geladen und nach dem Verweis auf ein XRDS-Dokument durchsucht, der sich entweder im HTTP-Header oder im Quelltext der Webseite befindet. Das XRDS-Dokument wiederum enthält den URI des Identity Providers, sowie eine Liste der vom unterstützten Erweiterungen, beispielsweise:

```
<?xml version="1.0" encoding="UTF-8"?>
<xrds:XRDS xmlns:openid="http://openid.net/xmlns/1.0"
  xmlns:xrds="xri://$xrds" xmlns="xri://$xrd*(v*2.0)">
  <XRD>
    <Service priority="1">
      <URI>https://openid.tzi.de/server</URI>
      <Type>http://specs.openid.net/auth/2.0/server</Type>
```

42 Vgl. Fitzpatrick, B. (et al. 2007): OpenID Authentication 2.0, http://openid.net/specs/openid-authentication-2_0.html

43 Miller, J. (2006): Yadis Specification Version 1.0, <http://yadis.org/papers/yadis-v1.0.pdf>

44 Vgl. Fitzpatrick, B. (et al. 2007): OpenID Authentication 2.0, Discovery, http://openid.net/specs/openid-authentication-2_0.html#discovery

```

<Type>http://openid.net/srv/ax/1.0</Type>
<Type>http://openid.net/extensions/sreg/1.1</Type>
<Type>http://openid.net/sreg/1.0</Type>
</Service>
</XRD>
</xrds:XRDS>

```

Wird kein XRDS-Dokument gefunden, so wird eine Discovery des Identity Providers über Angaben im HTML-Quelltext versucht. Dafür muss im HEAD der Webseite ein Link-Element mit dem Verweis auf die URL des Identity Providers enthalten sein, zum Beispiel:

```

<link rel="openid2.provider openid.server" href="https://openid.tzi.de/server" />

```

Ist die Discovery erfolgreich, sendet die Relying Party ihre Anfrage an die gefundene URL des Identity Providers. Wird die URL im Rahmen der Discovery nicht gefunden, kann keine Anfrage gesendet werden und der Authentifizierungsvorgang scheitert.

2.3.1.2 Simple Registration

Die Protokollerweiterung Simple Registration (SReg) dient dem Austausch von Benutzerprofilinformationen.⁴⁵ Simple Registration wurde entwickelt, um zusätzlich zur Authentifizierungsbestätigung auch Profildaten des Benutzers zu übertragen und dadurch den Registrierungsvorgang bei einer Relying Party zu überspringen – daher auch der Name. Bei den Benutzerprofilinformationen handelt es sich um die folgenden acht Attribute: *nickname*, *email*, *fullname*, *dob* (Date of Birth, Geburtsdatum), *gender*, *postcode*, *country*, *timezone* und *language*. Wenn eine Relying Party eine Authentifizierungsanfrage an den Identity Provider des Benutzers stellt, kann sie zusätzlich auch diese Benutzerinformationen anfordern. Dabei wird außerdem festgelegt, welche der Attribute für eine Registrierung des Benutzers erforderlich und welche optional sind.

Der Nachteil von Simple Registration ist, dass es sich bei den acht Attributen um eine fixe Menge handelt, welche nicht erweitert werden kann. Das bedeutet, dass nur die festgelegten Attribute ausgetauscht werden können und Relying Parties somit

⁴⁵ Vgl. Hoyt, J. (et al. 2006): OpenID Simple Registration Extension 1.0, http://openid.net/specs/openid-simple-registration-extension-1_0.html

beispielsweise nicht den Wohnort, die Telefonnummer oder die URL der Website des Benutzers erfragen können. Daher eignet sich Simple Registration nur für den Austausch der rudimentären Benutzerdaten.

Der Vorteil von Simple Registration ist, dass es von vielen Identity Providern unterstützt wird, weil sie schon sehr früh Bestandteil der Entwicklung von OpenID war. Mittlerweile ist Simple Registration jedoch veraltet, da es mit Attribute Exchange eine viel flexiblere Lösung zum Austausch von Attributen gibt.

2.3.1.3 Attribute Exchange

Attribute Exchange (AX) definiert im Gegensatz zu Simple Registration kein fixes Set von Attributen, sondern stellt einen Namensraum dar, in welchem eigene Attribute definiert werden können. Ein Attribut besteht dabei aus den vier Komponenten Type Identifier (URL als Definition), Title (Attributbezeichner für den Endnutzer), Count (Anzahl der gewünschten Werte) und Value (Wert der Eigenschaft).⁴⁶ Das Attribut E-Mail-Adresse eines Benutzers kann sich beispielsweise folgendermaßen zusammensetzen:

- *Type Identifier*: <http://axschema.org/contact/email>
- *Title*: E-Mail-Adresse
- *Count*: 1
- *Value*: dbloete@tzi.de

Voraussetzung für den Attributaustausch zwischen Relying Party und Identity Provider ist, dass beide Seiten das gleiche Set an Attributen unterstützen. Ausschlaggebend dabei ist der Type Identifier: Die URL dient als Bezeichner und legt fest, um was für ein Attribut es sich genau handelt. Der bislang einzige Standard für definierte Attribute ist eine Liste von Type Identifiern auf axschema.org, worunter sich auch Type Identifier für die in Simple Registration definierten Attribute befinden.⁴⁷

Der Nachteil von Attribute Exchange ist, dass die Erweiterung erst im Dezember 2007 spezifiziert wurde und daher bislang sowohl von Relying Parties als auch von Identity Providern kaum unterstützt wird. Abhilfe dabei können die Type Identifier schaffen, welche

46 Vgl. Hardt, D. (et al. 2007): OpenID Attribute Exchange 1.0, http://openid.net/specs/openid-attribute-exchange-1_0.html

47 Sxip Inc. (2007): Schema for OpenID Attribute Exchange, <http://www.axschema.org/types/>

die schon in Simple Registration definierten Attribute abbilden: Identity Provider, die bereits Simple Registration unterstützen können dadurch auch Attribute Exchange Anfragen für diese Attribute beantworten, ohne ihren Datenbestand erweitern zu müssen.

Der Vorteil von Attribute Exchange ist, dass die Erweiterung die Definition eigener Attribute ermöglicht, was in besonderen Anwendungsfällen unerlässlich ist. Beispielsweise gibt es bei den auf axschema.org definierten Schemata keine Attribute wie Studiengang oder Matrikelnummer. Da solche Attribute aber für das Identitätsmanagement im akademischen Umfeld eine wichtige Rolle spielen, können sie – wie im Rahmen dieser Arbeit geschehen – von einem Identity Provider definiert und ausgetauscht werden.

2.3.2 Ablauf einer OpenID-Authentifizierung

Ausgangspunkt einer OpenID-Transaktion ist das Login eines Benutzers auf Seiten der Relying Party. Im Gegensatz zur herkömmlichen Authentifizierung mittels Benutzername und Passwort befindet sich auf OpenID-unterstützenden Websites ein Eingabefeld für den OpenID-Identifizier des Benutzers. Dieses Feld ist üblicherweise durch das OpenID-Symbol gekennzeichnet, um die Erkennung zu erleichtern und ein konsistentes Benutzungserlebnis zu gewährleisten. Abbildung 3 verdeutlicht dies.

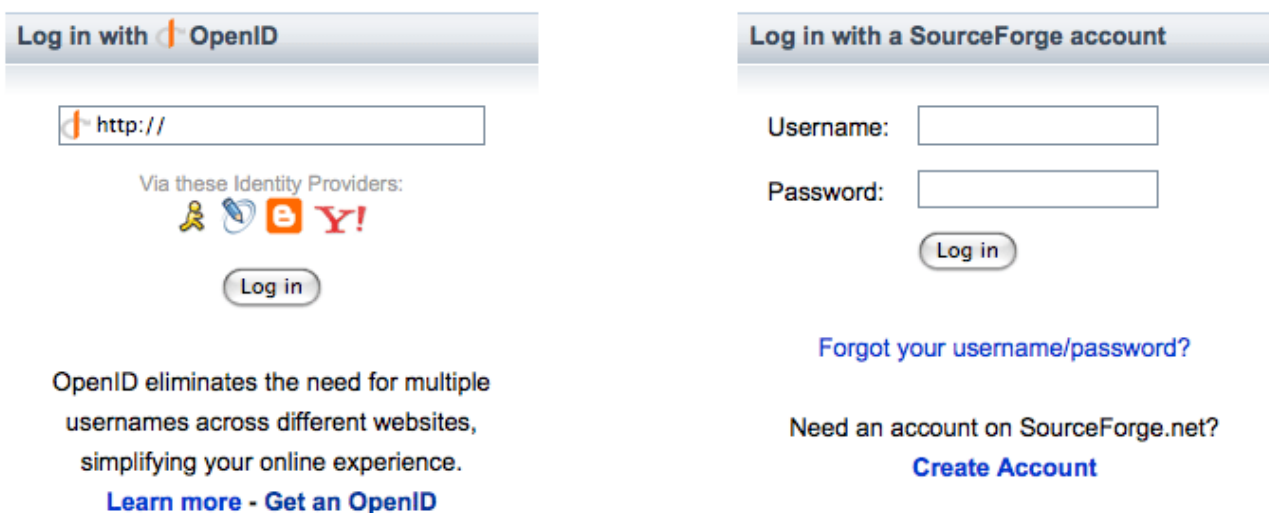


Abbildung 3: OpenID Login gegenüber der Angabe von Benutzername und Passwort

Der Benutzer gibt seinen OpenID-Identifizier an und die Relying Party leitet ihn zur eigentlichen Authentifizierung an den zuständigen Identity Provider weiter. Nachdem der Benutzer sich bei seinem Identity Provider authentifiziert hat, kann der Identity Provider die OpenID Anfrage der Relying Party beantworten. Das folgende Sequenzdiagramm verdeutlicht die einzelnen Schritte des Loginvorgangs.⁴⁸

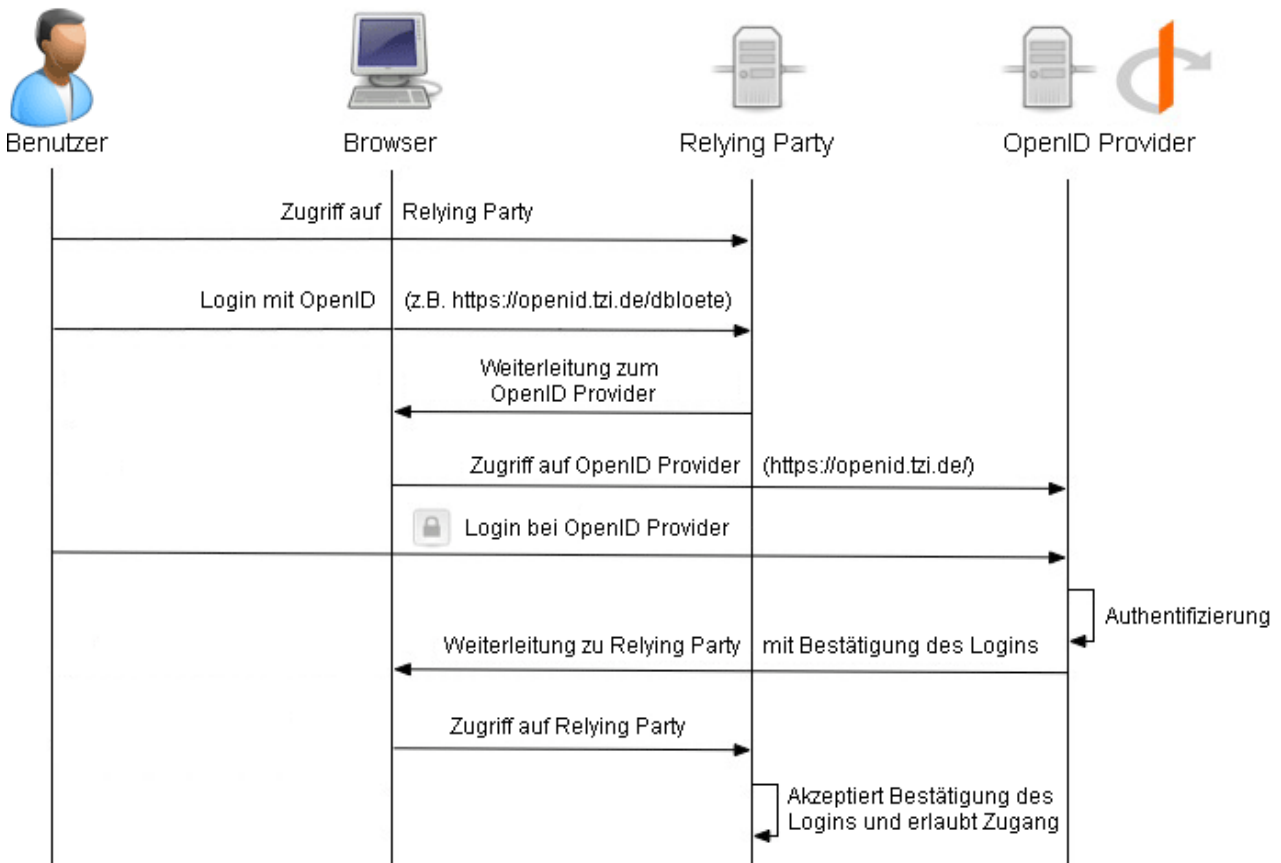


Abbildung 4: Sequenzdiagramm des Authentifizierungsvorgangs mit OpenID⁴⁸

Zusätzlich zur Authentifizierungsbestätigung können auch Benutzerinformationen ausgetauscht werden, sofern der Identity Provider die Erweiterungen Simple Registration oder Attribute Exchange unterstützt. Der genaue Ablauf dieses Vorgangs ist von der Implementierung auf Seiten des Identity Providers abhängig, läuft allerdings üblicherweise nach folgendem Muster ab: Der Benutzer wird nach dem Login nicht direkt an die Relying Party weitergeleitet, sondern bekommt eine Seite präsentiert, auf der er die von der

⁴⁸ Vgl. Govoni, R. (2008): OpenID and Rails, Authentication 2.0, <http://www.devx.com/opensource/Article/37692>

Relying Party angeforderten Attribute sieht und entscheiden kann, welche der Attribute er freigeben möchte. Der folgende Screenshot der im Rahmen dieser Arbeit entwickelten Software⁴⁹ zeigt ein Beispiel einer solchen Auswahl.

Identitätsanfrage

<http://ilvp.informatik.uni-bremen.de/> fragt persönliche Daten an.

Bitte wählen Sie aus, welche Informationen Sie übertragen möchten.

Diese Informationen stammen von ihrer Persona **Standard**. Um andere Daten zu übermitteln, können Sie die [Persona editieren](#) oder [eine andere Persona wählen](#).

Angeforderte Informationen		Datenfreigabe
Vorname	Dennis	<input type="checkbox"/> optional
Nachname	Bloete	<input type="checkbox"/> optional
E-Mail	dbloete@tzi.de	<input checked="" type="checkbox"/> erforderlich
Telefon	+49 163 6845699	<input type="checkbox"/> optional
Website	http://dennisbloete.de/	<input type="checkbox"/> optional
Kennung	dbloete	<input checked="" type="checkbox"/> erforderlich

Durch das Speichern der Datenfreigabe werden zukünftige Identitätsanfragen von ilvp.informatik.uni-bremen.de automatisch mit diesen Informationen beantwortet.

Abbildung 5: Auswahlmaske für die Freigabe von Attributen⁴⁹

Die freizugebenden Daten stammen üblicherweise aus einer Persona, welche der Benutzer beim Identity Provider anlegen kann. Eine Persona beinhaltet personenbezogene Daten und definiert dadurch eine bestimmte Rolle der Identität des Benutzers. Durch das Anlegen verschiedener Personae können somit unterschiedliche Rollen ausgeprägt werden – beispielsweise dienstlich und privat. Dadurch lässt sich die OpenID in verschiedenen Kontexten einsetzen.

⁴⁹ Vgl. OpenID-Server des Fachbereich 3, Universität Bremen: <https://openid.tzi.de/>

Im Anschluss an die Auswahl wird der Benutzer wie im normalen Authentifizierungsvorgang an die Relying Party weitergeleitet. Dabei enthält die Antwort des Identity Providers neben der Authentifizierungsbestätigung auch die vom Benutzer freigegebenen Informationen. Um die Benutzerfreundlichkeit zu erhöhen, kann der Identity Provider dem Benutzer anbieten, die Datenfreigabe für die Relying Party zu speichern. Dadurch kann der Schritt der Attributauswahl bei zukünftigen Anfragen übersprungen werden und der Identity Provider beantwortet die Anfrage auf Basis der gespeicherten Freigabe automatisch.

2.3.3 Potentielle Gefahren und Probleme

2.3.3.1 Phishing

Bei Phishing handelt es sich um ein Problem, das nicht nur OpenID betrifft, dort aber als der gefährlichste Angriffspunkt gilt: Erlangt ein Angreifer Zugriff auf die OpenID eines Benutzers, so stehen dem Angreifer sämtliche Benutzerkonten und Informationen des Opfers zur Verfügung.⁵⁰ Der Ablauf eines OpenID-Phishing Angriffs kann folgendermaßen aussehen⁵¹ und mit der OpenID Phishing Demo⁵² nachgestellt werden:

1. Der Benutzer besucht eine bösertige Relying Party, bei der ihm ein normales OpenID-Loginformular präsentiert wird
2. Der Benutzer gibt seinen OpenID-Identifizier an
3. Die Relying Party leitet den Benutzer zu einem gefälschten Identity Provider – welcher wie der Identity Provider des Benutzers aussieht – weiter
4. Der gefälschte Identity Provider bittet den Benutzer um seine Zugangsdaten
5. Der Benutzer bemerkt nicht, dass er sich nicht bei seinem eigentlichen Identity Provider einloggt und übermittelt die Zugangsdaten
6. Der gefälschte Identity Provider hat nun die Zugangsdaten des Benutzers und kann ihn an die Relying Party weiterleiten, damit der Vorgang nicht auffällt

Die Lösungsmöglichkeiten für dieses Problem sind nicht trivial und erfordern einen Mehraufwand des Benutzers: Der Benutzer muss sicherstellen, dass es sich nur bei seinem wirklichen Identity Provider einloggt. Dies kann durch Überprüfung der Adresszeile

50 Vgl. Kveton, S. (2007): OpenID 2.0 and Phishing, <http://kveton.com/blog/2007/01/21/openid-20-and-phishing/>

51 Vgl. OpenID Wiki: OpenID Phishing Brainstorm, http://wiki.openid.net/OpenID_Phishing_Brainstorm

52 OpenID Phishing Demo (2008): <http://idtheft.fun.de/>

des Browsers oder besser noch durch manuelle Eingabe der URL geschehen. Darüber hinaus gibt es Browserplugins, die den Benutzer unterstützen und ihn zum Login an den richtigen Identity Provider weiterleiten – zum Beispiel das von VeriSign für den Firefox entwickelte Plugin SeatBelt.⁵³

Identity Provider können zusätzliche Sicherheitsmechanismen implementieren, um den Benutzer vor Phishing zu schützen. Lösungsansätze dafür wären beispielsweise die Verwendung von Zertifikaten oder One-Time Passwords, die zur Authentifizierung des Benutzers eingesetzt werden können.

2.3.3.2 Erstellung eines umfangreichen Benutzerprofils

In OpenID werden alle Anfragen der Relying Parties durch den Identity Provider des Benutzers beantwortet: „So werden deren Server automatisch kontaktiert, wenn sich ein Anwender mit seiner OpenID bei einem OpenID-Konsumenten anmeldet. Speichert man diese Daten und wertet sie in Kombination mit den in den Profilen hinterlegten Daten aus, erhält man nicht nur umfangreiche Benutzerprofile, sondern auch weit reichende Informationen bezüglich des Surfverhaltens der Anwender.“⁵⁴

Identity Provider sollten in ihren Datenschutzzinformatoren Bezug auf diesen Punkt nehmen und ihre Nutzer über die Speicherung und gegebenenfalls Verwendung der Verbindungsdaten informieren. Mit der Wahl eines Identity Providers geht der Benutzer ein Vertrauensverhältnis ein, da der Benutzer nicht nur seine Identität, sondern auch Informationen zu seinem Nutzungsverhalten von Websites preisgibt. Identity Provider müssen daher alle erforderlichen Maßnahmen ergreifen, um die Daten der Benutzer zu schützen.

2.3.3.3 Verlust einer OpenID

Ein Benutzer erhält seinen OpenID-Identifizierer durch einen Identity Provider. Da der Identifizierer die Verbindung zwischen dem Benutzer und seinen Benutzerkonten auf Seiten der Relying Parties herstellt, kann der Benutzer sich nicht mehr bei einer Relying Party anmelden, wenn ihm der Identifizierer nicht mehr zur Verfügung steht. Dieser Fall tritt auf, wenn der Identity Provider des Benutzers nicht erreichbar ist – zum Beispiel durch

53 Verisign OpenID SeatBelt Plugin: <https://pip.verisignlabs.com/seatbelt.do>

54 Maaß, C. (et al. 2008): Schlagwort OpenID, http://www.christian-maass.com/wp-content/uploads/2008/05/openid_maas.pdf

temporäre Wartungsarbeiten, Downtime oder sogar dauerhaft durch Konkurs. Letzteres würde zu einem Verlust des Identifiers und somit auch zum Verlust der damit verbundenen Benutzerkonten führen.

Dieses Problem lässt sich durch Delegation beheben: Dabei verwendet der Benutzer eine URL als Identifier, über die er selbst die Kontrolle hat. Auf der unter dieser URL erreichbaren Website kann der Benutzer im HTML-Quelltext die URL seines Identity Providers angeben und die Anfragen dorthin delegieren, beispielsweise:

```
<link rel="openid2.provider openid.server" href="https://openid.tzi.de/server" />  
<link rel="openid2.local_id openid.delegate" href="https://openid.tzi.de/dbloete" />
```

Bei der Discovery ermittelt die Relying Party die URL der delegierenden Website und nutzt diese als OpenID-Identifier.⁵⁵

Darüber hinaus können Relying Parties ihren Nutzern anbieten, das Benutzerkonto mit mehreren OpenID-Identifiern zu verknüpfen. Verliert der Benutzer einen seiner Identifier, kann er sich immer noch mit einem anderen Identifier anmelden und dadurch das Benutzerkonto behalten.

2.4 Identitätsmanagement

Identitätsmanagement bezeichnet die Verwaltung von persönlichen Daten. Dazu zählt neben der Erfassung und Verwertung der identitätsbezogenen Informationen auch der vertrauensvolle Umgang mit ihnen, weshalb auch Datenschutz und Sicherheitsaspekte eine wichtige Rolle spielen.⁵⁶

Das Datenschutzzentrum des Landes Schleswig-Holstein schreibt dazu: „Identitätsmanagement ist in [...] der realen Welt ein von den Menschen seit Tausenden von Jahren eingeübtes Handeln. Je nachdem in welcher Situation oder Rolle man sich befindet, verhält man sich anders und gibt unterschiedliche Informationen von sich preis. So gibt sich derselbe Mensch im Beruf anders als im Privatbereich, in dem er oft eher bereit ist,

55 Vgl. Fitzpatrick, B. (et al. 2007): OpenID Authentication 2.0, Relying Parties, http://openid.net/specs/openid-authentication-2_0.html#anchor38

56 Vgl. Dinger, J.; Hartenstein, H. (2008): Netzwerk- und IT-Sicherheitsmanagement, Eine Einführung, <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/142064>

auch persönliche Dinge von sich zu erzählen. Aber auch einzelnen Menschen gegenüber passt man sein Verhalten instinktiv an. Engen Freunden berichtet man persönlichste Angelegenheiten, während der Verkäuferin im Supermarkt möglichst nur die für die Kaufabwicklung unbedingt erforderlichen Daten mitgeteilt werden. Auch dem Kollegen, von dem man weiss, dass er alles weiterberichtet, erzählt man möglichst wenig und wählt eine reserviertere Identität, um einen Missbrauch seiner persönlichen Daten zu vermeiden.“⁵⁷

Ein Identitätsmanagementsystem soll den Benutzer dabei unterstützen, die Kenntnisse aus dem alltäglichen Umgang mit seiner Identität auf die Anwendung in der digitalen Welt zu übertragen. Es soll dem Anwender ein möglichst hohes Maß an Sicherheit und Kontrollmöglichkeiten bieten, so dass die Interaktion und Kommunikation zu einem transparenten und nutzerseitig gestaltetem Prozess wird. Aus technischer Sicht umfasst der Begriff Identitätsmanagementsystem sämtliche an diesem Prozess beteiligten Komponenten: Neben immateriellen Bestandteilen wie der benötigten Software, Passwörter und Zertifikate zählt auch Hardware dazu – beispielsweise Schlüssel oder Token.

Die fortschreitende Digitalisierung führt dazu, dass das Thema Identitätsmanagement auch im Alltag zunehmend präsenter wird und bedingt die Entwicklung von Systemen, die den Benutzer in den Mittelpunkt stellen und ihm die Kontrolle über seine Identität und Privatsphäre ermöglichen.

2.4.1 Anwendungsfälle im akademischen Umfeld

Das akademische Umfeld stellt ein sehr großes und repräsentatives Anwendungsfeld für Identitätsmanagement dar. Universitäten und Hochschulen haben seit jeher das Problem der Verwaltung von Personendaten, welche zudem aus einer Vielzahl von unterschiedlichen Quellen zusammengeführt werden müssen. Einzelne Einrichtungen wie beispielsweise die Fachbereiche, Institute, Bibliothek oder Rechenzentrum betreiben in der Regel ihre eigenen Systeme, um die Daten der Studenten, des Personals und gegebenenfalls weiterer Personen (zum Beispiel Gasthörer oder Alumni) zu verwalten. Dadurch ergeben sich allein an einer Universität oft eine Vielzahl eigenständiger Benutzerverwaltungen, auf deren Basis Informationssysteme (Raumplanung, Telefon-

⁵⁷ Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein: Was ist Identitätsmanagement?, <https://www.datenschutzzentrum.de/projekte/idmanage/was.htm>

listen, E-Mail-Verteiler) sowie Zugangsberechtigungen (beispielsweise zu Computern und Räumen) geschaffen werden.⁵⁸

Insbesondere im Bereich der Provisionierung – also der Ausstattung der Anwender mit infrastrukturellen Notwendigkeiten – verursacht die getrennte Datenhaltung einen nicht unerheblichen Mehraufwand: Neu hinzukommende Studenten und Mitarbeitern müssen Benutzerkonten und Zugänge erhalten, die beim Austritt aus der Universität auch wieder entzogen werden müssen. Dies birgt zudem die Gefahr potentieller Sicherheitslücken, die durch vergessene oder unterlassene Deprovisionierung an den einzelnen Systemen entstehen können.⁵⁹ Dieser Umstand erschwert nicht nur die Datenpflege, sondern auch die Entwicklung und Integration neuer Dienste und verbindet sie mit zusätzlichen Kosten.

„Das Identitätsmanagement soll diese historisch gewachsene Vielfalt durch eine einheitliche Verwaltung von Personendaten einschließlich zugehöriger Kontaktinformationen, Rollen und (Zugriffs)Rechten sowie von anderen Ressourcen ablösen. Es geht über die reine Identifikation in einem übergeordneten Verzeichnis deutlich hinaus, indem es dafür sorgt, dass dem Nutzer auf allen Systemen, auf denen ihm Rechte zustehen, diese Rechte ohne weitere Anträge und Verwaltungsvorgänge automatisch eingerichtet und ggf. auch wieder entzogen werden (Provisioning). Gleichzeitig werden die ausschließlich von der Anwendung genutzten Daten auch nur von der betreffenden Anwendung gespeichert und genutzt.“⁶⁰

Ziel des Identitätsmanagement im akademischen Umfeld ist jedoch nicht nur die einheitliche Datenhaltung an der jeweiligen Hochschule, sondern auch die Ermöglichung eines (organisationsübergreifenden) Single Sign-Ons. So soll es durch das System möglich sein, durch einmalige Authentifizierung an der Heimatorganisation auch die Dienste anderer Organisationen zu nutzen und entsprechend seiner Berechtigungen darauf zuzugreifen. Ein im akademischen Umfeld übliches Szenario dafür ist, dass ein Student Kurse an verschiedenen Hochschulen belegt. Die Hochschulen bilden in diesem

58 Universität Bielefeld (2006): Zentrale Speicherung von Nutzerdaten, http://www.uni-bielefeld.de/hrz/projekte/znttr_speicherung.html

59 Vgl. Dinger, J.; Hartenstein, H. (2008): Netzwerk- und IT-Sicherheitsmanagement, Eine Einführung, <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/142064>

60 Universität Bielefeld (2006): Zentrale Speicherung von Nutzerdaten, http://www.uni-bielefeld.de/hrz/projekte/znttr_speicherung.html

Fall einen Verbund, in dem sich die einzelnen Parteien gegenseitig vertrauen und Mitgliedern der anderen Organisationen den Zugriff auf die jeweils eigenen Ressourcen gestatten. Durch das Login an der Heimatorganisation haben die Benutzer somit auch Zugang zu Ressourcen der anderen Organisationen, ohne sich dort erneut authentifizieren zu müssen.⁶¹

Auf technischer Seite setzt ein solches System einen Standard voraus, welcher das Protokoll für den Datenaustausch und die auszutauschenden Informationen definiert. Dies ist nötig, da die einzelnen Organisationen oftmals Benutzerdaten zur Verfügung stellen, welche sich durch Inhalt und Benennung unterscheiden. Der für den Verbund gewählte Standard legt die für den Austausch verwendete Bezeichnung und Form der Daten fest, auf deren Basis dann Informationen übermittelt und Berechtigungen vergeben werden können. Der im akademischen Umfeld am häufigsten eingesetzte Standard ist Shibboleth.⁶²

Das Deutschen Forschungsnetz⁶³ hat im November 2007 die DFN-AAI⁶⁴ – eine Infrastruktur für Authentifizierung und Autorisierung – in Produktivbetrieb genommen. Sie stellt eine Föderation auf Basis von Shibboleth dar, der sich Forschungseinrichtungen anschließen können, um Zugangsmechanismen zu vereinheitlichen und somit den Austausch mit anderen Organisationen zu ermöglichen: „In der DFN-AAI müssen die Verantwortungen und Modalitäten der Kommunikation zwischen den beteiligten Partnern geregelt werden. Dies geschieht mithilfe von vertraglichen Vereinbarungen zwischen DFN-Verein, Anwendern und Anbietern, die die Verlässlichkeit der Kommunikationbeziehungen und des Datenaustausch zum Gegenstand haben. Bei der DFN-AAI tritt der DFN-Verein als zentraler Vertragspartner für alle Teilnehmer auf. Für die Anwender bedeutet die Teilnahme an der DFN-AAI, dass sie mit Abschluss eines Vertrages kontrollierten Zugang zu verschiedenen Anbietern bekommen. Umgekehrt erreichen Anbieter mit einem Kontrakt Zugang zu einer Vielzahl von Anwendern.“⁶⁵

61 Vgl. Dinger, J.; Hartenstein, H. (2008): Netzwerk- und IT-Sicherheitsmanagement, Eine Einführung, <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/142064>

62 Shibboleth, <http://shibboleth.internet2.edu>

63 DFN-Verein, <http://www.dfn.de/>

64 DFN-AAI - Authentifikation Autorisierungs Infrastruktur, <https://www.aai.dfn.de/>

65 DFN-AAI (2007): Der Dienst, <https://www.aai.dfn.de/der-dienst/>

Eine solche Föderation stellt immer einen Verbund von Partnerorganisationen dar, die sich vertraglich zur Erfüllung der Rahmenbedingungen verpflichten und sich gegenseitigen Zugang erlauben. Zu diesen Rahmenbedingungen zählen unter anderem die Einhaltung von Sicherheitsstandards und „organisatorische und technische Prozesse, die durchlaufen werden, wenn eine Person in die Einrichtung aufgenommen wird, innerhalb der Einrichtung die Rolle ändert oder die Einrichtung verlässt.“⁶⁶

2.4.2 Identitätsmanagement an der Universität Bremen

An der Universität Bremen gibt es aktuell Bestrebungen, ein zentrales Identitätsmanagementsystem einzuführen, auf dessen Basis auch eine Integration in die DFN-AAI mittels Shibboleth möglich sein soll. Grundlage dafür ist eine Benutzerdatenbank, welche derzeit vom ZfN⁶⁷ geplant und realisiert wird.

Die Informationen der erfassten Personen (Studenten, Mitarbeiter, externe Dozenten, Alumni, etc.) stammen aus verschiedenen Quellen mit jeweils unterschiedlicher Datenqualität: So enthält beispielsweise die Telefondatenbank nur sehr wenige Daten, welche zusätzlich nicht normiert sind und daher manueller Überarbeitung bedürfen. Andere Datenquellen wie zum Beispiel die Studentendatenbank werden über einen festgelegten Workflow erfasst und enthalten bereits einen Großteil der benötigten Daten, weshalb sie einfacher zu integrieren sind.⁶⁸

Die Primärdatenbestände werden aktuell bereits in einer Oracle-Datenbank zusammengeführt – ausgenommen davon sind die Personaldaten, deren Import aus datenschutzrechtlichen Gründen bisher nicht erlaubt ist. Durch die Erfassung der Informationen in der zentralen Oracle-Datenbank erhält jeder Datensatz einen Schlüssel, über den er eindeutig referenziert werden kann. Dadurch wird eine redundante Speicherung von Informationen vermieden und ein regelmäßiger Datenabgleich erleichtert. Die Oracle-Datenbank wird zukünftig als Importquelle eines LDAP-Servers genutzt, welcher der vertrauenswürdigen Benutzerverwaltung dient. Dieser LDAP-Server wird den Anforderungen der DFN-AAI

66 DFN-AAI (2007): IdentityManagement, <https://www.aai.dfn.de/der-dienst/identitymanagement/>

67 Universität Bremen – Zentrum für Netze und verteilte Datenverarbeitung,
<http://www.zfn.uni-bremen.de/zfn>

68 Bergst, O. (2006): Schrittweise Einführung eines Ident-Management Systems in der Universität Bremen (nicht veröffentlichte Arbeitsversion)

entsprechen⁶⁹ und somit eine Integration in die Föderation der DFN-AAI ermöglichen. Zusätzlich soll die Benutzerverwaltung Self-Service Dienste bereitstellen, über welche die Benutzer einen Teil ihrer Daten eigenständig pflegen können – Abbildung 6 verdeutlicht die Verteilung der Daten im geplanten Ident-Management-System des ZfN.⁷⁰ Die Fertigstellung des Systems ist für Ende Juni 2009 geplant. Anschließend soll die Einbindung in die DFN-AAI erfolgen.

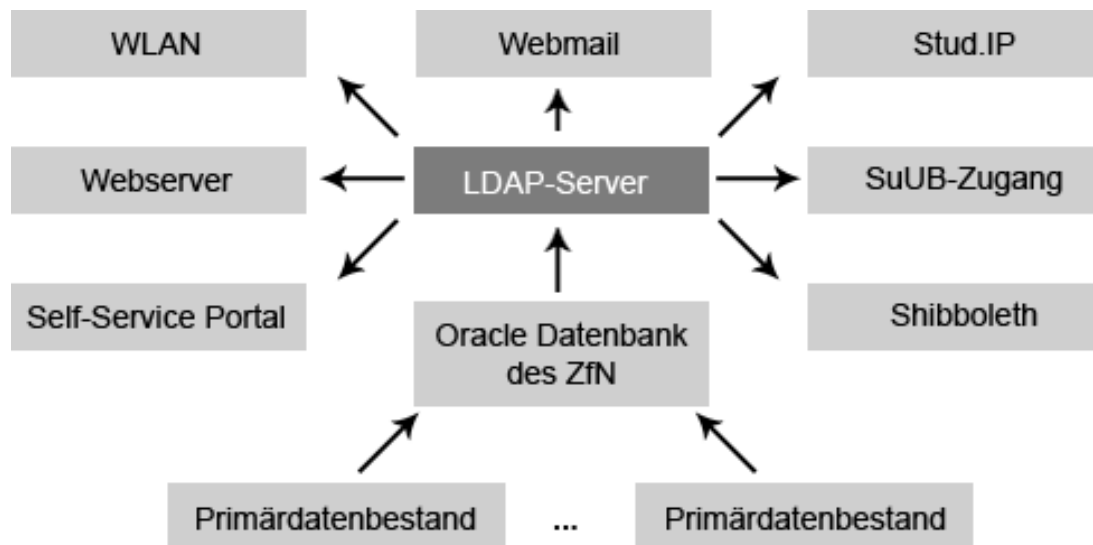


Abbildung 6: Verteilung der Daten im geplanten Ident-Management-System des ZfN⁷⁰

2.4.3 Identitätsmanagement an anderen Hochschulen

Die zentrale Problemstellung des Identitätsmanagements ist wie am Beispiel der Universität Bremen gesehen der Datenimport und -abgleich. Da jede Hochschule unterschiedliche Datenquellen hat, gibt es im akademischen Bereich kein Standard-system, welches zum Identitätsmanagement genutzt wird. An deutschen Hochschulen werden oftmals kommerzielle Lösungen verwendet, welche im Zusammenhang mit einer Beratungsleistung zur Zusammenführung der Daten aus verschiedenen Quellen stehen: „Die Entscheidung für ein bestimmtes kommerzielles Produkt wird meist von vorhandenen oder angestrebten Landeslizenzen bestimmt, da andernfalls die entstehenden Kosten für

69 DFN-AAI (2007): Voraussetzungen zur Teilnahme an der DFN-AAI, <https://www.aai.dfn.de/der-dienst/identitymanagement/#c53>

70 Vgl. Bergst, O. (2006): TaskForce Ident-Management, Künftige Workflows (nicht veröffentlichte Arbeitsversion)

eine Hochschule kaum finanzierbar wären. In Nordrhein-Westfalen etwa wird der Tivoli Identity Manager eingesetzt und seit der Einführung der SUN-Landeslizenz auch der SUN Identity Manager. Dagegen wird in Thüringen und Bayern der Novell Identity Manager bevorzugt. Einige Hochschulen haben sich, meist auf Landesebene, zu Verbänden zusammengefunden (z. B. in NRW und Thüringen), die aufgrund des Einsatzes des gleichen Produkts zusammenarbeiten.⁷¹

So setzt beispielsweise die RWTH Aachen⁷² seit Juli 2004 als weltweit erste Hochschule den angesprochenen Tivoli Identity Manager von IBM⁷³ ein – in Hamburg wird im Rahmen des eCampusII⁷⁴ Projekts eine Identitätsmanagementlösung von Novell verwendet⁷⁵. Als Alternative zur Verwendung einer kommerziellen Lösung kann auch die Eigenentwicklung gesehen werden: Zum Beispiel setzt die Universität Augsburg auf ein selbstentwickeltes Identitätsmanagementsystem auf Basis von OpenLDAP⁷⁶.

Grundlegendes Bestreben bei der Einführung einer Identitätsmanagementlösung – egal ob kommerziell oder selbstentwickelt – ist meistens eine Integration in die DFN-AAI, da somit eine organisationsübergreifende Interoperabilität sichergestellt wird. So nehmen bislang schon über 50 deutsche Hochschulen und angehörige Organisationen an der Föderation des DFN teil⁷⁷.

71 Universität Augsburg (2007): Identity Management - Analyse und Anforderungen, http://www.uni-augsburg.de/einrichtungen/its/teilprojekte/im/paket_im1.html

72 RWTH-Aachen (2007): Tivoli Identity Manager, <http://www.rz.rwth-aachen.de/ca/c/pys/lang/de/>

73 IBM: Tivoli Identity Manager, <http://www-306.ibm.com/software/tivoli/products/identity-mgr/>

74 Multimedia Kontor Hamburg - eCampus II: IT-Modernisierung der Hochschulen, <http://www.mmkh.de/index.php?idcat=43>

75 Moenig, M.; Winklmeier, S. (2007): Single Sign on, Identitätsmanagement, <http://www.campus-innovation.de/idm>

76 Universität Augsburg (2007): Identity Management, <http://www.uni-augsburg.de/einrichtungen/its/teilprojekte/im/index.html>

77 DFN-AAI (2007): Teilnehmerliste, <https://www.aai.dfn.de/verzeichnis/teilnehmer/>

3 Entwicklung des Pilotsystems

3.1 Beschreibung des Pilotsystems

3.1.1 Anforderungen

Mit Hilfe der zu entwickelnden Lösung soll es zukünftig möglich sein, auf der bestehenden Datenbasis aufzusetzen und die Einbindung der zentralen Benutzerkonten in Anwendungen des Fachbereich 3 zu vereinfachen. Das Identitätsmanagementsystem soll ein Single Sign-On für alle unterstützten Anwendungen bieten, so dass diese mittels des Fachbereich 3-Benutzerkontos zugänglich sind und keine erneute Registrierung notwendig ist. Auf Seiten der Anwendungen soll dadurch das Einrichten einer eigenständigen Benutzerverwaltung entfallen, damit neue Software einfacher und schneller entwickelt werden kann (vgl. Situated Software).

Zusätzlich soll das Identitätsmanagementsystem dem Benutzer erlauben, seine persönlichen Daten in begrenztem Umfang selbst zu pflegen und diese gezielt an die verschiedenen Anwendungen zu übertragen. Dadurch kann das System zukünftig beispielsweise von Studenten genutzt werden, um sich zu Lehrveranstaltungen eintragen oder diese anonym evaluieren zu können. Darüber hinaus lassen sich die Attribute auch zur einfachen Berechtigungsvergabe nutzen, beispielsweise um den Studenten- oder Mitarbeiterstatus oder die Zugehörigkeit zu einem Studienfach nachzuweisen.

3.1.2 Vorgaben und Rahmenbedingungen

Der Fachbereich 3 verfügt über ein LDAP-Verzeichnis, mit dem die Benutzerdaten von Studenten, Mitarbeitern und weiteren Personen (beispielsweise Gasthörer) verwaltet werden. Die Einträge des LDAP-Verzeichnis dienen als zentrales Benutzerkonto für die jeweilige Person, auf dessen Basis der Zugriff auf Dienste wie E-Mail und WLAN gewährt wird. Die Benutzerdaten werden bei der Immatrikulation oder Anstellung erfasst und das Benutzerkonto wird gelöscht, wenn ein Student oder Mitarbeiter die universitären Laufbahn verlässt.

Eine Anbindung des LDAP-Verzeichnis erlaubt es, auf den bestehenden Benutzerkonten aufzubauen, so dass das Verzeichnis weiterhin als zentrale Quelle dient und die Anwender

der Identitätsmanagementlösung kein separates Benutzerkonto benötigen. Die folgende Tabelle enthält eine Liste der im LDAP-Verzeichnis zur Verfügung stehenden Attribute, die sich für den Einsatz im Identitätsmanagement eignen.

Attributname	Beschreibung	Beispiel
uid	Login-ID / Benutzername	dbloete
uidNumber	Eindeutige Benutzernummer	1234
cn	Name des Benutzers	Bloete, Dennis
displayName	Anzeigename des Benutzers	Dennis Bloete
UDBstatus	Art der Zugehörigkeit einer Person zur Universität	S (Student), M (Mitarbeiter), G(Gasthörer), X (Sonderstatus)
UDBdate	Datum des Studienbeginns	2005-10-04
UDBstuga	Studiengang	Medieninformatik
UDBmatnr	Matrikelnummer	2083225

Darüber hinaus lässt sich aus dem Attribut *uid* die E-Mail-Adresse des Benutzers ableiten, da sie sich aus *uid@tzi.de* beziehungsweise *uid@informatik.uni-bremen.de* zusammensetzt.

3.1.3 Motivation für den Einsatz von OpenID

Der Fachbereich 3 benötigt eine Lösung, welche auf der bestehenden Datenbasis des LDAP-Servers aufsetzt und ein leichtgewichtiges Identitätsmanagement ermöglicht. Die Wahl der Technologie für die Implementierung des Pilotsystems fiel auf OpenID, da es sich dabei im Vergleich zu Shibboleth um eine einfach zu integrierende Lösung handelt: Es wird keine komplexe Infrastruktur oder Zertifizierung benötigt und die Komponenten Identity Provider und Relying Party lassen sich unabhängig von der verwendeten Programmiersprache integrieren. Der wichtigste Unterschied zwischen OpenID und Shibboleth ist in diesem Fall, dass sich die Einbindung der Lösung in Relying Party Anwendungen mit OpenID einfacher gestaltet als mit Shibboleth und bereits Bibliotheken und Referenzimplementierungen⁷⁸ für eine Vielzahl von Programmiersprachen vorhanden sind.

⁷⁸ OpenID Wiki: Libraries, <http://wiki.openid.net/Libraries>

Da es sich bei OpenID um einen offenen und auf Dezentralität ausgelegten Standard handelt, lässt sich die vom Fachbereich 3 bereitgestellte OpenID Identität auch außerhalb des universitären Kontext einsetzen. Die OpenID kann zukünftig auch genutzt werden, um sich bei anderen Websites einzuloggen, vorausgesetzt diese unterstützen OpenID.

3.1.4 Technische Anforderungen

Für die Integration eines OpenID Identity Providers wird eine entsprechende Server-Software benötigt. Die Software wurde im Rahmen dieser Arbeit mit der Programmiersprache Ruby⁷⁹ entwickelt und basiert auf dem Webapplikationsframework Ruby on Rails⁸⁰. Zusätzlich wurden die Ruby Gems `ruby-ldap`⁸¹, `ruby-yadis` und `ruby-openid`⁸² verwendet. Der Kern der Software wurde unter dem Projektnamen `masquerade`⁸³ als Open Source Software veröffentlicht. Die speziell für den Fachbereich 3 entwickelten Komponenten (beispielsweise die Anbindung an das LDAP-Verzeichnis) wurden als Erweiterungen hinzugefügt. Abbildung 7 zeigt die Schichten der Identity-Provider Software.

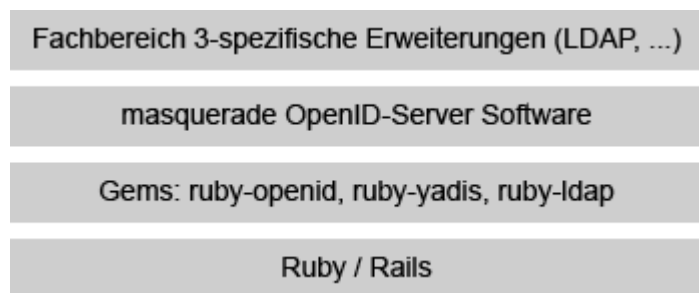


Abbildung 7: Software-Schichten des Identity-Providers

Der dokumentierte Quelltext der Anwendung befindet sich in einem Git⁸⁴-Repository, das den Projektbetreuern zugänglich ist. Die Dokumentation des Quelltexts enthält die für das Verständnis und die Pflege nötigen Kommentare, so dass sich die weiteren Ausführungen in dieser Arbeit nur auf einzelne Details beschränken.

79 Ruby: <http://www.ruby-lang.org/de/>

80 Ruby on Rails: <http://www.rubyonrails.org/>

81 Ruby/LDAP: `ruby-ldap`, <http://ruby-ldap.sourceforge.net/>

82 OpenID Enabled: `ruby-openid`, <http://openidenabled.com/ruby-openid/>

83 Blöte, D. (2008): `masquerade` Documentation, <http://dennisbloete.de/projects/masquerade>

84 Git: <http://git.or.cz/>

Installiert wurde die Identity Provider-Software auf dem Apache-Webserver des Fachbereich 3 – zusätzlich wird der Applikationsserver Mongrel⁸⁵ benötigt. Als Datenbankmanagementsystem kommt MySQL⁸⁶ zum Einsatz.

3.1.5 Der Identity Provider

Der Identity Provider ist unter der URL *https://openid.tzi.de/* erreichbar. Alternativ wäre es möglich gewesen, die Domain *openid.informatik.uni-bremen.de* zu verwenden. Davon wurde abgesehen, da es bei manueller Eingabe der Identity-URL mehr Schreibaufwand bedeutet hätte, was aus Sicht der Benutzerfreundlichkeit nicht erwünscht ist. Die Identity-URL eines Benutzers setzt sich aus *https://openid.tzi.de/* und der jeweiligen UID zusammen, beispielsweise *https://openid.tzi.de/dbloete*.

Die Identity Provider-Software setzt auf dem bestehenden LDAP-Verzeichnis auf. Dieser Ansatz wurde gewählt, weil dadurch die bereits existierenden Benutzerkonten genutzt werden können und keine erneute Registrierung am Identity Provider notwendig ist. Das Login beim Identity Provider erfolgt somit über die Zugangsdaten des normalen Benutzerkontos (UID und Passwort). Eine anderer Lösungsansatz wäre die Einführung eines separaten Benutzerkontos für die Nutzung des Identity Providers gewesen: Dies hätte allerdings eine erneute Registrierung und eine Verifizierung der Identität – zum Beispiel über einen Aktivierungslink an die E-Mail-Adresse des Benutzers – bedeutet.

Der gewählte Lösungsansatz erfordert dennoch eine Benutzerdatenbank am Identity Provider, in welcher die benötigten Attribute der Benutzer vorgehalten werden. Dies ist notwendig, da der LDAP-Server nur zur Authentifizierung der Benutzer und zur Bereitstellung ihrer Attribute verwendet werden kann. Die Lösung mit einer eigenständigen Benutzerdatenbank ist notwendig, da dem Identity Provider aus Sicherheitsgründen keine generelle Sicht auf das LDAP-Verzeichnis zur Verfügung steht und die Verbindung nur im Kontext des jeweiligen Benutzers stattfinden kann. Der Ablauf dabei ist folgendermaßen: Der Benutzer loggt sich am Identity Provider mit seinen Fachbereich 3 Benutzerdaten ein, woraufhin sich der Identity Provider mit diesen Benutzerdaten zum LDAP-Server verbindet. Ist die Authentifizierung erfolgreich, so werden die benötigten Attribute⁸⁷ vom

85 Mongrel: <http://mongrel.rubyforge.org/>

86 MySQL: <http://www.mysql.de/>

87 Vgl. 3.1.2 Vorgaben und Rahmenbedingungen

LDAP-Eintrag des Benutzers in sein Benutzerkonto am Identity Provider übernommen und der Benutzer eingeloggt. Es findet somit bei jedem Login auch ein Datenabgleich zwischen LDAP-Verzeichnis und Identity Provider statt. Das erste erfolgreiche Login am Identity Provider erstellt automatisch ein Benutzerkonto für die jeweilige Person.

Abbildung 8 verdeutlicht den Aufbau des Identitätsmanagementsystems und das Zusammenspiel der beteiligten Komponenten.⁸⁸

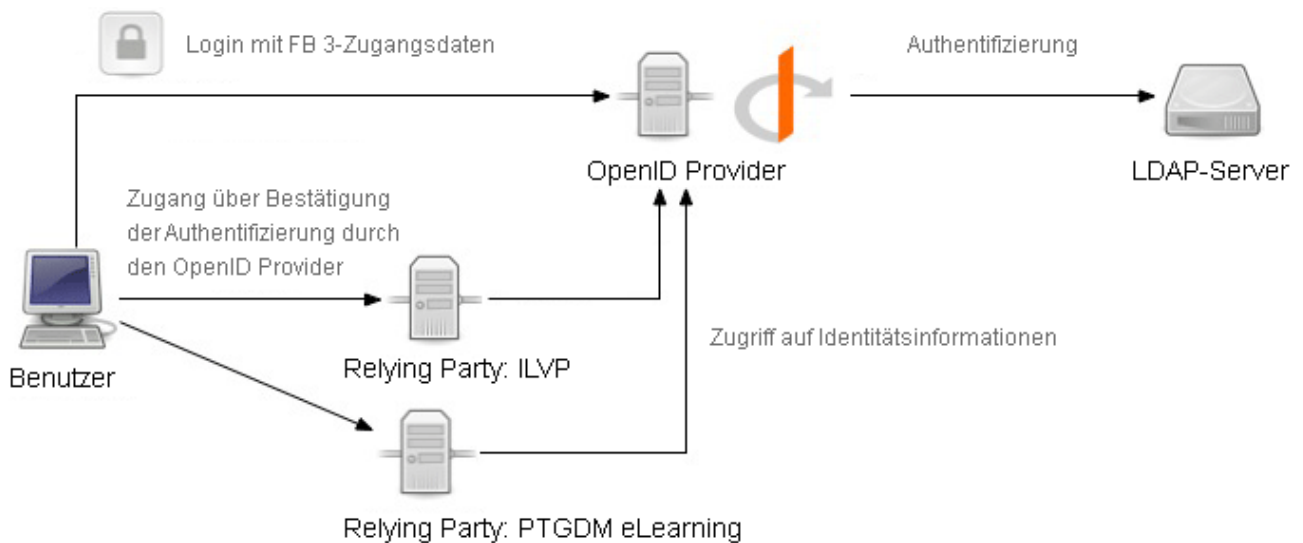


Abbildung 8: Aufbau des Identitätsmanagementsystems⁸⁸

3.2 Integration in eine Relying Party

3.2.1 Login-Formular

Die Integration des Login-Formulars bei der Relying Party kann auf zwei Arten geschehen: Entweder die Relying Party präsentiert dem Benutzer ein Textfeld zur Eingabe seiner OpenID-URL (vgl. Abbildung 3: OpenID Login gegenüber der Angabe von Benutzername und Passwort) oder einen einfachen Button, der ihn zu seinem Identity Provider weiterleitet. Das Login ohne Angabe der OpenID-URL wird in diesem Fall dadurch ermöglicht, dass der Identity Provider des Benutzers bekannt ist und OpenID seit Version 2.0 das Konzept Directed Identity unterstützt: „With OpenID 2.0, the discovery process may tell the [Relying Party] that the URL identifies the [Identity Provider] rather than the *user*. If this

⁸⁸ Vgl. Govoni, R. (2008): OpenID and Rails, Authentication 2.0, <http://www.devx.com/opensource/Article/37692>

happens, the RP proceeds with the authentication request using the special 'http://specs.openid.net/auth/2.0/identifier_select' value as the identity URL. The [Identity Provider] will then fill in the user's actual identity URL in the subsequent authentication response."⁸⁹

Dem Benutzer wird die manuelle Eingabe seiner OpenID-URL erspart, da es im Fachbereich 3 nur einen Identity Provider gibt, an den somit direkt weitergeleitet werden kann. Der Identity Provider sendet bei der Beantwortung der Authentifizierungsanfrage dann den eigentlichen Identifier mit, so dass der Benutzer auf Seiten der Relying Party eindeutig zu identifizieren ist. Dieser Lösungsansatz wurde zur Verbesserung der Benutzbarkeit bereits in die Relying Party Anwendung ILVP⁹⁰ integriert – Abbildung 9 verdeutlicht dies.

Login mit OpenID

Wenn Sie ein FB 3-Benutzerkonto haben, können Sie sich mit Ihrer OpenID am ILVP anmelden. In diesem Fall entfällt auch der Registrierungsvorgang für den ILVP.



Login mit Benutzername und Passwort

Login

Passwort

angemeldet bleiben

[Registrieren](#)

Abbildung 9: Formular mit OpenID-Button und herkömmlichem Login⁹⁰

Die Lösung mit der manuellen Eingabe des Identifiers eignet sich, wenn eine Relying Party auch OpenIDs anderer Identity Provider akzeptiert und Personen außerhalb des Fachbereich 3 das Login ermöglicht werden soll.

Da mittels OpenID selbst nur die Verifikation des Identifier-Besitzes möglich ist, erfolgt der Datenaustausch über die Protokoll-Erweiterungen Attribute Exchange und Simple Registration. Die Voraussetzung dafür ist jeweils, das Relying Party und Identity Provider das selbe Set an Attributen unterstützen. Die folgenden Abschnitte spezifizieren die Attribute, die mittels des Identitätsmanagementsystems ausgetauscht werden können.

⁸⁹ Henstridge, J. (2007): OpenID 2.0, <http://blogs.gnome.org/jamesh/2007/10/23/openid-20/>

⁹⁰ Interner Lehrveranstaltungs-Planer: Login, <http://ilvp.informatik.uni-bremen.de/login>

3.2.2 Attribute Exchange

Für den Datenaustausch per Attribute Exchange gibt es bislang nur AXSchema.org⁹¹ als einzigen Standard, der Type Identifier für Attribute festlegt. Die dort definierten Type Identifier decken jedoch nur einige Attribute eines Benutzerprofils ab, weshalb im Rahmen dieser Arbeit eigene Type Identifier für die Attribute des Fachbereich 3 festgelegt werden mussten. Dies geschah im Hinblick auf bestehende Standards, um bereits existierende Bezeichner zu verwenden und so Portabilität zu ermöglichen. Als Grundlage dafür wurde die Objektklassen eduPerson und ihre Erweiterung swissEduPerson genommen. swissEduPerson⁹² wurde von SWITCH für den Schweizer AAI Verbund definiert und beinhaltet zusätzliche Attribute, welche im deutschsprachigen universitären Umfeld benötigt werden (beispielsweise Matrikelnummer und Studiengang). Da der Attributspezifikation der DFN-AAI⁹³ einige der im Fachbereich 3 verfügbaren Attribute fehlen, wurde auf die Spezifikation der SWITCH-AAI zurückgegriffen.

Die folgende Tabelle zeigt die Attribute, welche sich mittels Attribute Exchange anfragen lassen. Der Type Identifier setzt sich dabei aus *http://openid.tzi.de/spec/schema/* und dem Attributnamen zusammen, beispielsweise also *http://openid.tzi.de/spec/schema/uid*.

Attributname	Beschreibung	Beispiel
uid	Login-ID / Benutzername	dbloete
mail	E-Mail-Adresse	dbloete@tzi.de
principalName	Netz-ID: Der Wert besteht aus einem linken und einem rechten Teil getrennt durch @, wobei der rechte Teil die Domain und der linke Teil eine innerhalb der Domain eindeutige ID beschreibt.	dbloete@informatik.uni-bremen.de
degreeProgram	Studiengang eines Studenten	Medieninformatik
matriculationNumber	Matrikelnummer	1234567

91 Sxip Inc. (2007): Schema for OpenID Attribute Exchange, <http://www.axschema.org/>

92 Vgl. SWITCH AAI (2007): Attribute Specification, http://www.switch.ch/aai/docs/AAI_Attr_Specs.pdf

93 Vgl. DFN-AAI (2006): Technische und organisatorische Voraussetzungen, <https://www.aai.dfn.de/fileadmin/documents/vertraege/attribute.20061130.pdf>

affiliation	Zugehörigkeit: Spezifiziert verschiedene Kategorien für die Art der Zugehörigkeit einer Person zur Heimatorganisation	faculty, student, staff, alum, member, affiliate, employee
displayName	Angezeigter Name	Dennis Blöte
givenName	Vorname	Dennis
surName	Nachname	Blöte
telephoneNumber	Telefonnummer: Landesvorwahl + Ortsvorwahl ohne Null + Teilnehmernummer	+49 421 1234567
postalAddress	Postadresse, Zeilen durch \$ voneinander getrennt.	Musterstr.78\$28199 Bremen

Zusätzlich dazu können die Attribute Adresse, Postleitzahl, Stadt, Bundesland, Land, Sprache, Geburtsdatum, Geschlecht und Website-URL auch über die auf AXSchema.org festgelegten Type Identifier angefragt werden. Dies wurde ermöglicht, damit die OpenID auch zum Datenaustausch mit Relying Parties außerhalb des Fachbereich 3 genutzt werden kann. Eine vollständige Liste aller Attribute und ihrer Type Identifier findet sich in der Spezifikation der Attribut Schemata⁹⁴.

3.2.3 Simple Registration

Die Erweiterung Simple Registration ist durch die Verwendung von Attribute Exchange veraltet, wird jedoch vom Identity Provider noch unterstützt, um eine Abwärtskompatibilität zu gewährleisten und um die Nutzung der OpenIDs mit Relying Parties außerhalb des Fachbereich 3 zu ermöglichen. Im Kontext des Fachbereich 3 sollten die Attributanfragen jedoch möglichst mit Attribute Exchange gestellt werden, da darüber alle Attribute verfügbar sind.

Die folgende Tabelle zeigt die Bezeichner für den Teil der Attribute, der sich mittels der Simple Registration Erweiterung anfragen lässt.⁹⁵

94 Identity Provider: Spezifikation der Attribut Schemata, <https://openid.tzi.de/spec/schema>

95 Vgl. Identity Provider: Spezifikation der Simple Registration Daten, <https://openid.tzi.de/spec/sreg>

Attributname	Beschreibung	Beispiel
nickname	Login-ID, welche den Benutzernamen spezifiziert	dbloete
email	E-Mail-Adresse	dbloete@tzi.de
fullname	Anzeigename des Benutzers	Dennis Bloete
postcode	Postleitzahl	28199
country	Land	DE
language	Sprache	de
dob	Geburtsdatum	1980-12-31
gender	Geschlecht	m / f

3.2.4 Referenzimplementierung einer Relying Party

Um die Integration in neue Anwendungen zu vereinfachen, wurde eine Referenzimplementierung für Relying Parties entwickelt. Dieses Beispiel ist ebenso wie der Identity Provider mit Ruby on Rails programmiert und kann in der Dokumentation eingesehen werden.⁹⁶

3.2.5 Einbindung in bestehende Anwendungen

Auch bei bereits existierenden Anwendungen besteht die Möglichkeit, nachträglich ein Login über OpenID zu integrieren. Dazu müssen die Benutzerkonten mit der OpenID des jeweiligen Benutzers verbunden werden, so dass diese beim Login mit ihrer OpenID identifiziert werden können. Ein kommentiertes Beispiel für diesen Anwendungsfall befindet sich ebenfalls in der Dokumentation.⁹⁶

⁹⁶ Identity Provider: Dokumentation, Referenzimplementierung: <https://openid.tzi.de/spec/example>

3.3 Sicherheitsaspekte

3.3.1 Authentifizierung

Aus Sicht der Relying Party Anwendungen stellt sich die Frage, welchen OpenIDs vertraut werden soll, da die Authentifizierung an den Identity Provider ausgelagert wird: „By accepting OpenID on your site you are *outsourcing the security of your users* to an unknown third party, and you can't guarantee that your users picked a good home for their OpenID. If you're a bank or a healthcare provider that's not a risk you want to take; whitelisting providers that you have audited for security means you don't have to rule out OpenID entirely.“⁹⁷

Für viele Relying Party Anwendungen des Fachbereich 3 empfiehlt es sich daher, nur OpenIDs des lokalen Identity Providers zu akzeptieren und Identitäten anderer Identity Provider auszuschließen – dieses Vorgehen ist exemplarisch bereits in der Referenzimplementierung einer Relying Party integriert. Für den Fall, dass auch andere OpenIDs akzeptiert werden sollen, kann die Whitelist der Identity Provider erweitert werden – zum Beispiel um Personen außerhalb des Fachbereich 3 das Login zu ermöglichen.

3.3.2 Verifizierte Attribute

In OpenID gibt es für Identity Provider bislang noch keine Möglichkeit, zusätzlich zu den Attributen auch Informationen zur Verifizierung dieser zu übermitteln. Dies wäre insbesondere im vorliegenden Fall wünschenswert, da der Identity Provider des Fachbereich 3 nicht nur Attribute aus dem LDAP-Verzeichnis, sondern auch vom Benutzer selbst gepflegte Informationen anbietet. Die aus dem LDAP-Verzeichnis übernommenen Attribute können vom Benutzer nicht editiert werden und könnten somit als verifiziert gelten, wohingegen die vom Benutzer gemachten Angaben nicht auf Echtheit überprüft werden können. Bei den verifizierten Attributen handelt es sich um *uid*, *mail*, *affiliation*, *principalName*, *degreeProgram*, *matriculationNumber*, *displayName*, *givenName* und *surName*.

⁹⁷ Willison, S. (2008): The point of “Open” in OpenID, <http://simonwillison.net/2008/Jun/24/openid/>

Um den beschriebenen Anwendungsfall abzudecken, gibt es bisher lediglich den Entwurf einer Spezifikation⁹⁸, welche im November 2006 veröffentlicht und seitdem nicht mehr aktualisiert wurde. Da es somit aktuell noch keine Lösung für die Verifizierung von Attributen gibt, sollten die Relying Parties nur bei den genannten Attributen auf deren Korrektheit vertrauen.

3.3.3 Autorisierung

Auf Basis der vom Identity Provider angebotenen Attribute gibt es bislang wenig Möglichkeiten zur Autorisierung. Das Attribut *affiliation* (Zugehörigkeit) kann genutzt werden, um zwischen Studenten und Mitarbeitern zu unterscheiden und mittels des Attributs *degreeProgram* (Studiengang) kann eine Zugangsbeschränkung auf einzelne Studiengänge festgelegt werden. Darüber hinaus können Benutzer über ihre UID eindeutig identifiziert und mit individuellen Berechtigungen ausgestattet werden. Um umfangreichere Autorisierungsmöglichkeiten anzubieten, könnte das LDAP-Verzeichnis um Attribute, welche sich zur Ableitung von Berechtigungen eignen, erweitert werden.

3.3.4 Phishing

Zur Lösung des Phishingproblems konnte bislang kein zufriedenstellender Ansatz gefunden werden. Die Lösung, den Benutzer vor dem Login zunächst auf eine Seite ohne Loginformular oder Links darauf weiterzuleiten und ihn von dort aus manuell zur Loginseite navigieren zu lassen wurde aus Gründen der Benutzerfreundlichkeit verworfen.

Eine gute Möglichkeit Phishing zu verhindern, ist der Einsatz des von Verisign entwickeltes Firefox Plugins SeatBelt⁹⁹, was jedoch erfordert, dass der Benutzer den Firefox verwendet und das Plugin eigenständig installiert. Fügt man, wie in Abbildung 10 gezeigt, einen Identity Provider hinzu, so kann man mittels SeatBelt immer auf die korrekte Loginseite gelangen, beziehungsweise diese automatisch beim Start der Browsersession besuchen und sich beim richtigen Identity Provider anmelden.

98 Hardt, D. (2006): OpenID Signed Assertions 1.0 – Draft 1, <http://www.mail-archive.com/specs@openid.net/msg00907.html>

99 Verisign OpenID SeatBelt Plugin: <https://pip.verisignlabs.com/seatbelt.do>

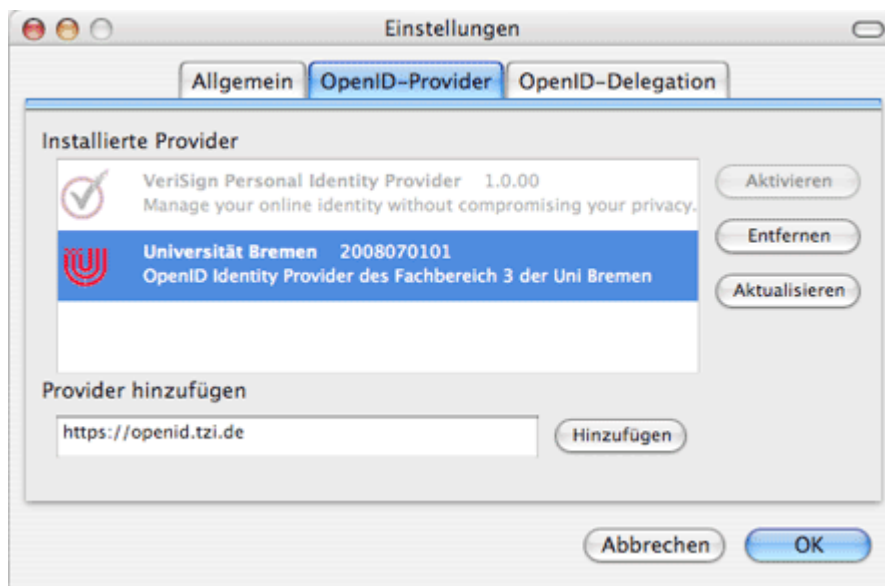


Abbildung 10: SeatBelt: Hinzufügen eines Identity Providers

3.3.5 Haltbarkeit bzw. Verlust der OpenID

Die OpenID steht einem Benutzer bislang nur für den Zeitraum der Verbundenheit mit der Universität zur Verfügung. Mit Beendigung des Studiums beziehungsweise der Kündigung wird das Benutzerkonto im LDAP-Verzeichnis entfernt, so dass die Person sich nicht mehr am Identity Provider anmelden kann. Dadurch verliert der Benutzer die OpenID und somit auch den Zugang zu den Relying Parties.

Dies mag im Kontext der Fachbereich 3-Anwendungen wünschenswert sein, kann aber ein Problem darstellen, falls die OpenID auch genutzt wurde, um sich bei Anwendungen außerhalb des Fachbereich 3 einzuloggen. Der Zugang zu Relying Parties im Internet ginge somit ebenfalls verloren, da der Identifier, mit dem man sich ehemals dort eingeloggt hat, nicht mehr zur Verfügung steht. Umgehen lässt sich dieses Problem nur durch die bereits beschriebene Delegation, mit der der Benutzer eine eigene URL als OpenID nutzen kann und von dort aus auf den Identity Provider des Fachbereich 3 verweist. Eine entsprechende Beschreibung befindet sich im Hilfe-Bereich des Identity Providers.

3.3.6 Erstellung eines Benutzerprofils

Am Identity Provider werden keine Daten vorgehalten, welche das Nutzerverhalten protokollieren, der Benutzer kann sich jedoch dazu entscheiden, die Attributfreigaben für einzelne Relying Parties zu speichern. Darüber ließe sich feststellen, welcher Benutzer welche Anwendung bereits genutzt hat, jedoch nicht zu welchem Zeitpunkt und wie oft. Die Anwendung selbst protokolliert jeden Seitenzugriff in einer Logdatei: Diese ist jedoch für das Debugging der Anwendung gedacht und eignet sich nicht, um Nutzerverhalten auszuwerten, da nicht protokolliert wird, welcher Benutzer welchen Seitenzugriff tätigt.

4 Zusammenfassung und Ausblick

Das Identitätsmanagementsystem wurde fertiggestellt und der Kern des Projekts wurde als Open Source Software unter dem Projektnamen *masquerade*¹⁰⁰ veröffentlicht. Mit den darüber hinaus für den Fachbereich 3 entwickelten Erweiterungen konnte die Software in die bestehende Infrastruktur des Fachbereich 3 integriert werden, so dass zukünftig entwickelte Anwendungen das zentrale Benutzerkonto anbinden können. Um ein Testszenario für die aufgestellten Anwendungs-fälle zu haben, wurde die OpenID-Unterstützung bereits in eine Anwendung (ILVP¹⁰¹) integriert. Der interne Lehrveranstaltungsplaner kann zukünftig als Beispiel für die Integration in das Identitätsmanagementsystem dienen.

Um die Integration von OpenID in weitere Relying Parties zu vereinfachen wurde eine umfangreiche Dokumentation¹⁰² erstellt, welche auf der Website des Identity Providers abrufbar ist. Die Dokumentation umfasst neben den Spezifikationen der Attribute auch Referenzimplementierungen für die Anfrage von Attributen und die nachträgliche OpenID-Integration in bereits bestehende Anwendungen¹⁰³. Somit kann die Dokumentation als Basis für den Ausbau des Identitätsmanagementsystems und die Implementierung weiterer Relying Parties genutzt werden.

Das Identitätsmanagementsystem wurde bis zum Abschluss dieser Arbeit nur im Rahmen von Tests eingesetzt, wodurch zumindest sichergestellt werden konnte, dass es die zuvor definierten Anwendungsfälle erfüllt. Ein Praxiseinsatz war bis zum Ende der Arbeit leider nicht möglich, so dass keine abschließende Bewertung des Projekts auf Basis von Evaluierungsergebnissen oder Befragung von Anwendern vorgenommen werden konnte. Durch die im Testeinsatz gewonnenen Erkenntnisse lässt sich jedoch die Behauptung aufstellen, dass OpenID sich für den Einsatz zum leichtgewichtigen Identitätsmanagement eignet und die erforderlichen Anwendungsfälle mittels OpenID abgedeckt werden können.

100Bloete, D. (2008): *masquerade* Documentation, <http://dennisbloete.de/projects/masquerade>

101Interner Lehrveranstaltungsplaner des Fachbereich 3, <http://ilvp.informatik.uni-bremen.de/>

102Identity Provider: Dokumentation, <https://openid.tzi.de/spec/>

103Identity Provider: Referenzimplementierungen, <https://openid.tzi.de/spec/example>

Sollte das Identitätsmanagementsystem zukünftig praktisch eingesetzt werden, kann der aktuelle Stand ausgebaut und erweitert werden. Der wichtigste Punkt dabei ist, dass die OpenID-Unterstützung in weitere Anwendungen integriert wird. Technische Erweiterungsmöglichkeiten des Identitätsmanagementsystems wären die Anbindung an das kommende Uni-weite Ident-Management System oder die Ergänzung des LDAP-Verzeichnisses um weitere eduPerson-Attribute, wie beispielsweise Entitlements, um strukturierte Rechtevergabe zu ermöglichen. Zudem ließe sich die Benutzerfreundlichkeit der Identity Provider-Software durch Benutzerbefragungen oder Usabilitytests verbessern.

5 Quellenverzeichnis

- Bergst, O. (2006): Schrittweise Einführung eines Ident-Management Systems in der Universität Bremen (nicht veröffentlichte Arbeitsversion)
- Bergst, O. (2006): TaskForce Ident-Management, Künftige Workflows (nicht veröffentlichte Arbeitsversion)
- Cameron, K. (2006): The laws of identity, <http://www.identityblog.com/?p=352>, aufgerufen am 11.07.2008
- Dr. Degenhardt, W.: Benutzerzentriertes Identitätsmanagement, <http://www.netigator.de/netigator/live/show.php3?id=47&aid=31227966>, aufgerufen am 11.07.2008
- DFN-AAI (2006): Technische und organisatorische Voraussetzungen, <https://www.aai.dfn.de/fileadmin/documents/vertraege/attribute.20061130.pdf>, aufgerufen am 10.07.2008
- DFN-AAI (2007): Der Dienst, <https://www.aai.dfn.de/der-dienst/>, aufgerufen am 10.07.2008
- DFN-AAI (2007): IdentityManagement, <https://www.aai.dfn.de/der-dienst/identitymanagement/>, aufgerufen am 10.07.2008
- DFN-AAI (2007): Teilnehmerliste, <https://www.aai.dfn.de/verzeichnis/teilnehmer/>, aufgerufen am 10.07.2008
- DFN-AAI (2008): Identity-Provider, <https://www.aai.dfn.de/dokumentation/identity-provider/>, aufgerufen am 10.07.2008
- DFN-AAI (2008): Service-Provider, <https://www.aai.dfn.de/dokumentation/service-provider/>, aufgerufen am 10.07.2008
- Dinger, J.; Hartenstein, H. (2008): Netzwerk- und IT-Sicherheitsmanagement, Eine Einführung, <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/142064>, aufgerufen am 31.07.2008
- Drebes, L.; JanRain (2008): Relying Party Stats as of July 1st 2008, <http://janrain.com/blog/2008/07/08/relying-party-stats-as-of-july-1st-2008/>, aufgerufen am 20.07.2008
- EDUCAUSE (2008): eduPerson Object Class, <http://www.educause.edu/eduperson/>, aufgerufen am 23.07.2008
- Fitzpatrick, B. (et al. 2007): OpenID Authentication 2.0, http://openid.net/specs/openid-authentication-2_0.html, aufgerufen am 30.07.2008

- Govoni, R. (2008): OpenID and Rails, Authentication 2.0,
<http://www.devx.com/opensource/Article/37692>, aufgerufen am 15.07.2008
- Hardt, D. (2006): OpenID Signed Assertions 1.0 – Draft 1,
<http://www.mail-archive.com/specs@openid.net/msg00907.html>, aufgerufen am 03.08.2008
- Hardt, D. (et al. 2007): OpenID Attribute Exchange 1.0, http://openid.net/specs/openid-attribute-exchange-1_0.html, aufgerufen am 30.07.2008
- Henstridge, J. (2007): OpenID 2.0, <http://blogs.gnome.org/jamesh/2007/10/23/openid-20/>, aufgerufen am 30.07.2008
- Hoyt, J. (et al. 2006): OpenID Simple Registration Extension 1.0,
http://openid.net/specs/openid-simple-registration-extension-1_0.html, aufgerufen am 30.07.2008
- Internet2 Middleware (2007): eduPerson Object Class Specification, <http://www.nmi-edit.org/eduPerson/internet2-mace-dir-eduperson-200712.html>, aufgerufen am 17.07.2008
- Kveton, S. (2007): OpenID 2.0 and Phishing, <http://kveton.com/blog/2007/01/21/openid-20-and-phishing/>, aufgerufen am 18.07.2008
- Maaß, C. (et al. 2008): Schlagwort OpenID,
http://www.christian-maass.com/wp-content/uploads/2008/05/openid_maas.pdf, aufgerufen am 18.07.2008
- Miller, J. (2006): Yadis Specification Version 1.0,
<http://yadis.org/papers/yadis-v1.0.pdf>, aufgerufen am 19.07.2008
- Moenig, M.; Winklmeier, S. (2007): Single Sign on, Identitätsmanagement,
<http://www.campus-innovation.de/idm>, aufgerufen am 07.07.2008
- OpenID.net (2008): What is OpenID? <http://openid.net/what/>, aufgerufen am 12.07.2008
- OpenID Foundation (2008): <http://openid.net/foundation>, aufgerufen am 12.07.2008
- OpenID Wiki: OpenID Phishing Brainstorm,
http://wiki.openid.net/OpenID_Phishing_Brainstorm, aufgerufen am 23.07.2008
- Recordon, D. (2006): Moving OpenID Forward,
<http://lists.danga.com/pipermail/yadis/2006-June/002631.html>, aufgerufen am 11.07.2008
- RWTH-Aachen (2007): Tivoli Identity Manager,

- <http://www.rz.rwth-aachen.de/ca/c/pys/lang/de/>, aufgerufen am 10.07.2008
- Shibboleth (2008): About Shibboleth, <http://shibboleth.internet2.edu/about.html>,
aufgerufen am 17.07.2008
- Shibboleth (2008): High Level Introduction to Shibboleth,
<http://shibboleth.internet2.edu/HighLevelIntro.html>, aufgerufen am 17.07.2008
- Shirkey, C. (2004): Situated Software,
http://www.shirky.com/writings/situated_software.html, aufgerufen am 12.07.2008
- SWITCH (2007): AAI Introductory Tutorial,
<http://www.switch.ch/proxy/aai/support/presentations/infoday-2007/AAI-ID07-20-Intro.pdf>, aufgerufen am 15.07.2008
- SWITCH (2007): Authentication and Authorization Infrastructure,
http://www.switch.ch/aai/docs/AAI_Attr_Specs.pdf, aufgerufen am 15.07.2008
- SWITCH (2008): Simple demo, <http://switch.ch/aai/demo/2/simple.html>, aufgerufen am
15.07.2008
- Sxip Inc. (2007): Schema for OpenID Attribute Exchange, <http://www.axschema.org/types/>,
aufgerufen am 30.07.2008
- Unabhängiges Landeszentrum für Datenschutz Schleswig-Holstein: Was ist
Identitätsmanagement?,
<https://www.datenschutzzentrum.de/projekte/idmanage/was.htm>, aufgerufen am
06.07.2008
- Universität Augsburg (2007): Identity Management - Analyse und Anforderungen,
http://www.uni-augsburg.de/einrichtungen/its/teilprojekte/im/paket_im1.html, aufgerufen
am 07.07.2008
- Universität Augsburg (2007): Identity Management,
<http://www.uni-augsburg.de/einrichtungen/its/teilprojekte/im/index.html>, aufgerufen am
07.07.2008
- Universität Bielefeld (2006): Zentrale Speicherung von Nutzerdaten,
http://www.uni-bielefeld.de/hrz/projekte/zntr_speicherung.html, aufgerufen am
06.07.2008
- Willison, S. (2008): The point of "Open" in OpenID,
<http://simonwillison.net/2008/Jun/24/openid/>, aufgerufen am 03.08.2008
- Windley, P. (2005): Digital Identity, 1. Auflage, O'Reilly

Zeilenga, K. (2006): RFC 4512 - Lightweight Directory Access Protocol (LDAP), Directory Information Models, <http://tools.ietf.org/html/rfc4512>, aufgerufen am 08.08.2008

6 Abbildungsverzeichnis

- Abbildung 1: Ablauf eines Authentifizierungsvorgangs in Shibboleth..... 11
- Abbildung 2: Von MyOpenID (Identity Provider) eindeutig gezählte Relying Parties..... 15
- Abbildung 3: OpenID Login gegenüber der Angabe von Benutzername und Passwort.. 19
- Abbildung 4: Sequenzdiagramm des Authentifizierungsvorgangs mit OpenID..... 20
- Abbildung 5: Auswahlmaske für die Freigabe von Attributen..... 21
- Abbildung 6: Verteilung der Daten im geplanten Ident-Management-System des ZfN... 29
- Abbildung 7: Software-Schichten des Identity-Providers..... 33
- Abbildung 8: Aufbau des Identitätsmanagementsystems..... 35
- Abbildung 9: Formular mit OpenID-Button und herkömmlichem Login..... 36
- Abbildung 10: SeatBelt: Hinzufügen eines Identity Providers..... 42

7 Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Bremen, den 10. August

Dennis Blöte