

HU-IAM – Identitätsmanagementsystem der Humboldt-Universität zu Berlin

Michail Bachmann | michail.bachmann@cms.hu-berlin.de

Das Feststellen der Identität einer Person und die darauf basierende, für die Nutzung von IT-Diensten erforderliche Verwaltung und Umsetzung von Rollen sind wichtige Voraussetzungen für den Betrieb einer modernen IT-Infrastruktur. Ob Personal-, Studierenden- und Prüfungsverwaltung, E-Mail-Systeme oder Datenbanken – alle diese Systeme sind darauf angewiesen, dass personenidentifizierende Daten korrekt, konsistent und ständig verfügbar sind, denn nur so ist eine individualisierte Nutzung der Anwendungen möglich. Den Teil der Infrastruktur, der diese Daten zur Verfügung stellt und der sich um die Verwaltung und die Anbindung an andere Systeme kümmert, nennt man Identitätsmanagement (IdM). Der folgende Artikel stellt kurz das Design des IdM-Systems der Humboldt-Universität vor, sowohl aus Sicht des IdM-Modells als auch aus Systemsicht.

Eine Grundvoraussetzung für den Betrieb einer individualisierten IT-Anwendung ist die Möglichkeit, den Zugriff zu kontrollieren und darauf basierend den Nutzern bestimmte Rechte innerhalb der Anwendung zuzuweisen. Dabei kann die Granularität der zugewiesenen Rechte stark schwanken: von streng individuellen Rechten, wie z. B. dem Zugriff auf den persönlichen E-Mail-Account, bis zu sehr generischen Rechten, wie z. B. dem Recht, auf eine organisationsinterne Intranetseite zugreifen zu können. Zuweisung und Entzug von Zugriffsrechten und das Zusammenfassen von einzelnen Rechten zu sogenannten Rollen müssen durch ein IdM-System unterstützt werden. Eine große Hochschule stellt besondere Herausforderungen an ein Identitätsmanagement. Diese reichen von technischen Aspekten, wie der Kopplung des IdM-Systems an eine Vielzahl von bereits vorhandenen – teilweise nicht mehr vom Hersteller unterstützten – Systemen, über hohe Anforderungen an die Skalierbarkeit (man denke z. B. an die dynamischen und stark fluktuierenden Rollenzuordnungen bei den Studierenden), bis hin zur organisatorischen Herausforderung der Umsetzung des Konzeptes in einem sowohl technisch als auch administrativ stark dezentralisierten Umfeld.

IdM-Modell

Im Folgenden sollen einige der Anforderungen vorgestellt und näher erläutert werden, die das IdM-Modell der HU auszeichnen.

Personenzentrierung bei der Identifizierung

Wenn von Identifizierungsdaten in Bezug auf IT-Systeme gesprochen wird, ist in den meisten Fällen ein Benutzerkonto (Account), bestehend aus einem Benutzernamen und einem Nutzerkennwort, gemeint. Daher ist es leider wenig verwunderlich, dass die meisten IdM-Systeme alle auf die Verwaltung von Accounts ausgerichtet sind, sollen doch die Administratoren angesprochen und überzeugt werden. Da in der Praxis eine Zuordnung des Accounts zu einer realen Person vorhanden sein muss, ist natürlich immer die Möglichkeit vorgesehen, zusätzliche Informationen abzulegen, sei es der Name oder die Telefonnummer der Person, die den Account nutzt. Es wird viel Aufwand getrieben, damit jeder (potentielle) Nutzer durch genau einen Account repräsentiert wird, der dann später in den IT-Systemen verwendet werden kann. Dieses idealisierte Modell kann jedoch erfahrungsgemäß in der Praxis nicht lange aufrechterhalten werden, so dass es gute Gründe gibt, die 1:1-Zuordnung zwischen Account und Person aufzuheben. So sollte beispielsweise unter dem Aspekt der Sicherheit für den Zugriff auf den Webmailer von unterwegs nicht der gleiche Account genutzt werden wie für den Zugriff auf die Personaldatenbank. Umgekehrt sollte auch die Nutzung eines Accounts durch mehr als eine Person möglich sein, notwendig z. B. bei der Verwendung eines gemeinsam nutzbaren Funktionsaccounts für die Pflege von Webseiten. Außerdem ist es manchmal erforderlich, bestimmte Accounts einer anderen Person temporär

zur Nutzung zu überlassen, ohne die Verantwortung für diesen Account abzugeben. Das ist z. B. bei Praktikantenaccounts der Fall, die zwar jeweils wechselnden Praktikanten zugewiesen werden, jedoch immer in der Verantwortung des Praktikumsbetreuers bleiben. Abweichungen von dem „Ein Nutzer – ein Account“-Schema werden in der Praxis bisher meist dadurch umgesetzt, dass parallel mehrere Accounts mit gleichen Personeninformationen angelegt werden. Die Mehrfachhaltung dieser Informationen erfordert jedoch einen hohen Pflegeaufwand, da z. B. Namensänderungen aufwendig in allen zu der Person gehörenden Accounts parallel aktualisiert werden müssen. Wird dieser Aufwand nicht betrieben, verschlechtert sich die Qualität des Accountdatenbestandes im Laufe der Zeit rapide. Auch die (IT-)Sicherheit der Organisation als Ganzes sinkt, da durch die fehlende Zuordnung eines Accounts zu einer Person deren Accounts auch nach dem Ausscheiden aus der Organisation als „Leichen“ in der Datenbank verbleiben und so für Angriffe auf die IT-Systeme missbraucht werden können. Ein modernes IdM-System muss sich also vom üblichen Modell, den Account als Repräsentanten und Ersatz der eigentlichen Person zu nutzen, entfernen und stattdessen die Person direkt als zentrales identifizierendes Element verwenden. Der Account als sekundäres Identifizierungsdatum wird der Person dann lediglich zugeordnet.

Flexible Authentifizierung

Authentifizierung ist der Beweis des Vorliegens einer bestimmten Identität. So besteht der „normale“ Login-Vorgang aus der Behauptung, ein berechtigter Nutzer des Accounts „xyz“ zu sein, und des Beweises dieser Behauptung, indem das zu dem Account gehörige Passwort eingegeben wird, welches nur dem berechtigten Nutzer bekannt ist.

Wie schon aus dem Beispiel erkennbar, ist die bei IT-Systemen mit Abstand verbreitetste Authentifizierung mithilfe eines Passworts. Ein modernes IdM-System muss jedoch verschiedene Möglichkeiten der Authentifizierung unterstützen, d. h.

neben dem Passwort auch Zertifikate (sowohl SmartCard, als auch Soft-Zertifikat), TAN-Listen, Passwortgeneratoren (z. B. RSA SecurID) usw. Idealerweise sollte ein IdM-System im Modell keine Einschränkungen bezüglich der verwendeten Authentifizierung haben, so dass auch derzeit unbekannt oder einfach nur zum Entstehungszeitpunkt nicht vorgesehene Authentifizierungsmöglichkeiten in das System eingebunden werden können. Zur flexiblen Authentifizierung gehört auch eine möglichst anpassungsfähige Verknüpfung von Identifizierungs- und Authentifizierungsinformationen. So sollte es möglich sein, sowohl ein bestimmtes Passwort mit mehreren Accounts zu verwenden als auch umgekehrt einen Account durch mehrere Passwörter zu authentifizieren (vgl. Abb. 1).

menarbeit zwischen den Kooperationspartnern aus IT-Sicht nur unter erschwerten Bedingungen möglich und führt zu ineffizienten Ad-hoc-Lösungen, die zu einem späteren Zeitpunkt Probleme bereiten können. Ein möglicher Ausweg kann sein, sich auf ein dezentrales (z. B. OpenID [1]) oder föderiertes (z. B. Shibboleth [2], vgl. auch den Artikel in dieser Ausgabe) Authentifizierungssystem zu einigen.

Verschiedene Identitätsquellen und Identitätskonsumenten

Eine Identitätsquelle ist ein Lieferant für Identitätsdaten. An der HU sind unterschiedliche Identitätsquellen u. a. für Studierende, wissenschaftliche und nicht-wissenschaftliche Angestellte, Lehrbeauf-

Person

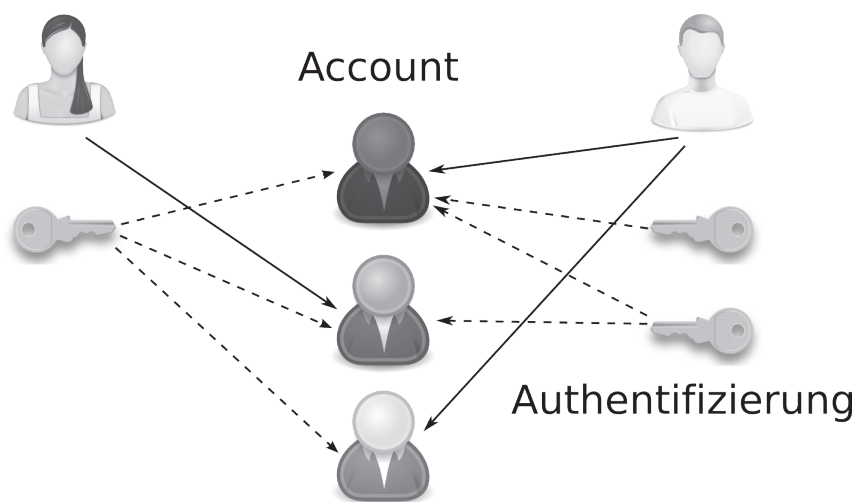


Abb. 1: Flexible Authentifizierung

Eine Besonderheit von Hochschulen im Vergleich zu anderen Organisationen ist die oft geforderte Möglichkeit zur organisationsübergreifenden Authentifizierung. Diese Forderung kann unterschiedlichste Gründe haben, sei es, weil Studierende anderer Hochschulen über das Internet Zugriff auf bestimmte Kurse oder Kursmaterialien erhalten sollen oder weil eine Arbeitsgruppe unter Beteiligung mehrerer Hochschulen gegründet wurde und die Modalitäten des Zugriffs auf die verschiedenen IT-Ressourcen der Hochschulen geregelt werden müssen. Fehlt diese Möglichkeit, ist eine Zusam-

tragte, Kooperationspartner usw. vorhanden. Das IdM-System muss daher in der Lage sein, diese und weitere (sowohl interne als auch externe) Datenquellen einzubinden. Verursacht durch die verschiedenen Datenquellen kann (und wird) häufig der Fall der Personenidentität zwischen verschiedenen Quellen vorliegen, d. h. das IdM-System muss auch mit gleichen Personen aus verschiedenen Datenquellen (Duplikaten) umgehen können. Erschwerend kommt hinzu, dass das Problem nicht so einfach lösbar ist. Ungenügende oder schlicht falsch erfasste Daten verhindern in den einzelnen

Quellen eine sichere Duplikatserkennung. Eine besondere Schwierigkeit dabei ist, dass derselbe Name einer Person in verschiedenen Quellen unterschiedlich bzw. sogar falsch geschrieben sein kann oder eine Person im Laufe der Zeit ihren Namen z. B. durch Heirat geändert hat. Ein modernes IdM-System muss die Daten aus den verschiedenen Identitätsquellen möglichst automatisiert konsolidieren können.

Die Erfassung der Identitätsinformationen dient dem Zweck, diese Daten später zur Identifizierung und Authentifizierung in anderen IT-Systemen, sogenannten Identitätskonsumenten, zu verwenden. Dabei werden die vom IdM-System gelieferten Informationen meist über ein Vermittlersystem den Identitätskonsumenten zur Verfügung gestellt. Ein solches System wird als Identitätsprovider bezeichnet. Beispiele für Identitätsprovider sind ein LDAP-Verzeichnis oder eine Windows-Domäne, die hauptsächlich von anderen Systemen als Adressbuch oder zur Anmeldung genutzt werden. Wie in jeder größeren Organisation gibt es meist schon mehrere, historisch gewachsene, Identitätsprovider. Diese können und sollen meist nicht abgelöst werden, so dass es die Aufgabe des IdM-Systems ist, sich grundsätzlich in die vorhandene Infrastruktur zu integrieren. Dabei können u. a. Inhalt der Daten sowie deren Format oder deren Übertragungsweg von einem zum anderen Identitätsprovider stark variieren. Ein IdM-System muss in der Lage sein, Änderungen auf die benötigte Art und Weise an den oder die Identitätsprovider zu übermitteln.

Provisionierung und Autorisierung

Als Provisionierung wird allgemein der Vorgang der Bereitstellung von Systemressourcen bezeichnet, der Spezialfall der Einräumung von Zugriffsrechten in Systemen wird Autorisierung genannt. Neben den bereits erwähnten Identifizierungs- und Authentifizierungsdaten kann ein IdM-System auch Daten zur Provisionierung von bestimmten IT-Systemen speichern. Diese Provisionierungsdaten gestatten es einem Dienst, eine identitätsbezogene Konfiguration des Dienstes für den konkreten Einzelfall vorzunehmen.

So könnten beispielsweise die Provisionierungsdaten für die Erstellung eines Accounts in einer Windows-Domäne die Größe und Position des bereitzustellenden Speicherplatzes, eine Auflistung zu gewährender Systemrechte usw. umfassen. Auch die Information, dass ein bestimmter Dienst nur von einer bestimmten Person verwendet werden darf, ist eine Provisionierungsinformation, da so ein minimales Zugriffsrecht ("Darf nutzen" oder "Darf nicht nutzen") provisioniert wird.

Es gibt Systeme, bei denen diese grobe Einteilung der Zugriffsrechte ausreichend ist, z. B. Informationsdienste, die nur organisationsintern, nicht jedoch öffentlich verfügbar sein sollen. Im Normalfall werden jedoch feingranularere Rechte benötigt. Das IdM-System enthält jedoch keine Mechanismen, um diese konkreten Systemrechte zu speichern oder gar durchzusetzen. Diese fehlende Funktionalität mutet kontraintuitiv an, scheint es doch zunächst eine genuine Aufgabe eines IdM-Systems zu sein, dieses sicherzustellen. Die internen Rechte eines bestimmten IT-Systems sind jedoch im Allgemeinen zu systemspezifisch und zu vielfältig, um zentral vorgehalten werden zu können. Eine Verwaltung und Durchsetzung detaillierter Rechte kann daher realistischerweise nur in den einzelnen IT-Systemen selbst geschehen.

In den meisten Fällen bekommt jedoch nicht jeder einzelne Nutzer eine individuelle Rechtezusammenstellung, vielmehr werden die Rechte zu Rollen oder Gruppen zusammengefasst und vergeben. Im Bereich eines Prüfungsanmeldesystems kommen z. B. die Rollen "Administratoren", "Prüfende" und "Studierende" mit jeweils unterschiedlichen Rechten zum Einsatz. In einem solchen Fall wird eine entsprechende Rechteabstraktion für einen Dienst im IdM-System gespeichert und diesem Dienst zur Verfügung gestellt; die Umsetzung in konkrete Rechte und deren Durchsetzung bleibt aber den einzelnen Diensten überlassen.

Die Verwaltung von Einzelzuordnungen zwischen Nutzern und Diensten ist bei einer größeren Anzahl sehr aufwendig. Der Normalfall sind jedoch IT-Systeme, die von einer bestimmten Gruppe von Personen genutzt werden sollen. Die Gründe dafür sind mannigfaltig: So könnte z. B. ein bestimmter Dienst nur Studierenden zur Verfügung stehen, nur Mitarbeitern einer bestimmten Abteilung oder nur Mitgliedern einer bestimmten organisationsweiten Arbeitsgruppe.

Auch die IT-Dienste selbst werden oft zu Gruppen zusammengefasst, z. B. die Dienste "Empfangen von E-Mails" und "Versand von E-Mails" zur Dienstegruppe "Mail". Ein modernes IdM-System

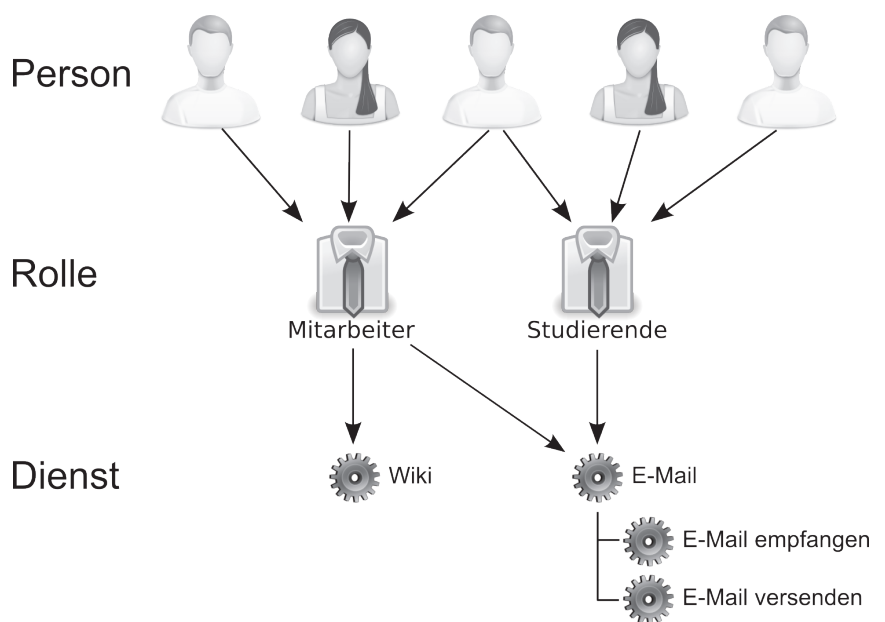


Abb. 2: Provisionierung

sollte es daher ermöglichen, beliebige Dienste und Dienstgruppen einer beliebigen Personengruppe zuzuordnen, so dass beispielsweise Regelungen der Art "Alle Studierenden dürfen die Mail-Systeme nutzen" umsetzbar werden. Die Zuordnungen zwischen Dienstgruppen und Personengruppen müssen dabei flexibel sein, so dass sowohl einer Dienstgruppe als auch einer Personengruppe neue Mitglieder hinzugefügt oder aus ihr entfernt werden können. Ein Beispiel findet sich in Abbildung 2.

Systemdesign

Ging es bisher um die Anforderungen, die an ein modernes Identitätsmanagement gestellt werden, so soll im Folgenden ein kurzer Überblick über die technische Funktionsweise des IdM-Systems der HU (HU-IAM) gegeben werden.

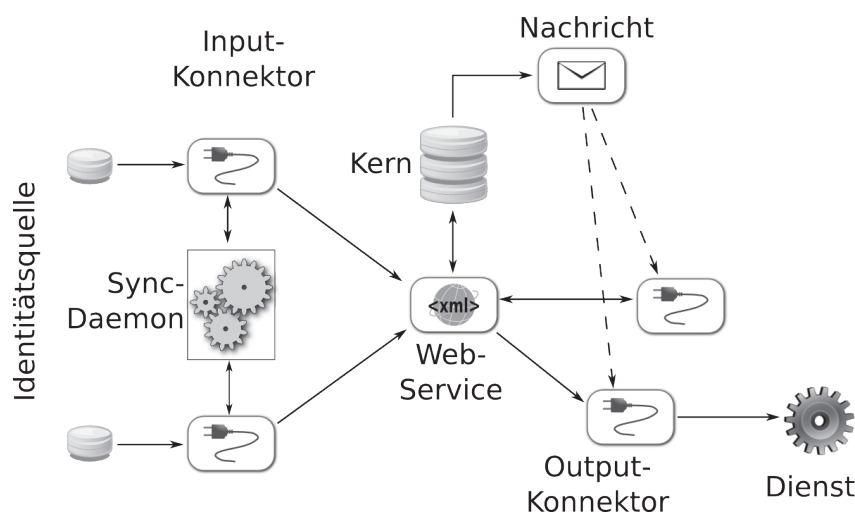


Abb. 3: Schema Systemdesign

Die Abbildung 3 stellt einen groben Überblick über die beteiligten technischen Komponenten dar und soll als Übersichtskarte für eine kleine Tour durch HU-IAM dienen, die am Beispiel der Immatrikulation eines neuen Studenten erfolgt.

Herbert Ungeheuer möchte an der Humboldt-Universität studieren. Er bewirbt sich, wird immatrikuliert und bekommt mit der Immatrikulation in der Studierendendatenbank SOSPOS die Matrikelnummer 666666 zugeteilt. Das IdM-System der HU führt die SOSPOS-

Datenbank als Identitätsquelle für Studierendendaten, d. h. es existiert eine Software, genannt Input-Konnektor, die das IdM-System mit Hilfe des sogenannten Sync-Daemons mit der Studierendendatenbank abgleicht. Dabei wird durch einen Vergleich mit den bereits im HU-IAM-Kern enthaltenen Daten erkannt, ob neue Studierende hinzugekommen sind, bereits Bekannte die Universität verlassen haben oder Änderungen an den Daten (z. B. Namensänderungen) erfolgt sind. Alle erkannten Änderungen werden dann an den HU-IAM-Kern übertragen und dort in einer Datenbank gespeichert. Der Zugriff auf die gespeicherten Daten erfolgt ausschließlich über Webservices, d. h. spezielle Schnittstellen, die dem automatisierten Datenaustausch zwischen verschiedenen IT-Systemen dienen. Für alle Operationen auf dem Datenbestand werden geeignete Methoden zur Verfügung gestellt.

In Herberts Fall hat nun der SOSPOS-Input-Konnektor festgestellt, dass dem Eintrag in der SOSPOS-Datenbank kein Eintrag im HU-IAM-Kern entspricht. Also erteilt der Konnektor dem HU-IAM-Kern den Auftrag, eine neue Person mit dem Namen „Herbert Ungeheuer“ und der Identitätsquelle „SOSPOS“ anzulegen und dieser Person auch gleich die Rolle „Studierende“ zuzuweisen. Der HU-IAM-Kern führt den Auftrag aus und erzeugt aus den Daten einen neuen Personeneintrag mit der ID XYZ.

Nun sind die Daten von Herbert im HU-IAM-Kern angekommen. Da die Daten dort jedoch nicht zum Selbstzweck gespeichert werden, müssen sie anderen Systemen zur Verfügung gestellt werden. Zu diesem Zweck führt jede Änderung am Kerndatenbestand zu mindestens einer Nachricht, die dann von den sogenannten Output-Konnektoren empfangen und weiterverarbeitet wird. Die Nachrichten werden mit Hilfe des sogenannten Publish-Subscribe-Verfahrens verteilt. Dabei besitzt der Sender einer Nachricht kein Wissen darüber, wer bzw. überhaupt jemand die versendete Nachricht empfängt („publish“). Die potentiellen Empfänger bekunden im Gegenzug Interesse am Empfang bestimmter Nachrichten („subscribe“), ohne jedoch zu wissen, ob es überhaupt Sender für Nachrichten dieses Typs gibt. Der Sender übermittelt Nachrichten damit nicht direkt an die Empfänger, sondern an eine zwischengeschaltete Instanz („broker“), die sich um die ordnungsgemäße Verteilung der Nachrichten kümmert.

Der Eintrag von Herbert Ungeheuer wird u. a. in folgende Nachrichten umgewandelt: „Es gibt eine neue Person mit der ID XYZ“ und „Die Gruppe 'Studierende' hat als neues Mitglied die Person mit der ID XYZ“. Auf die erste Nachricht hat der Output-Konnektor „Accountgenerierung“ gewartet. Über die Webservices des HU-IAM-Kerns ruft er die Daten von Herberts Eintrag ab, erzeugt daraus den Accountnamen „ungeheuer“ und schickt dann an den HU-IAM-Kern den Auftrag, diesen Account zu erstellen und ihn Herbert zuzuweisen. Die aus diesen Änderungen resultierende Nachricht ist u. a. „Es gibt den neuen Account 'ungeheuer'“, die vom Output-Konnektor „Neuer Account“ empfangen wird und diesem Account die aus Herberts Rolle „Studierende“ resultierenden Dienste zur Nutzung zuweist, woraufhin der HU-IAM-Kern die Nachrichten „Der Account 'ungeheuer' darf die Dienste 'E-Mail', 'E-Mail empfangen' und 'E-Mail versenden' nutzen“ verschickt. Eine solche Nachricht empfängt der Output-Konnektor für den Dienst „E-Mail empfangen“, ruft daraufhin Daten aus dem HU-IAM-Kern ab und teilt dem Mailsystem mit, dass

ein neues E-Mail-Konto für den Account „ungeheuer“ generiert werden soll und legt auch einen E-Mail-Alias „herbert.ungeheuer@student.hu-berlin.de“ für dieses Konto an. Von nun an kann Herbert E-Mails empfangen und – sobald er ein Passwort für den Account festgelegt hat – diese auch lesen.

Auch wenn das System der Output-Konnektoren auf den ersten Blick kompliziert aussieht, hat ein solches nachrichtenorientiertes asynchrones Systemdesign einige Vorteile. So müssen z. B. die Output-Konnektoren nicht immer sofort verfügbar sein, da die Nachrichten zwischengespeichert werden und bei Bedarf abgerufen werden können, wodurch auch die Fehlertoleranz des Systems erhöht wird. Zwischen den Konnektoren und dem Kern besteht nur eine lose Kopplung, so dass das Gesamtsystem besser skaliert, z. B. durch parallele Verarbeitung der Nachrichten durch mehrere Instanzen eines Konnektors. Da eine Nachricht auch an mehrere unterschiedliche Konnektoren ausgeliefert werden kann, sind Erweiterungen des IdM-Systems sehr einfach zu bewerkstelligen, ohne in den Kern des Systems eingreifen zu müssen.

Fazit

Der vorliegende Artikel versucht in aller Kürze sowohl das Modell als auch das technische Design des IdM-Systems der Humboldt-Universität etwas näher zu beleuchten. Nach der vollständigen Integration der Datenbestände soll demnächst die vorhandene Accountdatenbank durch das IdM-System abgelöst werden. Im weiteren Verlauf können dann neue und bestehende Dienste mit aktuellen Identitätsinformationen versorgt werden.

Literatur

- [1] *OpenID Foundation website.*
<http://openid.net/>
- [2] *shibboleth.*
<http://shibboleth.internet2.edu/>