Integrierter Shibboleth Identity Provider auf Basis verteilter Identitätsdaten

Sebastian Labitzke, Michael Simon, Jochen Dinger

{sebastian.labitzke, michael.simon, jochen.dinger}@kit.edu

Abstract: Typischerweise erlauben Shibboleth-basierte Authentifikations- und Autorisationsinfrastrukturen (AAI), wie die DFN-AAI, nur *einen* Identity Provider (IdP) für *eine* teilnehmende Organisation, um den Wartungsaufwand seitens des AAI-Betreibers möglichst gering zu halten. Ferner wahrt dies die Benutzerfreundlichkeit, da so das IdP-Verzeichnis minimal ausfällt, aus dem Nutzer vor der Authentifikation die passende Organisation auswählen müssen. Allerdings liegen in großen Einrichtungen Identitätsdaten häufig in verteilten Datenquellen vor und sind nicht über eine zentralisierte Schnittstelle verfügbar. Die Shibboleth IdP-Implementierung ist jedoch in der Anbindung verteilter Datenquellen limitiert. In dieser Arbeit werden mögliche Konzepte zur Integration eines IdP in eine Organisation mit verteilten Identitätsdaten vorgestellt und bewertet. Dabei werden für die Authentifikation bestehende Ansätze untersucht. Bislang nicht erfüllten Anforderungen konnte durch zwei Entwicklungen, einem *Shibboleth Login Handler* und einem *JaaS Dispatcher Module*, nachgekommen werden. Darüber hinaus wird gezeigt, wie sich die Shibboleth-Attributlieferung in ein bestehendes Identitätsmanagementsystem integrieren lässt. Die Umsetzbarkeit der vorgestellten Integrationslösungen wird abschließend am Beispiel des Karlsruher Instituts für Technologie verdeutlicht.

1 Einleitung

Als dezentrales Authentifikations- und Autorisationssystem für Browser-basierte Web-Dienste hat sich im Bereich der Forschung und Lehre die Spezifikation Shibboleth und das zugehörige Softwareprodukt der „Internet2 Middleware Initiative“[1] etabliert. Nutzer werden dabei durch Shibboleth Identity Provider (IdP) authentifiziert. Ferner liefern IdPs Attribute als Basis für eine Autorisationsentscheidung an Web-Dienste. Somit kann die Implementierung des Web-Dienstes auf dessen Kernfunktionalitäten beschränkt bleiben und ein dienstübergreifendes Single-Sign On wird ermöglicht. Weiterführende Informationen zum Ablauf von Shibboleth-Authentifikationen und -Autorisationen finden sich unter anderem auf den Web-Seiten der Switch AAI[2].

Zur Kollaboration über Organisationsgrenzen hinweg werden zudem Authentifikations- und Autorisationsinfrastrukturen (AAIs) aufgebaut. Diese übernehmen die Verwaltung der Shibboleth Meta-Daten, die für den Einsatz von Shibboleth notwendig sind und ansonsten bidirektional zwischen Betreibern von Web-Diensten und IdP-Betreibern ausgetauscht

[1] http://shibboleth.internet2.edu/
[2] http://switch.ch/aai/demo/

werden müssten. Die Betreiber solcher AAIs, wie das Deutsche Forschungsnetz (DFN), stellen einen sogenannten Discovery Service zur Verfügung, über den einem Nutzer eine Auswahl an teilnehmenden Organisationen und damit deren IdPs zur Verfügung gestellt wird. So kann ein Nutzer die Einrichtung wählen, bei der für ihn ein entsprechendes Nutzerkonto eingerichtet wurde, und wird zur Authentifikation zu dessen IdP weitergeleitet.

Da die Auswahl des IdPs möglichst übersichtlich gehalten werden soll, streben AAIs an, nur jeweils einen IdP als Authentifikationsdienst einer Organisation zu verzeichnen. Diese Restriktion kann Organisationen, die an der AAI teilnehmen wollen, vor Herausforderungen stellen. Shibboleth IdPs sind in den Möglichkeiten zur Anbindung verteilter Identitätsdaten eingeschränkt und die Forderung nach einem IdP, der alle relevanten Nutzergruppen einer Organisation authentifizieren und Attribute für diese zur Verfügung stellen kann, ist insbesondere dann problematisch, wenn keine organisationsweite Nutzerkontenverwaltung existiert oder ein Identitätsmanagement (IdM) einen einheitlichen Zugriff gewährleistet. Zur Authentifikation von Nutzern aus verschiedenen Nutzerverwaltungen sieht Shibboleth das AAI-Konzept vor. Würde demnach organisationsintern auf mehreren Nutzerverwaltungen eine AAI aufgebaut, müssten alle darin etablierten IdPs im Discovery Service einer übergeordneten AAI verzeichnet werden, um auch an dieser teilzunehmen. Dies würde jedoch der Forderung nach einem IdP pro teilnehmender Organisation widersprechen. Abbildung 1 visualisiert das dargestellte Problem noch einmal.

Wie in Abschnitt 2 näher erläutert wird, gibt es bereits Möglichkeiten, einen IdP mit verteilten Datenquellen zu konfigurieren. Diese Möglichkeiten sind jedoch entweder in ihrer Flexibilität eingeschränkt, gehen mit Einbußen bezüglich der Leistung einher oder bedingen zusätzliche Nutzerinteraktionen bei der Authentifikation. Die Entwicklung individueller Module und die IdP-Integration nach den in diesem Papier vorgestellten Konzepten erfüllen entsprechende Anforderungen und bedeuten eine Reduktion des betrieblichen Aufwands. Es werden zwei flexibel einsetzbare Module zur Realisierung der Authentifikation sowie ein Konzept zur Integration eines IdP in bestehende IdM-Systeme vorgestellt. Dabei wird diskutiert unter welchen Voraussetzungen und an welcher Stelle individuelle Module zum Einsatz kommen können und entsprechende Umsetzungen präsentiert.

Der folgende Abschnitt 2 untersucht und bewertet ausgewählte bestehende Ansätze. Das erarbeitete Lösungskonzept und die implementierten Authentifikationsmodule werden im Abschnitt 3 vorgestellt. Bevor in Abschnitt 5 die Ergebnisse zusammengefasst werden und auf die zukünftigen Shibboleth-Vorhaben am Karlsruher Institut für Technologie (KIT)

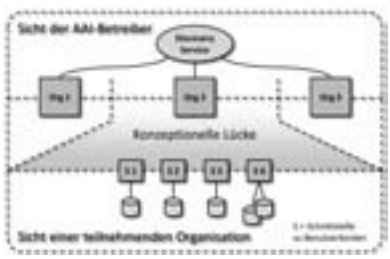

Abbildung 1: Konzeptionelle Lücke zwischen Sicht der AAI-Betreiber und Organisationen

eingegangen wird, wird in Abschnitt 4 ein Blick auf die Shibboleth-Infrastruktur am KIT geworfen und Aspekte wie das Clustering skizziert.

2 Bestehende Lösungsansätze

Die Authentifikation und Attributlieferung durch einen einzelnen Shibboleth IdP ist genau dann problematisch, wenn eine Einrichtung aus Organisationseinheiten mit eigenen Nutzerverwaltungen und damit verteilt vorliegenden Identitätsdaten besteht. Im Folgenden werden existierende Möglichkeiten vorgestellt und bewertet, durch die der Zugriff auf verteilte Identitätsdaten realisiert werden kann.

Eine Möglichkeit bietet der bei Shibboleth zum Einsatz kommende *JAVA Authentication and Authorization Service* (JaaS) [Mah03]. Schnittstellen zu multiplen Nutzerverwaltungen können damit sukzessiv durchlaufen werden, um die passende Quelle für eine Authentifikation zu finden. Hierfür wird eine Liste verschiedener Authentifikationsschnittstellen aufgebaut. Es existieren bereits JaaS-Module, um zum Beispiel gegen LDAP-Schnittstellen zu authentifizieren oder Kerberos Authentifikationen zu integrieren. Den konfigurierten Schnittstellen werden zusätzlich Attribute hinzugefügt, die beispielsweise anzeigen, ob eine erfolgreiche Authentifikation gegen eine Schnittstelle hinreichend oder notwendig ist.

Die Anbindung von Authentifikationsschnittstellen mit JaaS funktioniert jedoch nur dann ohne individuelle Implementierung, wenn bereits JaaS-Module für die zu konfigurierenden Schnittstellen verfügbar sind und keine komplexere Logik als das sequentielle Durchlaufen dieser notwendig ist. Soll mittels proprietärer Schnittstellen wie Web Services, CGIs, Skripten etc. auf eine Quelle zugegriffen oder zusätzliche Restriktoren überprüft werden, können individuelle JaaS-Module implementiert werden. Das JaaS-Modul wird analog den existierenden Modulen in der Datei *login.config* konfiguriert und das als *.jar* kompilierte Modul in das *war*-File des IdP in den Ordner *WEB-INF\lib* abgelegt.

Mit dem Einsatz mehrerer JaaS-Module, wächst jedoch die Latenz einer Authentifikation um die Länge der Antwortzeiten von Datenquellen, gegen deren Schnittstellen eine Authentifikation nicht erfolgreich verlief, bevor die Authentifikationsanfrage an die passende Quelle gereicht wurde. Eine Vorauswahl der Authentifikationsquelle, zum Beispiel anhand eines Schemas für Benutzernamen, ist nicht vorgesehen.

Alternativ wurden bereits verschiedene proprietäre Module, vor allem Servlet-Filter und Shibboleth Login Handler, entwickelt. Die Universitätsbibliothek Freiburg setzt das Modul *myLogin*[3] als Erweiterung zum Shibboleth IdP ein. *myLogin* ermöglicht die Anbindung mehrerer Nutzerkontenverwaltungen und bietet dem Nutzer im Anschluss an die Auswahl der Heimateinrichtung beim Discovery Service der AAI eine Auswahl der zur Verfügung stehenden organisationsinternen Quellen an. So kann ein Nutzer stets den Nutzerkontenpool auswählen, gegen den er sich authentifizieren möchte, respektive für den er einen Login besitzt. Dieses Modul ist von besonderem Vorteil, wenn ein Nutzer verschiedene

[3] https://mylogin.uni-freiburg.de

Nutzerkonten besitzt, mit denen jeweils unterschiedliche Attribute und Rechte verknüpft sind. Nach erfolgreichem Login werden so je nach zuvor getätigter Auswahl die zum Nutzerkonto gehörigen Attribute an die Service Provider geliefert.

Mit *myLogin* wurde eine Hierarchisierungsstufe bei der Auswahl der Authentifikationsquelle eingeführt. Nachteilig ist jedoch, dass unter Einsatz dieses Moduls den Nutzern nach dem Discovery Service eine weitere System-Interaktion aufgebürdet wird. Die Wiederverwendbarkeit von *myLogin* ist zudem dann eingeschränkt, wenn interne Organisationsstrukturen vorhanden sind, bei denen die Nutzer ihr Nutzerkonto nicht eindeutig einem angegebenen Nutzerkontenpool (z.B. Rechenzentrum, Bibliothek, Klinikum...) zuordnen können, so dass ein Ausprobieren der Quellen notwendig wäre. Insbesondere ist dies der Fall, wenn mit einem Konto zentrale Dienste genutzt werden und der zugehörige Nutzername keinen Aufschluss über die das Nutzerkonto verwaltende Organisationseinheit gibt.

Um *eine* Schnittstelle zum Zugriff auf verteilte Daten zu etablieren, kann alternativ ein virtuelles Verzeichnis oder ein Meta-Directory aufgebaut werden, in das alle notwendigen Attribute sowie Informationen zu Nutzerkonten und Passwörter repliziert werden. Ein derartiges Verzeichnis würde als einzige Quelle im IdP konfiguriert werden. Für Einrichtungen, die bereits eines dieser beiden Verzeichnisarten betreiben, kann die Anbindung dieses eine Alternative zu oben genannten Lösungen sein. Gegen die Etablierung eines solchen Verzeichnisses speziell für die Shibboleth-Infrastruktur spricht jedoch der zu erwartende erhebliche Aufwand für Betrieb und Wartung. Insbesondere ohne ein umfassendes IdM-System kann der Datenbestand eines Meta-Directory nur mit zusätzlichem Aufwand auf einem aktuellen Stand gehalten werden.

3 Authentifikationsmodule und Attributprovisionierung

Um den Aspekten Flexibilität, Geschwindigkeit und Nutzerkomfort bei der Anbindung verteilter Identitätsdaten gerecht zu werden, kann die Notwendigkeit zu einer individuellen Lösung bestehen. Die erarbeiteten Lösungsvorschläge für die Authentifikation sowie die integrative Bereitstellung von Attributen für den Shibboleth IdP werden im Folgenden getrennt voneinander vorgestellt.

3.1 Authentifikationsmodule

Für die konzipierten Authentifikationsmodule werden folgende Annahmen bzw. Anforderungen zu Grunde gelegt. Organisationseinheiten, die ihre Accounts eigenständig verwalten, sollen dazu auch nach der Etablierung eines IdP in der Lage sein, ohne zusätzliche Prozesse etablieren zu müssen, um die Shibboleth-Infrastruktur mit aktuellen Daten zu versorgen. Ferner ist die Replikation eines Passworts in eine zweite Datenquelle, neben dem Sicherheitsfaktor und dem erhöhten Aufwand für Passwortänderungsprozesse, technisch oft nicht möglich. Außerdem ist zu beachten, dass im Bibliotheksbereich häufig mit einer IP-Überprüfung als hinreichende Authentifikation der Nutzer gearbeitet wird

[ORBL09]. Mit dem Einsatz von Shibboleth kann die IP-basierte gegen eine personalisierte Authentifikation ersetzt werden, jedoch dürfen z.B. sogenannte Library-Walk-In-Nutzer nur von Rechnern der Bibliothek auf lizensierte Inhalte zugreifen, so dass eine aus beiden Verfahren kombinierte Authentifikation ermöglicht werden sollte.

Ferner war das Ziel der Entwicklung von Authentifikationsmodulen die passende Nutzerverwaltung zu identifizieren, bevor die Authentifikationsanfrage an eine der konfigurierten Schnittstellen gerichtet wird. Da die Datenquellen so nicht sequentiell abgearbeitet werden müssen, vermindert dies die Verzögerungszeit der Authentifikation. Voraussetzung für die automatisierte Auswahl der Datenquelle ist, dass die unterschiedlichen Identifikatoren wie E-Mail-Adressen und Nutzerkennungen eindeutig den Datenquellen zuordenbar sind. Für die Entwicklung der im folgenden vorgestellten Module sollte die Identitfikation über reguläre Ausdrücke in einer Konfigurationsdatei eingestellt werden können.

Insofern ergeben sich folgende spezifischen Anforderungen:

- Wahrung der Autonomie von Organisationseinheiten bezüglich ihrer Nutzerkonten
- Zugriff auf verteilte Identitätsdaten mit minimaler Latenz beim Nutzerlogin
- Keine Replikation von Passwörtern in Shibboleth-spezifische Datenbanken
- *Ein* IdP, der alle abzudeckenden Nutzergruppen authentifizieren kann
- Automatisierte, konfigurierbare Identifikation der zum Nutzer gehörigen Quelle
- Einbezug der Nutzer-IP-Adresse in die Authentifikationsentscheidung (optional)

Abbildung 2 zeigt zum einen die Anbindung verteilter Identitätsdaten mit bestehenden Lösungen, wie sie in Abschnitt 2 vorgestellt wurden. Zum anderen werden die im Rahmen dieser Arbeit erweiterten Konzepte sowie entstandene Komponenten visualisiert, die durch einen Stern gekennzeichnet sind. Dargestellt ist im ersten Teil eine Anbindung verteilter Datenquellen mit JaaS und einer sequentiellen Abarbeitung der Authentifikationsquellen, wie es ein IdP vorsieht. Im zweiten Teil der Abbildung ist visualisiert, wie ein Zugriff auf verteilte Daten durch den Einsatz eines virtuellen Verzeichnisses ermöglicht wird.

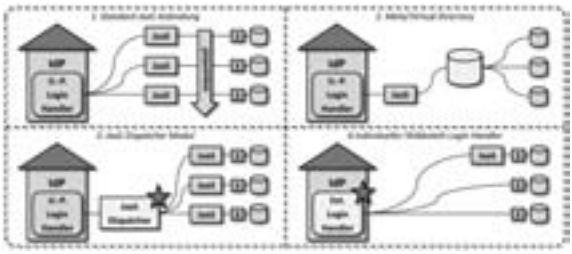

Abbildung 2: Konzepte zur Anbindung verteilter Identitätsdaten

JaaS Dispatcher Module: Um die zu erwartende Verminderung der Authentifikationsgeschwindigkeit durch ein sequentielles Abarbeiten mehrerer in der Datei *login.config* konfigurierter JaaS-Module zu umgehen, wurde ein *JaaS Dispatcher Module* entwickelt, wie es im dritten Teil der Abbildung 2 visualisiert ist. Dieses wendet reguläre Ausdrücke, die in der Datei *login.config* konfiguriert werden können, auf die Nutzernamen an und entscheidet, an welches weitere JaaS-Modul die Authentifikation delegiert werden muss.

Das Dispatcher-Modul ist ebenfalls ein JaaS-Modul und implementiert das Interface *LoginModule*. Für die nachgelagerten JaaS-Module wird jeweils ein eigener Namensraum in der Datei *login.config* konfiguriert und diese in der Methode *initialize* des Dispatcher-Moduls, zusammen mit den regulären Ausdrücken, eingelesen. In der Methode *login* wird anschließend ein entsprechender *LoginContext* aufgebaut und die Authentifikation an diesen delegiert. Der Aufruf des zum *LoginContext* gehörigen Moduls verläuft analog dem Aufruf des ersten JaaS-Moduls durch den Shibboleth IdP selbst.

Extended Login Handler: Die Shibboleth IdP-Implementierung stellt den JaaS-Modulen lediglich den Nutzernamen und das Passwort mittels sogenannter *Callback Handler* zur Verfügung, jedoch keine weiteren Attribute aus der Anfrage des Nutzers. Um zusätzlich die IP-Adresse des Nutzers in den Authentifikationsprozess einbeziehen zu können, kann der *Extended Login Handler* eingesetzt werden. Dieser individuell implementierte Shibboleth Login Handler ist im vierten Teil der Abbildung 2 visualisiert. Er ist durch die Replikation und entsprechende Anpassungen der Klassen[4] des *Username Password-Login Handler* in separate Namensräume, einer Erweiterung der Klasse *BaseSpringNamespaceHandler* und dem Ausbau der Methode *authenticateUser* der Klasse *UsernamePasswordLoginServlet* realisiert worden. In diese Methode wurde, analog dem Ansatz mit einem JaaS-Dispatcher-Modul, die Auswahl der Datenquelle und die Logik für die Authentifikation eingebracht. Auf die IP-Adresse des Nutzers wird über den Http-Request durch den Aufruf *request.getRemoteAddr* zugegriffen und der entsprechende Wert optional in die Authentifikationslogik integriert. Die Authentifikation gegen die unterschiedlichen Schnittstellen der Identitätsdaten kann über das konfigurierte JaaS-Modul oder über direkte Zugriffe auf proprietäre Schnittstellen implementiert werden.

Alternativ ist es möglich weiterhin das *JaaS Dispatcher Module* zu nutzen und lediglich die Übergabe der IP-Adresse des Nutzers in den Shibboleth Login Handler zu integrieren. Die Bereitstellung der IP-Adresse kann als zusätzlicher Callback implementiert werden, indem die Methode *handle* der Klasse *UsernamePasswordLoginServlet* entsprechend erweitert wird. In jedem Fall ist für den Einbezug der IP-Adresse des Nutzers die Ergänzung eines individuellen Login Handler notwendig.

Rückblickend auf die gestellten Anforderungen wahren beide vorgestellten Lösungen die Autonomie der Organisationseinheiten bezüglich ihrer Nutzerkonten, da direkt gegen diese authentifiziert wird und aus einer Kontensperrung eine umgehende Sperrung des Shibboleth-Zugangs folgt. Der Zugriff auf bestehende Nutzerverwaltungen erspart des Weiteren die Replikation von Passwörtern in eine dedizierte Shibboleth-Authentifikationsquelle.

[4] UsernamePasswordLoginHandler, UsernamePasswordLoginServlet, UsernamePasswordLoginHandlerBeanDefinitionParser, UsernamePasswordLoginHandlerFactoryBean

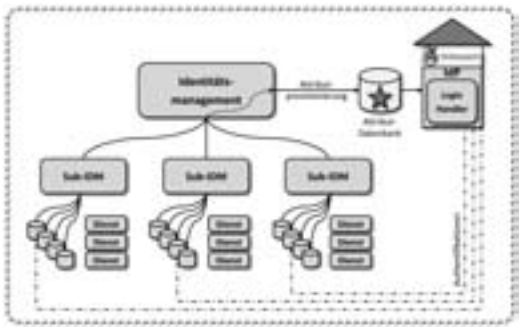

Abbildung 3: Integrierte Attributprovisionierung für einen Shibboleth IdP

Durch den Einsatz der vorgestellten Lösungen, können mit *einem* IdP verteilt vorliegende Nutzerkonten zur Authentifikation genutzt werden. Da beide Module automatisiert und anhand konfigurierbarer regulärer Ausdrücke eine Vorauswahl der Authentifikationsquelle treffen, werden lediglich Anfragen an für ein Nutzerkonto zuständige Quellen gestellt und andere Quellen übersprungen, wodurch eine Leistungsoptimierung erreicht werden konnte. Durch die Möglichkeit des Zugriffs auf die IP-Adresse im *Extended Login Handler* konnte abschließend auch der optionalen Anforderung an den Einbezug dieses Merkmals in die Authentifikationsentscheidung gerecht werden. Ist ein solcher Einbezug nicht notwendig, ist der Einsatz des *JaaS Dispatcher Module* vorzuziehen, da dieser keinen Eingriff in den Shibboleth-Code erfordert und zur Integration lediglich konfiguriert werden muss.

3.2 Integrierte Attributprovisionierung

Für das Konzept zum Zugriff auf Nutzerattribute wird im Folgenden angenommen, dass eine Organisation in Einheiten aufgespalten und ein IdM-System im Einsatz ist [SHH08]. Dieses System versorgt die Organisationseinheiten mit personenbezogenen Daten der Verwaltungseinheiten und kann Daten eines Sub-Systems in andere replizieren. Die Verwaltung lokaler Nutzerkonten sowie die Anbindung von Diensten obliegen jedoch den einzelnen Einheiten. Abbildung 3 zeigt einen solchen IdM-Aufbau mit einem bereits integrierten Shibboleth IdP (rechts im Bild). Der IdP wurde mit einem individuell implementierten Authentifikationsmodul versehen, das nach den Konzepten aus Abschnitt 3.1 implementiert wurde und die Schnittstellen zu Nutzerverwaltungen der einzelnen Organisationseinheiten anbindet. Attribute, die an Web-Dienste ausgeliefert werden sollen, werden dagegen in einer dedizierten Attribut-Datenbank dem IdP aufbereitet und im erforderlichen Schema zur Verfügung gestellt. Im Unterschied zu einem Meta-Directory werden hier nur die für die Shibboleth-Attributlieferung benötigten Daten abgelegt, andere Daten und Passwörter werden nicht repliziert.

Es bietet sich an benötigte Attribute zu allen Nutzerkonten in einer solchen dedizierten Datenbasis zu aggregieren, da einerseits eine Shibboleth-seitige Konfiguration des Mappings von Attributen in das gewünschte Zielschema der AAI (beispielsweise *eduPerson*)

entfällt. Die Übersichtlichkeit der Shibboleth-Konfiguration bleibt gewahrt und der initiale Konfigurationsaufwand dafür wird minimiert. Ferner liegen einige benötigte Attribute, insbesondere für das *eduPerson*-Schema, nicht in den verteilten Nutzerverwaltungen vor und können demnach nicht ad-hoc eingeholt werden, sondern sind in Abhängigkeit der Nutzergruppe lediglich implizit bekannt. Hierzu zählen zum Beispiel Attribute, die Beziehungen der Nutzer zur Organisation oder Rechte innerhalb der AAI repräsentieren.

3.3 Bewertung

Zunächst widersprüchlich könnten die auseinanderlaufenden Konzepte für die Authentifikation und die Attributlieferung scheinen. Bei der Authentifikation ist das verfolgte Ziele auf verteilt vorliegende Identitätsdaten zuzugreifen, Attribute werden hingegen aggregiert und zentral vorgehalten. Die Alternative auf Seiten der Authentifikation wäre, die Nutzerkonten und damit auch sensible Informationen wie Passwörter zentral und redundant vorzuhalten. Dagegen spricht jedoch nicht nur das Risiko des Transports der Passwörter von einer in eine andere Datenbank, sondern auch die verminderte Entscheidungsfreiheit der Organisationseinheiten über die Kontoberechtigungen. Würden die Nutzerkonten in eine zentrale Datenbasis repliziert, müssten Mechanismen geschaffen werden, die bei einer Kontensperrung oder einem Rechteentzug auch die Möglichkeit der Shibboleth-Authentifikation für betroffene Nutzer unterbinden. Mit der verteilten Haltung der Nutzerkonten wird dies implizit vollzogen, wenn in den Organisationseinheiten ein Kontostatus verändert wird. Ein virtuelles Verzeichnis würde dem entgegen kommen. Jedoch bedeutet der Einsatz dieser zusätzlichen Komponente mehr betrieblichen Aufwand und damit höhere Betriebskosten, als ein in Shibboleth integriertes Modul aufwirft. Auf technischer Seite stellt der Einsatz eines virtuellen Verzeichnisses auch einen Aufbau einer weiteren Fehlerquelle für die Authentifikation dar, die stets überwacht werden muss. Der Integrationsaufwand für den Einsatz eines Authentifikationsmoduls fällt hingegen sehr gering aus.

Der Aufwand für den Aufbau und Betrieb einer dedizierten Attribut-Datenbank ist ebenfalls gering. Alternativ könnte auch die Attributlieferung durch einen dezentralen Ansatz realisiert werden. Wie bereits beschrieben, ist jedoch einer der Vorteile, dass mit einem aufbereiteten Satz von Daten die Geschwindigkeit einer Authentifikation erheblich gesteigert werden kann. Zum Teil aufwändige Berechnungen von Attributen zur Laufzeit würden die Bereitstellung von Attributen erheblich mindern. Ein weiterer Vorteil der dedizierten Attributquelle birgt die Verlagerung der Schemakonvertierung zum IdM-System, anstatt dies durch den Shibboleth IdP durchführen zu lassen. Attribute können so nicht nur automatisiert aktuell gehalten werden, die zum Teil komplexe Schemakonvertierung, zum Beispiel in ein AAI-weit gültiges Attributschema durch die Konfiguration des IdPs, wird durch die wesentlich flexiblere Konvertierung durch ein IDM-System ersetzt. Der Verzicht auf eine zusätzliche Erweiterung der Shibboleth-Software und damit der Einsatz einer dedizierten Datenbasis für Attribute ist abschließend auch dadurch begründet, dass es Nutzergruppen geben kann, für die Attribute nur implizit bekannt sind und nicht explizit in den Datenquellen der verschiedenen Nutzerverwaltungen vorliegen.

4 Shibboleth am KIT

Am KIT[5] ist ein föderatives IdM-System etabliert, das Organisationseinheiten mit Daten versorgt, die ihrerseits diese Daten anreichern und in den eigenen IT-Systemen einsetzen können, um Dienste anzubieten. Diesem Identitätsmanagement kommt eine besondere Bedeutung zu, da sowohl der Campus Süd (ehem. Universität) mit ca. 4.000 Mitarbeitern und ca. 20.000 Studierenden sowie der Campus Nord (ehem. Forschungszentrum) mit ca. 4.000 Mitarbeitern jeweils eine eigenständige Personalverwaltung betreiben. Durch die Gründung des Steinbuch Centre for Computing (SCC) als gemeinsames Rechenzentrum und dem Aufbau eines föderativen IdM-Systems sind viele Herausforderungen bezüglich der bestehenden und neu hinzukommenden Nutzerkonten überwunden worden. Dennoch liegt hier eine bereits gegebene Aufteilung in Organisationseinheiten vor, die Vorhaben im Bereich des Identitätsmanagement erschweren und für die Etablierung von Shibboleth den Einsatz der skizzierten Lösung nötig machen.

Aus dem Verständnis der Einrichtung als eine Föderation seiner eigenen Satelliten entstand das föderative IdM-System des KIT [SHH09]. Die personenbezogenen Daten, die Organisationseinheiten zur Verwaltung von Nutzerkonten benötigen, werden vom zentralen IdM-System an die Satelliten IdM-Systeme verteilt. Dienste, die den Verantwortungsbereich einer einzelnen Organisationseinheit übersteigen, wie ein Shibboleth IdP, werden direkt vom föderativen IdM-System gespeist. Eine Integration eines Shibboleth IdPs sowie dessen Zugriff auf getrennt verwaltete Nutzerkonten der beiden Organisationsteile des KIT und darüber hinaus deren Organisationseinheiten konnten mit dem oben vorgestellten Konzept realisiert werden. Die verschiedenen Datenquellen wurden für Authentifikationen durch einen *Extended Login Handler* angebunden und eine dedizierte Attribut-Datenbank eingerichtet, die durch das IDM-System provisioniert wird.

Um den Shibboleth IdP hochverfügbar anbieten zu können, wurden zwei Hardware-basierte sowie zwei virtuelle Maschinen aufgesetzt. Die IdPs werden in einem dedizierten virtuellen LAN betrieben und ein F5 BIG-IP Loadbalancer[6] voran geschaltet. Diese Hardware stellt den Zugangspunkt zur Shibboleth-Infrastruktur. Mit Terracotta[7] wird zudem die Sitzungsverwaltung geclustert, so dass bestehende Nutzersitzungen bei Ausfall einer Maschine auf eine andere übertragen werden und der Nutzer innerhalb der Gültigkeit seiner Sitzung keinen erneuten Loginvorgang durchlaufen muss.

Der betriebliche Aufwand blieb durch den Einsatz des *Extended Login Handler* auf übliche Shibboleth-Betriebsaspekte (Konfiguration neuer Service Provider etc.) beschränkt und es wurden keine zusätzlichen Komponenten etabliert. Zusätzliche Authentifikationsquellen können durch die eingesetzten Module effizient hinzu konfiguriert werden. Initial nicht vorgesehene Attribute lassen sich durch den Einsatz des IdM-Systems flexibel und ebenfalls zeitnah in die etablierte Shibboleth-Attributdatenbank integrieren und durch minimale Konfigurationsänderungen des IdP an entsprechende Service Provider ausliefern.

[5] Zusammenschluss der ehem. Einrichtungen Universität Karlsruhe (TH) und Forschungszentrum Karlsruhe.
[6] http://www.f5.com
[7] http://www.terracotta.org

5 Zusammenfassung und Ausblick

Die Integration eines Shibboleth Identity Provider in bestehende IdM-Infrastrukturen erfordert unter Umständen spezifische Konzepte sowie individuelle Implementierungen, um der Forderung von AAI-Betreibern nach einem IdP pro teilnehmender Organisation gerecht zu werden. In dieser Arbeit wurden Konzepte vorgestellt, mit denen die Authentifikationsmechanismen und Attributlieferung eines Shibboleth IdP an durch Organisationseinheiten verteilt verwaltete Identitätsdaten angebunden werden können. Es wurden bestehende Ansätze diskutiert, die jedoch nicht allen Anforderungen gerecht werden. Anschließend wurden zwei Entwicklungen vorgestellt, welche die zu einem Nutzernamen passende Nutzerverwaltung anhand regulärer Ausdrücke identifizieren und die Authentifikationsanfrage direkt an die entsprechende Datenquelle richten. Durch das vorgstellte *JaaS Dispatcher Module* oder alternativ dem Einsatz des *Extended Login Handler* wird so vermieden, dass Authentifikationsanfragen an für einen Nutzer nicht zuständige Datenquellen gerichtet werden und damit die Latenz einer Authentifikation verringert. Ferner wurde eine Möglichkeit mittels des *Extended Login Handler* vorgestellt, um die IP-Adresse des Nutzers in die Authentifikationsentscheidung einzubeziehen. Abschließend wurde gezeigt, wie das Konzept am KIT umgesetzt wurde, um Shibboleth trotz verteilt vorliegender Identitätsdaten als Authentifikations- und Attributlieferdienst anbieten zu können.

Das vorgestellte Konzept zur Attributprovisionierung kann in Zukunft weiter verfeinert werden, indem die Attribut-Datenbank nicht vorprovisioniert, sondern erst nach der ersten erfolgreichen Authentifikation eines Nutzers mit dessen Attributen befüllt wird. Mit dieser Erweiterung hätten Nutzer die Entscheidungsfreiheit, ob ihre Attribute auch der Shibboleth-Infrastruktur zur Verfügung stehen sollen. Darüber hinaus würde die Attribut-Datenbank nicht mit Attributen von Nutzern gefüllt, die eine Shibboleth-Authentifikation nicht benötigen. Ferner wäre eine Optimierung denkbar, mit der nur die Attribute provisioniert werden, die für den gewünschten Dienst angefordert werden. Herausforderungen für dieses Konzept stellen die notwendige Verfügbarkeit aller potentiellen Quellen für Attribute und etwaige Minderungen der Latenz durch das Aggregieren und Konvertieren von Daten zwischen einer Authentifikation und der Auslieferung von Attribute.

Literatur

[Mah03] Qusay H. Mahmoud. Java Authentication and Authorization Service (JAAS) in Java 2, Standard Edition (J2SE) 1.4. URL: http://java.sun.com/developer/technicalArticles/Security/jaasv2/, September 2003.

[ORBL09] B. Oberknapp, A. Ruppert, F. Borel und J. Lienhard. From a pile of IP addresses to a clear authentication and authorization with Shibboleth. *Serials*, 1:28–32, 2009.

[SHH08] F. Schell, T. Höllrigl und H. Hartenstein. Federated and Service-Oriented Identity Management at a University. In *Proceedings of the 14th European University Information Systems (EUNIS 2008)*, Juni 2008.

[SHH09] F. Schell, T. Höllrigl und H. Hartenstein. Federated Identity Management as a Basis for Integrated Information Management. *it - Information Technology*, 1:14–23, 2009.