Orientation in Objects

Vergleich von Java SSO Lösungen

CAS, OpenSS und Atlassian Crowd

) Schulung)

) Beratung)

) Entwicklung)

) Artikel)

AUTOR

Björn Feustel
Orientation in Objects GmbH

Veröffentlicht am: 10.6.2009

ABSTRAKT

Identity Management(IdM) und Federation Lösungen haben sich in den letzten Jahren zu einem zentralen Teil der Sicherheitsstrategien vieler Firmen entwickelt. Da ein Großteil der im Unternehmen aus Nutzersicht existierenden Anwendungen zumeist webbasiert sind, bildet eine webbasierte Single Sign-On(SSO) Authentifizierung dabei oftmals eine zentrale Komponente und den ersten Schritt bei der Einführung umfassender IdM Lösungen. Der Markt bietet mittlerweile auch eine Vielzahl von SSO-Frameworks und -Produkten als Basis für die Implementierung einer SSO-Infrastruktur an, sowohl kommerzielle als auch freie. Doch was kann man von aktuellen Frameworks erwarten, worin unterscheiden sie sich und welche Vorteile bringen kommerzielle Produkte?

Orientation in Objects GmbH

Weinheimer Str. 68
D-68309 Mannheim

Tel. +49 (0) 6 21 - 7 18 39 - 0
Fax +49 (0) 6 21 - 7 18 39 - 50

www.oio.de info@oio.de

ÜBERBLICK

Dieser Artikel vergleicht exemplarisch drei Vertreter der Gattung:

- **JA-SIG Central Authentication Service (CAS)**- ein freies, verbreitetes und flexibles SSO-Framework
- **Sun Open Web SSO project (OpenSSO)**- eine umfassende IdM Lösung aus dem Hause Sun
- **Atlassian Crowd**- ein kommerzielles, kostengünstiges und leicht zu integrierendes SSO und IdM-Produkt

Kriterium	JA-SIG CAS	Atlassian Crowd	OpenSSO
Homepage	http://www.ja-sig.org/products/cas/	http://www.atlassian.com/software/crowd/	https://opensso.dev.java.net/
Lizenz	JA-SIG Lizenz	Kommerziell, $600(25 Users) - $8000(unbegrenzt) pro Jahr	CDDL
Support	-	Atlassian selbst (In Lizenz enthalten)	Sun selbst für Enterprise und Express Builds
Doku/Community	Dokumentation, Wiki, Foren, Mailinglisten, Artikel und Vorträge, Bugtracking	Dokumentation, Wiki, Foren, Bugtracking, Artikel und Vorträge	Dokumentation, Bucktracking, Mailinglisten

Tabelle 1: SSO-Lösungen im Vergleich

ANFORDERUNGEN

Enterprise Single Sign On Systeme(E-SSO) müssen je nach konkretem Einsatzszenario und vorhandenen Sicherheitsrichtlinien den folgenden funktionalen und nicht funktionalen Anforderungen genügen.

HOHE SICHERHEIT

Die zentrale Rolle eines SSOs mit seinen weitreichenden Rechten macht einen lückenlosen Schutz gegen Manipulation und Missbrauch unerlässlich, denn wird das SSO-System selbst oder ein SSO-Account kompromittiert, kann das weitreichende Folgen für die gesamte Unternehmenssicherheit haben. Ein Angreifer erhielte unter Umständen Zugriff auf sämtliche, dem SSO-Kontext zugeordnete Daten und Applikationen. Um dem entgegenzuwirken, existieren neben üblichen systemseitigen Schutzmechanismen verschiedene anwendungsspezifische Ansätze, die zumeist entsprechend den konkreten Sicherheitsanforderungen kombiniert werden.

SINGLE SIGN-OUT

Einer davon ist das *Single Sign-Out*. Es sorgt für die Beendigung der gesamten SSO-Session eines Nutzers, wenn dieser sich bei einer einzelnen Anwendung abmeldet und sorgt so dafür, dass Dritte nicht die als beendet geglaubte SSO-Session missbrauchen können. Dieses Feature stellte in der Vergangenheit oftmals gerade bei Web-SSO Systemen ein Problem dar, da es ein Management der SSO-Sessions und eine komplexere Kommunikation zwischen SSO-Server und SSO-Client (*Policy Enforcement Point*, kurz *PEP*) voraussetzt. Letzterem Umstand wurde im Übrigen auch bei der Entwicklung der SAML V2.0 Spezifikation [1] Rechnung getragen, welche ein eigenes Single Logout Protocol enthält.

STRONG AUTHENTICATION

Eine weitere Anforderung an E-SSO Systeme ist die Fähigkeit, mehrere Authentifizierungsverfahren kombinieren zu können, was oftmals auch als *Strong Authentication* bezeichnet wird. Die Idee dahinter ist, Nachteile einzelner Authentifizierungsverfahren durch die Kombination verschiedenartiger Methoden (z.B. beruhend auf Wissen und Besitz) auszugleichen und so die Sicherheit zu erhöhen. Ein typisches Beispiel im Falle von webbasierten SSO-Systemen ist die Kombination von Nutzerpasswörtern (Wissen) und Client-Zertifikaten (Besitz), die auf dem Rechner des Nutzers installiert werden.

FORCED RENEWAL:

In einigen Situationen wie dem Ändern der Nutzer-Credentials kann es aus Sicherheitsgründen erforderlich sein, trotz einer bestehenden SSO-Session eine erneute Authentifizierung zu erzwingen. Dieses Verfahren wird auch *Forced Renewal* genannt und bietet sich zur selektiven Erhöhung der Sicherheit in der Ausführung einzelner Prozessschritte an.

GUTE INTEGRATIONSMÖGLICHKEITEN

Eine zentrale Authentifizierung und ein unternehmensweites Sicherheitskonzept lässt sich nur etablieren, wenn sich alle relevanten und gewünschten Systeme in einen gemeinsamen SSO-Kontext integrieren lassen. Demnach ist eine gute Integrierbarkeit natürlich eine elementare Anforderung an ein SSO-Framework.

INTEGRATION BESTEHENDER SYSTEM

Aktuelle Frameworks bieten hierfür auch fertige Adapter für verbreitete Anwendungen wie Application Server, LDAP Verzeichnisse oder Standardsoftware. Allerdings ist die zugrunde liegende Problemstellung in aller Regel zu komplex, um sie nur durch die Konfiguration fertiger Komponenten lösen zu können, so dass die fehlenden Bindeglieder und Funktionalitäten individuell implementiert werden müssen. Dazu muss das Framework jedoch die relevanten Funktionen in Form von APIs zur Verfügung stellen, entweder speziell für einzelne Programmiersprachen oder universell als Web Services.

Welches die potentiellen Integrationspunkte einer bestehende Infrastruktur sind, zeigt folgende Abbildung:

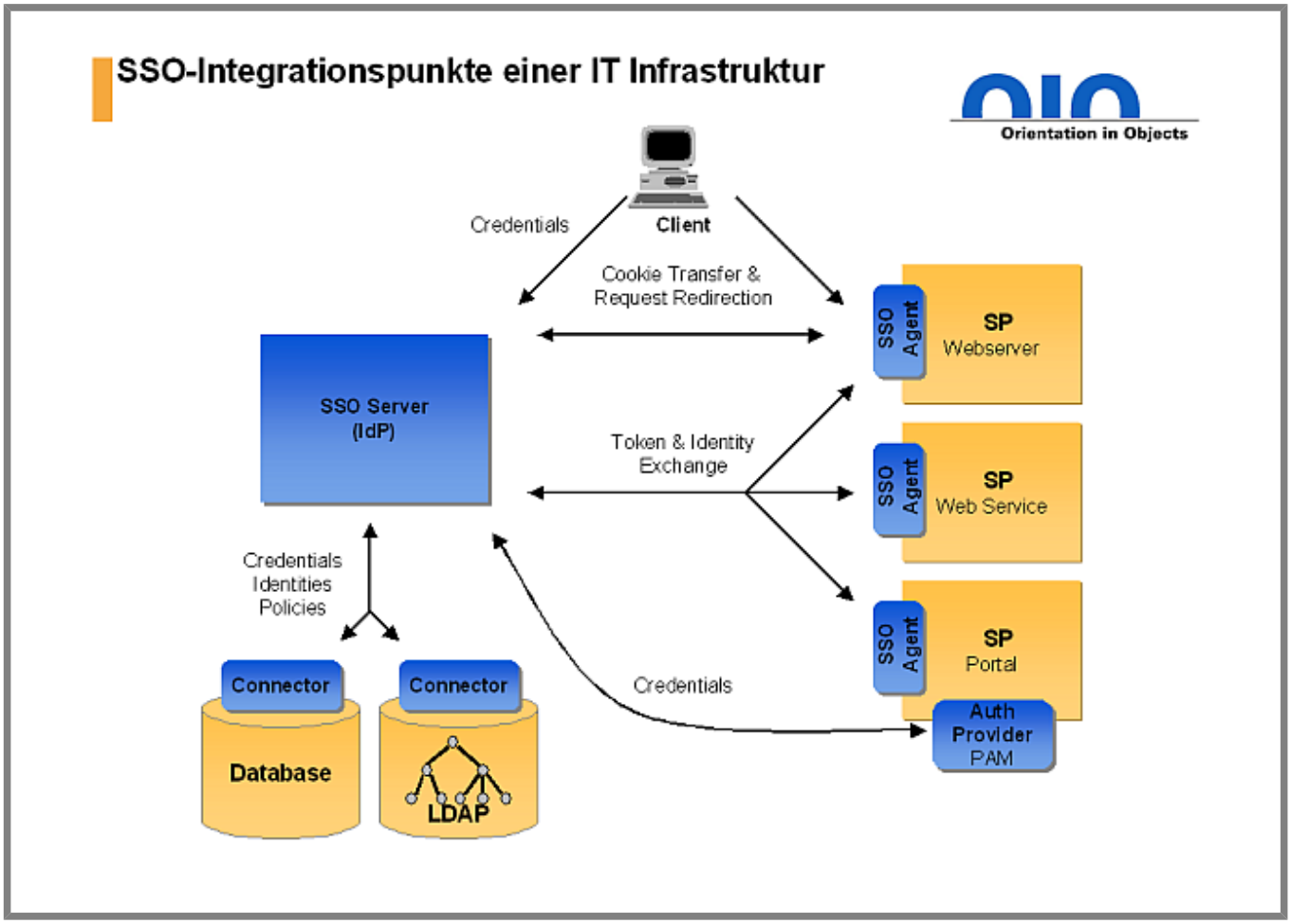

Abbildung 1: SSO-Integrationspunkte einer IT Landschaft

Der SSO-Server, der als zentrale Authentifizierungsinstanz arbeitet und als einziger die Überprüfung der Nutzer-Credentials vornimmt, muss über Adapter, oftmals *Connectors* genannt, an bestehende Datenquellen gebunden werden. Diese Datenquellen beinhalten zum einen die Nutzerdaten, welche für die eigentliche Autorisierung notwendig sind (Login, Passwort, Zertifikate), können aber - je nach Mächtigkeit der verwendeten SSO/IdM-Lösung - auch umfassendere Daten wie Rollenzuweisungen, Berechtigungsschemata oder weitergehende Identitätsdaten enthalten. Die technische Basis sind hier meist proprietäre Datenbanken oder Verzeichnisdienste. Neben der Anbindung des SSO-Servers gilt es natürlich auch, die SSO Lösung mit den Anwendungen (*Service Provider*, kurz *SP*) zu integrieren, die in den zu schaffenden SSO-Kontext einbezogen werden sollen. Dies geschieht in aller Regel durch sogenannte *SSO-Agents*, welche mit dem SSO-Server kommunizieren und als *Policy Enforcement Points (PEP)* die von ihm vorgegebenen Regeln und Berechtigungen in den jeweiligen Anwendungen durchsetzen.

Üblicherweise bieten SSO-Lösungen bereits spezifische SSO-Agents für Web- und Applicationserver oder Standardsoftware, so dass sich der Integrationsaufwand hier zumindest bei einem Teil der vorhandenen Systeme in Grenzen hält. Möchte man nicht unterstützte Anwendungen integrieren, muss man entweder auf Reverse-Proxy-Mechanismen zurückgreifen oder eigene, angepasste SSO-Agents entwickeln.

CROSS-DOMAIN SSO (CDSSO)

Erstreckt sich ein SSO-Kontext über mehrere DNS Domains, spricht man von *Cross-Domain-Single Sign-On*. Dieses bedingt im Gegensatz zu einem Web-SSO-System für eine Domain, welches den Authentifizierungsstatus durch Browser-Cookies halten und übertragen kann, eine komplexere technische Implementierung, denn Cookies können nur innerhalb derselben Domain gespeichert und wieder gelesen werden. In der Vergangenheit haben SSO-Systeme dazu eigene, nicht standardisierte Mechanismen verwendet, die die Identitätsinformationen von einer Domain in die andere transportierten. Mittlerweile hat sich aber auch hier SAML als Standard etabliert und wird auch zumindest teilweise von den hier verglichenen Kandidaten implementiert.

ROBUSTHEIT UND GUTE ADMINISTRIERBARKEIT

SSO-Systeme spielen ihrer Natur nach eine zentrale Rolle in der IT-Infrastruktur und müssen deshalb äußerst robust arbeiten und hoch verfügbar sein. Darüber hinaus müssen sie aber natürlich auch Möglichkeiten zur Überwachung und Nachvollziehbarkeit des Betriebs bieten, um den administrativen Anforderungen zu genügen.

HIGH AVAILABILITY

Ein Ausfall des SSO-Systems, selbst für kurze Zeit, würde u.U. das komplette Unternehmen lahmlegen. Ebenso bliebe eine schlechte Performance nicht ohne Folgen für die Produktivität. Daher sind eine gute Skalierbarkeit und ein funktionierendes Failover unverhandelbare Anforderungen, die bei der Auswahl eines SSO-Systems vor dem konkreten nicht-funktionalen Anfoirderungsprofil berücksichtigt werden müssen.

ADMINISTRIERBARKEIT

Aber auch administrative Features wie ein Monitoring der Betriebsparameter, Backup/Restore-Möglichkeiten für die Konfigurations- und Nutzerdaten oder Audit-Funktionen sind hoch interessant, um beispielsweise gegebene Sicherheitsstandards oder Richtlinen zu erfüllen.

DIE KANDIDATEN

JA-SIG CAS

Der Central Authentication Service wurde ursprünglich an der University of Yale unter dem Namen *Yale CAS* als einfaches Single Sign-On System entwickelt. Nachdem das System auf breites Interesse stieß, wurde es im Jahre 2004 Teil der durch die *Java Architectures Special Interest Group* gesponserten Projekte und fortan unter dem Namen *JA-SIG CAS* als Open Source Projekt weiterentwickelt.

INSTALLATION

Installation Die CAS Serverkomponente lässt sich als leichtgewichtiges Installationsarchiv von der JA-SIG Webseite laden. Es enthält neben verschiedenen Erweiterungen zur Systemintegration auch den eigentlichen Identity Provider, der als WAR Archiv direkt in einem Servlet-Container installiert werden kann, welcher die Servlet Spezifikation in der Version 2.4 implementiert. Im Gegensatz zu den anderen beiden Kandidaten beinhaltet die Basisinstallation jedoch keine Administrationsoberfläche. Anpassungen am Aussehen und der Funktion erfolgen ausschließlich über Konfigurationsdateien. Da die Qualität und Vollständigkeit der Dokumentation von CAS nicht homogen ist, kann dies auch schon mal etwas kniffliger und mit einiger Recherche verbunden sein.

INTEGRATION

Die Integration von Nutzerverzeichnissen und Authentifizierungsquellen wie LDAP-Verzeichnisse, RADIUS-Server oder Datenbanken erfolgt über sogenannte *AuthenticationHandler*, die in den Laufzeitpfad der Serverkomponente gelegt und per Konfigurationsdatei bekanntgegeben werden.

Die Integration von Anwendungen in den SSO-Kontext erfolgt in CAS ähnlich den anderen Kandidaten über spezifische Client-Adapter, wobei bereits Adapter und Anleitungen für verschiedene Anwendungen und Container wie Apache Webserver, Magnolia CMS oder Oracle Portal zur Verfügung stehen. Reicht dies nicht, lassen sich verschiedene Client-APIs nutzen, um die Integration passgenau durch Eigenentwicklungen umzusetzen.

Eine besondere Form der Integration bietet CAS mit der sogenannten *Proxy Authentication*. Dabei räumt CAS einer vermittelnden Komponente das Recht ein, im Auftrag eines Users unter dessen Identität auf per CAS geschützte Services zuzugreifen, wie es beispielsweise ein Portalserver muss, der persönliche Daten des angemeldeten Nutzers aus Drittsystemen konzentriert.

SSO FEATURES

CAS hat sich im Verlauf seiner Entwicklung von einem reduzierten und spezialisierten SSO System zu einem mittlerweile ausgereiften SSO Framework entwickelt, das auch fortgeschrittene Features wie serverseitiges Session Management für eine Single Log-Out Funktionalität oder eine fragmentarische SAML 2.0 Unterstützung bietet. Mit der "Remember Me" Funktionalität unterstützt CAS auch eine Langzeitauthentifizierung, die die Authentifizierung eines Nutzers auch über den Zeitraum einer Session hinaus bestehen lässt, wodurch bei einem späteren Zugriff ein erneutes Anmelden entfällt. Auch wenn dieses Feature Sicherheitsrisiken birgt, kann es in passenden Situationen die Benutzerfreundlichkeit des Systems erhöhen.

ERWEITERBARKEIT

Grundsätzlich lässt sich CAS über bereitgestellte APIs erweitern, mit denen sich beispielsweise eigene *AuthenticationHandler* Implementierungen schreiben lassen. Andere Aspekte wie zum Beispiel der serverseitige Authentifizierungprozess lassen sich auf technischer Ebene ebenso anpassen. Allerdings wird man auch hier immer wieder mit Lücken oder qualitativen Unterschieden der Dokumentation leben müssen, was oftmals den Blick in die Quellen nötig macht.

LIZENZ UND SUPPORT

JA-SIG CAS steht unter der eigenen "JA-SIG License", die eine Verwendung, Verteilung und Modifikation der Software erlaubt. Neben den üblichen und im Falle von CAS auch gut funktionierenden Supportkanälen von Open Source Software wie Wiki, Mailingliste oder Issue-Tracker bieten mittlerweile ein paar Firmen Supportdienstleistungen rund um CAS an [2].

OPENSSO

Das *Open Web SSO project (OpenSSO)* ist ein von Sun Microsystems gesponsertes Open Source Projekt, welches auf dem kommerziellen Sun Java System Access Manager Version 7.x beruht. Sun bietet es selbst als Teil seiner IdM Strategie in zwei kommerziellen Editionen an.

Technisch bildet OpenSSO eine umfassende IdM Lösung, welche neben Single Sign-On Funktionalitäten auch solche zu Autorisierung, Identity Federation (z.B. SAML und WS-Federation) und Web Service-Security umfasst.

INSTALLATION

Wie sich vielleicht in Anbetracht des großen funktionellen Umfangs und das Hintergrunds erahnen lässt, hat man es bei OpenSSO nicht unbedingt mit einer leichtgewichtigen Softwarekomponente zu tun. Und so umfasst der Download des aktuellen OpenSSO Enterprise 8.0 auch knapp 300 MB, enthält dafür allerdings neben den Serverkomponenten auch Client-Bibliotheken, Tools und Dokumentation.

Die Basisinstallation von OpenSSO erfolgt anschließend jedoch erfreulicherweise durch das einfache Deployen eines einzelnen WAR-Files in einem der zahlreichen unterstützten Web Container und der anschließenden Konfiguration mittels des dann verfügbaren webbasierten Wizards, der Schritt für Schritt durch den Konfigurationsprozess führt. Die eigentliche Verwaltung des SSO Kontexts erfolgt danach über eine zentrale, webbasierte Administrationskonsole, welche Zugriff auf die Detailkonfiguration sämtlicher Komponenten bietet. Alternativ stehen Kommandozeilentools zur Verfügung.

INTEGRATION

Dank des soliden Produkthintergrundes und der mittlerweile etablierten Community um das Open Source Projekt bietet OpenSSO eine hohe technische Flexibilität und eine umfangreiche Integrationspalette.

User- und Konfigurationsdaten lassen sich für Testzwecke im integrierten OpenDS Verzeichnis ablegen, sollten in Produktivumgebungen jedoch in externe Repositories ausgelagert werden. Dabei werden bereits verschiedene Systeme wie Microsoft Active Directory oder IBM Tivoli Directory Server unterstützt und über Plugins lassen sich beliebige andere anbinden.

Authentifizierungsverfahren lassen sich über sogenannte *Authentication Modules* abbilden, wobei die mit OpenSSO gelieferten bereits einen großen Teil der üblichen Anwendungsfälle abdecken. Als Beispiele seien Module für die Kerberos-, RADIUS- und zertifikatbasierte Authentifizierung genannt, wobei auch hier beliebige Erweiterungen über Plugins möglich sind.

Für die Anwendungsintegration in den SSO-Kontext stellt OpenSSO für viele Standard-Container wie Apache, IIS, BEA Weblogic oder SAP Enterprise Portal bereits PEPs (sogenannte *Policy-Agents*) zur Verfügung, die sich ebenfalls über entsprechende Plugins erweitern lassen. Die Konfiguration dieser Agents wird dabei zentral über die Administrationskonsole verwaltet.
Bietet die Zielanwendung keine Möglichkeit, einen Agent zu installieren, kann OpenSSO auch über einen Reverse-Proxy Mechanismus vor diese geschaltet werden und so einen transparenten PEP bilden.

Wem die Plugin-basierte Integration nicht reicht oder zu spezifisch ist, dem stellt OpenSSO zusätzlich seine gesamten Identity-Services über universelle SOAP und REST-Schnittstellen zur Verfügung.

SSO FEATURES

Der Anspruch von OpenSSO, eine umfassende Sicherheitslösung im Enterprise-Umfeld zu bieten, lässt bereits vermuten, dass die Software viele SSO-spezifische Funktionalitäten bietet. Und so gehören Features wie Cross-Domain-Fähigkeit, Session Verwaltung mit *Single Sign-Out* und Strong Authentication natürlich auch zur Grundfunktionalität. Erweitert wird diese durch einige nicht so selbstverständliche Funktionen. Beispielsweise lässt sich zur Erhöhung der Gesamtsicherheit die User Interface Komponente für die Authentifizierung getrennt vom eigentlichen Authentifizierungservice installieren (Stichwort DMZ). Ein eigenständiger Security Token Service ist nicht nur in der Lage, Security Tokens auszustellen und zu verwalten, er bietet auch einen Plugin-Mechanismus, um Token anderer Access Management Systeme wie CA Siteminder oder Oracle Access Manager zu integrieren.

ERWEITERBARKEIT

Mit den schon mehrfach erwähnten Plugins lassen sich Service Provider Interfaces (SPI) der verschiedenen Services implementieren, womit sich OpenSSO ganz hervorragend erweitern und speziellen Anforderungen anpassen lässt. Daneben existieren von Hause aus Client APIs für Java und C, eine PHP Erweiterung ist auch bereits verfügbar.

LIZENZ UND SUPPORT

Die Codebasis von OpenSSO steht unter der CDDL [3] und steht damit zur freien Verfügung, auch für kommerzielle Zwecke

Besitzt man eine *Sun OpenSSO Enterprise perpetual license*, eine *Sun Identity Management Suite subscription* oder eine *Java Enterprise System subscription*, bietet Sun Support für die beiden angebotenen, kommerziellen Editionen von OpenSSO: *Sun OpenSSO Enterprise* (vormals *Federated Access Manager 8.0*) und *Sun OpenSSO Express*.

Beide basieren auf der OpenSSO Codebasis, werden jedoch in unterschiedlichen Zyklen released. Die Enterprise-Edition soll dabei die Basis für unternehmenskritische Produktivsysteme bilden, erscheint alle 12-15 Monate und wird einem umfangreichen QA-Prozess unterzogen. Die Brücke zu den häufig erscheinenden und wenig gestesteten Community Releases ohne Support bildet die Express-Edition. Sie erscheint alle 3 Monate und macht so neue Features im kommerziellen Zweig verfügbar, allerdings ohne Hotfix-Support und mit einem schwächeren QA-Prozess.

Positiv lässt sich die Qualität der Dokumentation anmerken. Sie erklärt in mehreren Dokumenten ausführlich die verschiedenen Aspekte des Einsatzes von OpenSSO wie z.B. Deployment, Administration, Integration und Erweiterung.

BESONDERHEITEN

Identity Management

Wenn es darum geht, unterschiedliche Identitäten aus unterschiedlichen Kontexten zu integrieren oder Vertrauensverhältnisse zwischen Systemen zu vermitteln, bietet OpenSSO out-of-the-box deutlich mehr Möglichkeiten als die beiden anderen vorgestellten Kandidaten. Durch die Unterstützung von Standards und Protokollen wie SAML, WS-Trust, WS-Federation oder Liberty ID-FF lässt sich relativ einfach ein flexibler *Circle of Trust* etablieren, also ein föderierter Zusammenschluss verschiedener Service und Identity Provider wie er beispielsweise für eine funktionierende *SaaS* Landschaft (*Software as a Service*) notwendig ist. Als Basiskonzept führt Sun dazu publikumswirksam die *Fedlets* ins Feld, bei denen es sich um leichtgewichtige SAML2 Service Provider Implementierungen handelt, welche z.B. direkt in der Administrationskonsole erzeugt werden können.

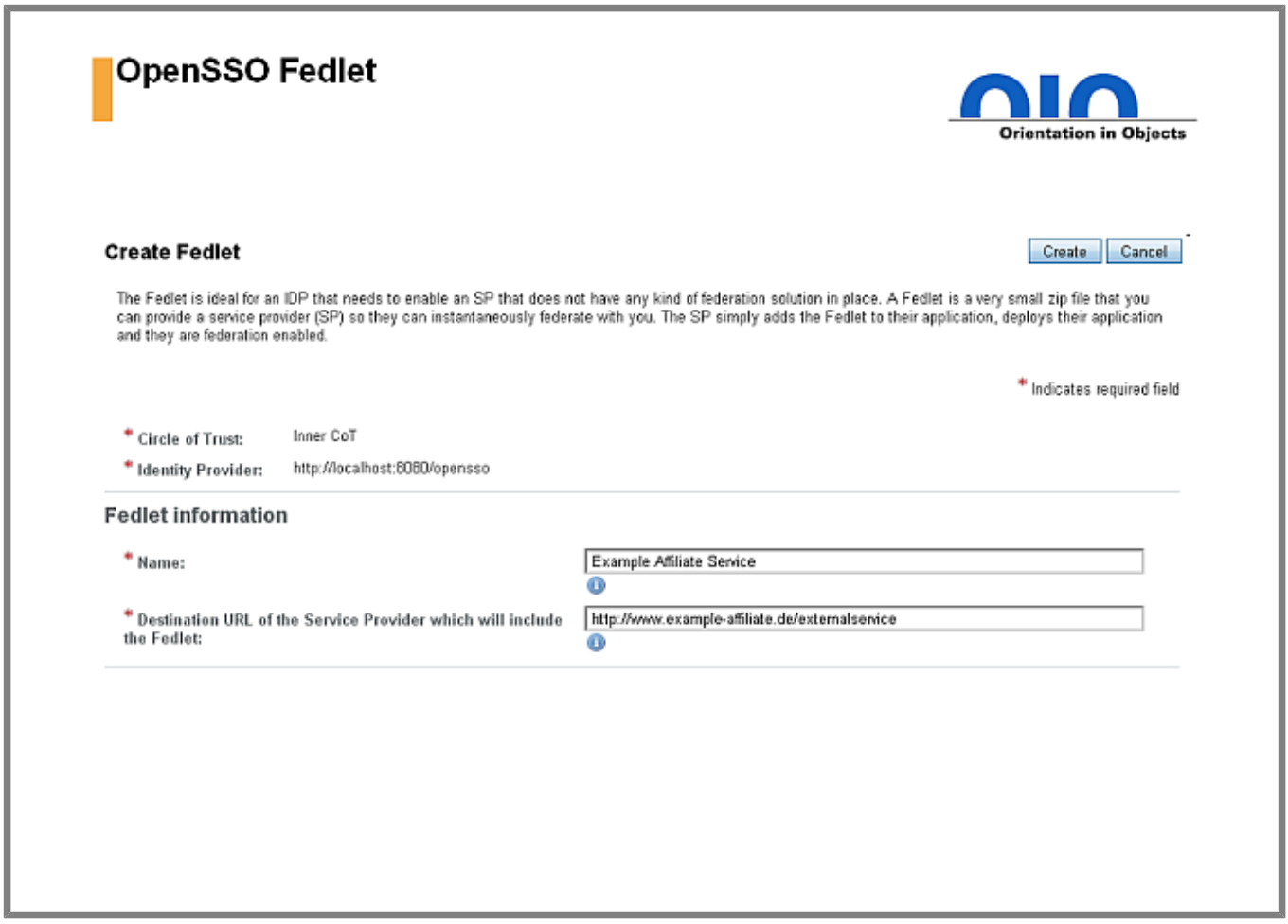

Abbildung 2: Erzeugen eines Fedlet

Deployed in einem Java EE Container oder einer .NET Anwendung ermöglichen sie es einer Applikation, per SAML mit einem Identity Provider zu kommunizieren und so beispielsweise als Service Provider in einem SSO-Kontext zu partizipieren und Zugriff auf ausgewählte Nutzerattribute zu erlangen.

Web Services Sicherheit

Um Sicherheitskonzepte in einer serviceorientierten Architektur zu realisieren, bietet OpenSSO auch Unterstützung für die Absicherung von Web Services. Dabei wird die Kommunikation zwischen Web Service Client (WSC) und Web Service Provider (WSP) durch Security Agents kontrolliert, die als Security Provider im Sinne der *Java Authentication Service Provider Interface for Containers* Spezifikation (JSR-196) realisiert wurden und sich somit leicht in eine JAX-WS 2.0 basierte Web Service Umgebung integrieren lassen. Als Basis für einen gemeinsamen Security Context zwischen WSC und WSP dienen Security Token, die der bereits erwähnte und auf der WS-Trust Spezifikation aufbauende Security Token Service generiert.

ATLASSIAN CROWD

Das von der australischen Firma Atlassian angebotene *Crowd* bietet neben reinen SSO-Features auch Identity Management Funktionalitäten und eignet sich damit für eine umfangreichere Integration in bestehende Systemlandschaften als beispielsweise *JASIG-CAS*. Crowd ist die Weiterentwicklung der Software *Identity Exchange* der Firma Authentisoft, welche Ende 2006 von Atlassian aufgekauft und in ihr Portfolio integriert wurde. Es spielt die zentrale Rolle zur Integration anderer Atlassian Tools wie das Enterprise Wiki *Confluence* oder das Issue Tracking und Projektmanagement Tool *JIRA* aber bietet mittlerweile auch genügend Anknüpfungspunkte, um in Nicht-Atlassian Systemlandschaften als zentrale SSO und IdM-Komponente zum Einsatz zu kommen.

INSTALLATION

Wie auch die anderen Produkte von Atlassian glänzt Crowd durch seine einfache Erstinstallation. Es lässt sich als fertiges WAR-File direkt in einem existierenden J2EE Application Server oder gebündelt mit einem Apache Tomcat als Standalone-Version installieren. Die Grundkonfiguration lässt sich innerhalb von Minuten mit Hilfe eines web basierten Wizards durchführen. Anschließend steht sowohl die Crowd-Administrationskonsole als auch ein OpenID-Server zur Verfügung.
Die Administrationskonsole gibt einen umfassenden Einblick in die Möglichkeiten von Crowd und offenbart dabei den über spezifische SSO-Fähigkeiten hinausreichenden Funktionsumfang, da sich hier alle zentralen Funktionalitäten einfach und schnell verwalten lassen.

INTEGRATION

Die Integration von Nutzerverzeichnissen erfolgt in Crowd durch sogenannte Directory Connectors. Dabei handelt es sich im Normalfall um LDAP-Adapter für gängige Verzeichnisdienste (siehe Tabelle: Vergleich der Integrationsmöglichkeiten)), es können aber auch interne Verzeichnisse verwendet werden, die Crowd in der konfigurierten Datenbank verwaltet. Andere Systeme wie Drittanwendungen oder Datenbanksysteme lassen sich über eigene Connector-Implementierungen integrieren und so als Quelle für Nutzerdaten heranziehen.

Die Integration auf Anwendungsebene erfolgt in Crowd durch Application Connectors, also anwendungsspezifische Komponenten, welche die Rolle des Policy Enforcement Points (PEP) übernehmen. Crowd bietet dazu fertige Konnektoren für verschiedene Anwendungen, zumeist Produkte aus dem Hause Atlassian aber auch für universellere Szenarien, z.B. für die Integration von Apache Webservern oder bestehender Spring Acegi-basierten Sicherheitskontexten.

SSO FEATURES

Da Crowd von Atlassian als eine praktische und einfache SSO und IdM Lösung entwickelt und angeboten wird, verwundert es nicht, dass komplexe SSO/IdM Technologien und Standards wie Shibboleth oder XACML nicht durch das Kernprodukt bedient werden. Anhand der in der aktuellen Version als Plugin realisierten Google Application Anbindung, welche auf SAMLv2 Features beruht, lässt sich jedoch erkennen, dass die vorhandenen Erweiterungsmöglichkeiten viel Raum für zukünftige und von dritten stammende Erweiterungen bietet.

Ein besonderes Feature bietet Crowd mit der eigenen OpenID Provider-Komponente *CrowdID*. OpenID [4] im weitesten Sinne ist ein Verfahren, eine digitale Identität für verteilte Single Sign-On Szenarien mit "OpenID-fähigen" Websites [5] zu nutzen. Im engeren Sinne wird der Begriff OpenID auch für die digitale Identität eines Users selbst benutzt, wobei diese Identität durch eine URI repräsentiert wird. *CrowdID* kann nun als Provider für diese Identitäten verwendet werden und übernimmt dann ebenfalls die notwendige Kommunikation auf Basis des OpenID-Protokolls mit den anfragenden Websites.

ERWEITERBARKEIT

Crowd, eher Produkt als Framework, bietet hierfür drei Möglichkeiten:

- Java APIs: Wie oben erwähnt, stehen für die Implementierung eigener Directory oder Application Connectoren Java-APIs zur Verfügung.
- SOAP Web Service: Maximale Flexibilität und Plattformunabhängigkeit bietet die wohl dokumentierte SOAP API, die sowohl Authentifizierungsfunktionen als auch Funktionen zur Verwaltung des Application Provisioning und des Identity Managements umfasst.
- Plugins: Crowd enthält das auch in anderen Atlassian Anwendungen verwendete Plugin Framework, welches in der aktuellen Version auf OSGi aufbaut. Mit seiner Hilfe ist es möglich, Crowd umfassend funktional zu erweitern. So könnte man beispielsweise durch ein Plugin einen neuen Passwort-Encryption Mechanismus integrieren, die Administrationsoberfläche erweitern oder Event-Listener implementieren, die gezielt auf Authorizationsereignisse reagieren.

LIZENZ UND SUPPORT

Atlassian bietet Crowd ähnlich den anderen Produkten mit unterschiedlichen Lizenzen an. Neben der freien Lizenzierung für offizielle, nicht kommerzielle Organisationen gibt es unterschiedliche Lizenzierungsmodelle für den akademischen oder kommerziellen Bereich, wobei jeweils unterschiedliche Ausbaustufen (nach Nutzerzahlen) verfügbar sind.
Mit einer Lizenz, welche nach 12 Monaten zum halben Anschaffungspreis verlängert werden muss, erhält der Käufer auch immer Upgrades und Quellcode der innerhalb des Lizenzzeitraumes verfügbaren Programmversionen.
Neben der Software bietet Atlassian dem Lizenznehmer einen vielfach gelobten 24/5 Support, den *Atlassian Legendary Service*.

BESONDERHEITEN

Application Provisioning

Eine der funktional über SSO-spezifische Anforderungen hinausgehenden Features von Crowd ist das Application Provisioning, also die Definition und Verwaltung der Nutzerbasis der integrierten Anwendungen. Wer darf eine Anwendung benutzen, in welcher Rolle agiert der Nutzer in dieser und welche Rechte hat die Anwendung selbst, um auf die Daten des Nutzers zuzugreifen? All das lässt sich problemlos über die Administrationskonsole konfigurieren, so denn der jeweilige Application Connector diese erweiterten Funktionen unterstützt.

Mit der Möglichkeit, einer Anwendung mehrere Nutzerverzeichnisse zuzuweisen, bietet Crowd eine sehr mächtige und flexible Funktion, wie sie normalerweise durch Meta-Directories bereitgestellt wird. Für die Anwendung erfolgt dabei der Zugriff auf die Nutzerdaten völlig transparent, d.h. sie arbeitet auf einem großen, virtuellen Verzeichnis.

Identity Management

Neben dem Application Provisioning bietet Crowd mit seinen Identity Management Funktionen eine weitere, über reines SSO hinausgehende Funktionalität. So können bestehende Nutzerdaten zentral gepflegt, neue angelegt und mit Attributen versehen werden, die den integrierten Anwendungen zur Verfügung stehen. Die Organisation der Nutzer erfolgt über Gruppen, die je nach Unterstützung des zugrundeliegenden Verzeichnisses auch geschachtelt sein können.

Um eine bestehende Nutzerbasis in Crowd selbst zu verwalten, weil beispielsweise das eigentliche Verzeichnis keine Speicherung von zusätzlichen Attributen erlaubt, lassen sich Bestandsdaten mit den bereitgestellten Importern (z.B. für LDAP Verzeichnisse und CVS Dateien) recht einfach migrieren.

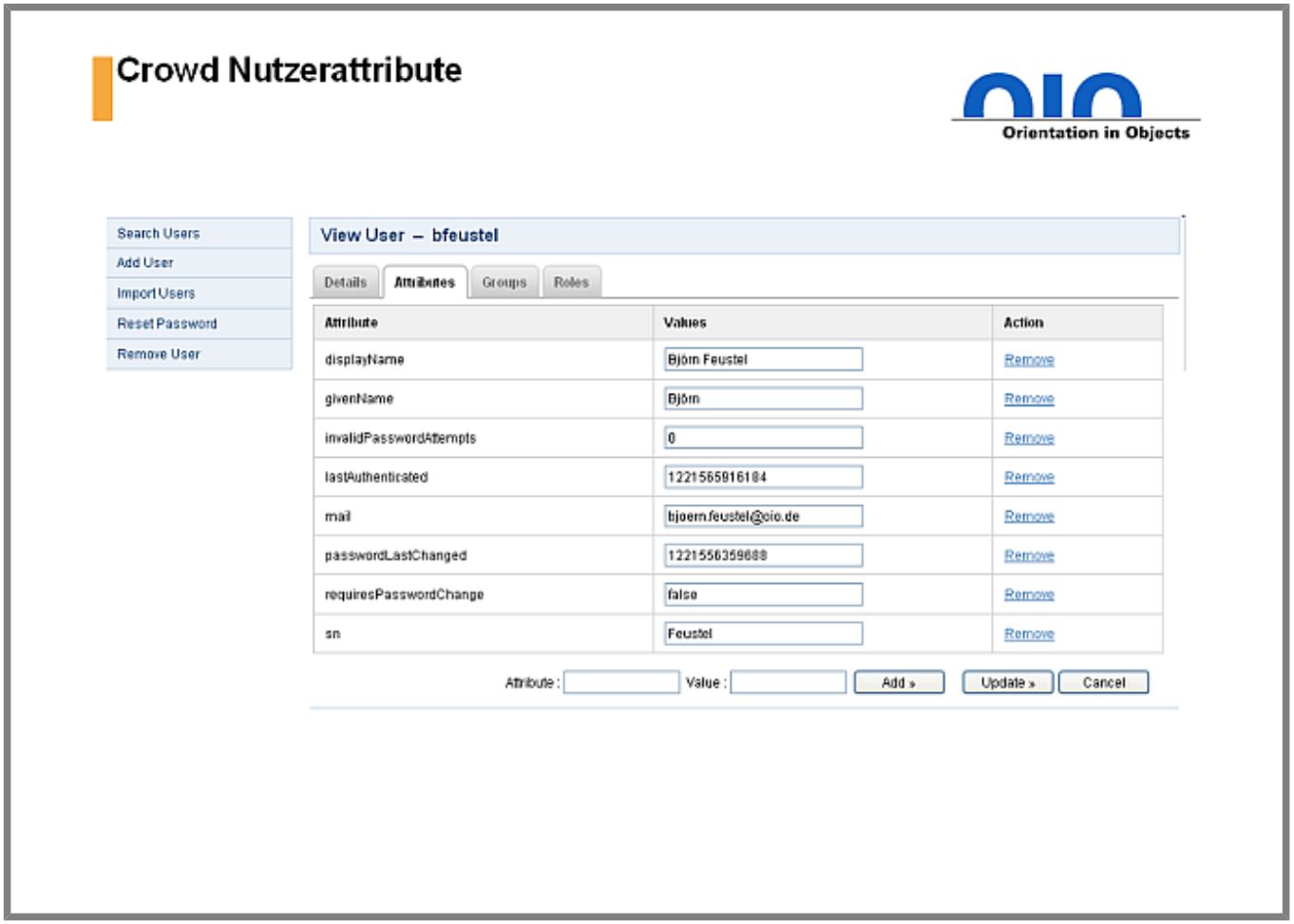

Abbildung 3: Pflege der Attribute eines Nutzers

Für die dezentrale Verwaltung der eigentlichen Nutzerdaten bietet Crowd eine "Self-Service Console". Sie ermöglicht die Profilpflege durch den jeweiligen Nutzer selbst und gibt diesem zusätzlich Aufschluss über die eigenen Gruppenmitgliedschaften und Rollenzuweisungen.

FAZIT

Grundsätzlich lässt sich sagen, dass sich alle drei der hier vorgestellten Kandidaten für die Einführung eines webbasiertes Single Sign-On Systems eignen. Alle drei sind stabil und ausgereift genug und bieten genügend Referenzen, um mit gutem Gewissen im Unternehmensumfeld erfolgreich eingesetzt zu werden. Allerdings variieren ihre Features deutlich. Sowohl in den SSO-relevanten Eigenschaften (siehe Tabelle "Vergleich SSO-relevanter Funktionen") als auch der Unterstützung spezifischer Systeme und Standards (siehe Tabelle "Vergleich der Integrationsmöglichkeiten"), dem Grad ihrer Komplexität und der über einfaches SSO hinausgehenden Funktionalität unterscheiden sie sich deutlich, so dass am Ende doch die konkreten Anforderungen die Auswahl bestimmen müssen.

Kriterium	JASIG CAS	Atlassian Crowd	OpenSSO
Single Sign-Out/Single Log-Out	+	+	+
Strong Authentication	+	- *(1)*	+
Forced Renewal	+	-	+
Cross-Domain	+	-	+
HA/Failover	+	+ *(2)*	+

Tabelle 2: Tabelle: Vergleich SSO-relevanter Funktionen

(1) Clientanfragen lassen sich auf IP-Adressen beschränken
(2) Clustering nicht offiziell supported

Kriterium	JASIG CAS	Atlassian Crowd	OpenSSO
Container and Application Agent	Apache Webserver, IIS, Tomcat, JSR 168/Pluto, Oracle	Apache Webserver, Google Applications, Subversion, Atlassian Produkte, MediaWiki, JForum, Grails	IBM WebSphere, BEA WebLogic, JBoss AS, GlassFish, Apache Tomcat, Apache Webserver, IIS, Lotus, PeopleSoft, SAP und weitere
Service-Protocols		SOAP	REST, SOAP
Client-APIs	Java, .NET, Perl, PHP, Ruby, PAM und weitere	Java, .NET *(1)*, PHP *(1)*, Ruby *(1)*	Java, C, PHP
Identity architectures and Protocolls	SAML 1.0, SAML 2.0, OpenID, RestfulApi,Custom Protocols, cas, sso	OpenID, SAML 2.0 *(2)*	SAML 1.0, SAML 2.0, OpenID, XACML, WS-Federation
Directory Connectors	LDAP, MS Active Directory, Novell eDirectory, Apache DC	Generic LDAP, verschiedene spezifische LDAP-Server (MS Active Directory, Novell eDirectory, usw.)	LDAP, OpenDS
Forms of Authentication	JAAS, JDBC, RADIUS, SPNEGO/Kerberos/NTLM, X.509 based	JAAS *(1)*, Grails *(1)*, NTLM *(1)*, Kerberos *(1)*	• X.509, JAAS, SPNEGO/Kerberos, RADIUS • SafeWord • ActivIdentity 4TRESS • Hitachi Finger Vein Biometric • Information Card Relying Party • Verisign Identity Protection • RSA Access Manager
Other	• Spring Security (bietet selbst CAS Unterstützung) • Atlassian Seraph	• Spring Acegi	• WS-Trust • Spring Security • Atlassian Seraph • JASIG CAS • CA SiteMinder

Tabelle 3: Tabelle: Vergleich der Integrationsmöglichkeiten

(1) externes Plugin
(2) begrenzter Support, nur getestet mit Google Apps

Insgesamt ist OpenSSO sicherlich die mächtigste Lösung im Starterfeld. Es bietet Funktionalitäten, die weit über normales Web-SSO hinaus gehen, umfassende Integrationsmöglichkeiten und direkten Support für die Einbeziehung von Web Services in den SSO-Kontext. Damit eignet sich OpenSSO am besten als Basis für umfangreiche Identity Management Lösungen mittlerer bis großer Ausdehnung in einer serviceorientierten Systemlandschaft. In solchen Szenarien wird man aber nicht umhin kommen, den Support von Sun in Anspruch zu nehmen oder aber selbst Know-How aufzubauen, um die komplexen Anforderungen wie höchste Ausfallsicherheit, Skalierbarkeit und Adaption an die Systemlandschaft bedienen zu können.

Dem gegenüber ist JA-SIG CAS anzusiedeln: Eine leichtgewichtige Architektur, großes Renommee vor allem im universitären Bereich und eine aktive Community machen aus CAS einen idealen Anwärter für webbasierte SSO Szenarien. CAS bietet genügend Potential und Anknüpfungspunkte, um durch Eigenentwicklungen an spezifischere, nicht nur webbasierte Anforderungen angepasst zu werden. Einzig die Dokumentation könnte homogener sein, wobei jedoch der gute Support aus der Community diesen Nachteil aufwiegt.

Den überschaubarsten SSO-bezogenen Funktionsumfang der drei Kandidaten bietet sicherlich Crowd. Atlassians Produkt besticht jedoch durch seine interessante Schnittmenge an Funktionen, die eine praxistaugliche Lösung für webbasiertes SSO bieten und zusätzlich wichtige Teilbereiche des Application Provisioning und Identity Managements abdecken. Gepaart mit seinen administrativen Möglichkeiten, seiner guten Erweiterbarkeit, der sehr guten Dokumentation und dem günstigen Support stellt Crowd damit eine sehr interessante Option für viele Anwendungsfälle dar.

REFERENZEN

- [1] SAML Specifications | SAML XML.org
  (http://saml.xml.org/saml-specifications#samlv20)
- [2] CAS | Jasig Community
  (http://www.ja-sig.org/products/cas/community/support/index.html)
- [3] Common Development and Distribution License (CDDL) Information
  (http://www.sun.com/cddl/)
- [4] OpenID
  (http://openid.net/)
- [5] The OpenID Directory
  (http://openiddirectory.com/)