Zusatzkapitel zum Buch:

de Boer, Essenpreis, García Laule, Raepple:

Single Sign-on mit SAP

Lösungen für die Praxis

ISBN 978-3-8362-1627-2

SAP PRESS 2011

»SAP, bitte authentifizieren Sie sich« – in diesem Kapitel dreht es sich um Single Sign-on mit einem AS ABAP-System als WS-Consumer, das sich per SAML-Assertion an Java- und .Net-WS-Providern anmeldet.

10 Single Sign-on mit AS ABAP als WS-Consumer

In Kapitel 8 des Buches »Single Sign-on mit SAP. Lösungen für die Praxis«, werden Optionen für Single Sign-on zwischen .Net- und Java-Anwendungen mit AS ABAP als WS-Provider erläutert. Dieses Kapitel baut auf Kapitel 8 auf und zeigt Beispiele für Single Sign-on zwischen AS ABAP als WS-Consumer und Apache Axis2, Sun Metro und Microsoft .Net in der Rolle des WS-Providers.

Die Anmeldung eines Benutzers am SAP-System wird dabei für die nachfolgende Anmeldung mit SAML am WS-Provider genutzt. Als Beispiel zeigen wir die Nutzung eines ABAP-Reports (siehe ❶ in Abbildung 10.1), der Bestände aus dem in Kapitel 8 behandelten Inventory-Service-WS-Provider ermittelt – der Inventory-Service-WS-Provider wird in den Workshops mit Java und .Net realisiert. Die Implementierungen für Metro, Axis und .Net sind im Buch-Download zu diesem Kapitel (*Workshops/10*) enthalten.

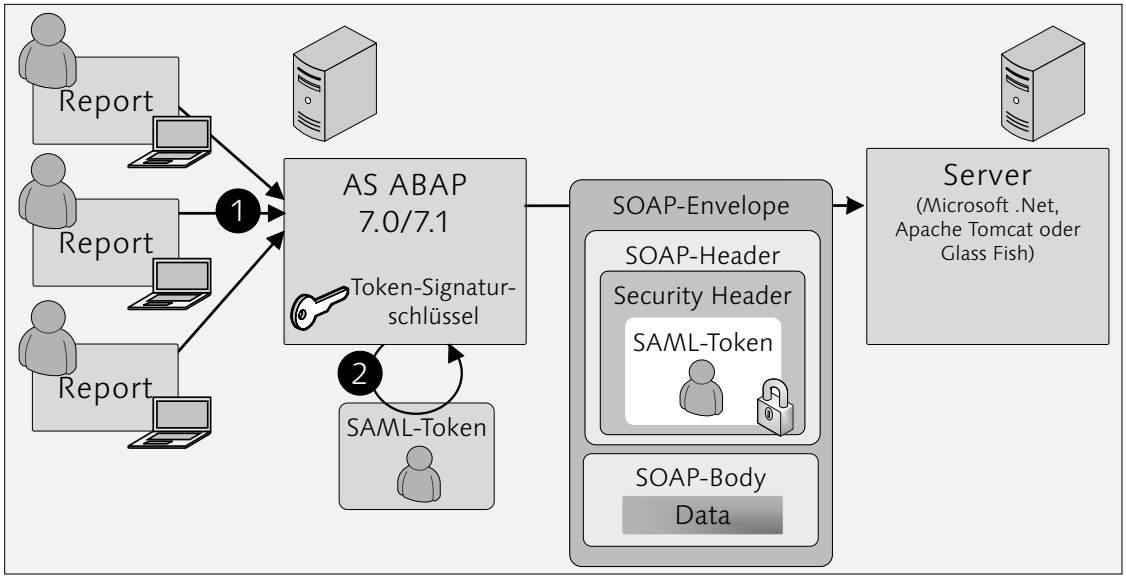

Abbildung 10.1 Inventory-Service-Szenario mit einem AS ABAP als WS-Consumer

In den Beispielen wird eine direkte Kommunikation zwischen dem WS-Consumer und dem AS ABAP verwendet ❷. Ähnliche Szenarien, in denen der WS-Adapter verwendet wird, sind auch mit SAP NetWeaver Process Integration 7.1 und 7.1 EhP1 möglich: Der ABAP-Report schickt Daten an den Integration Server, der sodann den Inventory Service aufruft und SAML als Authentifizierung verwendet.

Dieses Kapitel enthält Workshops für Single Sign-on zwischen dem AS ABAP und Java- sowie .Net-Providern:

- **Workshop: Single Sign-on mit Apache Axis2 (SAML-Sender-Vouches)**
  Single Sign-on mit Apache Axis2 beschreibt Single Sign-on zwischen AS ABAP 7.00 und Apache Axis2 auf Apache Tomcat (siehe Abschnitt 10.2).
- **Workshop: Single Sign-on mit Sun Metro 2.0 (SAML-Sender-Vouches)**
  Single Sign-on mit Sun Metro 2.0 zeigt eine Lösung für Single Sign-on zwischen AS ABAP 7.01 und Apache Tomcat (siehe Abschnitt 10.3).
- **Workshop: Single Sign-on zu .Net-Webservices**
  Bei Single Sign-on zu .Net-Webservices geht es um Single Sign-on zwischen AS ABAP 7.01 bzw. AS ABAP 7.11 mit vom AS ABAP als lokale STS ausgestellten SAML-Tokens (siehe Abschnitt 10.4).
- **Workshop: Single Sign-on zu .Net-Webservices mit Active Directory Federation Services 2.0**
  Hier wird das zuvor erwähnte Szenario erweitert, indem anstelle von lokalen Tokens ein SAML-Token von einem ADFS 2.0-STS verwendet wird (siehe Abschnitt 10.5).

Alle Workshops verwenden einen ABAP-Report zum Testen des Szenarios. Dieser wird im nun folgenden Abschnitt 10.1 zusammen mit der Konfiguration des AS ABAP-Systems vorgestellt.

10.1 WS-Consumer erstellen und konfigurieren

Zum Testen der Workshopszenarien dient ein ABAP-Report, der für alle Workshops identisch ist. In diesem Abschnitt geht es um den Testreport ZLIST_INVENTORY, den verwendeten Proxy, die Konfiguration des Systems und des WS-Consumers.

10.1.1 WS-Consumer-Proxy erzeugen

Erstellen Sie einen WS-Consumer-Proxy im AS ABAP, indem Sie in der Transaktion SE80 die folgenden Schritte durchführen:

1. Wählen Sie zunächst ein Paket; im Beispielprojekt werden die lokalen Objekte des Benutzers verwendet.
2. Im Kontextmenü des Pakets wählen Sie CREATE • ENTERPRISE SERVICE.
3. Im ersten Schritt des Wizards zum Anlegen eines Proxys verwenden Sie als OBJECT TYPE den SERVICE CONSUMER (siehe Abbildung 10.2). Wählen Sie im nächsten Schritt LOCAL FILE als WSDL-Quelle, und geben Sie anschließend den Pfad zur WSDL (*Workshops/10/InventoryService/Inventory Service.wsdl)* aus dem Download-Archiv zum Buch an.

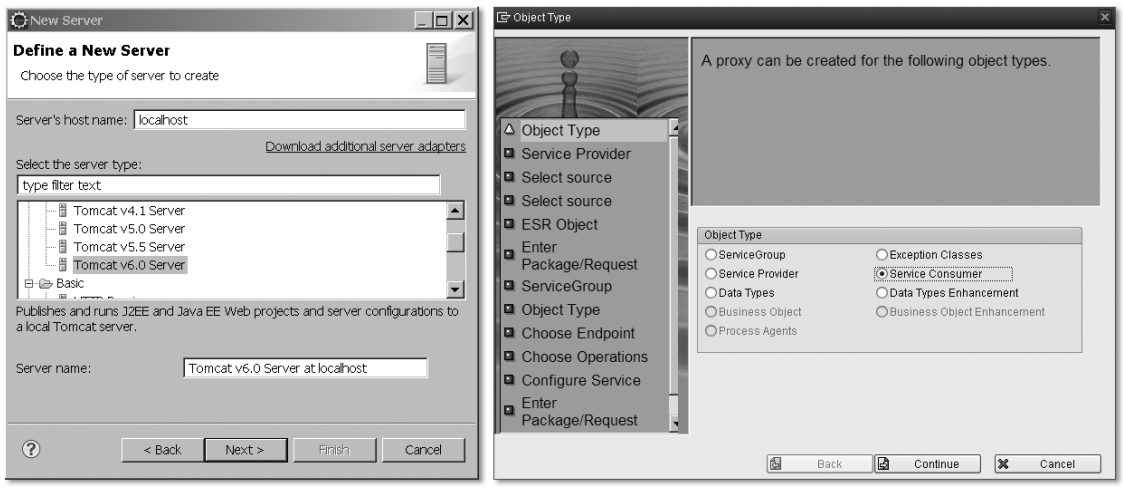

Abbildung 10.2 Anlegen eines WS-Consumer-Proxys in der Transaktion SE80 (links) und Angabe von Paket und Präfix im Schritt »Enter Package/Request«

4. Im Schritt Enter Package/Request wird die Angabe von Paket und Präfix benötigt. In unserem Beispielprojekt wird als Prefix »ZIS« und als Package »$TMP« eingetragen.
5. Beenden Sie nun den Wizard, und aktivieren Sie anschließend den Proxy.

10.1.2 Report

Zum einfacheren Verständnis wird auf komplexere Benutzeroberflächen wie Web Dynpro ABAP oder Web Dynpro verzichtet und ein ABAP-Report zum Aufrufen des Services verwendet.

Der Aufruf des WS-Consumers erfolgt unter der Angabe des logischen Ports über den Report ZLIST_INVENTORY, der sich in den Beispielprojekten unter *Workshops/10/ABAP* befindet (siehe Abbildung 10.3).

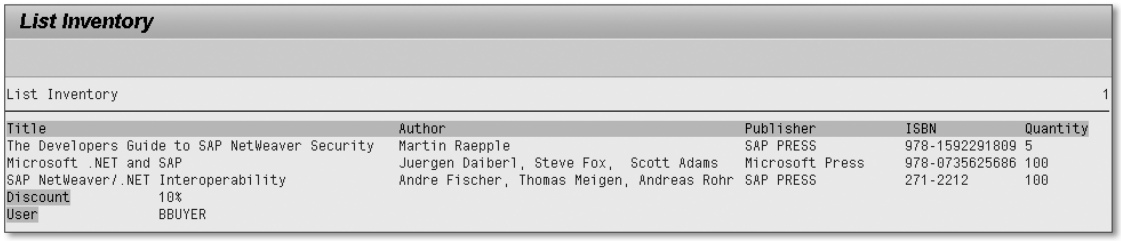

Abbildung 10.3 Aufrufen des ABAP-WS-Consumers über einen Report

10.1.3 RSA-Signaturschlüssel

Alle nachfolgenden Workshops benötigen einen RSA-Schlüssel für SAML-Signaturen. RSA-Signaturschlüssel werden in AS ABAP 7.00 SP21, 7.01 SP7, 7.02 SP4, 7.10 SP 9, 7.11 SP6 und 7.30 SP0 unterstützt.

In der Standardkonfiguration verwendet AS ABAP die System-PSE als Signaturschlüssel, um die SAML-Assertions zu signieren. Die System-PSE verwendet allerdings ein DSA-Schlüsselpaar, das von WS-Security nicht akzeptiert wird. WS-Security erfordert für alle Signaturen einen RSA-Signaturalgorithmus; daher ist die System-PSE für den Einsatz mit Metro nicht geeignet.

RSA-Schlüssel für AS ABAP 7.01 und 7.11

Um eine RSA-Signatur auszustellen, ist für die Releases AS ABAP 7.01 und 7.11 die SSF-Anwendung (SSF = Secure Store and Forward) S2SVP anzulegen. Wurde dieses Element nicht angelegt, so wird die System-PSE verwendet.

1. Melden Sie sich nun am SAP-WS-Consumer-System an, und starten Sie die Transaktion STRUST.
2. Wählen Sie im Menü Environment • SSF Parameters, woraufhin der Pflegebildschirm für die SSF-Parameter startet (siehe Abbildung 10.4).
3. Wählen Sie nun in der Symbolleiste New Entries ❶, und der Pflegebildschirm für die SSF-Applikationen startet.
4. Wählen Sie S2SVP als SSF Application, drücken Sie den Speichern-Button ❷, und geben Sie schließlich einen Transport-Request an, wenn Sie danach gefragt werden.
5. Navigieren Sie zurück zur STRUST-Hauptoberfläche, indem Sie zweimal auf den Zurück-Button klicken ❸.

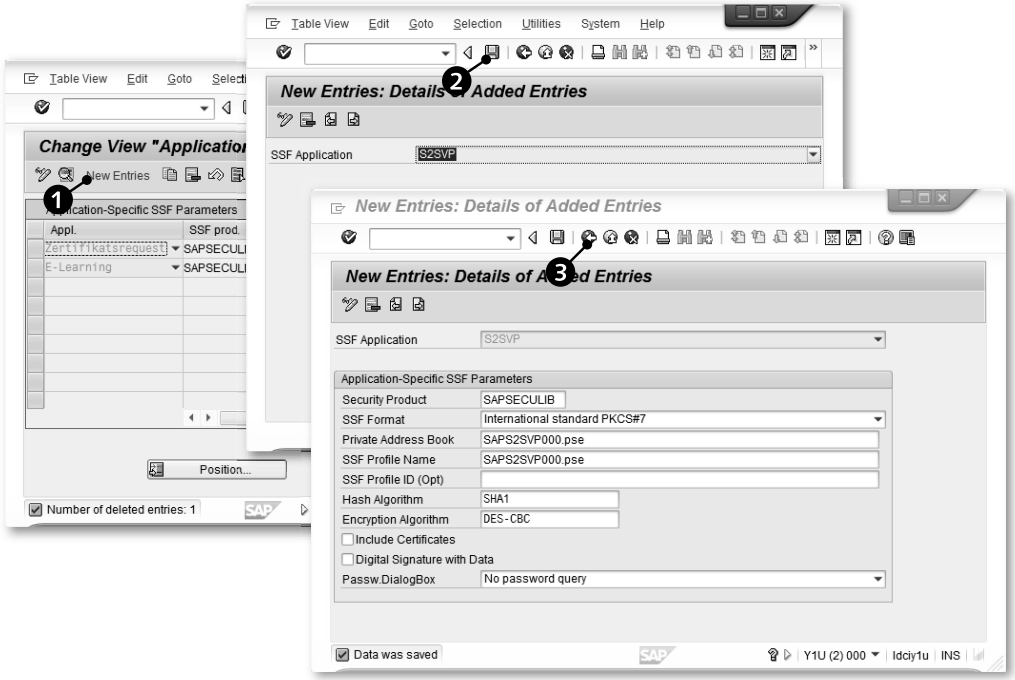

Abbildung 10.4 Anlegen der SSF-Applikation S2SVP

Sie sollten nun eine neue SSF-Applikation mit dem Namen »S2SVP« sehen (siehe Abbildung 10.5). Klicken Sie mit der rechten Maustaste darauf, und wählen Sie CREATE ❶.

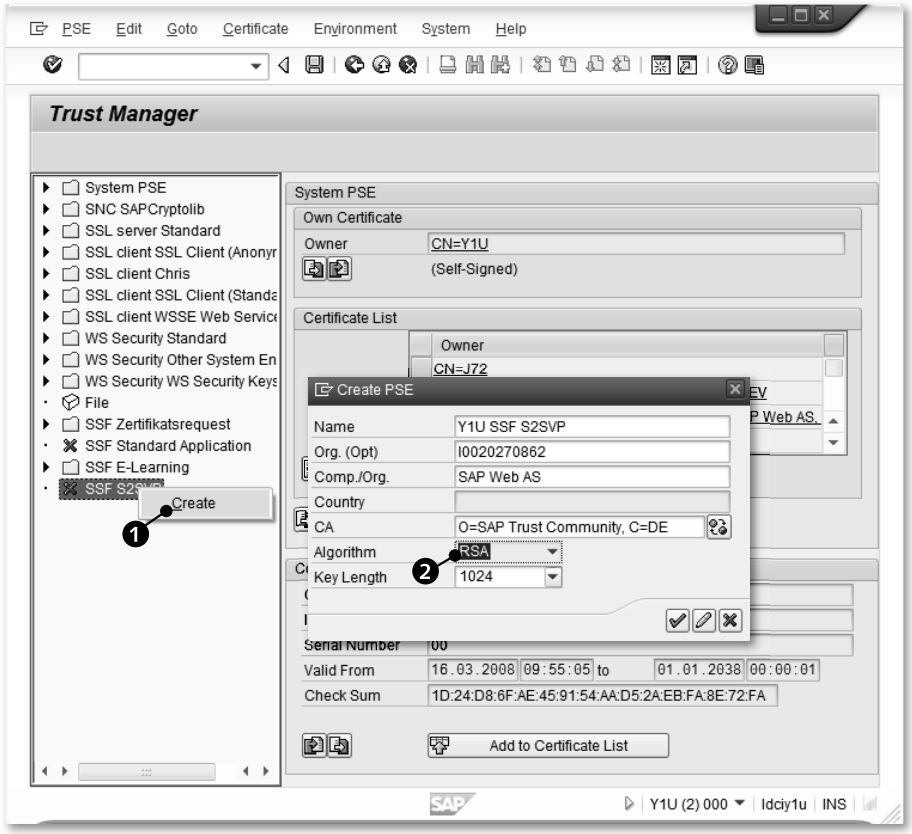

Abbildung 10.5 Anlegen der S2SVP-PSE zur Signatur der SAML-Assertion

Wählen Sie RSA als ALGORITHM, und bestätigen Sie das Pop-up ❷; die PSE ist jetzt angelegt.

RSA-Schlüssel für AS ABAP 7.02

Für die Releases AS ABAP 7.02 und 7.30 werden RSA-Schlüssel verwendet, wenn die Option SAML Trust im Report WSS_SETUP eingestellt wurde.

1. Melden Sie sich nun an Ihrem SAP-WS-Consumer-System an, und starten Sie den Report WSS_SETUP.
2. Wählen Sie anschließend die Option Use SAML Trust (siehe Abbildung 8.8 auf Seite 251 im Buch).
3. Deaktivieren Sie Test Run, und führen Sie den Report mithilfe der F8-Taste aus, um Ihre Einstellungen zu speichern.

10.1.4 Konfiguration des WS-Consumers mit dem SOA Manager

Im vorangegangenen Abschnitt wurde mit dem Anlegen des WS-Consumer-Proxys das Entwicklungsobjekt erzeugt. Im nächsten Schritt ist eine Laufzeitkonfiguration auf der Basis der WSDL anzulegen. Die Konfigurationsinformationen zu SAML sind in der WSDL enthalten und müssen bei Bedarf (z. B. für .Net und Sun Metro) mit dem Verschlüsselungszertifikat vervollständigt werden.

Zum Anlegen eines logischen Ports, also einer WS-Consumer-Konfiguration, führen Sie die folgenden Schritte durch:

1. Öffnen Sie zunächst die Transaktion Soamanager, und wählen Sie anschließend Single Service Administration • Single Service Configuration.
2. Zur Selektion des WS-Consumers wählen Sie Search By: Consumer Proxy und geben im Feld Search Pattern den Namen »InventoryService« an. Zur Vereinfachung kann dabei * als Platzhalter verwendet werden.
3. Nachdem der WS-Consumer ausgewählt worden ist, wählen Sie Configurations • Create Logical Port und geben den Namen, eine Beschreibung sowie die WSDL-URL an.

   Selektieren Sie den Reiter Consumer Security, um die Konfiguration zu vervollständigen. Wenn SSL für die Nachrichtensicherung verwendet wird, sind keine weiteren Konfigurationsangaben notwendig. Die Szenarien in den Abschnitten 10.3, »Workshop: Single Sign-on mit Sun Metro 2.0 (SAML-Sender-Vouches)«, und 10.4, »Workshop: Single Sign-on zu .Net-Webservices«, verwenden die XML-Verschlüsselung. Hierzu ist zusätzlich noch die Angabe eines Verschlüsselungszertifikats notwendig. Wählen Sie das Verschlüsselungszertifikat mit der Werthilfe aus.
4. Prüfen Sie nun die WS-Addressing-Einstellungen. Die WS-Addressing-Einstellungen werden teilweise nicht aus der WSDL übernommen. Nehmen Sie die in Tabelle 10.1 aufgeführten Einstellungen für Axis2, Metro und .Net vor, indem Sie im Reiter Messaging das Message-ID-Protokoll auswählen.

WS-Consumer	Message-ID-Protokoll-Einstellung
Sun Metro	WS-A Message ID 1.0
Apache Axis2	Suppress ID Transfer
.Net	WS-A Message ID

Tabelle 10.1 WS-Addressing-Einstellungen für .Net, Apache Axis2 und Sun Metro

10.1.5 Konfiguration des WS-Consumers mit dem SAP NetWeaver PI Integration Directory

SAML-Sender-Vouches wird in SAP NetWeaver PI 7.10 SP9 unterstützt; SAML-Holder-of-Key wird in SAP NetWeaver PI 7.11 SP5 unterstützt.

Die Konfiguration besteht aus dem Kommunikationskanal (Communcation Channel) und der Empfängervereinbarung (Receiver Agreement).

- **Legen Sie nun den Empfänger-Kommunikationskanal an.**
  Achten Sie darauf, dass die empfangende Kommunikationskomponente nicht als SAP-System deklariert wurde. Denn nur, wenn die Kommunikationskomponente als Fremdsystem deklariert wurde, erhalten Sie das Eingabefeld WSDL Access URL (siehe Abbildung 10.6).

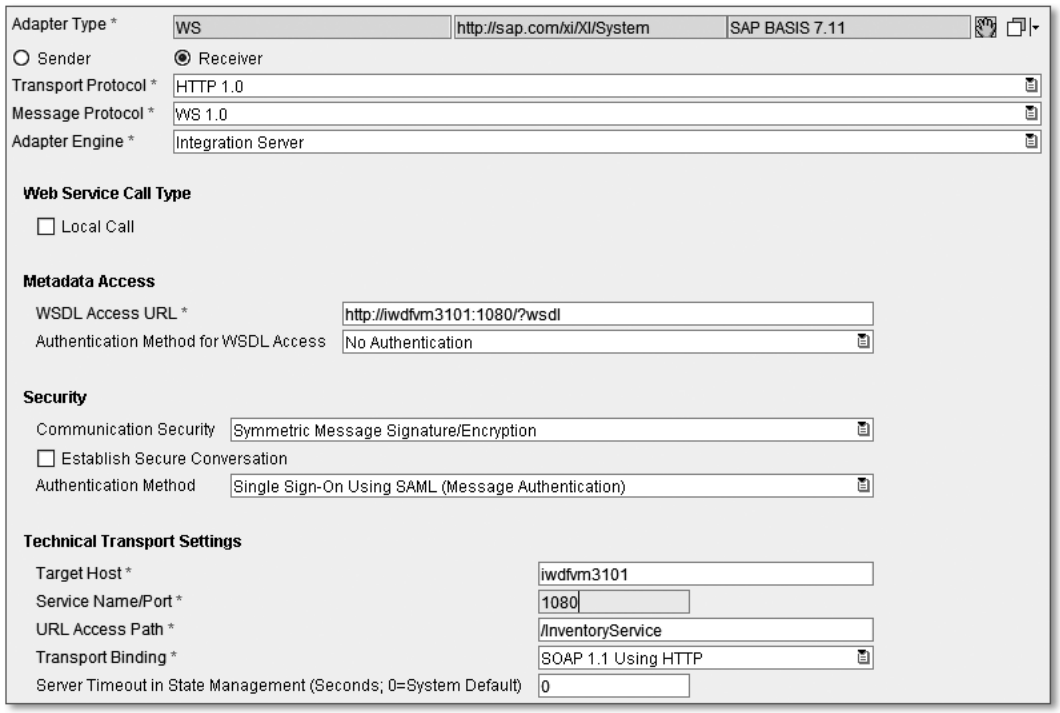

Abbildung 10.6 Konfiguration des Empfänger-Kommunikationskanals Receiver Agreement

  Wählen Sie den Adapter Type* »WS«, mit dem Namensraum »http://sap.com/xi/XI/System« der Version SAP BASIS 7.11.

- **Pflegen Sie nun die WSDL-URL des Webservices.**
  Wählen Sie hierzu »Single Sign-On Using SAML (Message Authentication)« als Authentication Method.

  Füllen Sie anschließend die Felder Target Host, Service Name/Port und URL Access Path aus der Aufruf-URL des Webservices (siehe Abbildung 10.6).

  Die Einstellungen für das Feld Communication Security sind für SAML-Sender-Vouches und den SAML-Holder-of-Key unterschiedlich. Entnehmen Sie die notwendigen Einstellungen der Tabelle 10.2.

Szenario	Communication Security	Message-ID-Protokoll-Einstellung
1.2 Workshop: Single Sign-on mit Apache Axis2 (SAML-Sender-Vouches)	HTTPS	Suppress ID Transfer
1.3 Workshop: Single Sign-on mit Sun Metro 2.0 (SAML-Sender-Vouches)	Asymmetric Message Signature/Encryption	WS-A Message ID 1.0
1.4 Workshop: Single Sign-on zu .Net-Webservices	Symmetric Message Signature/Encryption	WS-A Message ID
1.5 Workshop: Single Sign-on zu .Net-Webservices mit Active Directory Federation Services 2.0	(wird von SAP NetWeaver PI nicht unterstützt)	

Tabelle 10.2 Einstellungen für Communication Security und den ID-Message-ID-Transfer

- **Legen Sie nun die Empfängervereinbarung an.**
  Die Szenarien aus den Abschnitten 10.3, »Workshop: Single Sign-on mit Sun Metro 2.0 (SAML-Sender-Vouches)«, und 10.4 »Workshop: Single Sign-on zu .Net-Webservices«, verwenden die XML-Verschlüsselung, für die zusätzlich noch die Angabe eines Verschlüsselungszertifikats notwendig ist. Wählen Sie hierzu mit der Wertehilfe das Verschlüsselungszertifikat aus (siehe Abbildung 10.7).

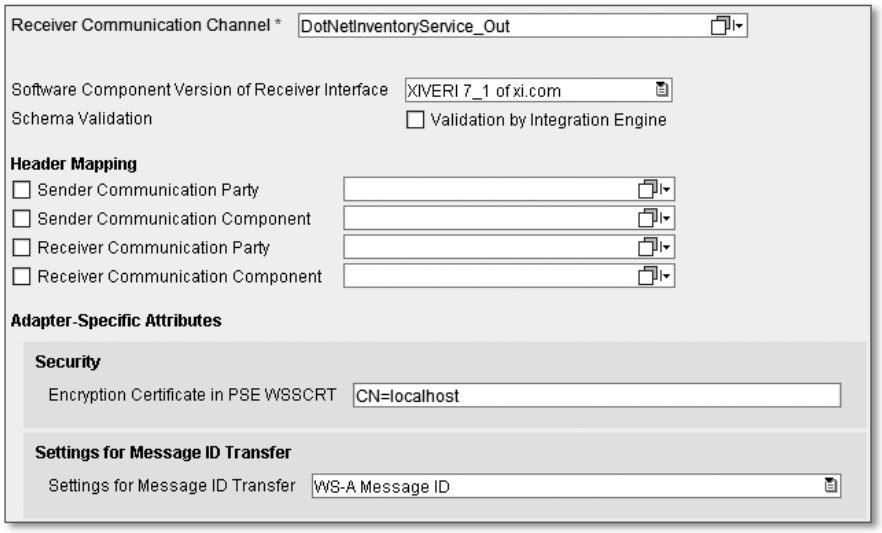

Abbildung 10.7 Konfiguration der Empfänger-Konfigurationsvereinbarung

10.2 Workshop: Single Sign-on mit Apache Axis2 (SAML-Sender-Vouches)

Aufbauend auf Abschnitt 8.3 im Buch »Single Sign-on mit SAP. Lösungen für die Praxis« geht es in diesem Workshop um Single Sign-on zwischen einem AS ABAP 7.00-WS-Consumer und Apache Axis2 mit dem WS-Security-Modul Apache Rampart. Als Authentifizierungsmethode wird SAML-Sender-Vouches verwendet.

In diesem Abschnitt wird zunächst der WS-Provider mit Eclipse angelegt und konfiguriert. Anschließend wird der WS-Consumer konfiguriert und schließlich das Szenario getestet.

10.2.1 Systemvoraussetzungen

Das Szenario wird mit folgenden AS ABAP Releases unterstützt: AS ABAP 7.00 SP20, 7.01 SP05, 7.02 SP05, 7.10 SP09, 7.11 SP03, 7.20 SP04 und 7.30 SP00.

Rampart 1.4 enthält die Apache WSS4J Library in der Version 1.5.4. Diese Library erwartet einen inkorrekten WS-Security-Header, wenn SAML konfiguriert ist. Dieses Problem wurde allerdings mit der WSS4J-Bibliothek ab der Version 1.5.8 behoben. Beziehen Sie die WSS4J-Bibliothek über *http://www.apache.org/dist/ws/wss4j/1_5_8/*, und ersetzen Sie die Bibliothek *rampart-1.4/lib/wss4j-1.5.4.jar* durch *wss4j-1.5.8.jar*.

Zusätzlich ist es erforderlich, die SAP Cryptographic Library in der Version 1.555.24 (oder neuer) installiert zu haben (siehe Abschnitt »SAP Cryptographic Library prüfen« auf Seite 242 im Buch).

10.2.2 Erstellen des Providers

Für den WS-Provider wird mit Eclipse anhand einer WSDL eine Java-Klasse erstellt. Informationen zu Eclipse und dessen Installation finden Sie in Abschnitt 8.3.3 auf Seite 257 im Buch.

Führen Sie die folgenden Schritte durch, um die Beispielanwendung zu erstellen:

1. Starten Sie Eclipse, und setzen Sie den Pfad für die Apache Axis2-Installation. Öffnen Sie nun die PREFERENCES-Seite, indem Sie WINDOW • PREFERENCES auswählen, und selektieren Sie WEB SERVICES • AXIS2 PREFERENCES. Tragen Sie schließlich den Pfad zur Axis2-Installation ein (z. B. *c:\opt\axis2-1.4.1*).
2. Starten Sie wiederum Eclipse, und erstellen Sie ein Dynamic Web Project, indem Sie in Eclipse FILE • NEW • OTHER auswählen oder die Tastenkombination Strg + N verwenden. Wählen Sie unter WEB • DYNAMIC WEB PROJECT.
3. Im Wizard zum Anlegen des Projektes geben Sie einen Projektnamen an und wählen anschließend einen Server. In unserem Beispiel wird Tomcat 6.0 verwendet.
4. Speichern Sie die WSDL aus *Workshops/10/InventoryService/InventoryService.wsdl* im Projekt ab.
5. Erzeugen Sie nun einen WS-Provider-Proxy, indem Sie in Eclipse FILE • NEW • OTHER auswählen oder die Tastenkombination Strg + N verwenden. Wählen Sie WEB SERVICE • WEB SERVICE aus (siehe Abbildung 10.8), geben Sie als WEB SERVICE TYPE den Wert »Top down Java beans Web Service« ein, und spezifizieren Sie die WSDL *InventoryService.wsdl* aus dem Projekt. Wählen Sie anschließend WEB SERVICE RUNTIME: APACHE AXIS2, und verwenden Sie schließlich die Schaltfläche FINISH, um den WS-Provider zu erzeugen.
6. Implementieren Sie nun die Klasse `InventoryServiceSkeleton`. Die Beispielimplementierung finden Sie im Buch-Download unter *Workshops/10/Axis2/InventoryServiceAxis.*
7. Eclipse fügt beim Erzeugen des Webservices die Axis2-Laufzeitbibliotheken dem Webprojekt hinzu. Zusätzlich werden noch Rampart, WSS4J- und Apache Xalan-Bibliotheken benötigt. Kopieren Sie die Rampart-Bibliotheken aus *rampart-1.4/lib* in die Webanwendung unter *WebContent/WEB-INF/lib.* Zusätzlich wird noch aus *axis-1.4.1/lib* die Bibliothek *xalan-2.7.0.jar* benötigt. Kopieren Sie diese ebenfalls nach *WebContent/WEB-INF/lib.*
8. Kopieren Sie nun die Rampart-Module aus *rampart-1.4/modules*, und fügen Sie sie in die Webanwendung unter *WebContent/WEB-INF/modules* ein.

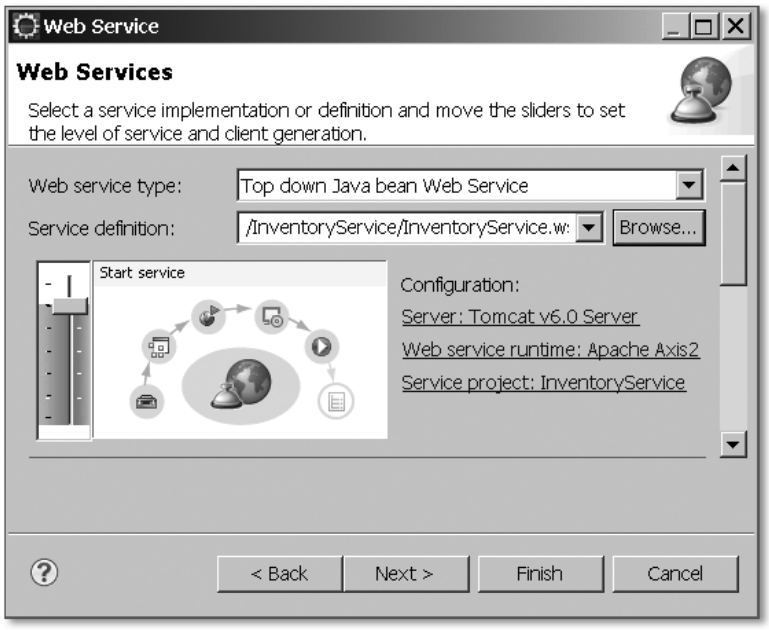

Abbildung 10.8 Erzeugen des WS-Providers

10.2.3 Konfiguration des Providers

Axis2 verwendet SSL für die Nachrichtenabsicherung zwischen dem WS-Consumer und dem WS-Provider. Die Konfiguration des Szenarios umfasst daher die folgenden Schritte, die in den nachfolgenden Abschnitten erläutert werden:

- Einen Java Truststore mit dem AS ABAP SAML-Signaturzertifikat anlegen und das Signaturzertifikat vom AS ABAP importieren.
- Den Inventory Service WS-Provider für die SAML-Authentifizierung konfigurieren.
- Die SAML-Assertion im Coding des Inventory Services validieren.

Truststore anlegen

Axis2 benötigt einen Keystore als Truststore mit den für die SAML-Signatur vertrauten Zertifikaten. Legen Sie hierzu einen Java Keystore (siehe Abschnitt 8.2.3 auf Seite 246 im Buch) an, und importieren Sie das Zertifikat *saml_cert.crt* (siehe Abschnitt 10.1.3, »RSA-Signaturschlüssel«). Führen Sie hierzu den nachfolgenden Befehl auf der Kommandozeile aus, und verwenden Sie als Alias den Wert `System ID/Mandant`, z. B. `UI2/960`.

```
keytool -import -file saml_cert.crt -alias <SID/Mandant> -keypass abcd1234 -store-
pass abcd1234 -keystore axis_trust.jks
```

Provider-Konfiguration anlegen

Nachdem der notwendige Java Keystore vorhanden ist, kann die Konfiguration des WS-Providers durchgeführt werden. Die Konfiguration erfolgt durch das Einfügen der Rampart-Anweisungen in die Konfigurationsdatei *services.xml* (siehe Listing 10.1). Diese befindet sich in der Datei *WEB-INF/services/<Service-Name>/META-INF/services.xml*. Axis2 (bzw. Apache Rampart) führt die folgenden Prüfungen durch:

- **Provider Request**
  Es wird über den Parameter `InflowSecurity` und die Actions `SAMLTokenUnsigned`, `Signature` und `Timestamp` sichergestellt, dass eine SAML-Assertion, ein Zeitstempel sowie eine XML-Signatur empfangen wurden. WSS4J verlangt eine strenge Einhaltung der Reihenfolge der SAML-Assertions. Der Eintrag für die Releases AS ABAP 7.00, 7.01, 7.10 und 7.11 lautet `Signature SAMLTokenUnsigned Timestamp`, während die Releases AS ABAP 7.02 und 7.30 `SAMLTokenUnsigned Signature Timestamp` verwenden.
- **Provider Response:**
  Es wird über den Parameter `OutflowSecurity` und die Action `Timestamp` der Zeitstempel in der Response geprüft.

```
<service name="InventoryService">
 <module ref="rampart" />
 <parameter name="OutflowSecurity">
  <action>
   <items>Timestamp</items>
   <enableSignatureConfirmation>false
   </enableSignatureConfirmation>
  </action>
 </parameter>
 <parameter name="InflowSecurity">
  <action>
   <!-- 7.00,7.01,7.10,7.11 <items>Signature SAMLTokenUnsigned Timestamp</items> -->
   <!-- 7.02/7.30 <items>SAMLTokenUnsigned Signature Timestamp</items> -->
   <items> Signature SAMLTokenUnsigned Timestamp
   </items>
   <signaturePropFile>cfg/crypto.properties
</signaturePropFile>
   <signatureParts>{Content}{http://schemas.xmlsoap.
org/soap/envelope/}Body;{Content}{http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-
1.0.xsd}Timestamp;{urn:oasis:names:tc:SAML:1.0:assertion}Assertion
   </signatureParts>
  <enableSignatureConfirmation>false
</enableSignatureConfirmation>
  </action>
 </parameter>
...
```

Listing 10.1 Konfiguration, um die SAML-Assertions mit Apache Rampart zu prüfen.

Validieren der SAML-Assertion

Rampart selbst sieht keine weiteren Prüfungen der SAML-Assertion – z. B. durch einen SAML-Token-Validator – vor. Die WS-Security-Verarbeitung führt eine einfache Prüfung durch und stellt die Ergebnisse dem WS-Provider zur weiteren Verarbeitung zur Verfügung. In unserem Beispielprojekt werden die Ergebnisse dazu genutzt, um die SAML-Assertion aus dem WS-Provider mit der Klasse `com.sap.sso.ws.axis.saml.validator.SAPSAMLTokenValidator` zu prüfen und den SAP-Benutzernamen zu ermitteln.

Listing 10.2 zeigt in einem Codebeispiel, wie eine SAML-Assertion validiert wird. Dabei wird ein Java Keystore als Basis für die Vertrauensbeziehung verwendet. Um eine SAML-Assertion richtig zu validieren, müssen die folgenden Voraussetzungen vorliegen:

1. Es wird im Keystore nach einem Signaturzertifikat mit dem SAML-Ausstellernamen (z. B. `U7A/000`) als Alias gesucht.
2. Das Signaturzertifikat wurde für die Nachrichtensignatur verwendet.

```
// SAML Token gegen crypto.properties validieren
Properties cryptoProperties = new Properties();
cryptoProperties.load(getClass().
getResourceAsStream("/cfg/crypto.properties"));
SAPSAMLTokenValidator validator =
    new SAPSAMLTokenValidator(cryptoProperties);
validator.validateTrust();
```

Listing 10.2 Prüfung der SAML-Assertion mit dem SAPSAMLTokenValidator

10.2.4 Konfiguration des WS-Consumers

Axis2 verwendet SSL für die Nachrichtenabsicherung zwischen dem WS-Consumer und dem WS-Provider. Die Konfiguration des Szenarios umfasst daher die folgenden Schritte, die in den nachfolgenden Abschnitten erläutert werden:

- SSL-Trust einrichten zwischen ABAP-WS-Consumer und Axis-WS-Provider. Dies erfolgt über den Import des CA-Zertifikats in die SSL-ANONYM PSE.
- Logischen Port in der Transaktion SOAMANAGER anlegen.

SSL-Trust konfigurieren

Wenn Sie für den Apache-Server selbst signierte Zertifikate verwenden, importieren Sie diese über die Transaktion STRUST in die SSL-client Anonymous-PSE. Im Falle, dass Sie CA-signierte Zertifikate verwenden, importieren Sie das CA-Zertifikat.

Öffnen Sie nun die Transaktion STRUST, und wählen Sie anschließend aus der Liste der SSL-Client PSEs durch Doppelklick client Anonymous aus. Importieren Sie das Zertifikat, indem Sie im Menü Certificate • Import wählen und die Datei mit dem Signaturzertifikat angeben. Anschließend wählen Sie Edit • Add Certificate und speichern Ihre Änderungen mit dem Sichern-Button.

Logischen Port in der Transaktion SOAMANAGER anlegen

Zur Konfiguration des WS-Consumers wird ein WSDL-Dokument mit WS-SecurityPolicy benötigt. Rampart erzeugt kein WSDL-Dokument mit WS-SecurityPolicy für SAML-Sender-Vouches; daher muss das WSDL-Dokument um WS- SecurityPolicy angereichert werden. Dies erfolgt über einen Servlet-Filter, der bei Angabe des zusätzlichen Parameters `policy=samlsv` die WSDL um die WS-SecurityPolicy anreichert. Hierzu muss die *web.xml*-Datei um einen Filter erweitert werden (siehe Listing 10.3):

```
<filter>
 <filter-name>WSDLFilter</filter-name>
  <filterclass>com.sap.sso.ws.axis2.wsdl.
WSDLRewriteFilter</filter-class>
</filter>
<filter-mapping>
 <filter-name>WSDLFilter</filter-name>
 <servlet-name>AxisServlet</servlet-name>
</filter-mapping>
```

Listing 10.3 Filter in der web.xml-Datei, um die WSDL mit WS- SecurityPolicy zu generieren

Die erforderlichen Dateien für den Servlet-Filter sowie eine *web.xml*-Beispielkonfiguration finden Sie unter *Workshops/10/Axis2/WSDLRewrite*.

Dem Beispielprojekt liegt ein Servlet bei, das die Axis2-WSDL um die WS-SecurityPolicy anreichert. Rufen Sie die angereicherte WSDL unter der folgenden URL auf: *http(s)://<host>:<port>/<WebApplication>/wsdl/<Servicename>?wsdl&policy=samlsv.*

Die URL für die WSDL zur Beispielapplikation lautet daher: *http:// <host>:<port>/InventoryService Axis/services/InventoryService?wsdl&policy=samlsv.*

Eine Liste der Services einer Webanwendung finden Sie unter *http:// <host>:<port>/<WebApplication>/axis2-web.*

Die Konfigurationsinformationen befinden sich in der angereicherten WSDL und werden für die WSDL-basierte Konfiguration verwendet.

Verwenden Sie das um WS-SecurityPolicy ergänzte WSDL-Dokument, um einen logischen Port AXIS2 für den WS-Consumer gemäß der Beschreibung in Abschnitt 10.1.4, »Konfiguration des WS-Consumers mit dem SOA Manager«, anzulegen. Damit ist die Konfiguration vollständig.

10.2.5 Testen des Szenarios

Verwenden Sie den Report ZLIST_INVENTORY aus dem Abschnitt 10.1.2, »Report«, um das Szenario zu testen. Geben Sie als Eingabe für den Parameter `LP` den Namen des logischen Ports an, den Sie bei der Konfiguration im SOA Manager angelegt haben.

10.2.6 Fehleranalyse und -behebung

Eine erfolgreiche SAML-Authentifizierung setzt die richtige Konfiguration von WS-Provider, WS-Consumer und Vertrauensbeziehung voraus. Ist diese Konfiguration fehlerhaft, so kommt es zu Laufzeitfehlern. Die häufigsten Fehler mit Apache Axis sind:

- **Must Understand check failed for header http://schemas.xmlsoap.org/ws/2004/08/addressing: To**
  Es ist die falsche Addressing-Version eingestellt, siehe Tabelle 10.1.
- **WSHandler: Certificate path verification failed for certificate with subject CN=…**
  Rampart traut dem AS ABAP-Signaturzertifikat nicht, siehe Abschnitt 10.2.3.
- **com.sap.sso.ws.axis.saml.validator.SAPSAMLValidationException: Certificate or issuer not trusted.(No alias <SAML Issuer> found in keystore /cfg/saml.jks,used signing certificate <certificate DN>)**
  Das Zertifikat wurde nicht korrekt in den Keystore importiert, siehe Abschnitt 10.2.3.

10.3 Workshop: Single Sign-on mit Sun Metro 2.0 (SAML-Sender-Vouches)

In diesem Workshop wird Single Sign-on zwischen einem AS ABAP 7.01-WS-Consumer und Metro 2.01 auf einem GlassFish v3-Applikationsserver umgesetzt. Dies ist das Gegenstück zu Abschnitt 8.4 im Buch, in dem es um Single Sign-on zwischen Sun Metro und dem AS ABAP geht. Das verwendete Authentifizierungsverfahren ist auch in diesem Fall SAML-Sender-Vouches.

In diesem Abschnitt wird zunächst der WS-Provider mit NetBeans angelegt und konfiguriert. Im Anschluss daran wird der WS-Consumer angelegt, konfiguriert und das Szenario getestet.

10.3.1 Systemvoraussetzungen

Die Metro-Implementierung von SAML-Sender-Vouches setzt WS-Addressing 1.0 und WS-Policy 1.5 voraus. Diese Funktionalitäten werden von AS ABAP 7.01 SP 8 bzw. AS ABAP 7.11 SP 6 unterstützt. Seitens Metro ist die Version 2.01 notwendig.

Zusätzlich ist es erforderlich, die SAP Cryptographic Library in der Version 1.555.29 installiert zu haben (siehe Abschnitt »SAP Cryptographic Library prüfen« auf Seite 242 im Buch).

10.3.2 Erstellen des Providers

Für den WS-Provider wird eine Java-Klasse erstellt, aus der über `EE5-@Webservice`-Annotationen ein Webservice erstellt wird. Den Source-Code dazu finden Sie in Listing 10.4.

Führen Sie die folgenden Schritte durch, um ein Webprojekt mit dem WS-Provider in NetBeans zu erstellen:

1. Starten Sie NetBeans, und erstellen Sie ein Web-Application-Projekt, indem Sie in NetBeans File • New Project auswählen. Selektieren Sie anschließend Java Web • Web Application, und betätigen Sie den Next-Button.
2. Im Abschnitt Name and Location geben Sie einen Projektnamen an und betätigen anschließend den Next-Button, um zum Abschnitt Server and Settings zu gelangen.
3. Wählen einen Server, auf dem Metro installiert ist. Für unser Beispiel wird GlassFish v3 verwendet.
4. Anschließend betätigen Sie den Next-Button, um zur Sektion Frameworks zu gelangen. Das Beispiel erfordert keine weiteren Frameworks.
5. Legen Sie nun eine neue Klasse in dem Projekt an. Wählen Sie hierzu File • New File, und selektieren Sie Web Services • Web Service. Geben Sie den Namen des Webservices sowie das entsprechende Paket an, und betätigen Sie wiederum den Next-Button.
6. Der WS-Provider ist nun angelegt und kann implementiert werden.
7. Die Implementierung des Services ist in Listing 10.4 skizziert. Die vollständige Implementierung findet sich im Buchdownload zu diesem Workshop. Für die Implementierung wird eine die Methode der Java Klasse `InventoryService` über die Java EE5-Annotationen `@WebService`, `@WebMethod` als WebService exponiert. Über die Annotationen `@WebResult` und `@WebParam` werden Namensraum und Namen der XML-Datenelemente festgelegt.

```
import javax.jws.*;
@WebService(targetNamespace = "http://com.sap.sso.ws.inventoryservice")
public class InventoryService {

    @WebMethod(operationName = "getInventory", action = "getInventory")
    @WebResult(name = "getInventoryResult", targetNamespace =
"http://com.sap.sso.ws.inventoryservice")
    public InventoryData getInventory(@WebParam(targetNamespace =
"http://com.sap.sso.ws.inventoryservice", name = "itemId") final String itemID)
...
```

Listing 10.4 Der Inventory Service verwendet Java-EE5-Annotationen

10.3.3 Konfiguration des Providers

Metro sichert den Nachrichtenaustausch zwischen dem WS-Consumer und dem WS-Provider durch die XML-Signatur und die XML-Verschlüsselung ab.

Dies setzt den Austausch von Schlüsseln für die XML-Verschlüsselung voraus. Die Konfiguration des Szenarios umfasst daher die folgenden Schritte, die in den nächsten Abschnitten erläutert werden:

- Schlüsselpaar erzeugen und im Java Keystore des Applikationsservers ablegen. Mit dem Schlüssel wird der SOAP- Request des WS-Consumers entschlüsselt und die Response von Metro signiert.
- Für die XML-Verschlüsselung muss das Metro-Zertifikat in den AS ABAP importiert werden.
- Das AS ABAP-SAML-Signaturzertifikat anlegen und als vertrauenswürdiges Zertifikat im Java Keystore des Applikationsservers ablegen
- WS-Provider konfigurieren

Metro-Schlüsselpaar erzeugen

Die Integration von Metro in den GlassFish Applikationsserver verwendet den Java Keystore *sges-v3/glassfish/domains/<domain>/config/keystore.jks* als Schlüsselablage. NetBeans erlaubt die Spezifikation eines anderen Keystores; dieser wird allerdings zur Laufzeit ignoriert und kann daher leicht zu Fehlkonfigurationen führen.

Der Keystore sowie alle Schlüsseleinträge sind mit dem GlassFish-Masterkennwort geschützt. In der Standardinstallation wird das Kennwort »changeit« verwendet, das über die ASADMIN-Konsole mit dem Befehl `change-master-password` änderbar ist.

Um im Keystore *keystore.jks* mit dem Kennwort »changeit« und dem Alias `metro_enc` ein Schlüsselpaar für die Verschlüsselung zu erzeugen, führen Sie auf der Kommandozeile den nachfolgenden Befehl aus; verwenden Sie als Kennwort das Masterkennwort, da es andernfalls zu Laufzeitfehlern kommt.

```
keytool -genkey -alias metro_enc -keyalg RSA -keysize 1024 -validity 1000 -keypass
changeit -storepass changeit -keystore keystore.jks
```

Um das zum Alias `metro_enc` abgelegte Zertifikat zu exportieren, wird die Option `-export` verwendet. Der folgende Befehl exportiert das erzeugte Zertifikat mit dem Alias `SAML` in die Datei *metro_cert.crt*.

```
keytool -export -file metro_cert.crt -alias metro -keypass changeit -storepass
changeit -keystore keystore.jks
```

Änderungen am Java Keystore erfordern einen Neustart des Servers. Weitere Informationen zur Schlüsselverwaltung mit Java finden Sie in Kapitel 8.2.3.

Metro-Schlüssel in den AS ABAP importieren

Für die XML-Verschlüsselung muss das Zertifikat *metro_crt.crt* in den ABAP-Schlüsselspeicher importiert werden. Dazu sind folgende Schritte notwendig:

1. Starten Sie nun die Transaktion STRUST.
2. Doppelklicken Sie anschließend auf die PSE WS SECURITY OTHER SYSTEM ENCRYPTION CERTIFICATES.
3. Wählen Sie im Menü CERTIFICATE • IMPORT. Geben Sie den Speicherort des Metro-Verschlüsselungszertifikats *metro_crt.crt* an, und bestätigen Sie anschließend den Dialog.

4. Wählen Sie ADD TO CERTIFICATE LIST, um das Zertifikat in die PSE aufzunehmen.
5. Speichern Sie schließlich die soeben veränderte PSE mit Strg + S.
6. Importieren Sie das Zertifikat ebenfalls in die PSE WS SECURITY WS SECURITY KEYS, denn dies ist für die Signaturprüfung der Response erforderlich.

SAML-Signaturzertifikat in GlassFish importieren

Die Metro-Integration in GlassFish verwendet den Java Keystore *sges-v3/glassfish/domains/<domain>/config/cacerts.jks* als Truststore mit den für die SAML-Signatur vertrauten Zertifikaten. Importieren Sie das Zertifikat *saml_crt.crt* in den Truststore.

Führen Sie dazu den folgenden Befehl auf der Kommandozeile aus; wie bereits im Abschnitt »Metro-Schlüsselpaar erzeugen« ausgeführt, ist dabei das Masterkennwort zu verwenden.

```
keytool -import -file saml_cert.crt -alias saml -keypass changeit -storepass changeit
-keystore cacerts.jks
```

Änderungen am Java Keystore erfordern einen Neustart des Servers.

Provider-Konfiguration anlegen

Nachdem die Zertifikate in die Java Keystores importiert worden sind, kann die Konfiguration des WS-Providers durchgeführt werden.

Mit dem Zertifikatimport des Signaturzertifikats in *cacerts.jks* wird eine Vertrauensbeziehung hergestellt. Weitere Prüfungen führt die Metro-Laufzeit nicht durch – sie überlässt dies einem Validator, dem WS-Provider. Um weiter gehende Prüfungen der Assertion durchzuführen, muss die Applikation, die den WS-Provider enthält, einen eigenen Validator spezifizieren. Der Validator wird von der Metro-Laufzeit gerufen, um applikationsspezifische Prüfungen zu implementieren. Technisch besteht der Validator aus der Implementierung des Interfaces `com.sun.xml.wss.impl.callback.SAMLValidator`.

In Java EE5 identifiziert ein Server einen erfolgreich angemeldeten Benutzer durch eine Instanz vom Typ `javax.security.auth.Subject`. Dabei kann es sich sowohl um eine Person als auch um ein System handeln. In jedem Fall ist ein *Subject* immer mit einer oder mehreren Identitäten in Form von sogenannten *Principals* assoziiert, die sich auf unterschiedliche Systeme, Technologien oder Anwendungen beziehen können (z. B. `WebPrincipal`, `SAMLPrincipal`).

Der SAMLValidator fügt bei erfolgreicher Prüfung einen *Principal* zum *Subject* hinzu. In unserem Beispielprojekt wird das `NameIdentifier`-Element aus der SAML-Assertion ausgelesen und als Principal eingefügt (siehe Listing 10.5). Innerhalb des WS-Providers werden anschließend Berechtigungsprüfungen über die JACC-Methoden (JACC = Java Authorization Contract for Containers) gegen diesen Principal durchgeführt.

```
public void validate(XMLStreamReader xmlStream, Map map, Subject subject) throws
SAMLValidationException {
Element domSamlAssertion = SAMLUtil.createSAMLAssertion(xmlStream);
SAMLUtil.validateTimeInConditionsStatement(domSamlAssertion);
//SAML:NameIdentifier element
NodeList nidList =
domSamlAssertion.getElementsByTagNameNS("urn:oasis:names:tc:SAML:1.0:assertion",
"NameIdentifier");
Node nid = null;
if (nidList.getLength() > 0)
    nid = nidList.item(0);
String child = nid.getFirstChild().getNodeValue();
```

```
//Principal setzen
if (child!=null) {
    Principal p = new SAPSAMLPrincipal(child);
    subject.getPrincipals().clear();
    subject.getPrincipals().add(p);
}
```

Listing 10.5 Validierung einer SAML-Assertion

Zur Konfiguration des Services wählen Sie im Kontextmenü des Webservices die Option Edit Webservice Attributes und nehmen anschließend die folgenden Einstellungen vor (siehe Abbildung 10.9): Security Mechanism: SAML Sender Vouches with Certificates

- Im Dialog Keystore:
  - Location: Pfad zu *sges-v3/glassfish/domains/<domain>/config/keystore.jks*
  - Keystore Password: Masterkennwort von GlassFish, im Beispiel »changeit«
  - Alias: Signaturschlüssel, im Beispiel *metro_enc*.
  - Key Password: Masterkennwort von GlassFish, im Beispiel »changeit«
  - Alias Selector Class: leer (wird in diesem Szenario nicht benötigt)
- Im Dialog Truststore:
  - Location: Pfad zu *sges-v3/glassfish/domains/<domain>/config/cacerts.jks*
  - Truststore Password: Masterkennwort von GlassFish, im Beispiel »changeit«
  - Certificate Selector Class: leer (wird in diesem Szenario nicht benötigt)
- Im Dialog Validators:
  - SAML Validator: Klasse, mit der die SAML-Assertion validiert wird; im Beispiel wird die Klasse `com.sap.sso.ws.metro.validator.SAPSAMLValidator` verwendet.

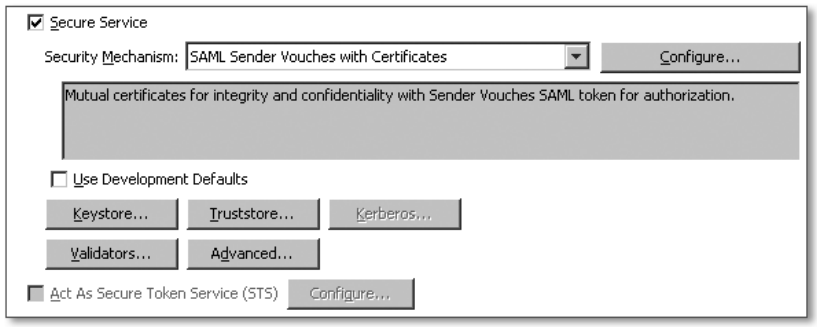

Abbildung 10.9 Provider-Konfiguration mit NetBeans für SAML-Sender-Vouches

10.3.4 Testen des Szenarios

Legen Sie mit der WSDL einen logischen Port an, so wie es in Abschnitt 10.1.4, »Konfiguration des WS-Consumers mit dem SOA Manager«, beschrieben wird.

Verwenden Sie anschließend Sie den Report ZLIST_INVENTORY aus Abschnitt 10.1.2, »Report«, um das Szenario zu testen. Geben Sie als Eingabe für den Parameter `LP` den Namen des logischen Ports an, den Sie bei der Konfiguration des logischen Ports im SOA Manager angelegt haben.

10.3.5 Fehleranalyse und -behebung

Eine erfolgreiche SAML-Authentifizierung setzt die richtige Konfiguration von WS-Provider, WS-Consumer und der Vertrauensbeziehung voraus. Ist diese Konfiguration fehlerhaft, so kommt es zu Laufzeitfehlern. Die häufigsten Fehler mit Sun Metro sind:

- **Es wird eine WS-Nachricht ohne XML-Signatur verschickt.**
  Metro 2.0 ist nicht im Klassenpfad eingebunden.
- **java.io.IOException: Key used to decrypt EncryptedKey cannot be null**
  AS ABAP verwendet ein X.509-Zertifikat für die Verschlüsselung, das Sun Metro unbekannt ist, siehe Abschnitt 10.3.3 und Abschnitt 10.1.4.

10.4 Workshop: Single Sign-on zu .Net-Webservices

Das Mittel zur Erstellung SOAP-basierter Webservices in der Microsoft-Welt heißt Windows Communication Framework (WCF). WCF-Webservices laufen traditionellerweise in einer Windows-Umgebung innerhalb einer Active-Directory-Domäne. Lange Zeit war Kerberos bzw. Windows Integrated Authentication das bevorzugte Authentifizierungsprotokoll. Um Single Sign-on zu erreichen musste ein WS-Consumer dieses Protokoll unterstützen. In der SAP-Welt behalf man sich z. B. mit dem SSO22KerbMap-Modul, das ein SAP Logon Ticket entgegennahm und dieses in ein Kerberos-Ticket umwandelte.

10.4.1 Einführung in das Szenario

Durch die Unterstützung des SAML-Token-Profile-Standards ist es ABAP-WS-Consumern möglich, sich an einem WCF-Service mit einer SAML-Assertion zu authentifizieren (siehe Abbildung 10.8).

1. Der Benutzer meldet sich am SAP-System an und startet eine Applikation, die einen Webservice aufruft.
2. Der AS ABAP-WS-Consumer authentifiziert sich am .Net-WS-Provider mit einer vom AS ABAP selbst ausgestellten SAML-Holder-of-key-Assertion. Im `NameIdentifier`-Element der Assertion wird der SAP-Anmeldename des Benutzers, der den Webservice-Aufruf durchführt, übertragen.
3. Anhand des SAP-Benutzernamens in der SAML-Assertion erfragt der .Net-WS-Provider die dazugehörigen Domänenbenutzerinformationen beim Active Directory. Die Abfrage erfolgt in unserem Beispielszenario über das Active-Directory-Schema-Attribut `sapUsername`, das für jeden Domänenbenutzer mit SAP-System-Zugriff gepflegt ist. Anhand der Benutzerinformationen aus dem Active Directory erfolgen die Authentifizierung und die Berechtigungsprüfung.

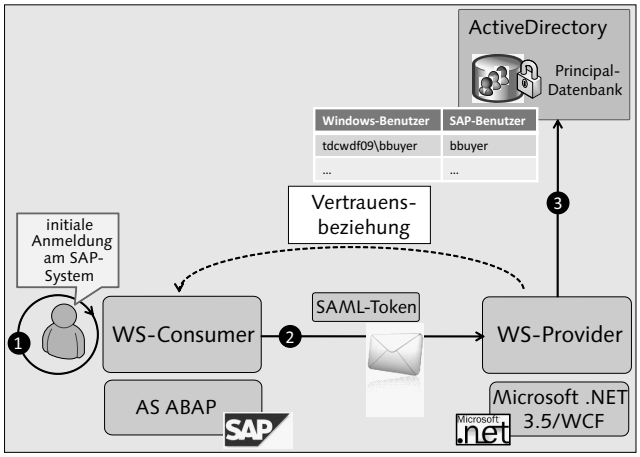

Abbildung 10.10 Single Sign-on von einem ABAP-WS-Consumer zu einem .Net-Webservice

Beispielanwendungen

Zu diesem Kapitel gibt es einen Beispiel-.Net-WS-Provider mit den dazugehörigen Visual Studio 2008-Projekten. Diesen Provider finden Sie im Buch-Download unter dem Verzeichnis *Workshops/10/DotNet*.

10.4.2 Systemvoraussetzungen

Beachten Sie folgende Systemvoraussetzungen:

Voraussetzungen im SAP-System

Dieses Szenario läuft auf einem SAP-System ab dem Basisrelease 7.01 mit dem Service Pack 8 bzw. 7.11 mit dem Service Pack 6. Zusätzlich ist die SAP Cryptographic Library in der Version 1.555.29 zu installieren. Die Prüfung der Version der SAP Cryptographic Library ist im Abschnitt »SAP Cryptographic Library prüfen« auf Seite 242 im Buch beschrieben.

Im SAP-System muss der im Active Directory eingetragene SAP-Benutzer vorhanden sein.

Voraussetzungen im Active Directory

Zur Identifikation eines Domänenbenutzers anhand eines SAP-Benutzernamens muss zu jedem Domänenbenutzer das Active-Directory-Attribut `sapUsername` gepflegt worden sein. Dieses Attribut enthält den maximal achtstelligen Benutzernamen für SAP-Systeme.

Erweiterung des Active-Directory-Schemas

Damit das `sapUsername`-Attribut gepflegt werden kann, muss im Active Directory eine Schemaerweiterung vorgenommen werden. In Abschnitt 6.2.3, »Konfigurationsschritte«, erfahren Sie, wie dies vonstattengeht.

Sollten Sie vor einer Schemaerweiterung zurückschrecken, so können Sie alternativ auch ein Standardattribut verwenden. Das Attribut `altSecurityIdentities` bietet sich hier an.

Für den Test des Szenarios wurde ein Windows Server 2008-Betriebssystem verwendet. Ein auf dem Windows Server 2003 basierendes Active Directory wird laut Microsoft ebenfalls unterstützt.

Voraussetzungen im .Net-WS-Provider-System

Der .Net-Webservice, den Sie im Download zu diesem Kapitel finden, nutzt die folgenden Microsoft-APIs, die auf Ihrem WS-Provider-Server installiert sein müssen:

- Microsoft .NET Framework 3.5 Service Pack 1
- Windows Identity Foundation (Grundlage von ADFS 2.0)
- Windows Identity Foundation SDK

Alle APIs können als Installationspakete direkt über *http://www.microsoft.com/downloads/* heruntergeladen werden.

Der Source-Code zum .Net-Webservice

Das Visual Studio 2008-Projekt für den Webservice liegt dem Buchdownload bei. Öffnen Sie die Solution mit dem Dateinamen *InventoryService.sln* aus dem Verzeichnis *Workshops/10/DotNet/InventoryService DotNet*.

10.4.3 Konfiguration des .Net-WS-Providers

Kopieren Sie das Verzeichnis *DotNet/InventoryServiceDotNet* aus dem Buch-Download für dieses Kapitel auf das vorgesehene WS-Provider-System.

Schlüsselpaar für die Verschlüsselung generieren

Die SAML-Assertion, die das SAP-System ausstellt, ist mit dem öffentlichen Schlüssel des .Net-WS-Providers verschlüsselt. Der dazugehörige private Schlüssel liegt auf dem WS-Provider-Server, damit nur dieser die SAML-Assertion entschlüsseln kann.

Sie können das Werkzeug makecert aus dem Microsoft .NET-Framework verwenden, um das Schlüsselpaar zu erzeugen:

1. Melden Sie sich mit Ihren Administrationsberechtigungen am .Net-WS-Provider-Server an.
2. Wählen Sie Startmenü • Run, und geben Sie »cmd« ein. Sodann erscheint der Command Prompt.
3. Wechseln Sie nun mit dem Befehl `cd C:\Program Files\Microsoft SDKs\Windows\v6.0A\bin` in das .Net-SDK-Verzeichnis.
4. Geben Sie makecert -# 42 -r -pe -n "CN=localhost" -e 01/01/2015 -a sha1 -sky exchange -ss my -sr localmachine c:\temp\localhost.crt ein.

Jetzt wird ein Schlüsselpaar mit dem Namen »CN=localhost« mit Gültigkeit bis zum 01.01.2015 erzeugt und im Bereich Local Computer *des Windows-Zertifikatspeichers* abgelegt. Außerdem wurde das Verschlüsselungszertifikat in die Datei *c:\temp\localhost.crt* exportiert; diese Datei müssen Sie später in das SAP-System importieren.

STS-URLs in der WS-Provider-WSDL auskommentieren

Die WSDL des .Net-WS-Providers enthält normalerweise URL-Referenzen zum STS, den der WS-Consumer verwenden soll, um eine SAML-Assertion anzufordern. Da in diesem Szenario kein zentraler STS zum Einsatz kommt, müssen die-URL-Referenzen in der .Net-WS-Provider-Konfigurationsdatei auskommentiert werden.

1. Öffnen Sie die Datei *DotNet/InventoryServiceDotNet/InventoryServiceHost/bin/Debug/InventoryServiceConsoleApp.exe.config* aus dem Buch-Download für dieses Kapitel.
2. Kommentieren Sie anschließend die URLs der XML-Elemente issuer und issuerMetadata. Abbildung 10.11 zeigt Ihnen die fertige Konfigurationsdatei.
3. Speichern Sie schließlich die Datei.

```
<bindings>
    <customBinding>
        <binding name="IST">
            <textMessageEncoding messageVersion="Soap11WSAddressing10" />
            <security authenticationMode="IssuedToken" messageSecurityVersion="WSSecurity11WSTrust13WSSecureConversat
                <issuedTokenParameters>
                  <!--issuer address="https://tdc09dc1/adfs/services/trust/13/issuedtokenmixedasymmetricbasic256" />
                    <issuerMetadata address="https://tdc09dc1/adfs/services/trust/mex" /-->
                </issuedTokenParameters>
                <secureConversationBootstrap authenticationMode="IssuedTokenOverTransport" />
            </security>
            <httpTransport />
        </binding>
    </customBinding>
</bindings>
```

Abbildung 10.11 Auskommentierte STS-URLs

10.4.4 Vertrauensbeziehung des .Net-WS-Providers zum SAP-System einrichten

Für die Vertrauensbeziehung zwischen dem .Net-WS-Provider und dem AS ABAP muss das SAP-System für das Signieren mit RSA-Schlüsseln konfiguriert werden (siehe Abschnitt 10.1.3, »RSA-Signaturschlüssel«). Außerdem muss das Signaturzertifikat für die SAML-Assertions in den Trusted-People-Bereich des Windows-Zertifikatspeichers im .Net-WS-Provider-System importiert werden.

Export des SAML-Signaturzertifikats aus dem SAP-System

1. Melden Sie sich zunächst am SAP-WS-Consumer-System an, und starten Sie die Transaktion STRUST (siehe Abbildung 10.12).
2. Doppelklicken Sie auf die PSE mit dem Namen »SSF S2SVP« ❶.
3. Doppelklicken Sie auf das OWNER-Zertifikat ❷.
4. Klicken Sie schließlich auf den Button EXPORT CERTIFICATE ❸, UND SPEICHERN SIE DAS ZERTIFIKAT IN EINER DATEI AB.

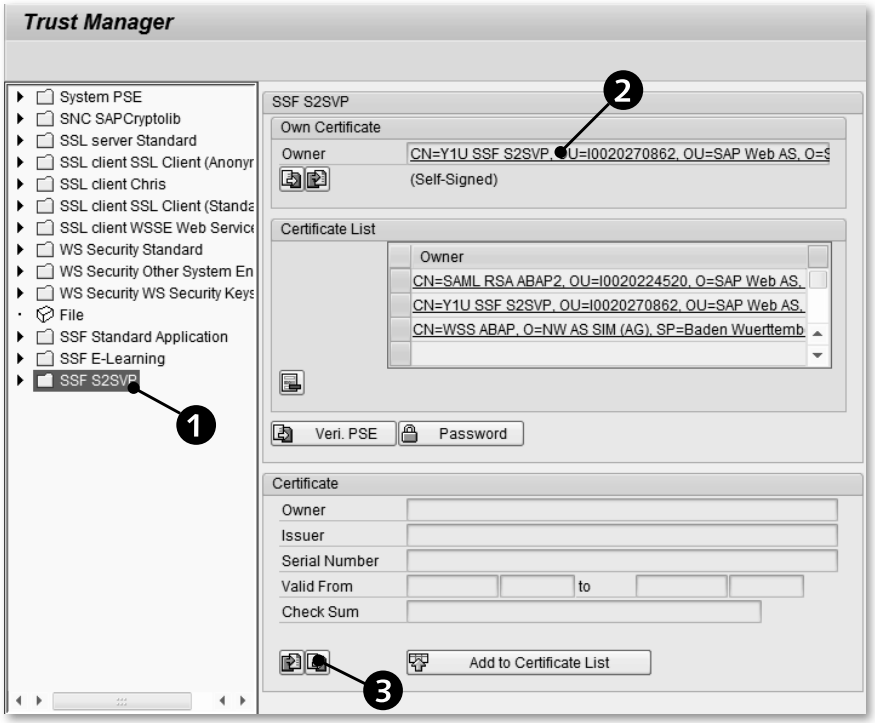

Abbildung 10.12 Export des Signaturzertifikats für die SAML-Assertion

Import des SAML-Signaturzertifikats in das .Net-WS-Provider-System

1. Wechseln Sie zunächst zurück in den .Net-WS-Provider-Server.
2. Starten Sie im Startmenü unter RUN die Microsoft Management Console MIT MMC.EXE, und laden Sie das Certificate Snap-in über das Menü FILE • ADD/REMOVE SNAP-IN …, das in Abbildung 10.13 dargestellt ist.
3. Wählen Sie anschließend den Eintrag CERTIFICATES, und klicken Sie auf den Button ADD. Schließlich wählen Sie COMPUTER ACCOUNT und beenden den Wizard mit FINISH.

Das Pop-up-Fenster zur Auswahl des Computer-Accounts erscheint nicht

Wenn das Pop-up-Fenster zur Auswahl des Computer-Accounts nicht erscheint, liegt es daran, dass der Benutzer, mit dem Sie sich angemeldet haben, nicht über die erforderlichen Berechtigungen verfügt. Verwenden Sie einen Benutzer mit Administrationsberechtigungen.

4. Beenden Sie den Dialog Add or Remove Snap-ins mit Close, und bestätigen Sie das Hinzufügen des Snap-ins Certificates (Local Computer) im Add or Remove Snap-ins-Fenster über den OK-Button.
5. Klappen Sie nun den Baum unter Console Root auf, und klicken Sie mit der rechten Maustaste auf Certificates (Local Computer) • Trusted People • Certificates (siehe Abbildung 10.13).
6. Wählen Sie anschließend All Tasks • Import, um das SAML-Signatur-Zertifikat zu importieren (siehe Abbildung 10.14).
7. Die Konfiguration der .Net-Provider-Seite ist jetzt fertig. Starten Sie nun den Webservice, so wie es im nächsten Schritt beschrieben ist.

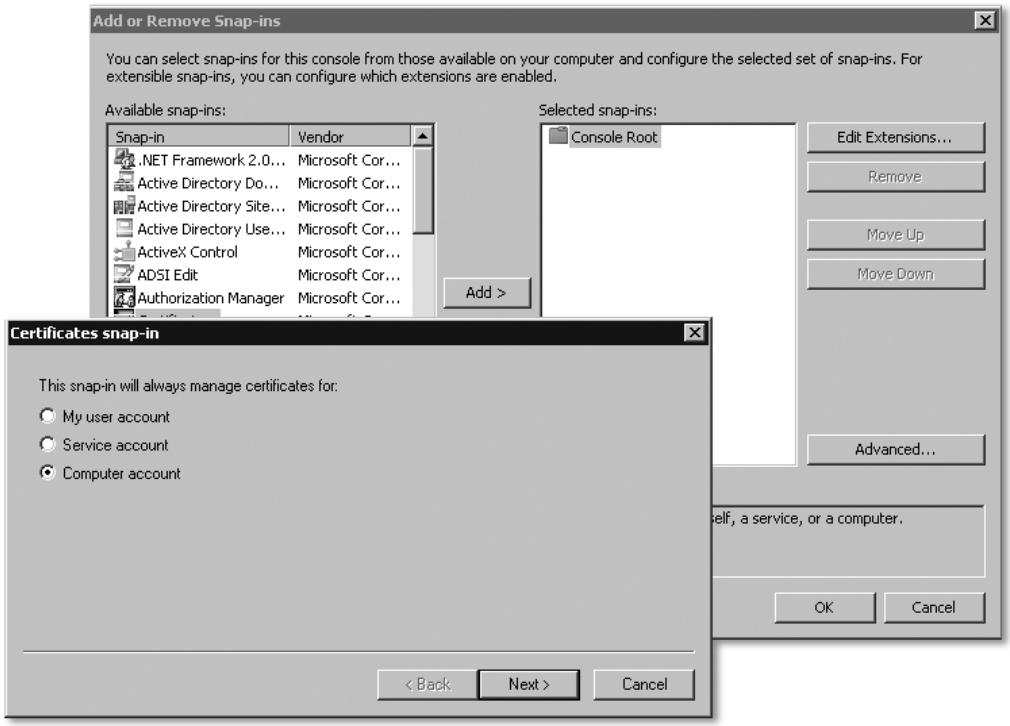

Abbildung 10.13 Laden des Certificate Snap-ins in der Microsoft Management Console (MMC)

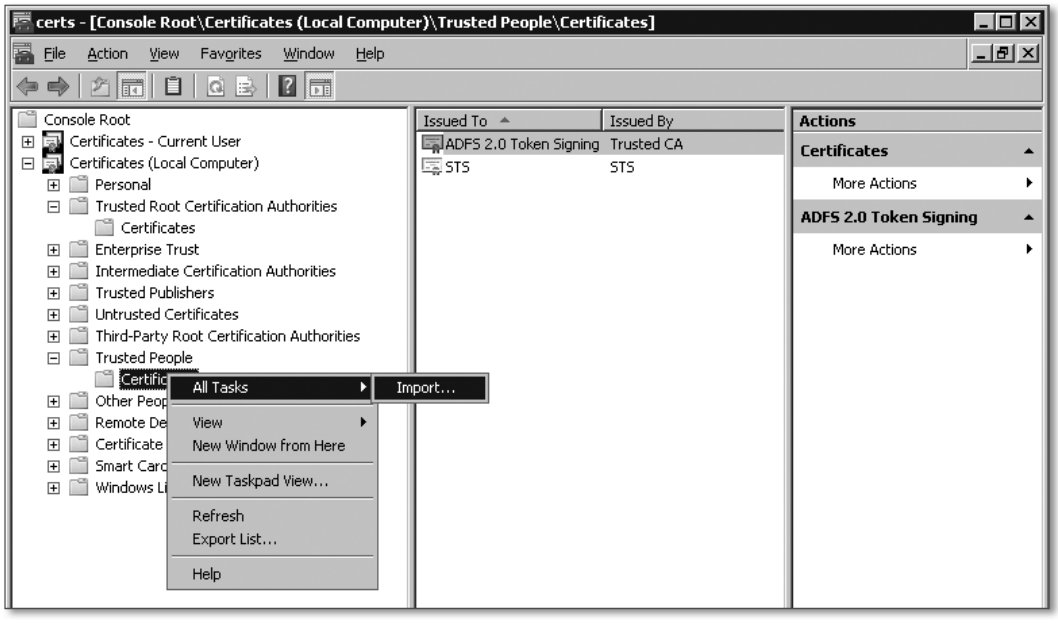

Abbildung 10.14 Import des Signaturzertifikats vom STS.

Webservice starten

Nachdem Sie alle Konfigurationsschritte durchgeführt haben, können Sie nun den Webservice starten.

1. Starten Sie die Anwendung `InventoryServiceConsoleApp.exe` im Verzeichnis *DotNet\InventoryServiceDotNet\InventoryServiceHost\bin\Debug* aus dem Buch-Download zu diesem Kapitel.

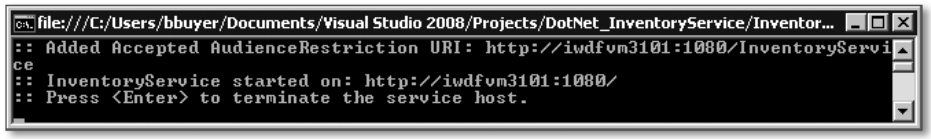

Abbildung 10.15 Erfolgreicher Start des Inventory-.Net-Webservices

2. Nun können Sie sich das Provider-WSDL-Dokument in Ihrem Browser anzeigen lassen, z. B. unter *http://iwdfvm3101:1080/?wsdl*. Die URL wird direkt in der Konsolenanwendung angezeigt (siehe Abbildung 10.15).

10.4.5 Konfiguration des SAP-WS-Consumers

Die Erstellung der Konfiguration umfasst den Import des Verschlüsselungszertifikats des .Net-WS-Providers in den AS ABAP und das Erstellen eines logischen Ports im SOA Manager.

Verschlüsselungszertifikats des .Net-WS-Providers importieren

1. Starten Sie nun die Transaktion STRUST.
2. Doppelklicken Sie anschließend auf die PSE WS Security Other System Encryption Certificates.
3. Wählen Sie im Menü Certificate • Import. Geben Sie den Speicherort des .Net-Verschlüsselungszertifikats an, und bestätigen Sie anschließend den Dialog.
4. Wählen Sie Add to Certificate List, um das Zertifikat in die PSE aufzunehmen.
5. Speichern Sie schließlich die soeben veränderte PSE mit Strg + S.

Logischen Port anlegen

Legen Sie mit der WSDL einen logischen Port, so wie es in Abschnitt 10.1.4, »Konfiguration des WS-Consumers mit dem SOA Manager«, beschrieben wird, an. Geben Sie dabei das soeben importierte Verschlüsselungszertifikat als Encryption Certificate an, und speichern Sie schließlich den logischen Port (siehe Abbildung 10.16).

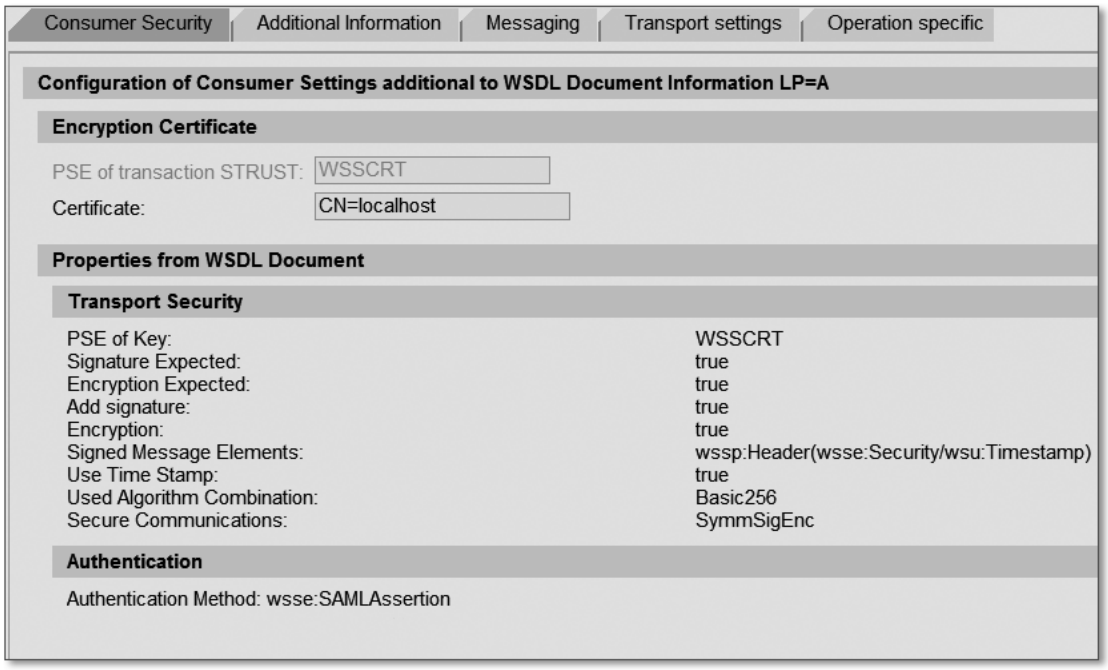

Abbildung 10.16 Der fertig konfigurierte logische Port

10.4.6 Testen des Szenarios

Melden Sie sich am SAP-System an und starten Sie mit der Transaktion SA38 den Report ZLIST_INVENTORY, wie in Abschnitt 10.1.2, »Report« beschrieben. Geben Sie als Eingabe für den Parameter `LP` den Namen des logischen Ports an, den Sie bei der Konfiguration im SOA Manager angelegt haben.

10.4.7 Fehleranalyse und -behebung

Eine Übersicht der verschiedenen Fehlersituationen in beteiligten Systemen finden Sie im nächsten Workshop (Abschnitt 10.5.9).

10.5 Workshop: Single Sign-on zu .Net-Webservices mit Active Directory Federation Services 2.0

Dieses Szenario ist der zweite Workshop zu Single Sign-on von AS ABAP zu .Net. Im ersten Szenario meldet sich der SAP-WS-Consumer direkt am .Net-WS-Provider an, und die Authentifizierungs- und Berechtigungsprüfung wird direkt vom .Net-WS-Provider durchgeführt. In diesem Szenario wird ein Microsoft ADFS 2.0 Security Token Service (ADFS-STS) verwendet, um das SAML-Token auszustellen. Die Verwendung eines STS hat mehrere Vorteile:

- **Zentrale Vertrauensverwaltung**
  Stellen Sie sich mehrere Provider und Consumer vor, die sich alle gegenseitig aufrufen. Ohne STS müsste jedes Provider-System jedem Consumer-System vertrauen. In diesem Szenario gibt es pro Provider lediglich eine Vertrauensbeziehung zum STS.
- **Zentrale Benutzerabbildung**
  Mit einem STS verringert sich die Anzahl der Benutzerabbildungen auf eine Benutzerabbildung pro Provider-System und pro angeschlossenem Benutzerspeicher. Beim Ausstellen der SAML-Assertion ermittelt der STS den jeweiligen SAP-Benutzer in den einzelnen Provider-Systemen zentral; Somit entfällt die Pflege der Benutzerabbildungen in den einzelnen Provider-Systemen.

10.5.1 Einführung in das Szenario

Abbildung 10.17 zeigt folgendes Szenario:

1. Der Benutzer meldet sich am SAP-System an und startet eine Applikation, die einen Webservice aufruft.
2. Der AS ABAP-WS-Consumer fordert beim ADFS-STS eine SAML-Assertion an, die er zur Authentifizierung am .Net-Webservice verwenden kann. Er authentifiziert sich am STS mit einer selbst ausgestellten SAML-Holder-of-Key-Assertion. Im `NameIdentifier` der Assertion wird der SAP-Benutzername des Benutzers, der den Webservice aufruft, übertragen.
3. Anhand des SAP-Benutzers in der SAML-Assertion ermittelt der ADFS-STS den dazugehörigen Domänenbenutzer im Active Directory. Die Abfrage erfolgt über das Active-Directory-Schema-Attribut `sapUsername`, das für jeden Domänenbenutzer mit SAP-Systemzugriff gepflegt ist. Schließlich stellt der STS eine SAML-Assertion aus, die den jeweiligen Windows-Benutzer im `NameIdentifier` beinhaltet. Des Weiteren enthält die SAML-Assertion auch die Information darüber, welchen Gruppen der Benutzer zugeordnet ist.
4. Der SAP-WS-Consumer ruft den .Net-Webservice auf und schickt die erhaltene SAML-Assertion mit.
5. Der .Net-WS-Provider prüft, ob der Aussteller der SAML-Assertion vertrauenswürdig ist; anhand der Domänen-Benutzerinformationen in der SAML-Assertion können Authentifizierung und Berechtigungsprüfung durchgeführt werden.

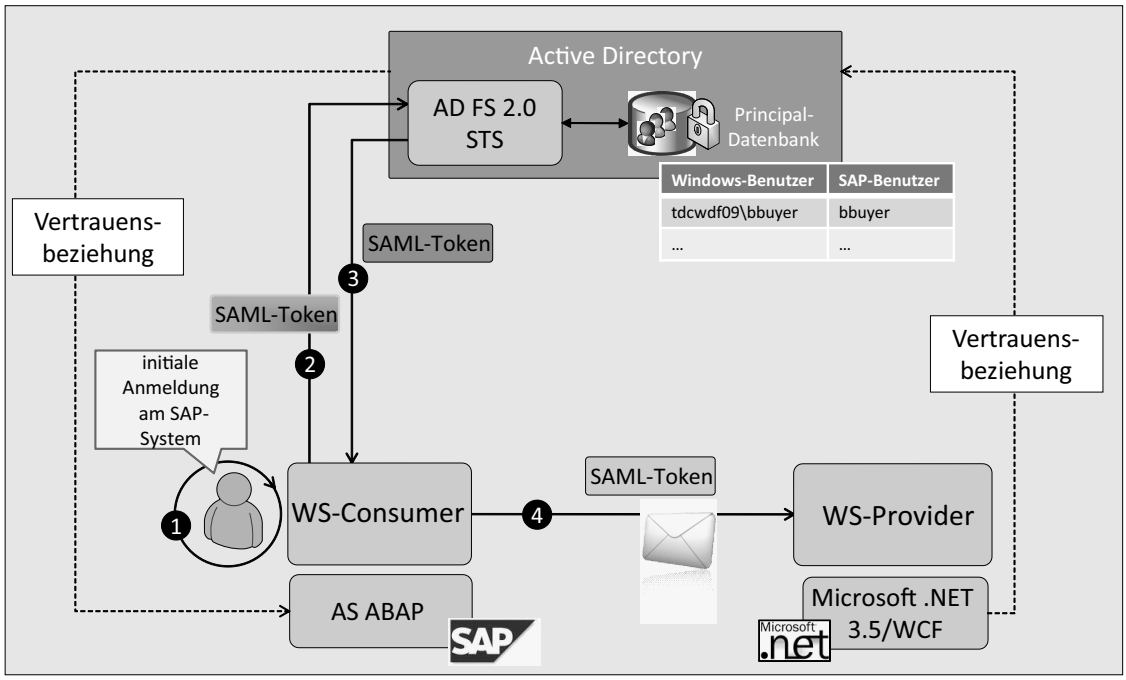

Abbildung 10.17 Single Sign-on von einem SAP-WS-Consumer zu einem .Net-Webservice mittels Active Directory Federation Services 2.0

Beispielanwendungen

Zu diesem Kapitel gibt es einen Beispiel-.Net-WS-Provider, mit den dazugehörigen Visual Studio 2008-Projekten. Den Beispiel-Provider finden Sie im Buch-Download im Verzeichnis *Workshops/10/DotNet*.

10.5.2 Systemvoraussetzungen

Die Voraussetzungen für das SAP-System, das Active Directory und für das .Net-WS-Providersystem sind die dieselben wie im vorigen Workshop.

Da in diesem Szenario der STS der Active Directory Federation Services 2.0 verwendet wird, sollten Sie ADFS 2.0 auf einem Server in Ihrer Landschaft installiert haben. Voraussetzung ist, dass der entsprechende Installationsserver Teil derselben Active-Directory-Domäne ist wie der Server, der den .Net-Webservice bereitstellt.

ADFS 2.0 kann direkt bei Microsoft für Betriebssysteme ab Windows Server 2008 bzw. Windows Vista heruntergeladen werden: *http://www.microsoft.com/adfs/*.

10.5.3 Konfiguration des .Net-WS-Providers

Die Beispielanwendung ist mit der Anwendung im zuvor behandelten Workshop identisch. Den .Net-WS-Provider konfigurieren Sie daher wie in Abschnitt 10.4.3 beschrieben. Die Konfiguration der STS-URL unterscheidet sich jedoch vom vorherigen Workshop.

ADFS-STS-URL in der WS-Provider-WSDL anpassen

Die WSDL des Providers enthält URL-Referenzen zu dem STS, den der WS-Consumer verwenden soll, um eine SAML-Assertion anzufordern.

Ändern Sie die betreffenden URLs in der WCF-Konfigurationsdatei, damit sie auf Ihren ADFS-STS verweisen.

1. Öffnen Sie hierzu die Datei */DotNet/InventoryServiceDotNet/InventoryServiceHost/bin/Debug/InventoryServiceConsoleApp.exe.config* aus dem Buchdownload.
2. Ersetzen Sie nun den Host-Namen im Attribut ADDRESS der XML-Elemente `issuer` und `issuerMetadata` mit dem Host-Namen Ihres ADFS-STS (siehe Abbildung 10.18).
3. Speichern Sie abschließend die Datei.

```
<bindings>
    <customBinding>
        <binding name="IST">
            <textMessageEncoding messageVersion="Soap11WSAddressing10" />
            <security authenticationMode="IssuedToken" messageSecurityVersion="WSSecurity11WSTrust13WSSecureConversation13WS
                <issuedTokenParameters>
                    <issuer address="https://tdc09dc1/adfs/services/trust/13/issuedtokenmixedasymmetricbasic256" />
                    <issuerMetadata address="https://tdc09dc1/adfs/services/trust/mex" />
                </issuedTokenParameters>
                <secureConversationBootstrap authenticationMode="IssuedTokenOverTransport" />
            </security>
            <httpTransport />
        </binding>
    </customBinding>
</bindings>
```

Abbildung 10.18 Anpassen der STS-URLs

10.5.4 Vertrauensbeziehung des WS-Providers zum ADFS-STS einrichten

Richten Sie nun eine Vertrauensbeziehung zum ADFS-STS ein, damit dessen SAML-Assertions vom WS-Provider verarbeitet werden können.

Export des Signaturzertifikats aus dem ADFS

Das primäre Signaturzertifikat des Security Token Services kann direkt aus der ADFS 2.0-Managementkonsole exportiert werden.

1. Melden Sie sich hierzu am STS-Server an, und starten Sie die Managementkonsole im Startmenü unter SETTINGS • CONTROL PANEL • ADMINISTRATIVE TOOLS • AD FS 2.0 MANAGEMENT.

2. Wählen Sie anschließend den Knoten CERTIFICATES im Baum unter AD FS 2.0 • SERVICE (siehe Abbildung 10.19).
3. Drücken Sie nun mit der rechten Maustaste auf das primäre Signaturzertifikat, und wählen Sie VIEW CERTIFICATE..., und ein neues Pop-up-Fenster mit den Zertifikatsinformationen öffnet sich.
4. Im Reiter DETAILS • COPY TO FILE können Sie das Zertifikat in einer Datei abspeichern.

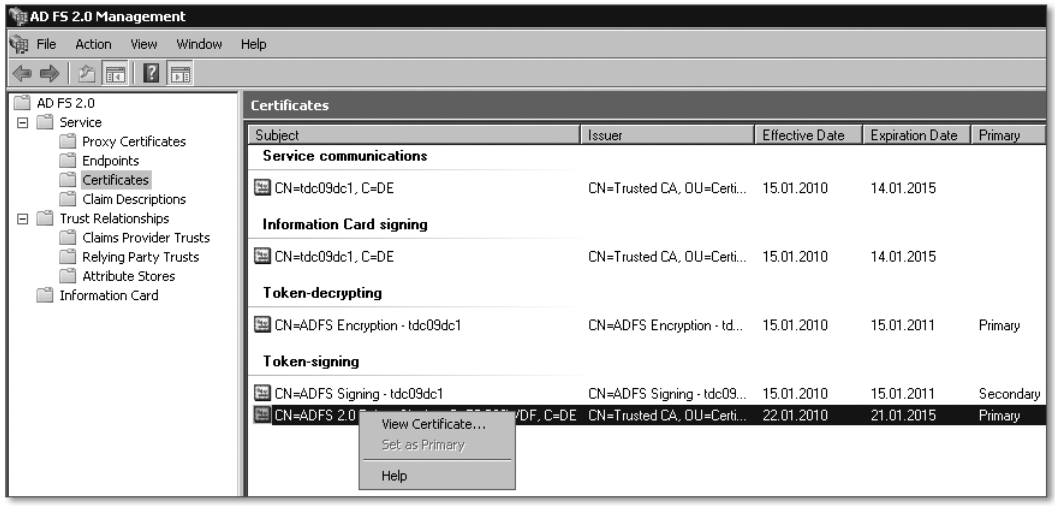

Abbildung 10.19 Export des ADFS-Signaturzertifikats

Import des STS-Signaturzertifikats in das WS-Provider-System

Stellen Sie nun die Vertrauensbeziehung vom WS Provider zum STS her, indem Sie das STS-Signaturzertifikat in das WS-Provider-System importieren.

1. Starten Sie im Startmenü unter RUN die Microsoft Management Console MIT MMC.EXE, und laden Sie das Certificate Snap-in über das Menü FILE • ADD/REMOVE SNAP-IN ..., so wie es in Abbildung 10.20 dargestellt ist.

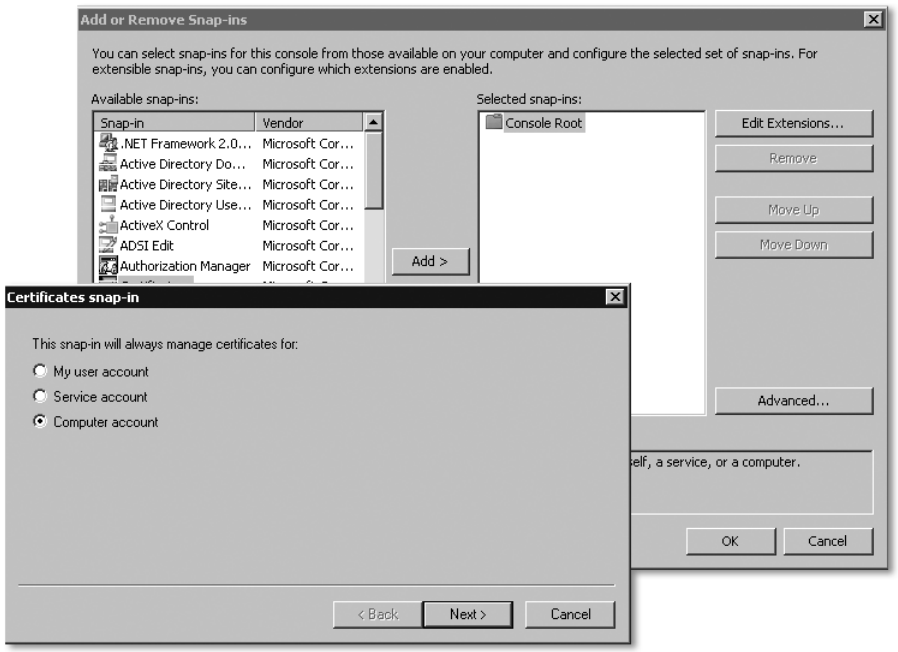

Abbildung 10.20 Laden des Certificate Snap-ins in der Microsoft Management Console (MMC)

2. Wählen Sie nun den Eintrag CERTIFICATES, und klicken Sie auf den Button ADD. Wählen Sie anschließend COMPUTER ACCOUNT und beenden Sie den Wizard mit FINISH.

Das Pop-up-Fenster zur Auswahl des Computer-Accounts erscheint nicht

Wenn das Pop-up-Fenster zur Auswahl des Computer-Accounts nicht erscheint, liegt es daran, dass der Benutzer, mit dem Sie sich angemeldet haben, nicht über die erforderlichen Berechtigungen verfügt; verwenden Sie daher einen Benutzer mit Administrationsberechtigungen für diesen Konfigurationsschritt.

3. Beenden Sie nun den Dialog ADD OR REMOVE SNAP-INS mit CLOSE, und bestätigen Sie das Hinzufügen des Snap-ins CERTIFICATES (LOCAL COMPUTER) im ADD OR REMOVE SNAP-INS-Fenster mit den OK-Button.

4. Klappen Sie nun den Baum unter CONSOLE ROOT auf, und klicken Sie mit der rechten Maustaste auf CERTIFICATES (LOCAL COMPUTER) • TRUSTED PEOPLE • CERTIFICATES (siehe Abbildung 10.21).

5. Wählen Sie anschließend ALL TASKS • IMPORT, um das Zertifikat der Zertifizierungsstelle zu importieren.

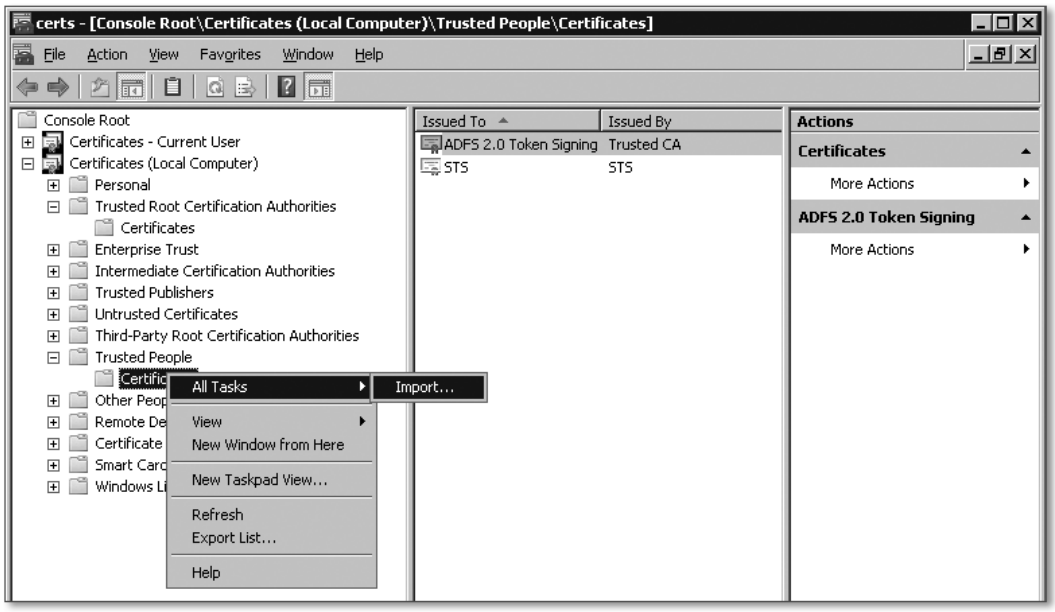

Abbildung 10.21 Import des Signaturzertifikats des STS

Webservice starten

Nachdem Sie alle Konfigurationsschritte durchgeführt haben, können Sie den Webservice nun starten.

1. Starten Sie die Anwendung *InventoryServiceConsoleApp.exe* aus dem Verzeichnis *DotNet/InventoryServiceDotNet/InventoryServiceHost/bin/Debug.* aus dem Buch-Download.

Abbildung 10.22 Erfolgreicher Start des Inventory-.Net-Webservices.

2. Nun können Sie sich die Provider-WSDL in Ihrem Browser anzeigen lassen, z. B. unter *http://iwdfvm3101:1080/?wsdl.* Die URL wird direkt in der Konsolenanwendung angezeigt (siehe Abbildung 10.22).

10.5.5 ADFS-STS für das Ausstellen von SAML-Assertions für den WS-Provider einrichten

STS-Endpunkt für asymmetrische SAML-Assertions freischalten

Das SAP-System verschickt eine Holder-of-Key-Assertion mit einem asymmetrischen Schlüssel an den ADFS-STS. Der ADFS-STS-Endpunkt, der die SAML-Assertion auswertet, muss allerdings zuvor freigeschaltet werden.

1. Starten Sie zunächst die ADFS 2.0-Managementkonsole im Startmenü unter Settings • Control Panel • Administrative Tools • ADFS 2.0 Management.
2. Wechseln Sie anschließend zur Endpoint-Konfiguration, und klicken Sie mit der rechten Maustaste auf den `/trust/13/issuedtokenmixedassymetricbasic256`-Endpunkt. Schalten Sie diesen schließlich mit Enable und Enable on Proxy frei (siehe Abbildung 10.23).

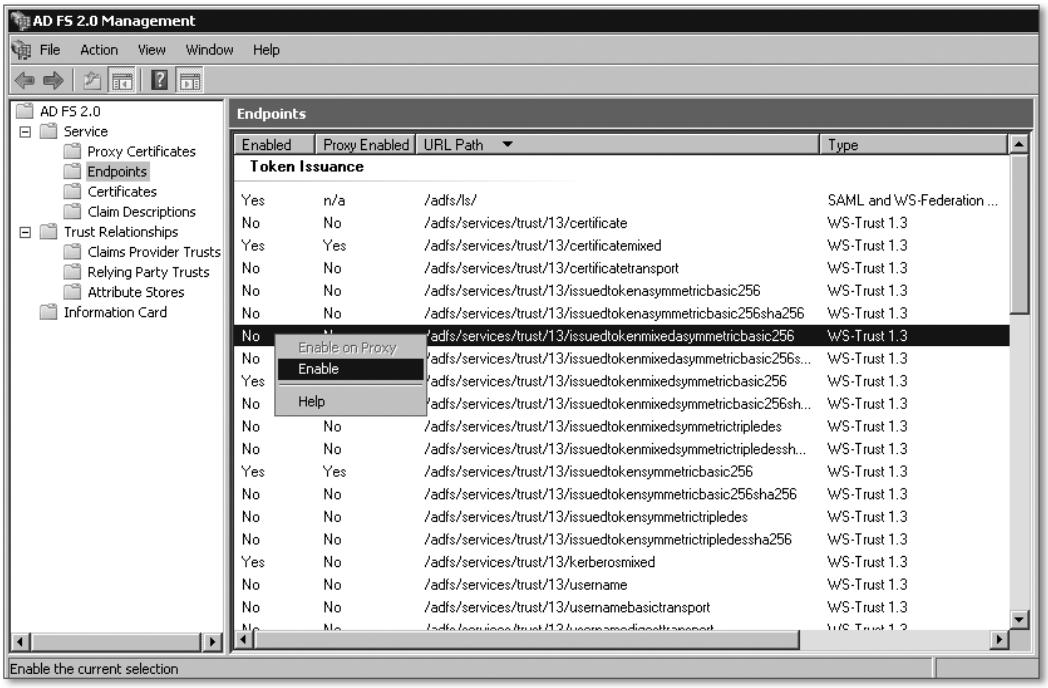

Abbildung 10.23 Aktivieren des asymmetrischen SAML-Endpunkts

3. Starten Sie nun ADFS 2.0 neu. Wählen Sie hierzu im Startmenü Settings • Control Panel • Administrative Tools • Services.
4. Klicken Sie nun auf den ADFS 2.0 Windows Service, und wählen Sie Restart (siehe Abbildung 10.24).

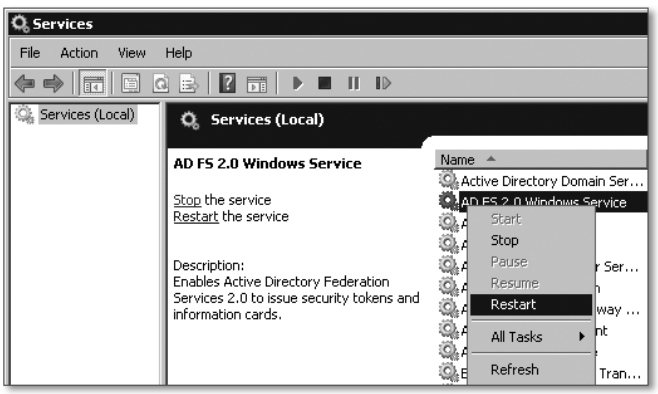

Abbildung 10.24 Neustarten von ADFS 2.0

Active Directory als LDAP-Attribute-Store einrichten

Um die SAML-Assertion, die ausgestellt werden soll, mit den jeweiligen Domänenbenutzerinformationen anreichern zu können, muss das Active Directory als Attribute Store in Form einer LDAP-Quelle (LDAP = Lightweight Directory Access Protocol) im ADFS eingebunden werden.

1. Wechseln Sie hierzu in die Verwaltung der ATTRIBUTE STORES unter TRUST RELATIONSHIPS.
2. Legen Sie nun einen neuen ATTRIBUTE STORE an. Klicken Sie hierzu mit der rechten Maustaste auf ATTRIBUTE STORES • ADD ATTRIBUTE STORE.
3. Wählen Sie jetzt AD AS LDAP als DISPLAY NAME (siehe Abbildung 10.25).
4. Geben Sie anschließend die LDAP-Verbindungs-URL im Format `ldap://<AD-Host>:<LDAP-Port>/Startknoten` zum Domänencontroller Ihrer Domäne an. Port-Nummer 389 ist der Standard-Port für das LDAP-Protokoll. Unter diesem Startknoten sollten Ihre Domänenbenutzer gespeichert sein.

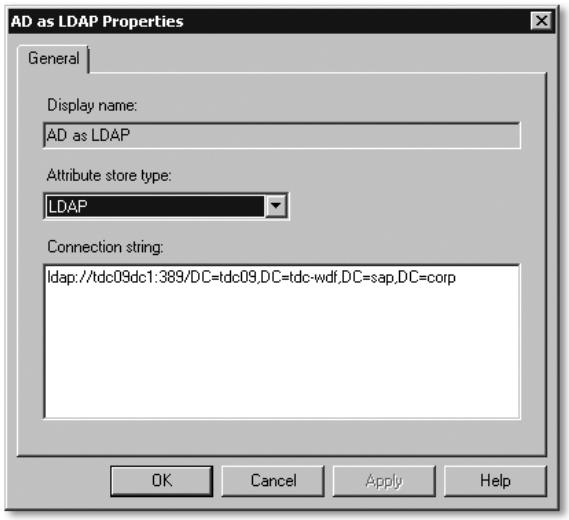

Abbildung 10.25 Domänencontroller als LDAP-Attribute-Store

Ermitteln des Domänencontrollers

Der Domänencontroller Ihrer Domäne steht in der Windows-Umgebungsvariable LOGONSERVER. Diese können Sie mithilfe der Windows-Kommandozeile abfragen.

- Wählen Sie im Startmenü RUN • CMD.EXE.
- Geben Sie anschließend den Befehl »set LOGONSERVER« ein, und bestätigen Sie mit der Enter-Taste.

5. Im Event Viewer von Windows können Sie überprüfen, ob der neue Attribute Store erfolgreich geladen werden konnte (siehe Abbildung 10.26). Diesen finden Sie im Startmenü unter PROGRAMS • CONTROL PANEL • ADMINISTRATIVE TOOLS • EVENT VIEWER.

Active Directory Federation Services 2.0 Logging und Tracing

Weiterführende Informationen zur Diagnose von ADFS 2.0 finden Sie unter:

- DIAGNOSTICS in AD FS 2.0
- *http://tinyurl.com/adfs20diagnostics*

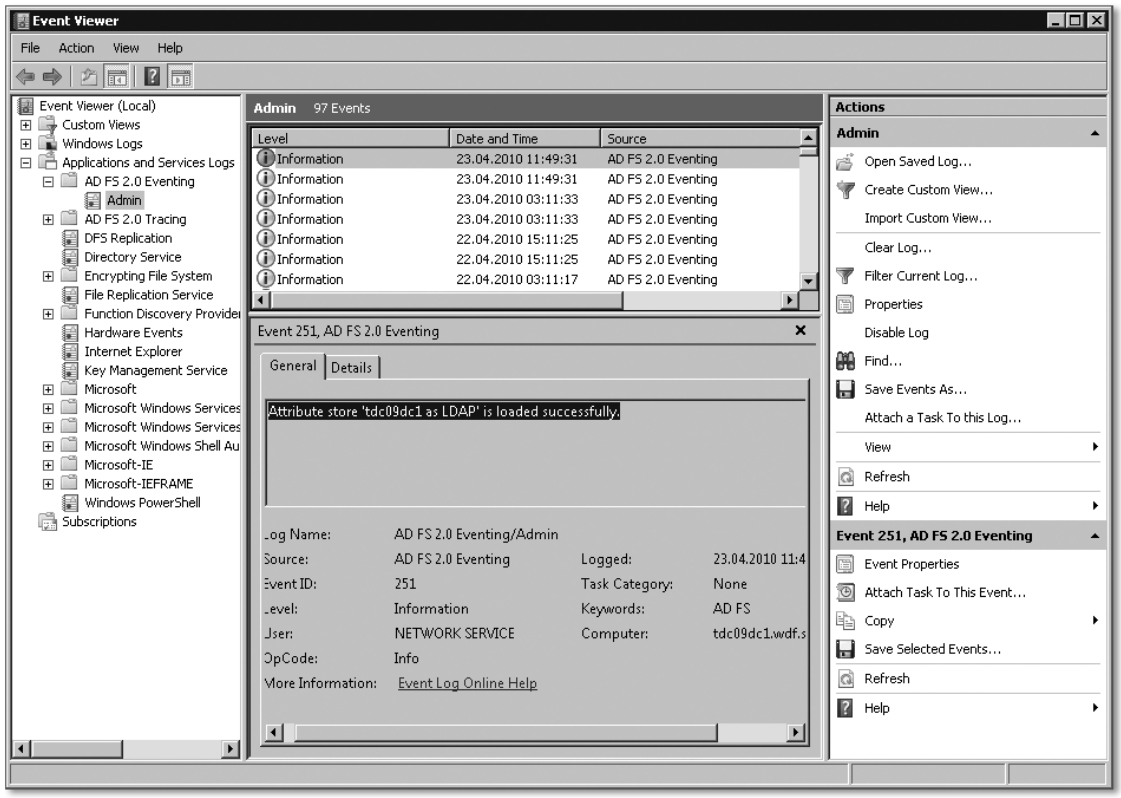

Abbildung 10.26 Erfolgreiches Laden des neuen Attribute Stores

.Net-WS-Provider als Relying Party hinzufügen

Bei der Verarbeitung einer Anfrage für ein Authentifizierungs-Token ist neben der Vertrauenswürdigkeit des WS-Consumers noch eine weitere Frage entscheidend, nämlich für welchen WS-Provider das Token ausgestellt werden soll. Anhand des konkreten Providers wird entschieden:

- welche Informationen (Claims) über den Benutzer in der SAML-Assertion enthalten sein sollen
- mit welchem Zertifikat das Token verschlüsselt werden soll, so dass nur der Empfänger es entschlüsseln kann

Im Umfeld der Active Directory Federation Services werden diese Einstellungen in sogenannten *Relying Partys* durchgeführt.

1. Starten Sie nun den Wizard für das Hinzufügen einer neuen Relying Party (siehe Abbildung 10.27).

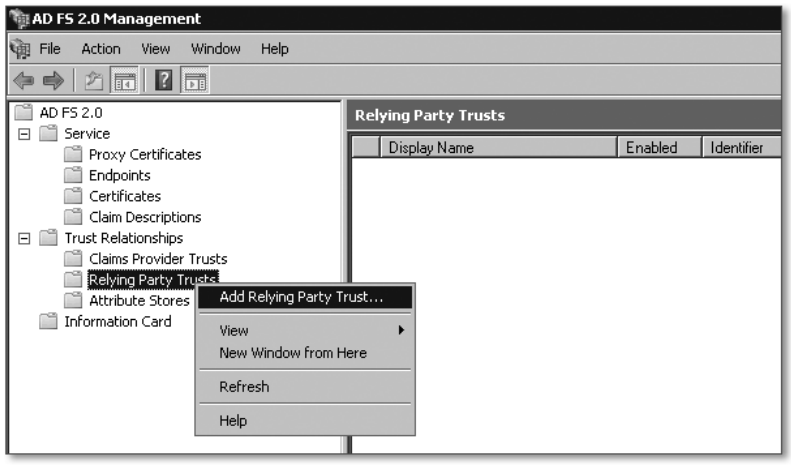

Abbildung 10.27 Wizard zum Hinzufügen einer Relying Party

2. Überspringen Sie den Willkommensbildschirm. Im nächsten Schritt wählen Sie Enter data about the relying pary manually.
3. In den nächsten Schritten hinterlegen Sie die Daten für den .Net-WS-Provider. Die Tabelle 10.3 erläutert die Eingaben der einzelnen Schritte im Konfigurations-Wizard

Display Name	.Net-Inventory-Service
Choose Profile	Wählen Sie das ADFS 2.0-Profil, um die SAML-Authentifizierung nutzen zu können.
Configure Certificate	Importieren Sie das Verschlüsselungszertifikat des .Net-WS-Providers, das Sie in Abschnitt 10.4.3, »Konfiguration des .Net-WS-Providers«, erzeugt haben.
Configure URL	Wenn Sie kein Browserszenario konfigurieren möchten, aktivieren Sie keine der beiden Checkboxen.
Relying party trust Identifier	Geben Sie die Stamm-URL (Host und Port) des .Net-WS-Providers an, z. B. *http://iwdfvm3101:1080/*. Dies hat zur Folge, dass die Relying Party zur Ausstellung von Security Tokens für alle Webservices unter der angegebenen URL verwendet wird. Wichtig: Die URL in der Relying Party muss mit der URL in der WSDL des Webservices übereinstimmen. Ansonsten weiß der STS nicht, welche Relying Party er verwenden soll.
Choose Issuance Authorization Rules	Wählen Sie Permit all users to access this relying party.

Tabelle 10.3 Manuelle Relying-Party-Konfiguration

4. Stellen Sie sicher, dass die Checkbox Open the Edit Claim Rules … aktiviert ist, um den Claim-Rules-Editor nach dem Abspeichern der Relying Party zu starten.

Claim Rule zur Abfrage der Benutzerinformation im AD einrichten

Legen Sie nun die Claim Rule an, die anhand des `NameIdentifier`-Claims, der mit der Pass-through-Regel aus Abschnitt 10.5.6, »Vertrauensbeziehung des ADFS-STS zum SAP-System einrichten«, durchgereicht wird, im Active Directory nach dem passenden Domänenbenutzer sucht.

1. Wählen Sie Add Rule, und geben Sie als Regel Typ Send Claims Using a Custom Rule an. Betätigen Sie anschließend den Next-Button.
2. Kopieren Sie den Inhalt der Datei *relying_party_claim_rule.txt*, die dem Buch-Download zu diesem Kapitel beiliegt, und fügen Sie sie in das Feld unter Custom rule ein (siehe Abbildung 10.28).
3. Klicken Sie schließlich auf Finish und anschließend auf OK, um die Konfiguration der Claims abzuschließen.

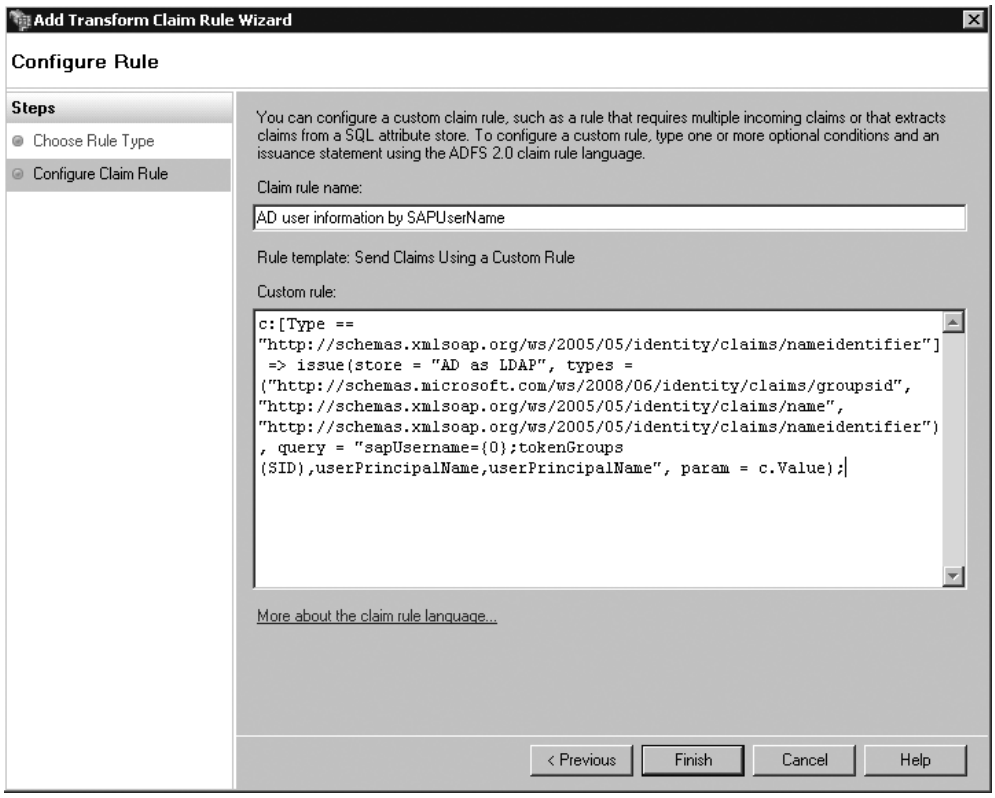

Abbildung 10.28 Claim Rule zur Identifikation des Windows-Benutzers anhand des SAP-Benutzernamens

10.5.6 Vertrauensbeziehung des ADFS-STS zum SAP-System einrichten

Vertrauensbeziehung auf Basis der Transaktion SAML2 einrichten

Zur Anmeldung am ADFS-STS stellt sich das SAP-System eine SAML-Assertion aus. Durch die Verwendung der Option SAML Trust im Report WSS_SETUP kann dem ADFS-STS der verwendete SAML-Issuer und das Signaturzertifikat per WS-Federation-Metadaten zur Verfügung gestellt werden.

1. Melden Sie sich nun an Ihrem SAP-WS-Provider-System an, und starten Sie den Report WSS_SETUP.
2. Wählen Sie anschließend die Option Use SAML Trust (siehe Abbildung 8.8 im Buch).
3. Deaktivieren Sie Test Run, und führen Sie den Report über die F8-Taste aus, um Ihre Einstellungen zu speichern.

Konfiguration der Vertrauensbeziehungen

Die WS-Security-Runtime kann bestehende Vertrauensbeziehungen wiederverwenden. Zwei Optionen stehen hier zur Auswahl:

- **Logon Ticket Trust**
  Vertrauensbeziehung über die PSE, die auch zur Validierung von Logon- bzw. Assertion-Tickets verwendet wird, z. B. die System-PSE.
- **SAML-Trust**
  Vertrauensbeziehung über die Transaktion SAML2 und deren PSEs.

Export der Metadaten aus dem SAP-System

Starten Sie nun die Transaktion SAML2, und es öffnet sich ein neues Browserfenster.

SAML erstmalig für den Mandanten einrichten

Sollte SAML 2.0 für Ihren Mandanten noch nicht eingerichtet worden sein, so erhalten Sie eine entsprechende Meldung beim Start der Transaktion. Betätigen Sie den Button ENABLE SAML2.0 SUPPORT, um die Einrichtung durchzuführen. Als PROVIDER NAME empfiehlt sich die SID und der Mandant, z. B. »UI2/000«. Für alle anderen Einstellungen können Sie die Vorschlagswerte übernehmen.

Sie befinden sich im Reiter LOCAL PROVIDER. Drücken Sie auf den Link METADATA, der sich ganz rechts auf der Buttonleiste befindet. Es öffnet sich sodann ein neues Browserfenster, in dem die Metadaten des Systems angezeigt werden (siehe Abbildung 10.29).

Metadaten signieren

Stellen Sie sicher, dass die beiden Optionen INCLUDE CERTIFICATE IN SIGNATURE und SIGN METADATA ausgewählt sind. Nur so kann der Security Token Service beim Import sicherstellen, dass das Metadatendokument während der Übertragung nicht verändert wurde und dass die Metadaten aus einer vertraulichen Quelle stammen.

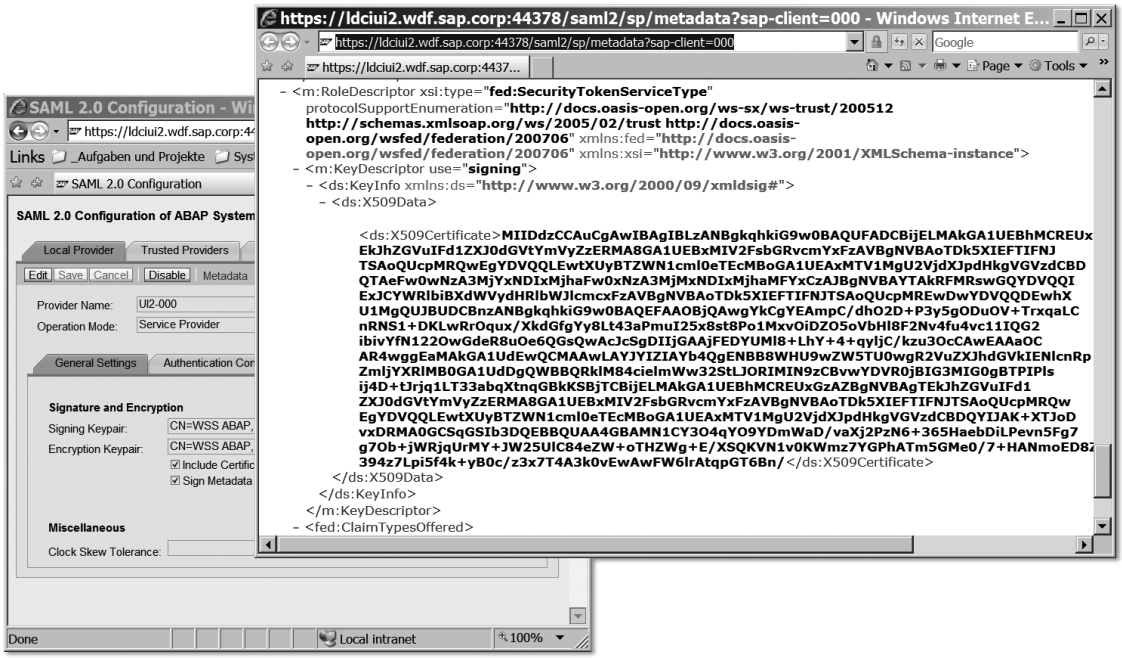

Abbildung 10.29 Metadaten des SAP-Systems

Speichern Sie die Metadaten in einer XML-Datei ab.

Import der Metadaten in den STS

Melden Sie sich nun an dem Server an, auf dem die Active Directory Federation Services 2.0 installiert sind.

1. Starten Sie anschließend die Managementkonsole im Startmenü unter SETTINGS • CONTROL PANEL • ADMINISTRATIVE TOOLS • AD FS 2.0 MANAGEMENT.
2. Starten Sie den Wizard für das Hinzufügen eines neuen *Claims Provider Trusts* (siehe Abbildung 10.30).
3. Geben Sie nun noch die Datei an, unter der Sie die Metadaten im letzten Schritt abgespeichert haben (siehe Abbildung 10.31), und betätigen Sie den NEXT-Button.

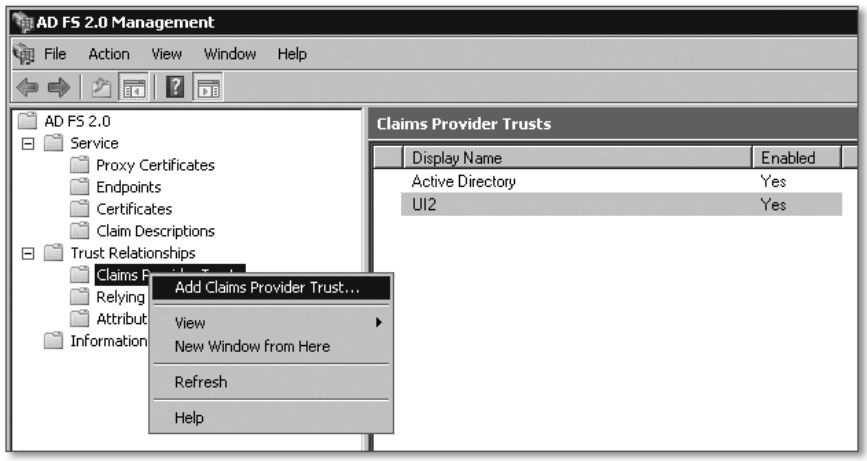

Abbildung 10.30 Hinzufügen einer neuen Vertrauensbeziehung.

Metadatenaustausch über das Netzwerk

Die Metadaten können auch direkt über die URL importiert werden (siehe die erste Option in Abbildung 8.18 im Buch). Die Konfiguration per Metadaten-URL hat den Vorteil, dass sie automatisch aktualisiert werden kann, wenn sich die Einstellung des SAP-Systems ändert.

Um sicherzustellen, dass die Metadaten von einem vertrauenswürdigen Server stammen, ist die Metadaten-URL mit SSL gesichert. Mithilfe der SSL-Serverauthentifizierung kann somit sichergestellt werden, dass die Metadaten von einem vertrauenswürdigen Server kommen.

Damit die Herkunft validiert werden kann, muss das Wurzelzertifikat der Zertifizierungsstelle, mit dem das SSL-Serverzertifikat des SAP-Systems signiert wurde, im Trusted Certification Authorities-Bereich des Windows-Schlüsselspeichers im ADFS-STS-Server vorliegen.

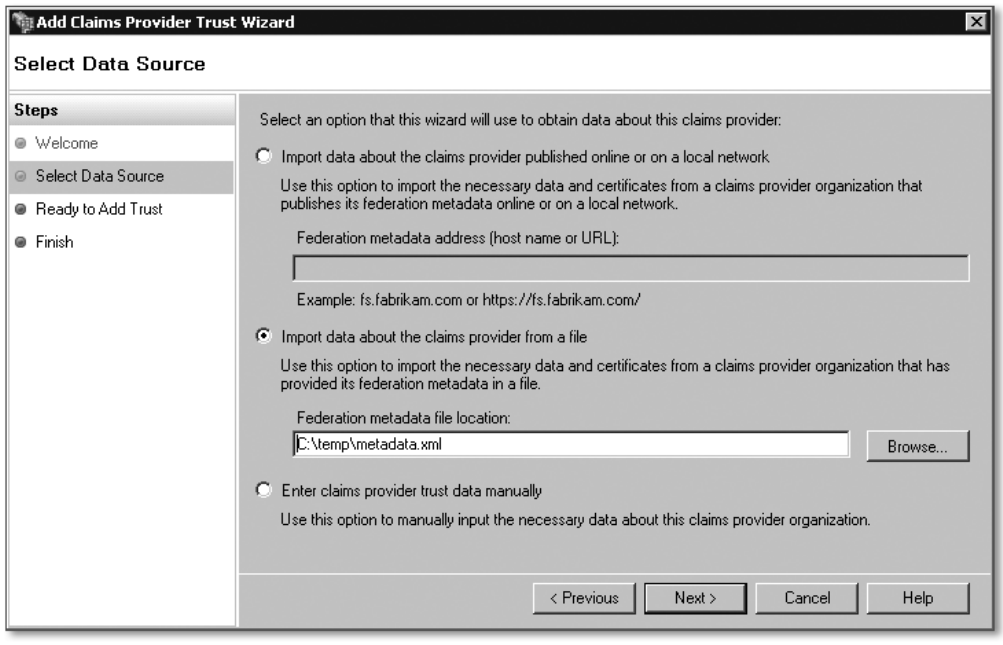

Abbildung 10.31 Import der Metadatendatei

4. Als Name für den Claims Provider Trust bietet sich hier der Name des SAP-Systems an.
5. Alle wesentlichen weiteren Einstellungen wurden durch das Einlesen der Metadaten vorgenommen. Navigieren Sie weiter durch den Wizard – bis zum letzten Schritt.
6. Stellen Sie anschließend sicher, dass die Checkbox Open the Edit Claim Rules aktiviert ist, um den Claim-Rules-Editor nach dem Abspeichern der Relying Party zu starten.

Der erste Teil der Konfiguration des Security Token Services ist jetzt abgeschlossen.

SAP-System-Support des SHA256-Algorithmus

ADFS 2.0 verwendet standardmäßig den SHA256-Algorithmus; SAP-Systeme unterstützen SHA256 ab der SAP Cryptographic-Library-Version 1.555.29 und dem SAP Basisrelease 7.02 bzw. 730.

Haben Sie eine ältere Version, so sollten Sie den *Claims Provider Trust* auf SHA1 ändern. Klicken Sie hierzu mit der rechten Maustaste auf den soeben importierten Claims Provider Trust und anschließend auf PROPERTIES. Im Reiter ADVANCED können Sie den Hash-Algorithmus einstellen. Die Prüfung der Version der SAP Cryptographic Library finden Sie in Abschnitt 8.2.1 im Buch.

Zugriffsmöglichkeit auf die Certificate Revocation Lists (CRL)

Enthält das Signaturzertifikat Verweise auf CRLs, so sollte es dem ADFS möglich sein, darauf zugreifen können. Ist dies nicht der Fall, so kommt es bei der Validierung der SAML-Assertion vom SAP-System zu einem Fehler. Testweise können Sie die Prüfung der CRLs auch abschalten. Starten Sie hierzu Windows PowerShell, und setzen Sie die folgenden Kommandos ab:

- `Add-PSSnapin Microsoft.Adfs.PowerShell`
- `Set-ADFSClaimsProviderTrust -SigningCertificateRevocationCheck none -targetidentifier` *<Name des Claims Provider Trusts>*

Pass-through-Claim-Rule für den NameIdentifier-Claim des SAP-Systems anlegen

Um eine SAML-Assertion für den .Net-Webservice zu erhalten, meldet sich der SAP-WS-Consumer ebenfalls mit einer SAML-Assertion beim ADFS-STS an. Im `NameIdentifier` dieser SAML-Assertion steht der jeweilige SAP-Benutzername, der den Webservice aufruft.

Legen Sie nun eine Claim Rule an, damit der `NameIdentifier`-Claim weiter zur Relying Party, dem .Net-Webservice, propagiert wird, von dem dann die dazugehörigen Domänenbenutzer-Attribute ausgewertet werden können:

1. Wählen Sie ADD RULE, und geben Sie PASS THROUGH OR FILTER AN INCOMING CLAIM als Regeltyp an. Bestätigen Sie dies anschließend über den NEXT-Button.
2. Konfigurieren Sie die Rule so, wie es in Abbildung 10.32 gezeigt wird, und drücken Sie FINISH.
3. Wählen Sie nun noch OK, um die Bearbeitung der Regeln abzuschließen.

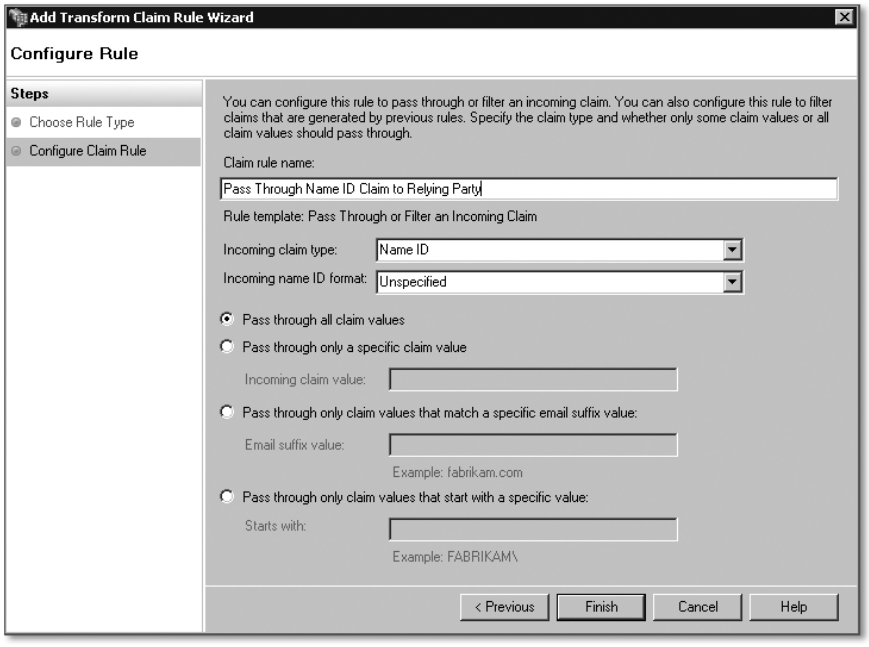

Abbildung 10.32 Pass-through-Regel für den NameID-Claim

10.5.7 Konfiguration des SAP-WS-Consumers

SSL-Vertrauensbeziehung zum ADFS-STS einrichten

Die WSDL des ADFS-STS ist mit SSL geschützt. Bevor die Verbindung zum ADFS-STS angelegt werden kann, muss das SAP-System dem SSL-Serverzertifikat bzw. dem Wurzelzertifikat der Zertifizierungsstelle des Provider-Systems vertrauen.

1. Folgen Sie den Anleitungen in Abschnitt 8.9.10 unter »Export des Wurzelzertifikats aus dem ADFS-STS«, und speichern Sie das Zertifikat in einer Datei.
2. Melden Sie sich anschließend am SAP-WS-Consumer-System an, und starten Sie die Transaktion STRUST (siehe Abbildung 10.33).
3. Doppelklicken Sie nun auf die PSE mit dem Namen »SSL Client Standard« (SSL DIENT SSL CLIENT/STANDARD) ❶.
4. Klicken Sie anschließend auf den Button IMPORT CERTIFICATE ❷, um in dem dann erscheinenden Pop-up-Fenster das SSL-Server- bzw. das Wurzelzertifikat der Zertifizierungsstelle zu importieren. Wählen Sie dabei dasselbe Speicherformat, das Sie auch beim Export benutzt haben, z. B. *Base64*.
5. Klicken Sie auf ADD TO CERTIFICATE LIST ❸, um das Zertifikat der Liste der vertrauten Zertifikate hinzuzufügen.
6. Speichern Sie schließlich die soeben veränderte PSE mit Strg + S oder dem SPEICHERN-Symbol ❹.

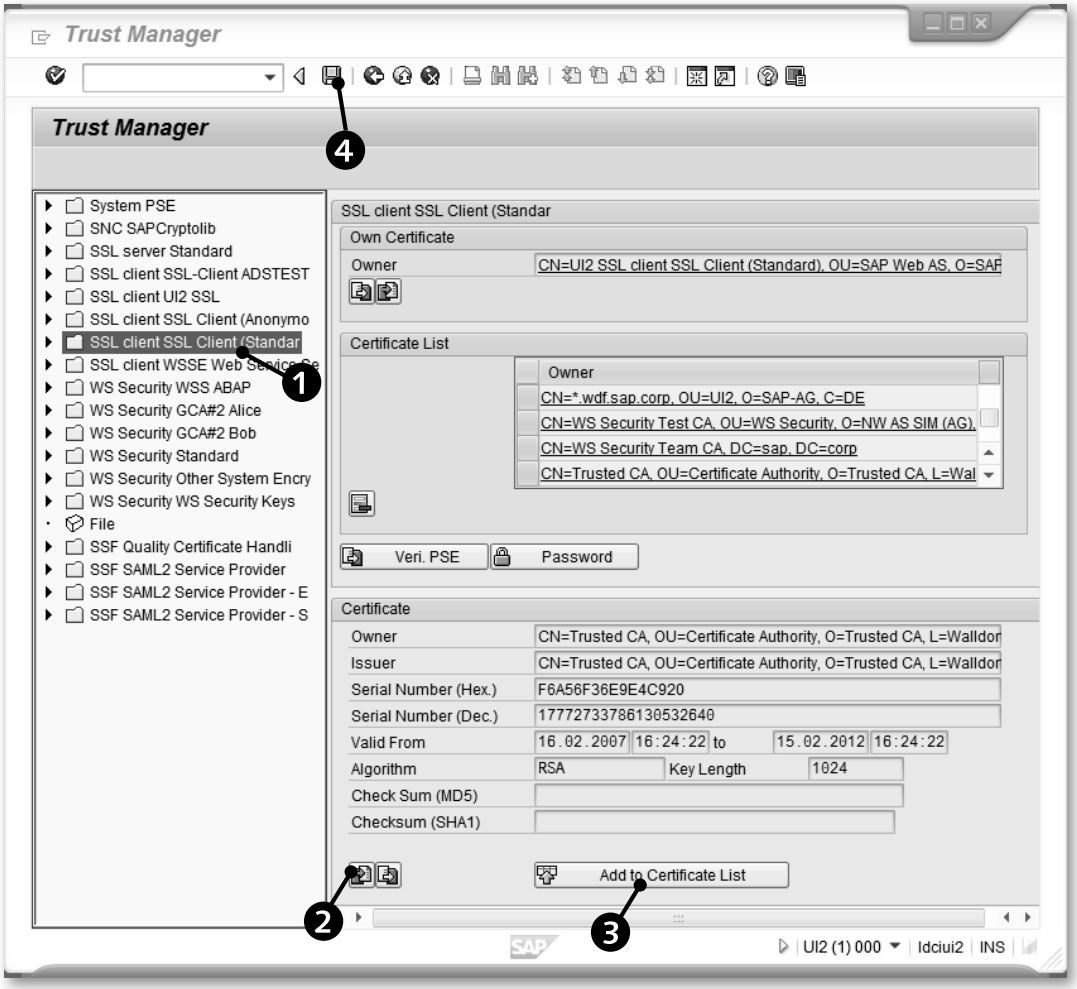

Abbildung 10.33 Import des SSL-Serverzertifikats

Logischen Port zum ADFS-STS einrichten

Um die Security Tokens von einem STS anzufordern, benutzt das SAP-System einen speziellen WS-Consumer-Proxy namens »CO_WSSESECURITY_TOKEN_SERVICE« (interner Name). Legen Sie einen logischen Port zum ADFS-STS an.

ADFS-STS-WSDL

Damit der SAP-WS-Consumer weiß, dass er sich beim ADFS-STS mit einer SAML-Assertion anmelden soll, wurde die WSDL des ADFS-STS angepasst. Die Datei *adfs_saml.wsdl* finden Sie in den Buch-Downloads zu diesem Kapitel.

1. Bearbeiten Sie die Datei *adfs_saml.wsdl*, und tragen Sie an den beiden durch Kommentaren markierten Stellen den Host und den Port Ihres ADFS-STS ein (siehe Abbildung 10.34).

```
73   <wsdl:types>
74     <xsd:schema targetNamespace="http://schemas.microsoft.com/ws/2008/06/identity/securitytokenservice/Imports">
75       <!-- Replace with your ADFS STS host/port. -->
76       <xsd:import schemaLocation="https://tdc09dc1/adfs/services/trust/mex?xsd=xsd0" namespace="http://schemas.microsoft.com/Message"/>
77       <xsd:import schemaLocation="https://tdc09dc1/adfs/services/trust/mex?xsd=xsd1" namespace="http://schemas.xmlsoap.org/ws/2005/02/trust"/>
78       <xsd:import schemaLocation="https://tdc09dc1/adfs/services/trust/mex?xsd=xsd2" namespace="http://docs.oasis-open.org/ws-sx/ws-trust/200512"/>
79     </xsd:schema>
80   </wsdl:types>
```

```
118  <wsdl:service name="SecurityTokenService">
119    <wsdl:port name="IssuedTokenWSTrustBinding_IWSTrust13Async1" binding="tns:IssuedTokenWSTrustBinding_IWSTrust13Async1">
120      <!-- Replace with your ADFS STS host/port. -->
121      <soap12:address location="https://tdc09dc1/adfs/services/trust/13/issuedtokenmixedasymmetricbasic256"/>
122      <wsa10:EndpointReference>
123        <!-- Replace with your ADFS STS host/port. -->
124        <wsa10:Address>https://tdc09dc1/adfs/services/trust/13/issuedtokenmixedasymmetricbasic256</wsa10:Address>
125      </wsa10:EndpointReference>
126    </wsdl:port>
127  </wsdl:service>
```

Abbildung 10.34 Anpassen der WSDL-Datei des ADFS-STS

Der Host-Name und der Inhabername des SSL-Serverzertifikats

Achten Sie bei der Angabe des STS-Host-Namens darauf, dass der Inhabername des SSL-Serverzertifikats mit dem Host-Namen übereinstimmt. Andernfalls wird das SAP-System die SSL-Verbindung zu Ihrem STS verweigern.

   Lautet der Host-Name Ihres STS z. B. *sts.dhcp.mydomain.com,* so muss der Inhabername entweder genauso lauten oder eine Wildcard enthalten (z. B. **.dhcp.mydomain.com*), um anzuzeigen, für welche Domain dieses Zertifikat gültig ist.

2. Starten Sie nun die Transaktion SOAMANAGER im WS-Consumer-System. Navigieren Sie zur Einzelkonfiguration unter Service Administration • Single Service Configuration.
3. Legen Sie anschließend einen logischen Port für den WS-Consumer CO_WSSESECURITY_TOKEN_SERVICE aus der WSDL-Datei an (siehe Abbildung 10.35). Der logische Port wird komplett aus den Einstellungen der WSDL konfiguriert; daher können Sie ihn direkt speichern (siehe Abbildung 10.36).

Freischalten der Verwendung von Security Token Services

Wenn Sie zum ersten Mal einen logischen Port zu einem STS anlegen, erhalten Sie eine Meldung, dass Sie diese Funktion erst freischalten müssen. Öffnen Sie hierzu die SAP Note, auf die in der Meldung verwiesen wird, und führen Sie den Report zur Freischaltung der Funktion aus.

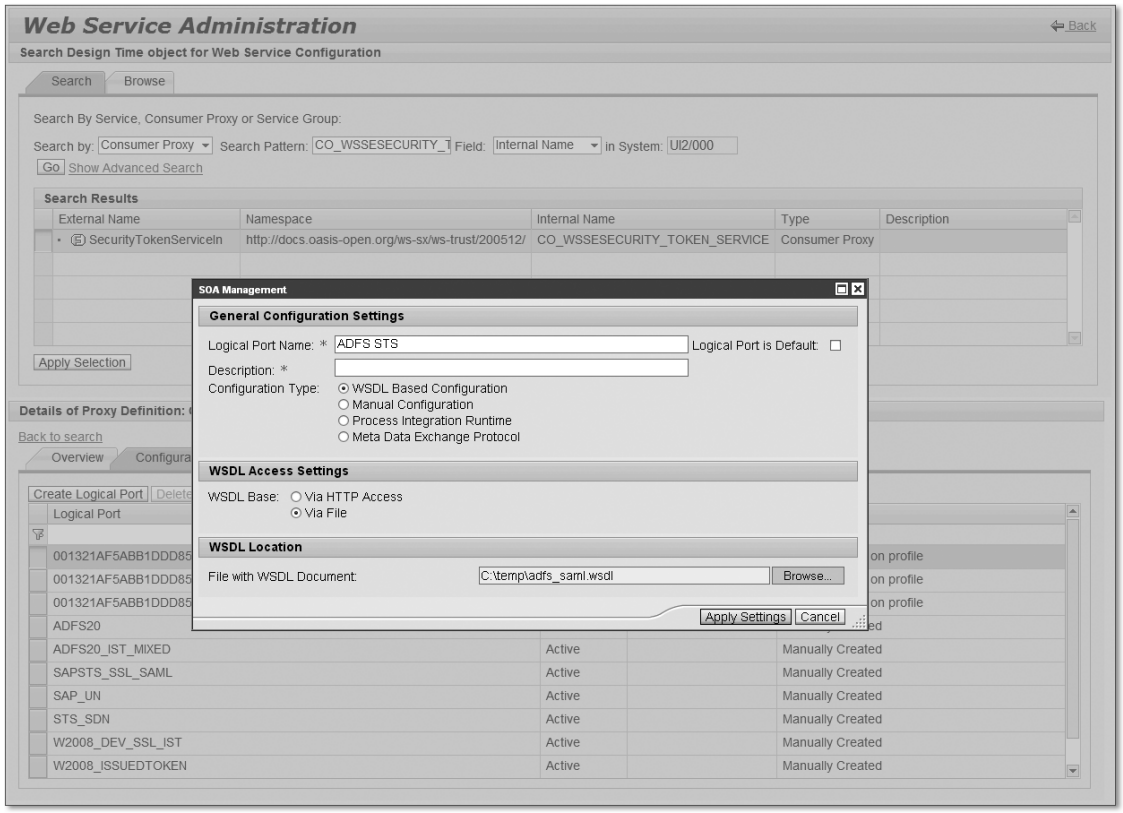

Abbildung 10.35 Anlegen des logischen Ports zum ADFS-STS

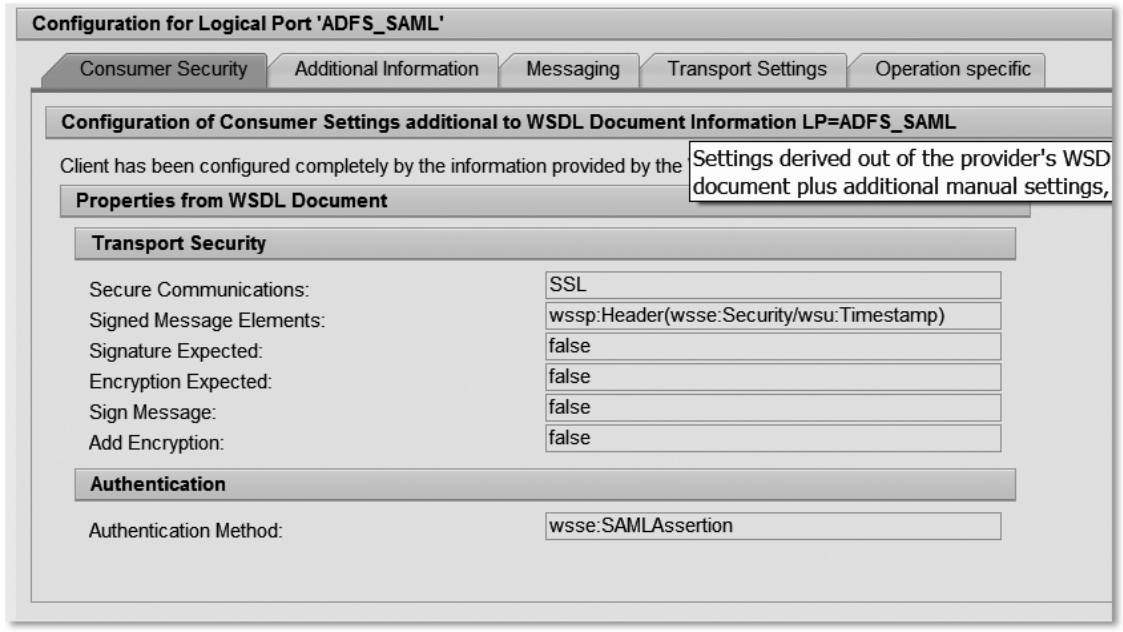

Abbildung 10.36 Die fertige Verbindung zum ADFS-STS

Logischen Port anlegen

Legen Sie mit der WSDL einen logischen Port, so wie es in Abschnitt 10.1.4, »Konfiguration des WS-Consumers mit dem SOA Manager«, beschrieben wird, an.

Legen Sie anschließend einen logischen Port aus der WSDL des .Net-Providers an. Der logische Port wird nun vollständig aus den WSDL-Daten konfiguriert (siehe Abbildung 10.37). Es sind keine weiteren Einstellungen Ihrerseits notwendig.

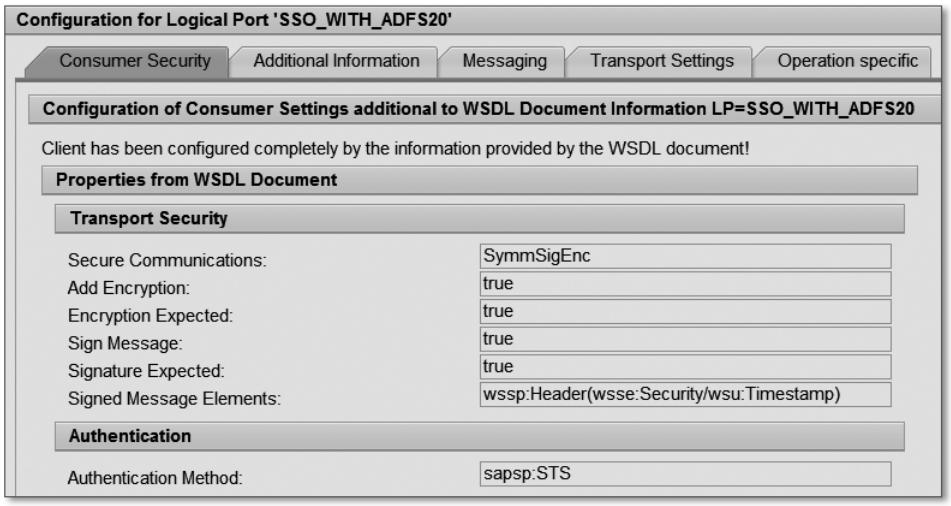

Abbildung 10.37 Der fertig konfigurierte logische Port

10.5.8 Testen des Szenarios

Verwenden Sie anschließend den Report ZLIST_INVENTORY aus Abschnitt 10.1.2, »Report«, um das Szenario zu testen. Geben Sie als Eingabe für den Parameter LP den Namen des logischen Ports an, den Sie bei der Konfiguration im SOA Manager angelegt haben.

10.5.9 Fehleranalyse und -behebung

Wie schon in Abschnitt 8.9.4 erwähnt wurde, ist es auch hier wichtig festzustellen, auf welcher Wegstrecke (WS-Consumer zum STS oder WS-Consumer zum WS-Provider) der Fehler auftritt. Die folgenden Fehlermeldungen können im ADFS-STS bzw. auf der WS-Provider-Seite auftreten:

- **InvalidSecurity: An error occurred when verifying security for the message**
  Dies ist ein generische Exception-Nachricht, die viele Gründe haben kann – im Log weiter unten finden Sie den tatsächlichen Grund. Die häufigsten Gründe sind im Folgenden aufgelistet:
  - **Cannot resolve KeyInfo for decryption: KeyInfo SecurityKeyIdentifier, oder:**
    Cannot resolve the SecurityKeyIdentifier to a SecurityKey with the given SecurityTokenResolver.
    Haben Sie das Verschlüsselungszertifikat des .Net-WS-Providers in das SAP-WS-Consumer System importiert (siehe Abschnitt 10.9.5)? Eine weitere Ursache für diesen Fehler: Der Webservice kann auf den privaten Schlüssel zur Entschlüsselung der SOAP-Nachricht/SAML-Assertion nicht zugreifen. Denn wahrscheinlich liegt dieser nicht mehr im Windows-Zertifikatspeicher *Personal* vor. Es kann allerdings auch sein, dass der Benutzer, unter dem der Webservice läuft, nicht die erforderlichen Berechtigungen hat, um auf den Schlüssel zugreifen zu dürfen. Im letzteren Fall starten Sie das Certificate Snap-in und wechseln zu Personal • Certificates. Klicken Sie anschließend mit der rechten Maustaste auf den jeweiligen privaten Schlüssel, und wählen Sie All Taks • Manage private Keys, um die Zugriffsberechtigungen zu setzen.
  - **Cannot create Servicehost**
    Diese Meldung tritt beim Starten des ServiceHosts für den Beispiel-Webservice auf. Entweder hat der angemeldete Benutzer nicht die erforderlichen Berechtigungen, um einen Ser-

viceHost und damit auch einen TCP-Port nach außen zu öffnen, oder auf dem Port läuft schon ein Webservice.

- **The audience restriction was not valid because the specified audience identifier is not present in the acceptable identifiers list of this Federation Service**
  Das SAP-System schreibt die Webservice-URL in die Audience Restriction innerhalb der SAML-Assertion, um anzuzeigen, dass die Assertion exklusiv für den Empfänger bestimmt ist. Stimmen Host, Port und URL-Pfad im logischen Port zum Webservice/STS nicht mit der Adresse überein, unter der der Webservice/STS erreichbar ist, erhalten Sie diese Fehlermeldung.
- **The key needed to verify the signature could not be resolved from the following security key identifier SecurityKeyIdentifier**
  Der passende öffentliche Schlüssel des STS-/SAP-Systems zur Verifikation der Signatur liegt nicht im Bereich Trusted People • Certificates im Windows-Zertifikatspeicher vor. Folgen Sie den Schritten in Abschnitt 10.5.4, »Vertrauensbeziehung des WS-Providers zum ADFS-STS einrichten«, bzw. 10.5.6, »Vertrauensbeziehung der ADFS-STS zum SAP-System einrichten«, wenn der Fehler im STS auftritt.

Ein anderer Grund für diesen Fehler kann ein fehlgeschlagener Zugriff auf die *Certificate Revocation List*, CRL sein. Zertifikate können eine URL auf diese Liste enthalten. Kann jedoch darauf aus irgendeinem Grund nicht zugegriffen werden, wird diese Fehlermeldung ausgeworfen.

Prüfen Sie die Gültigkeit des Signaturzertifikats und ob auf die URL im Zertifikatattribut `CRL Distribution Point` zugegriffen werden kann. Für Testzwecke können Sie die Revocation-Prüfung auch abschalten.

Starten Sie Windows PowerShell auf Ihrem AD FS 2.0, und setzen Sie die folgenden Befehle ab:

```
Add-PSSnapin Microsoft.Adfs.PowerShell
set-adfsclaimsprovidertrust -signingcertificaterevocationcheck none -
targetidentifier <SAML Issuer bzw. Identifier im Claims Provider Trust>
```

- **The request scope is not valid or is unsupported**
  Im ADFS-STS wurde keine Relying Party gefunden, deren Relying Party Identifier zur URL des gerufenen Webservices passt.

Stellen Sie sicher, dass Sie das SAP-System als Relying Party im ADFS angelegt haben. Abschnitt 8.9.4 bzw. 8.9.5 beschreiben deren Konfiguration. Es ist entscheidend, dass Sie den SAP-Webservice mit derselben URL ansprechen, die Sie als Relying Party Identifier angegeben haben. Geben Sie einen vollqualifizierten Host-Namen an, z. B. *td09dc1.wdf.sap.corp.* Diesen muss der .Net-Consumer ebenfalls verwenden.

Die häufigsten Fehlerursachen, die eine Konfigurationsänderung im SAP-WS-Consumer-System erfordern, sind folgende:

- **No Logical Port for this Secure Token Service URL: »https://... oder Configuration is invalid: No LP exists for STS: https://** Es existiert kein logischer Port zum STS mit der angegebenen URL. Legen Sie, wie es in Abschnitt 10.5.7, »Konfiguration des SAP-WS-Consumers«, beschrieben wird, über die Transaktion SOAMANAGER einen logischen Port zum ADFS-STS an.
- **ICM_HTTP_SSL_ERROR** Dieser Fehler deutet auf ein Problem hin, das während der SSL-Sitzung aufgetreten ist.

Starten Sie die Transaktion SMICM, und wählen Sie im Menü Goto • Trace File • Display End. SAP-Hinweis 1318906 gibt weitere Hilfestellungen zur Auswertung des Traces und zur Behebung des Problems.