Vorlesung am 23.6.2009

Serviceorientiertes E-Government

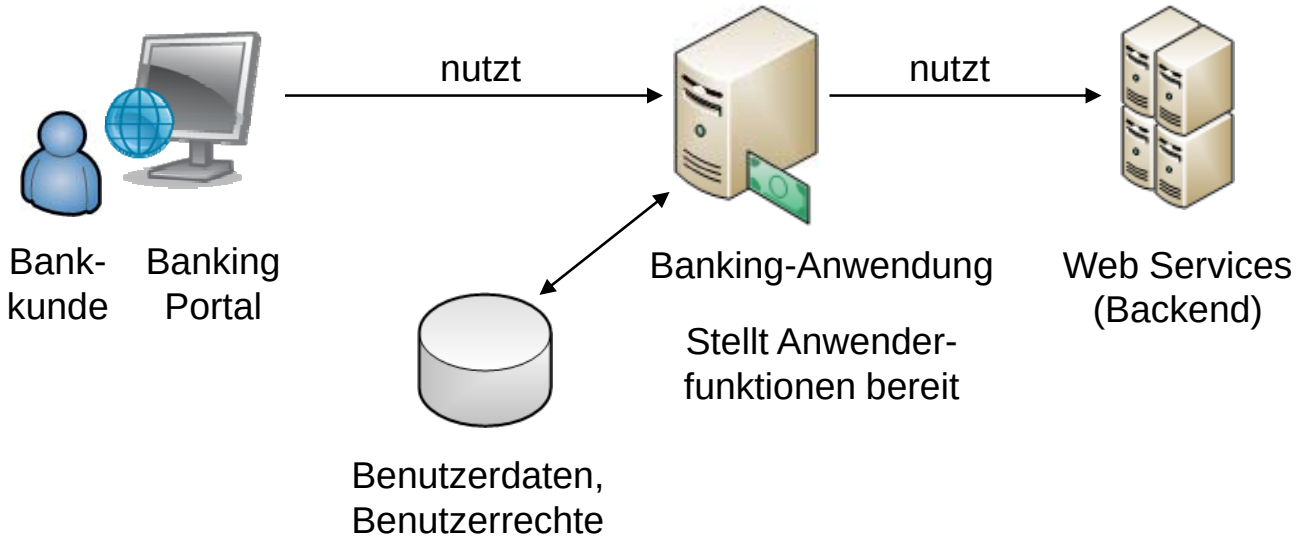
Identity Management und Berechtigungen

Dr. Frank Sarre
Lehrbeauftragter der LMU München

Ausgangssituation (1)

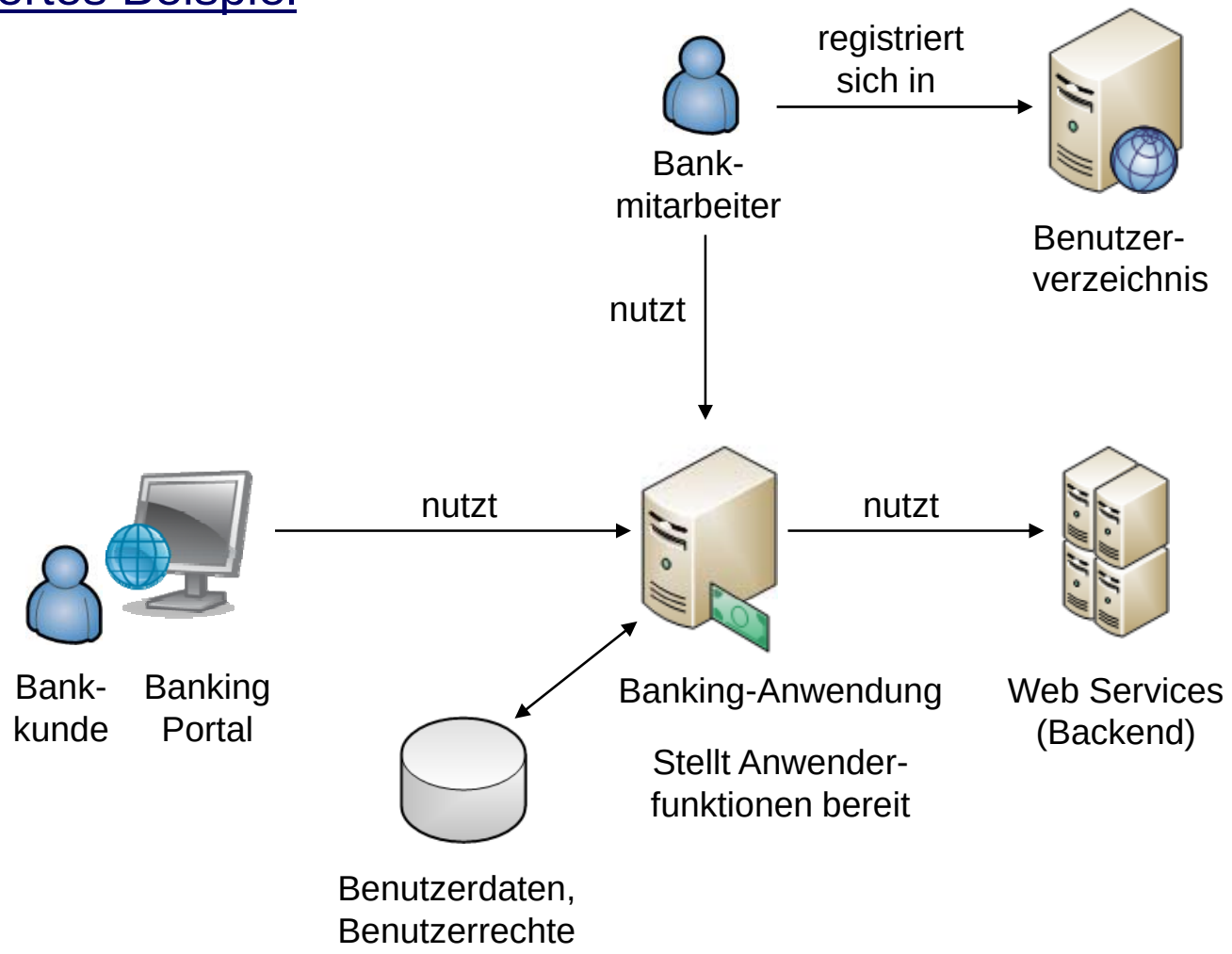
- Web Services und die verwendeten Protokolle bieten in der grundlegenden Form keine Sicherheitsmechanismen
- In der Praxis besteht jedoch immer die Notwendigkeit, dass Services gewissen Nutzungsbeschränkungen unterliegen

Beispiel



Ausgangssituation (2)

Erweitertes Beispiel



Sicherheitsanforderungen (1)

Anforderungen mit höchster Priorität

- Authentifizierung / Authentisierung
- Autorisierung
- Vertraulichkeit
- Integrität

Weitere Anforderungen geringerer Priorität

- Unleugbarkeit → Vorgang ist zweifelsfrei nachvollziehbar und kann nicht abgestritten werden
- Verfügbarkeit → Denial of Service muss verhindert werden
- Auditing → Protokollierung und Nachvollziehbarkeit

Sicherheitsanforderungen (2)

Authentifizierung / Authentisierung

- **Authentifizierung** ist die Verifikation einer behaupteten Identität
- **Authentisierung** ist die Vorlage eines Nachweises der eigenen Identität

→ Diese Ziele erreicht man üblicherweise durch **Identity Management** und **Zertifikate**

Autorisierung

- Festlegung bzw. Prüfung, ob ein Subjekt eine Aktion überhaupt ausführen darf bzw. welche Aktionen ein Subjekt ausführen darf

→ Dieses Ziel erreicht man üblicherweise durch **Identity Management in Verbindung mit Rollen und Rechten**

Sicherheitsanforderungen (3)

Vertraulichkeit

- Informationen und Daten, die über Organisationsgrenzen hinweg verschickt werden, **dürfen nicht durch unbefugte Dritte gelesen werden können**.

→ Dieses Ziel erreicht man üblicherweise durch **Verschlüsselung**.

Integrität

- Die übermittelten Informationen müssen **unversehrt übermittelt** werden bzw. dürfen zumindest **nicht unbemerkt verändert** werden können.

→ Dieses Ziel erreicht man üblicherweise durch **Signaturen**.

Identity Management (1)

Identität

- Systemobjekte haben in der Regel eine eindeutige Kennzeichnung und individuelle Attribute.

→ Daraus definiert sich eine sog. Identität (engl. Identity)

Objekttypen (Beispiele):

- Benutzer
- Services
- Maschinen

- Üblicherweise sind einer Identität verschiedene Rollen zugewiesen.

An diesen Rollen hängen in der Regel definierte Rechte.

Identity Management (2)

Beispiele für Identitäten

- Benutzer in einem Active Directory oder LDAP-Verzeichnis
- Computer in einem Active Directory oder LDAP-Verzeichnis
- Benutzer in einer Benutzerdatenbank einer Anwendung

Beispiele für Rollen

- Systemadministrator
- Finanzbuchhalter
- Administrator für die Finanzbuchhaltung
- Verkäufer

Beispiele für Rechte

- Benutzer anlegen
- Konten in der Finanzbuchhaltung anlegen
- Konto bebuchen

Identity Management (3)

Identity Management innerhalb einer Organisation

- Häufig findet man eine verteilte Verwaltung von Benutzern vor:
  Die unterschiedlichen Anwendungen halten Informationen über Benutzer in eigenen Datenbanken

  Technisch wäre es heute möglich, eine zentrale Benutzerdatenbank zu führen (z.B. mit LDAP oder Active Directory)

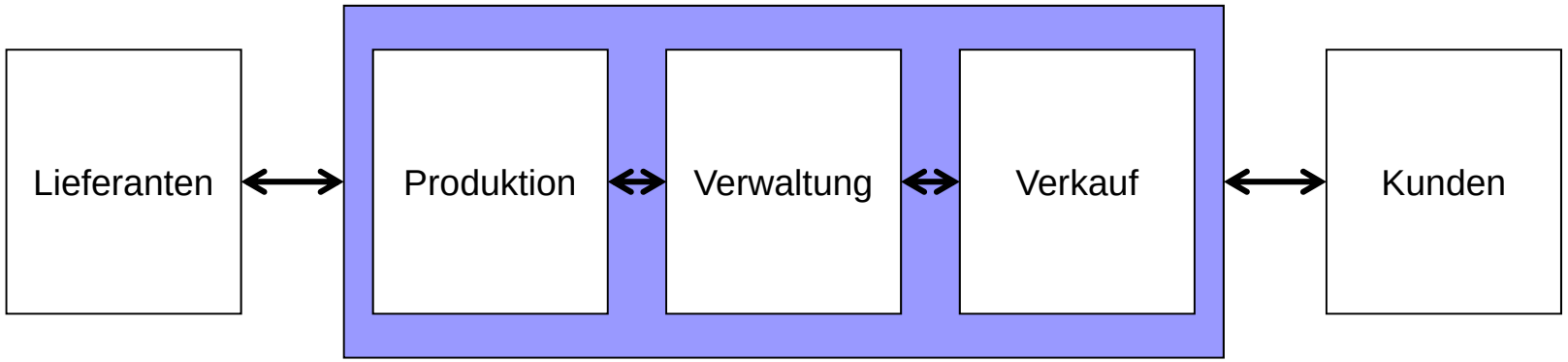
- Die anwendungsbezogenen Rechte werden in der Regel in den einzelnen Anwendungen verwaltet

- Meistens hat der Benutzer neben der Repräsentation im Verzeichnisdienst eine eigene Repräsentation im Anwendungssystem (z.B. für die Hinterlegung anwendungsspezifischer Einstellungen)

Identity Management über Organisationsgrenzen hinweg

- In einer SOA werden auch Services anderer Organisationseinheiten (Handelspartner, unabhängige Ressorts) genutzt, die in einem anderen technischen Umfeld existieren

  Es besteht kein direkter Zugriff auf das Identitätsmanagement des externen Services bzw. umgekehrt kann der Service nicht auf das Identitätsmanagement des Consumers zugreifen



→ **Federated Identity Management**

Identity Management (5)

Federated Identity Management

- Die Verantwortung für alle Identitäten einer Organisationseinheit obliegt der jeweiligen Organisationseinheit
- Die Organisationseinheiten vertrauen sich gegenseitig
- Der Service Provider hat keine Informationen über den Benutzer - Ausnahme bilden sog. „Security Credentials“, die bei der Nutzung des Services übergeben werden
- Der Service Provider lässt die Credentials des Nutzers bei der zuständigen Organisationseinheit überprüfen (diese Org-Einheit ist dann in der Rolle eines sog. **Identity Providers (IdP)**)

  Identity Provider können auch vertrauenswürdige Dritte sein.

Security Credentials

Übliche Security Credentials sind:

- Benutzername / Passwort
- X.509 Zertifikat
- Kerberos-Ticket
- SAML-Token

Single Sign On

- Jede Identität, zum Beispiel ein Benutzer, soll sich nur ein einziges Mal am System anmelden und dann alle Anwendungen und Services nutzen können, für die sie/er eine Berechtigung hat

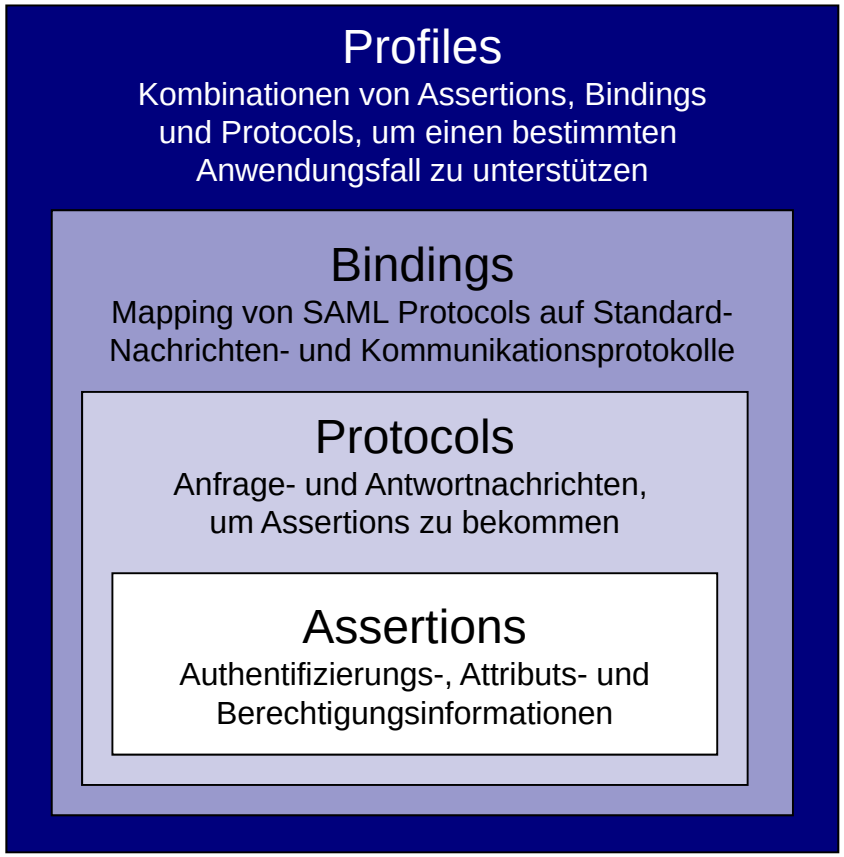
  Eine Technologie, die diese Zielsetzung im SOA-Umfeld unterstützt, ist **SAML** (Security Assertion Markup Language)

SAML (1)

Umsetzung des Federated Identity Management durch SAML

- SAML ist ein OASIS-Standard (http://saml.xml.org/saml-specifications)

Quelle:
XML-Spektrum 5/2007
SAML 2.0, Ein Tutorium



SAML (2)

Assertion

- Eine Assertion bestätigt die Authentizität eines Subjekts, die ein IdP einem Service garantiert
- Es wird auch die Information geliefert, wie das Subjekt authentifiziert wurde
- Zudem werden weitere Attribute über das Subjekt geliefert – dazu gibt es mehrere Statement -Typen:
  - Authentication → Authentifizierungsmethode
  - Attribute → Hinterlegung beliebiger Attribute
  - Autorization Decision → Rechte des Subjekts
- Weiterhin kann es weitere Informationen zur Gültigkeit einer Assertion geben, z.B. Zeitraum der Gültigkeit oder berechtigter Nutzer der Assertion usw.

SAML (3)

Protocols

Insgesamt gibt es 6 definierte Abfolgen von Abfragen und Bestätigungen.

Die wichtigsten sind:

- Assertion Query and Request
  - → Anfrage neuer Assertions
- Authentication Request
  - → Prüfung einer bestehenden Assertion
- Artifact Resolution
  - → Abfrage einer Assertion, die nicht direkt, sondern nur als Verweis übergeben wurde
- Single Logout
  - → (Beinahe) simultaner Logout aus verschiedenen aktiven Sessions

SAML (4)

Bindings

- SOAP 1.1
  - → SAML-Nachricht über SOAP
- HTTP Redirect
  - → SAML-Nachricht über HTTP-Redirect
- HTTP Post
  - → SAML-Nachricht base64-encoded in HTML-Formularen
- HTTP Artifact
  - → Referenz bzw. Adresse einer SAML-Nachricht
- SAML URI
  - → Referenz bzw. Adresse einer Assertion

SAML (5)

Profiles

- Insgesamt gibt es 13 „Profile“ für verschiedene Einsatzszenarien von SAML
- In einem Profil wird eine Kombination von Assertions, Bindings und Protokollen für den jeweiligen Anwendungszweck empfohlen

Beispiel: **Web Browser SSO Profile**

- Ein Internet-Nutzer greift auf eine Ressource eines Serviceproviders zu

  (dazu muss er sich zuvor bei einem Identityprovider authentisieren oder der Serviceprovider leitet eine noch nicht erfolgte Authentifizierung seinerseits ein)
- Dieser Vorgang erzeugt eine Assertion, die der Serviceprovider akzeptiert
- Die Implementierung erfolgt mit **Authentication Request Protocol** in Verbindung mit **HTTP Redirect**, **HTTP POST** oder **HTTP Artifact** Binding

SAML (6)

Verwendung digitaler Signaturen

- Um sicherzustellen, dass eine Assertion gültig und authentisch ist, signiert der IdP die Assertion
- Standard: Signaturen mit X.509-Zertifikaten
- Die Signatur wird mit Hilfe eines asymmetrischen Verschlüsselungsverfahrens (Public Key Verfahren) erzeugt

SAML (7)

Asymmetrische Verschlüsselung

- Verwendung eines **Schlüsselpaares**:
  - **Öffentlicher** Schlüssel (public key)
  - **Privater** Schlüssel (private key)
- **Verschlüsselung** von Daten / Nachrichten durch Verwendung des **öffentlichen Schlüssels des Empfängers**. Nur mit dem privaten Schlüssel des Empfängers können die Nachrichten wieder entschlüsselt werden.
- Die **Signatur** einer Nachricht ist ein Hash-Wert, der mit dem **privaten Schlüssel des Absenders** verschlüsselt wird.
- Der Empfänger erzeugt seinerseits den Hash-Wert der (entschlüsselten) Nachricht und vergleicht diesen Wert mit dem Hash-Wert der (entschlüsselten) Signatur

  → Anhand der Signatur lässt sich die **Integrität** einer Nachricht überprüfen

SAML (8)

Zertifikate und „Certification Authorities“

- Bei der Übermittlung einer signierten Nachricht wird auch ein **Zertifikat** übermittelt, das Informationen über den Aussteller des Zertifikats und den **öffentlichen Schlüssel des Absenders** enthält.

  Zertifikate sowie das Schlüsselpaar werden von einer sog. **Certification Authority** (CA) ausgestellt.

- Dieser Certification Authority müssen sowohl Sender als auch Empfänger von Nachrichten **vertrauen**.

- Anhand eines Zertifikats kann der Nachrichtenempfänger prüfen, ob der behauptete **Benutzer** und damit der übermittelte Schlüssel **authentisch** ist.

- Certification Authorities können externe Dritte sein, z.B. Verisign. Es können aber auch firmenintern eigene Zertifikate erstellt und herausgegeben werden.

Berechtigungen (1)

Entscheidung, einen Service nutzen zu dürfen

- Bei dem Aufruf eines Services muss geprüft werden, ob der Consumer die Berechtigung dazu hat

  → „**Policy Enforcement Point**“

- Ein Service kennt in der Regel keine User oder Berechtigungen, d.h. die Entscheidung muss außerhalb des Services getroffen werden

  → „**Policy Decision Point**“

- Eine Autorisierung kann verschiedenen Subjekttypen zugeordnet werden
  - Benutzer
  - Maschine
  - Anwendung / Service

- Es wird ein **Regelwerk** benötigt, mit dem Berechtigungen definiert und abgefragt werden können

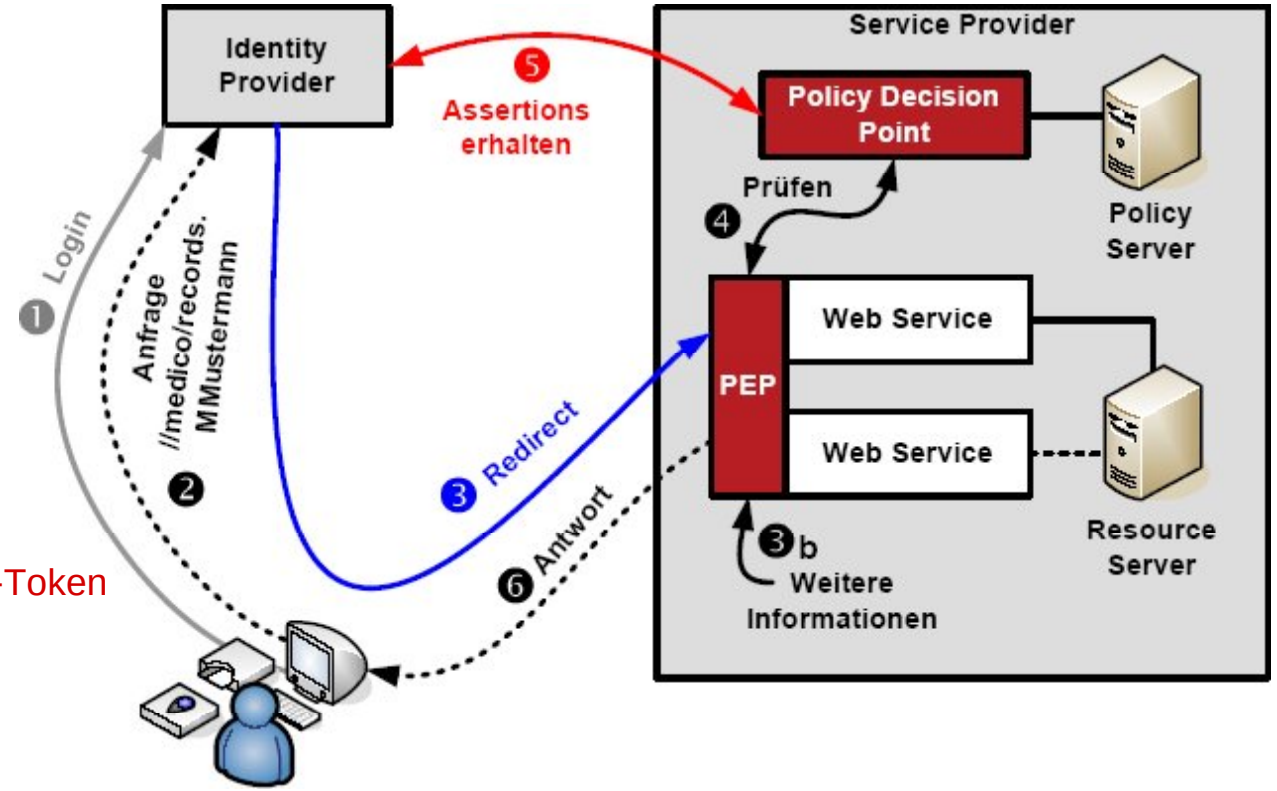
Berechtigungen (2)

XACML (eXtensible Access Control Markup Language)

- Beschreibung von Autorisierungsinformationen nach einem definierten Schema
- Verwendet SAML

Beispielszenario

Quelle: http://www.iam-wiki.org/XACML



Wichtige Schritte

Schritt 1: Authentisierung

Schritt 3: Liefert User-SAML-Token

Schritt 5: Autorisierung

Weitere Federated Identity Management Frameworks

- Liberty ID-FF 1.2
  - Zusammenschluss verschiedener Organisationen und Firmen, um ein weltweites Identity Management mit Single Sign On für das E-Business aufzubauen
  - Schaffung eines „Circle of Trust“
  - Mapping von Identitäten anhand von Pseudonymen
  - Viele Aspekte sind bereits in SAML 2.0 eingeflossen, deshalb wird SAML genutzt und unterstützt
- Shibboleth Projekt
  - Umsetzung von SAML für Webanwendungen
  - Erweiterungen um Privacy-Funktionen
  - Single Sign On und Attribute-Austausch

Sicherheit für Web Services (1)

Transport Layer Security

- Die Verbindung zwischen Consumer und Serviceprovider wird auf der Ebene der Transportprotokolle verschlüsselt
- Verwendung von HTTPS bzw. SSL / TLS

  → Aufbau eines sicheren Kanals (Secure Channel)
- Vorteil ist die für die Services die transparente Verschlüsselung
- Nachteilig ist, dass nur eine Punkt-zu-Punkt-Sicherheit möglich ist

  → keine oder nur beschränkte Verwendung von Intermediären möglich (z.B. ESB), da diese Subsysteme Teile der Information für die Verarbeitung / Weiterleitung benötigen
- Keine asynchrone Kommunikation möglich, da synchrones Protokoll
- Nur in relativ einfachen Umgebungen einsetzbar

Sicherheit für Web Services (2)

Message Layer Security

- Verschlüsselung und Signierung der Nachrichteninhalte
- Herstellung einer „End-to-End-Sicherheit“

→ Im Allgemeinen das geeignete Verfahren für Web Services:

- Verschlüsselung des SOAP-Bodys ganz oder in Teilen
  - → XML Encryption
- Signatur des SOAP-Bodys ganz oder in Teilen
  - → XML Signature
- Signatur von Teilen des SOAP-Headers

Sicherheit für Web Services (3)

XML Encryption

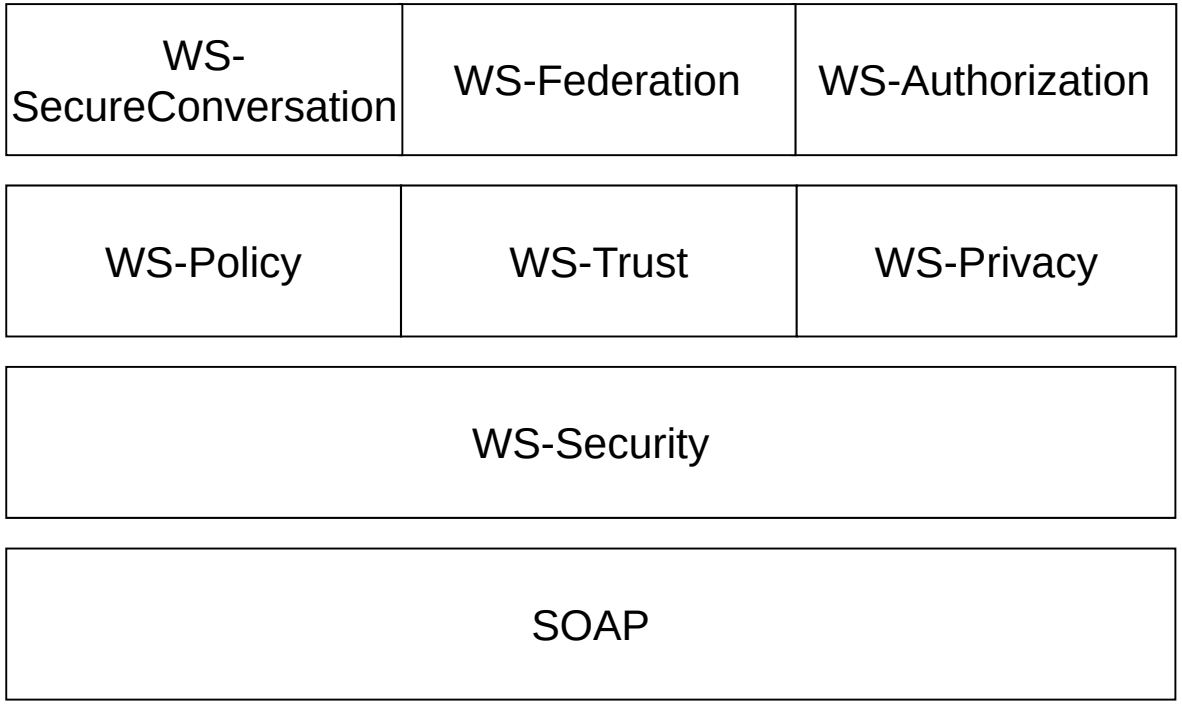
- Verschlüsselung ganzer XML-Dokumente oder nur einzelner Teilstrukturen
- Symmetrische oder asymmetrische Verschlüsselung möglich
- Mindeststandards sind Advanced Encryption Standard (AES) und TripleDES
- Die gewählte Methode wird in der XML-Datei selbst definiert
- Initialisierungsparameter für die Ver- bzw. Entschlüsselung werden ebenfalls integriert
- Bei symmetrischer Verschlüsselung werden die Schlüssel häufig vorab mit asymmetrischer Verschlüsselung ausgetauscht

Sicherheit für Web Services (4)

WS-Security

- Festlegung, wie Sicherheitsinformationen in SOAP-Nachrichten untergebracht werden
- Verwendung der verschiedenen Standards
  - XML Encryption
  - XML Signature
  - SAML

Erweiterungen von WS-Security



Sicherheit für Web Services (6)

WS-Policy

- Festlegung der **Sicherheitsanforderungen** für eine Kommunikation
- Definition ist bereits in der Servicebeschreibung möglich

WS-Trust

- Festlegung des **Formats** der sicherheitsrelevanten Daten, z.B. SAML-Token
- Definition von **Vertrauensketten**
- Definition von **Namensräumen**

WS-SecureConversation

- Verfahren zum Austausch eines Session Keys für symmetrische Verschlüsselung mit Hilfe eines asymmetrischen Verfahrens (Diffie-Hellman) → **Schaffung eines Sicherheitskontextes**
- Verwendung von Sicherheitstoken aus WS-Trust

Sicherheit für Web Services (7)

WS-Federation

- Unterstützung der **Herstellung gegenseitigen Vertrauens**
- Baut auf Methoden von WS-Policy, WS-Trust und WS-SecureConversation auf
- Wird für die Schaffung einer Single-Sign-On-Lösung benötigt

WS-Privacy

- Definition der Anforderungen an die **Vertraulichkeit** von Nachrichten
- Methoden zur Sicherstellung der Einhaltung der Anforderungen

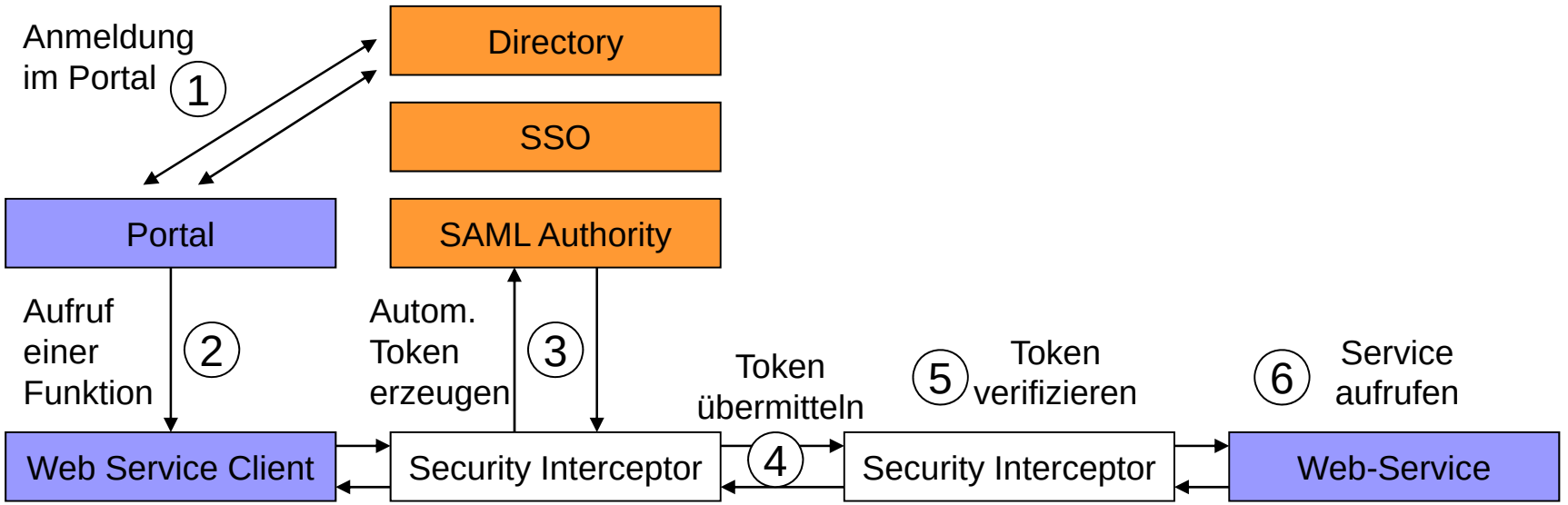
WS-Authorization

- Definition und Auswertung von **Zugriffsrechten**
- Erweitert WS-Trust um Berechtigungsattribute in den Sicherheitstokens

Security Proxy

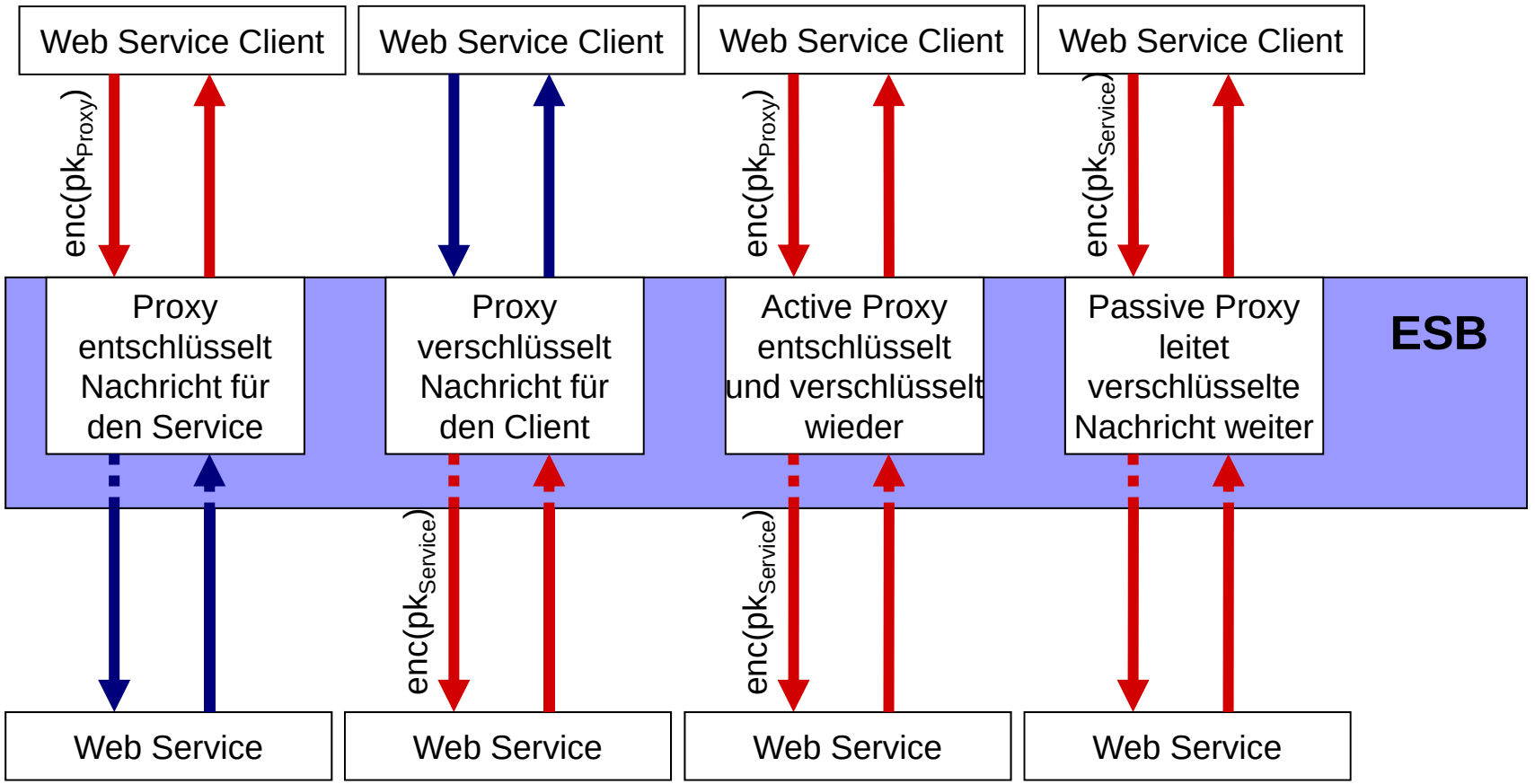
- Services müssen die Sicherheitsrichtlinien nicht selbst umsetzen
  - → Hierzu kann ein sog. **Security Proxy** vorgeschaltet werden (andere Bezeichnungen sind **SOAP-Proxy** oder **Security-Interceptor**)

Beispiel: Authentifizierung



Intermediary (Security) Proxy

Beispielszenarien für Verschlüsselung mit ESB



enc(pk …) = Verschlüsselung mit Public key von …