Sicherheit in Web Services

Dr. Eric Dubuis*
CHOOSE Talk / SWEN Vortragsreihe
Berner Fachhochschule

19. Februar 2007

* Unter Mitwirkung von Dr. Stephan Fischli und Dr. Bernhard Anrig

SWEN Fachverein

Ziele:

- Förderung der Vernetzung und Zusammenarbeit zwischen den (Fach)Hochschulen und der Wirtschaft
- Stärkung des Software Engineerings in schweizerischen Klein- und Mittelunternehmen (KMU)
- Fördern einer praxisorientierten Ausbildung an den Fachhochschulen

Aktivitäten:

- Veranstaltungsreihe zu Software Engineering Themen
- Talentierte Studierende unterstützen
- Unterstützung bei gemeinsamen Projekten, Publikation entsprechender Resultate.
- Diskussionsplattform Software Engineering
- Gemeinsame Unterrichtsunterlagen

www.swen-network.ch

Wer sind wir?

Dr. Bernhard Anrig

Dr. Stephan Fischli

Dr. Eric Dubuis

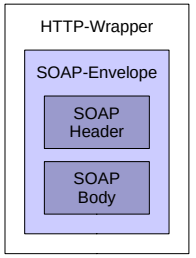
- Professoren des Fachbereichs Informatik des Departements Technik und Informatik der Berner Fachhochschule.
- Unterricht im Bereich verteilter Systeme, theoretischer Informatik und Security.
- Sie beschäftigen sich im Rahmen eines internen Forschungsprojekts mit Web Services Security.
- Ausserdem arbeitet Dr. Anrig am EU-Projekt FIDIS "Future of Identity in the Information Society" mit.

Inhalt

Web Service-Grundlagen

Simple Object Access Protocol (SOAP)

SOAP definiert den Aufbau von XML-Meldungen zum Aufruf eines Web Service



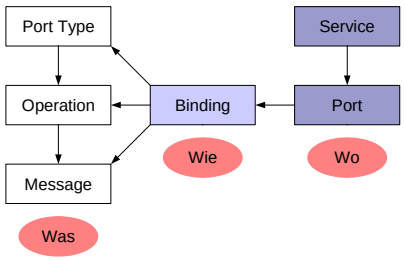
Beispiel einer SOAP-Meldung via HTTP

```
POST /purchaseorder/submit HTTP/1.1
Content-Type: text/xml; charset=utf-8
Host: wss.ph.com:80

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <order xmlns="http://wss.ph.com/order">
      <address>
        <name>Alice Smith</name>
        ...
      </address>
      <payment>
        ...
      </payment>
  </soap:Body>
</soap:Envelope>
```


Web Services Description Language (WSDL)

WSDL ist eine XML-Sprache, um die Schnittstelle eines Web Service zu beschreiben



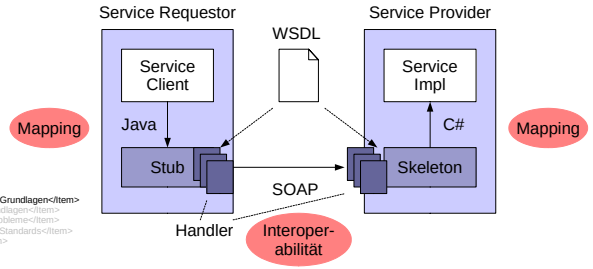
Struktur eines WSDL-Dokuments

```
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/">
  <import .../>
  <types>
    Datenformate
  </types>
  <message>
    Meldungsaufbau
  </message>
  ...
  <portType>
    Operationen des Service
  </portType>
  <binding>
    Protokollanbindungen
  </binding>
  <service>
    Adresse des Service
  </service>
</definitions>
```

XML header
XML schema definitions
Message declarations
Port type definitions (endpoints)
SOAP bindings section
Services declaration

Technische Architektur

Stub und Skeleton sind Hilfsobjekte, welche Programmaufrufe in SOAP-Meldungen umwandeln. Handler bewirken Nach- bzw. Vorverarbeitung.



Code first versus Contract first

Code first
- WSDL wird aus der Service-Implementation erzeugt
- Vorteil: einfache und schnelle Realisierung

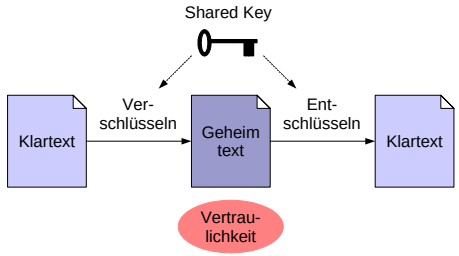
Contract first
- WSDL wird zuerst erstellt und dann der Web Service implementiert
- Vorteil: Interoperabilität eher gewährleistet

Auch:
- XML-Verarbeitung auf Ebene Applikationscode

Security-Grundlagen

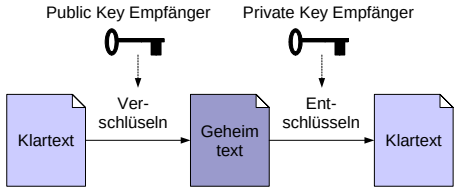
Symmetrische Verschlüsselung

- Ver- und Entschlüsseln erfolgt mit demselben Schlüssel
- Vorteil: schnelle Algorithmen, anwendbar auf beliebig lange Texte
- Nachteil: Schlüssel muss vorher ausgetauscht werden



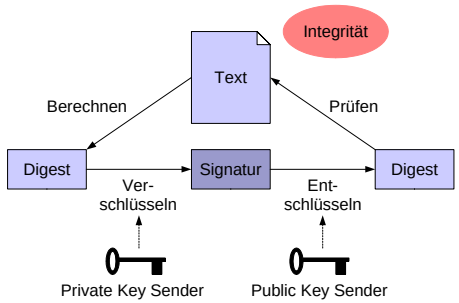
Asymmetrische Verschlüsselung

- Ver- und Entschlüsseln erfolgt mit verschiedenen Schlüsseln
- Vorteil: Public Key kann öffentlich bekannt gemacht werden
- Nachteil: aufwändige Algorithmen



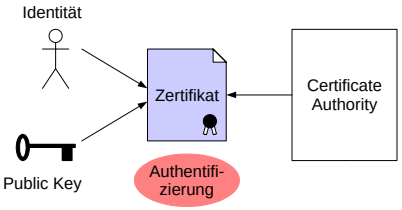
Digitale Signaturen

Eine Signatur ist ein verschlüsselter Digest eines Texts



Digitale Zertifikate

Ein digitales Zertifikat enthält eine Identität mit einem Public Key und ist von einer Zertifizierungsstelle unterschrieben



Sicherheitsebenen bei Web Services

Anwendungsebene	XML-Dokumente	XML-Signature XML-Encryption
Meldungsebene	SOAP-Meldungen	WS-Security
Transportebene	TCP-Datenstrom	SSL

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

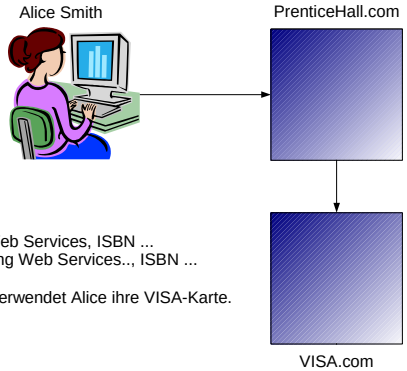
Welches sind die Security-Probleme?

Sicherheitsanforderungen

- **Authentifizierung**
  Meldung stammt von einem bestimmten Benutzer
- **Autorisierung**
  Nur bestimmte Benutzer dürfen einen Dienst verwenden
- **Integrität**
  Meldung wird auf ihrem Weg nicht verändert
- **Vertraulichkeit**
  Meldung kann nicht von Dritten gelesen werden
- **Unleugbarkeit**
  Sender und Empfänger können Meldung nicht abstreiten

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Ein Szenario



A.S. bestellt:
- 1 x Securing Web Services, ISBN ...
- 1 x Implementing Web Services.., ISBN ...

Zur Bezahlung verwendet Alice ihre VISA-Karte.

Original einer Bestellung

```
<?xml version="1.0"?>
<order xmlns="http://wss.ph.com/order">
  <address>
    <name>Alice Smith</name>
    ...
  </address>
  <payment>
    <amount>76.87</amount>
    <creditCard>Visa</creditCard>
    <creditCardDetails>
      <number>4019 2445 0277 5567</number>
      <holder>ALICE SMITH</holder>
      <expiration>2007-03-01</expiration>
    </creditCardDetails>
  </payment>
  <item id="isbn-12345-6789-0">...</item>
  <item id="isbn-12345-9876-1">...</item>
</order>
```

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Sicherstellung der Integrität, Vertraulichkeit und Unleugbarkeit

- **HTTPS**

- Dokumentsicherheit

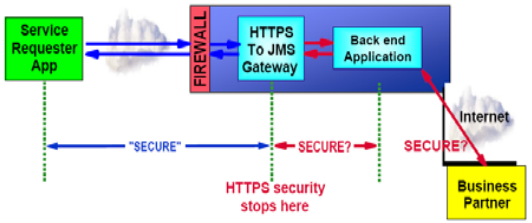
<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Vor- und Nachteile bei HTTPS

- gut verstanden, standardisierte Technologie
- funktioniert auch für die gegenseitige Authentifikation
- gesamter Datenstrom wird behandelt
- weiss nicht, dass XML im Datenstrom ist:
  - keine selektive Verschlüsselung, kein selektives Aufzeichnen, kein selektives Weiterleiten
  - kein Signieren der Daten
- Schutz ist begrenzt:
  - kein Schutz auf Ebene Betriebssystem oder über Tier-Grenzen hinweg
- Schutz ist nur Punkt-zu-Punkt
  - Konsument (z.B. VISA.com) und Produzent (z.B. PrenticeHall.com) sprechen nicht direkt miteinander

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Nachteile bei HTTPS illustriert



<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Sicherstellung der Integrität (Fortsetzung)

- HTTPS
- **Dokumentsicherheit**
  - Verschlüsselung
  - Signierung

Dokumentsicherheit: Verschlüsselung

PrenticeHall.com braucht die Kreditkartendetails nicht zu kennen.

Original Teildokument:

```
<payment>
  <amount>76.87</amount>
  <creditCard>Visa</creditCard>
  <creditCardDetails>
    <number>4019 2445 0277 5567</number>
    <holder>ALICE SMITH</holder>
    <expiration>2007-03-01</expiration>
  </creditCardDetails>
</payment>
```


Verschlüsselung (Fortsetzung)

Nach der Verschlüsselung des Teildokuments:

```
<payment>
  <amount>76.87</amount>
  <creditCard>Visa</creditCard>
  <EncryptedData ...>
    <EncryptionMethod ... "...aes128-cbc"/>
    <KeyInfo ...>
     <EncryptedKey ...>
       <EncryptionMethod ... "...rsa-1_5"/>
       <KeyInfo ...>VISA.com-Zertifikat ...</KeyInfo>
       <CipherData><CipherValue>cJql7...</CipherValue>
       </CipherData>
     </EncryptedKey>
    </KeyInfo>
    <CipherData><CipherValue>NlUQ...</CipherValue>
    </CipherData>
  </EncryptedData>
</payment>
```

öffentlicher Schlüssel von VISA.com

gemeinsamer Schlüssel

verschlüsselte Kredit-karteninformation von Alice

Dokumentsicherheit: Signierung

- Der Betrag bei der Bezahlungsinformation ist kritisch
- Die Warenangaben der Bestellung ist kritisch
- Die Angaben des Bestellers sind kritisch

Signierung am Beispiel der Bezahlungsinformation nach Verschlüsselung:

```
<payment>
  <amount>76.87</amount>
  <creditCard>Visa</creditCard>
  <EncryptedData ...>
    <...>...</...>
    <CipherData><CipherValue>NlUQ...</CipherValue>
    </CipherData>
  </EncryptedData>
</payment>
```

Signierung (Fortsetzung)

Die Bezahlungsinformation wird vom Besteller signiert:

```
<payment Id="#PI">
  <amount>76.87</amount>
  <creditCard>Visa</creditCard>
  <EncryptedData ...>...</EncryptedData>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>...
      <Reference URI="PI">
      <Transforms>...</Transforms>
        <DigestMethod Algorithm="...sha1"/>
        <DigestValue>CMX...</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>CC8L...</SignatureValue>
    <KeyInfo>
      z.B. Zertifikat von Alice Smith...
    </KeyInfo>
  </Signature>
</payment>
```

was wurde signiert?

Unterschrift von Alice Smith

Signierung (Fortsetzung)

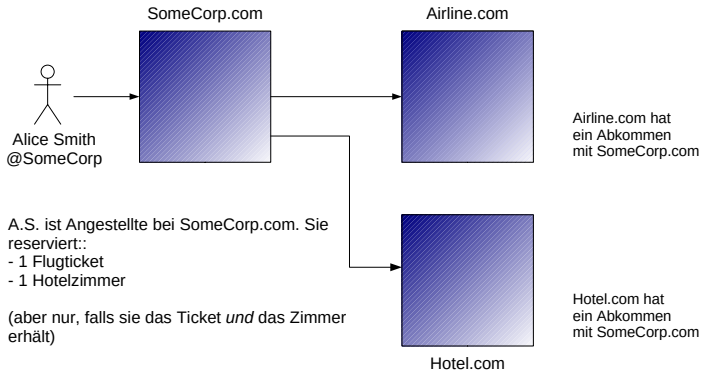
Was würde Alice sonst noch unterzeichnen?

- Die ganze Bestellung, damit:
  - Alice sicher sein kann, dass die Lieferung nur das umfasst, was bestellt wurde (Bücher, Quantität);
  - PrenticeHall.com sicher sein kann, dass nicht ein Dritter die Bestellung gemacht hat;
  - PrenticeHall.com eindeutig nachweisen kann, dass Alice die Bestellung getätigt hat.

Was wurde bisher erreicht?

- **Integrität** der Bestellung:
  - Signierung
- **Integrität** der Bezahlungsinformation
  - Signierung
- **Vertraulichkeit** der Kreditkarteninformation
  - Verschlüsselung
- **Vertraulichkeit** der Bestellung
  - die Bestellung ist noch nicht vertraulich
  - falls die Meldung über mehrere *hops* geht, so kann sie dort eingesehen werden
  - Meldung verschlüsseln (und Empfehlung gemäss [J. Rosenberg et.al.]: Verschlüsselte Meldung nochmals signieren)
- **Unleugbarkeit**:
  - PrenticeHall.com erhält signierte Bestellung

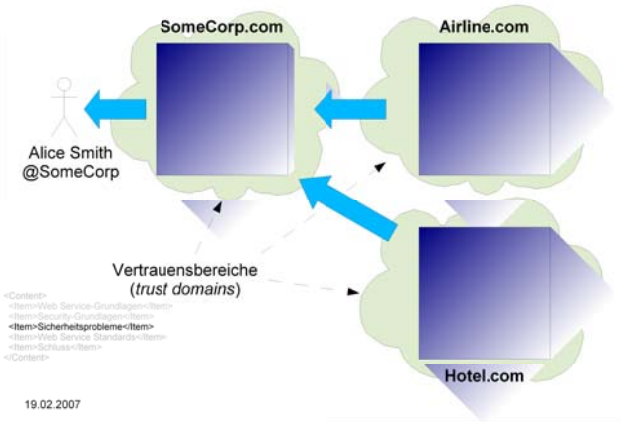
Ein anderes Szenario



A.S. ist Angestellte bei SomeCorp.com. Sie reserviert::
- 1 Flugticket
- 1 Hotelzimmer

(aber nur, falls sie das Ticket *und* das Zimmer erhält)

Etablierte Vertrauensbeziehungen



Etablierte Vertrauensbeziehungen (Forts.)

- SomeCorp.com vertraut Alice Smith:
  Nach erfolgter Authentifikation wird das Subjekt „Alice Smith“ mit der virtuellen Persönlichkeit alice@AD.SOMECORP.COM assoziiert.
- Airline.com besitzt den öffentlichen Schlüssel (oder das Zertifikat) von SomeCorp.com (hat aber kein a-priori-Vertrauen in Alice Smith).
  Die bilaterale Abmachung lautet, dass authentifizierte Mitarbeiter von SomeCorp.com Flugreservationen machen können.
- Hotel.com besitzt den öffentlichen Schlüssel (oder das Zertifikat) von SomeCorp.com (hat aber kein a-priori-Vertrauen in Alice Smith).
  Die bilaterale Abmachung lautet, dass authentifizierte Mitarbeiter von SomeCorp.com Zimmerreservationen machen können.

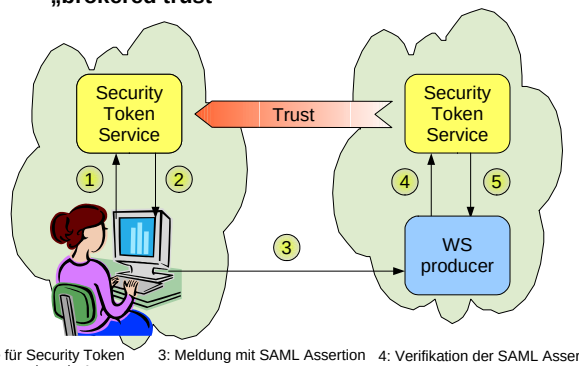
Authentifikation und Autorisation

Ansätze

- Alice Smith hat überall einen Account.
  - Probleme wie Skalierung, Passwörter, Kündigung, ...
- Eine globale Public Key-Infrastruktur (PKI) steht zur Verfügung und Alice Smith hat ein Zertifikat.
  (Und: Airline.com und Hotel.com wissen, dass Alice Smith Angestellte bei SomeCorp.com ist.)
  - „full“ PKI birgt eigene Probleme.
- Vertrauen wird „vermittelt“ (***brokered trust***):
  - SomeCorp.com stellt Security-Token aus
  - Jeweils Airline.com und Hotel.com prüfen die Security-Token und vertrauen den Angaben, die im Security-Token enthalten sind.

Der zuletzt aufgezählte Ansatz wird genauer vorgestellt.

Authentifikation und Autoristation auf der Basis „brokered trust“



1: Anfrage für Security Token
2: SAML Assertion als Antwort
3: Meldung mit SAML Assertion
4: Verifikation der SAML Assertion
5: Ok / Not Ok

SAML: Security Assertion Markup Language

Security Token Service

Merkmale:

- Implementiert die WS-Trust-Spezifikation
- Verarbeitet Issue-Anfragen mit:
  - Username/Password-Tokens
  - Kerberos-Tickets
  - X.509-Zertifikate
  - SAML-Assertions
- Generiert Issue-Antworten:
  - SAML-Assertions
- Verarbeitet Validate-Anfragen mit:
  - SAML-Assertions
- Generiert Validate-Antworten:
  - <status>ok</status>

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

SAML-Assertion

Was ist ein SAML-Assertion?

- eine Darlegung von Fakten betreffend einer „virtuellen Person“, gestützt auf einen Dritten

Welche Arten von Fakten gibt es?

- Fakten betreffend der Authentifikation (*authentication statement*)
- Fakten betreffend bestimmter Eigenschaften (Attribute) (*attribute statement*)
- Fakten betreffend bestimmter Autorisationen (*authorization statement*)

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Authentication Statement

Eine Ausstellungsautorität bezeugt, dass

- Subjekt S (z.B. Alice Smith) authentifiziert worden ist,
- durch den Vorgang M (z.B. Username/Passwort beim Active Directory)
- zum Zeitpunkt T

Kann verwendet werden für:

- Single Sign-On
- Überbrückung von verschiedenen Vertrauensbereichen

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Beispiel eines Authentication Statements

```
<saml:Assertion …>
  <saml:AuthenticationStatement
    AuthenticationMethod=“password” (By means M)
    AuthenticationInstant=“2001-12-03T10:02:00Z”>
                                        (At time T)
    <saml:Subject> (Subject S)
      <saml:NameIdentifier
        SecurityDomain=“AD.SOMECORP.COM”
        Name=“Alice Smith” />
      <saml:ConfirmationMethod>
        http://…core-25/sender-vouches
      </saml:ConfirmationMethod>
    </saml:Subject>
  </saml:AuthenticationStatement>
</saml:Assertion>
```

Empfänger muss existierende Vertrauens-beziehung haben

Die Assertion müsste noch unterzeichnet sein

Attribute Statement

Eine Ausstellungsautorität bezeugt, dass
- ein Subjekt S (z.B. Alice Smith)
- Attribute A, B, C (z.B. Kreditlimite), ... mit den
- Werten a1, b1, c1 (z.B. CHF 5000.00)

„besitzt".

Beispiel eines Attribute Statements

```
<saml:Assertion …>
  <saml:AttributeStatement>
    <saml:Subject>..Alice Smith..</saml:Subject>
    <saml:Attribute AttributeName="PaidStatus" (an attr.)
      AttributeNamespace="http://somecorp.com">
      <saml:AttributeValue> (with a value)
        PaidUp
      </saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute AttributeName="CreditLimit" (e.g., C)
      AttributeNamespace="http://somecorp.com">
      <saml:AttributeValue> (with a value)
        <ns:amount currency="CHF">5000.00</ns:amount>
      </saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

Die Assertion müsste noch unterzeichnet sein

Authorization Statement

Eine Ausstellungsautorität bescheinigt,
- die Anfrage des Subjekts S (z.B. Alice Smith)
- an die Ressource R (z.B. Flugreservation)
- für den Zugriffstyp A (z.B. Buchung)

ist erlaubt.

Beispiel eines Authorization Statements

```
<saml:Assertion …>
  <saml:AuthorizationStatement
    Decision="Permit" (Whether to grant request)
    Resource="http://airline.com/booking"> (for res. R)
    <saml:Subject>…</saml:Subject> (by Subject S)
    <saml:Actions
      ActionNamespace="http://…core-25/rwedc ">
      <saml:Action>Execute</saml:Action> (for acc. t. A)
    </saml:Actions>
  </saml:AuthorizationStatement>
</saml:Assertion>
```

Die Assertion müsste noch unterzeichnet sein

Was wurde mit SAML-Assertions erreicht?

Web Service-Aufrufe können z.B. mittels SAML-Assertions Informationen folgender Art besitzen:

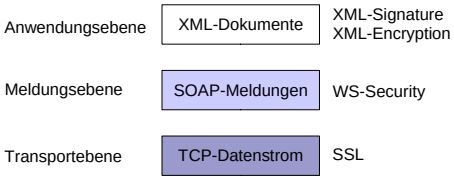
- Authentifikation
- Autorisation

Aber: Wie kann verhindert werden, dass Alice Smith nicht einfach zwei Buchungen durchführt?

```
<soapenv:Envelope ...>
  <soapenv:Header>
    <wsse:Security xmlns:wsse="...">
      <saml:Assertion ...>...</saml:Assertion>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body ...>
    <ns1:flight>...</ns1:flight>
  </soapenv:Body>
</soapenv:Envelope>
```

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Zur Erinnerung: Sicherheitsebenen bei Web Services



<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Aspekte, die noch geregelt werden müssen (1)

- Wo in SOAP-Meldungen sollen z.B. SAML-Assertions eingefügt werden?
  - Im Header einer SOAP-Meldung
  - Ein <wsse:Security>-Element ist zu verwenden. <Security>-Elemente können enthalten:
    - Security-Tokens wie SAML-Assertions, Username-Token, binäre Security-Tokens wie X.509-Zertifikate oder Kerberos-Tickets.
    - XML-Signatur-Elemente
    - XML-Verschlüsselungs-Elemente
  - Geregelt im WS-Security-Standard
- Wie soll die unberechtigte Zweitbuchung verhindert werden?
  - Kopplung des Body der Meldung mit der SAML-Assertion, z.B. mittels Digest.

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

Aspekte, die noch geregelt werden müssen (2)

- Wie kann sowohl die Buchung des Flugs wie des Zimmers sichergestellt werden?
  - Verwendung eines Transaktionsdienstes
- Wie weiss der Programmierer des WS-Consumers von SomeCorp.com, dass für die Buchung eines Flugs ein SAML-Assertion notwendig ist?
  - Airline.com assoziiert zum Buchungsdienst eine WS-Policy

<Content>
<Item>Web Service-Grundlagen</Item>
<Item>Security-Grundlagen</Item>
<Item>Sicherheitsprobleme</Item>
<Item>Web Service Standards</Item>
<Item>Schluss</Item>
</Content>

WS-Security-Spezifikation

- Eine Menge von SOAP-Erweiterungen für die End-zu-End SOAP-Meldungssicherheit
  - Security-Schemas auf Ebene der Meldungen
- Unterzeichnung und Verschlüsselung von SOAP-Meldungen und der Einbettung der entsprechenden Security-Elementen in die SOAP-Meldungen
  - Beliebige Kombination von Teilen einer Meldung wie Header, Body, Attachments, etc.

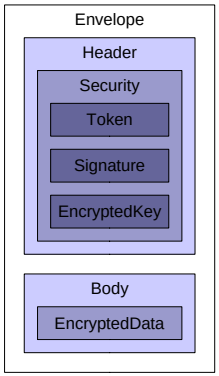
WS-Security-Mechanismen

WS-Security definiert, wie SOAP-Meldungen mit Security-Tokens, XML-Encryption und XML-Signature gesichert werden

Security-Tokens dienen der Authentifizierung oder Autorisierung:

- Username/Passwort
- X.509-Zertifikat
- Kerberos-Ticket
- SAML-Assertion
- XrML (Extensible Rights Markup Language)
- XCBF (XML Common Biometric Format)

Aufbau einer gesicherten SOAP-Meldung



Wie werden die Sicherheitsanforderungen eines Web Services publiziert?

- **Web Services Policy Framework** (WS-Policy): Modell zur Beschreibung der (Sicherheits-) Anforderungen (*policies*) eines Web Services.
- **Web Services Policy Attachment** (WS-Policy Attachment) Definiert u.a., wie obige (Sicherheits-) Anforderungen mit einer Web Service-Spezifikation (WSDL) verknüpft wird.

WS-Policy-Beispiel

Die Policy zur Durchführung einer Buchung könnte lauten:

```
<wsp:Policy ...>
  <wsp:ExactlyOne>
    <wsp:All ...>
      <!-- Alle hier gelisteten policy statements
           müssen erfüllt sein. -->
    </wsp:All>
    <wsp:All ...>
      <wsse:SecurityToken TokenType="wsse:SAML20"/>
      <wssw:Algorithm Type="wsse:AlgEncryption"
        URI="..xmlenc#aes-cbc"/>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

„All" bedeutet...

„ExactlyOne" bedeutet: „ODER"

WS-Policy-Beispiel (Fortsetzung)

Die **WS-Policy Attachment**-Spezifikation definiert, wie eine WS-Policy mit einem Web Service assoziiert wird. Es gibt zwei Möglichkeiten:

- Einfügen einer Referenz auf ein WS-Policy-Dokument innerhalb eines WSDL-Dokuments, oder das Einfügen der ganzen WS-Policy im WSDL-Dokument.
- Eigenständiges Dokument (*arbitrary resource attachment*):

```
<wsp:PolicyAttachment>
  <wsp:AppliesTo>
    <wsp:EndpointReference xmlns:ns1="...">
      <wsp:ServiceName Name="ns1:BookingService"/>
      <wsp:PortType Name="ns1:BookingPortType"/>
      <wsp:Address
          URI="http://server.airline.com/booking"/>
    </wsp:EndpointReference>
  </wsp:AppliesTo>
</wsp:PolicyAttachment>
```


Web Services Standards im Bereich Security

Standardorganisationen

W3C (www.w3.org)
- XML, SOAP, WSDL
- XML-Signature, XML-Encryption

OASIS (www.oasis-open.org)
- UDDI, ebXML (Electronic Business using XML)
- SAML, WSS (Web Services Security)

WS-I (www.ws-i.org)
- Interoperability Profile

ITU (www.itu.int)
- X509, ASN.1

Auszug aus der WS-*-Familie

Interoperabilität	Trust	Integration
WS-Conversation	WS-Privacy	WS-Federation
WS-Policy	WS-Trust	WS-Authorization
XKMS	WS-Security	XACML
XML-Signature	XML-Encryption	SAML


```
<Content>
 <Item>Web Service-Grundlagen</Item>
 <Item>Security-Grundlagen</Item>
 <Item>Sicherheitsprobleme</Item>
 <Item>Web Service Standards</Item>
 <Item>Schluss</Item>
</Content>
```


Detaillierte Liste befindet sich in der Beilage

Schluss

- HTTPS genügt nicht in allen Fällen
- XML-Dokumentsicherheit ist oft nötig (XML-Verschlüsselung, XML-Signierung)
- Authentifikation und Autorisation:
  - unterschiedliche Vertrauensbereiche (*trust domains*)
  - „*brokered trust*“ mittels Security Token Service
  - Sicherung der SOAP-Meldungen: WS-Security
- Aber: Weitere Standards etablieren sich wie
  - WS-Conversation
  - WS-Federation


```
<Content>
 <Item>Web Service-Grundlagen</Item>
 <Item>Security-Grundlagen</Item>
 <Item>Sicherheitsprobleme</Item>
 <Item>Web Service Standards</Item>
 <Item>Schluss</Item>
</Content>
```


Besten Dank

Für Fragen stehen wir Ihnen gerne zur Verfügung.