Auszug aus der Schulung Web Services Sicherheit

Dieses Dokument ist ein Auszug aus unserem Skript zur Schulung Web Services Sicherheit. Es dient als Beispiel für unsere Kursunterlagen.

Mehr zum Kurs finden Sie unter:
http://www.thomas-bayer.com/web-services-security-schulung.htm

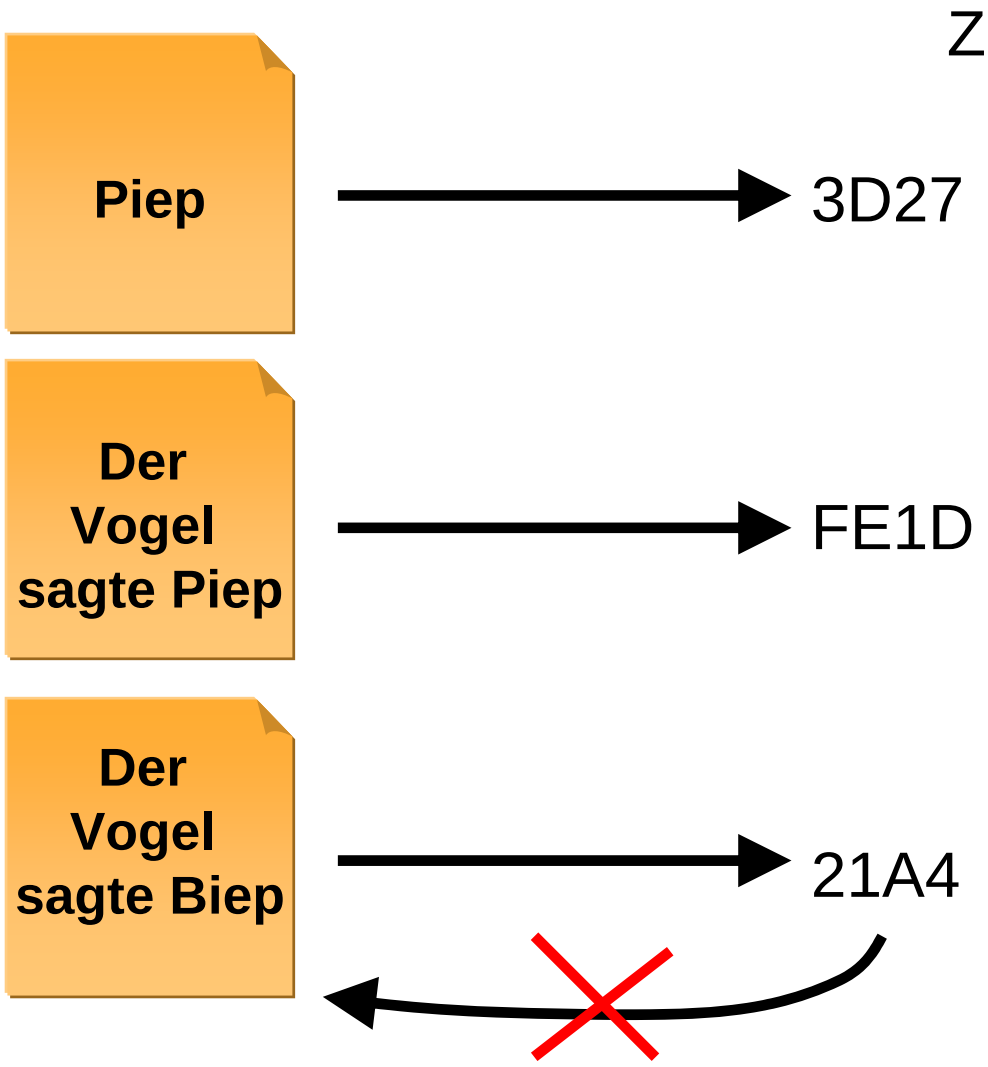
Thomas Bayer

Nikolausstraße 107
50937 Köln

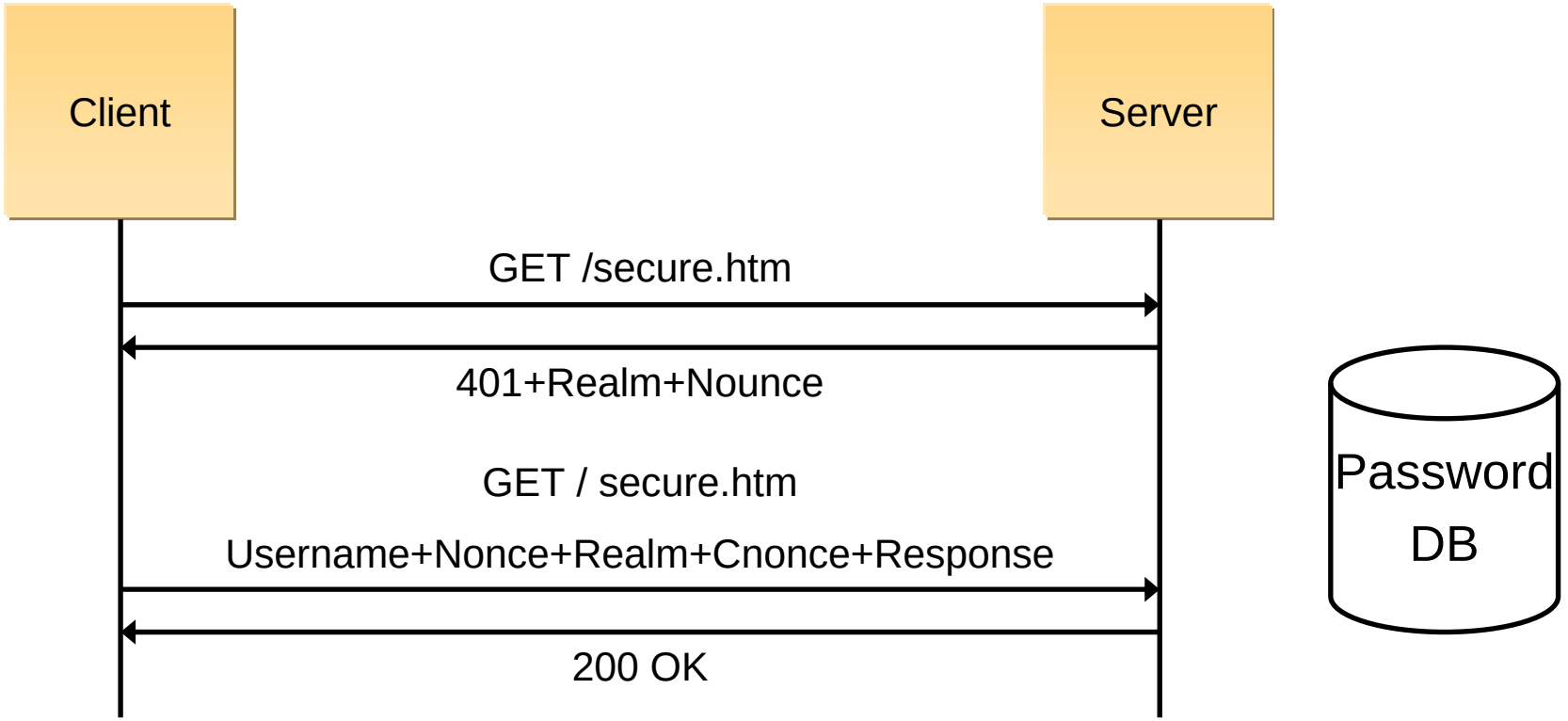
Tel. : +49 (221) 4249250
www.thomas-bayer.de
info@thomas-bayer.de

Version: 1.0

Hash Funktion



HTTP Digest Access Authentication



HA1 = md5 (username+Realm+Password)

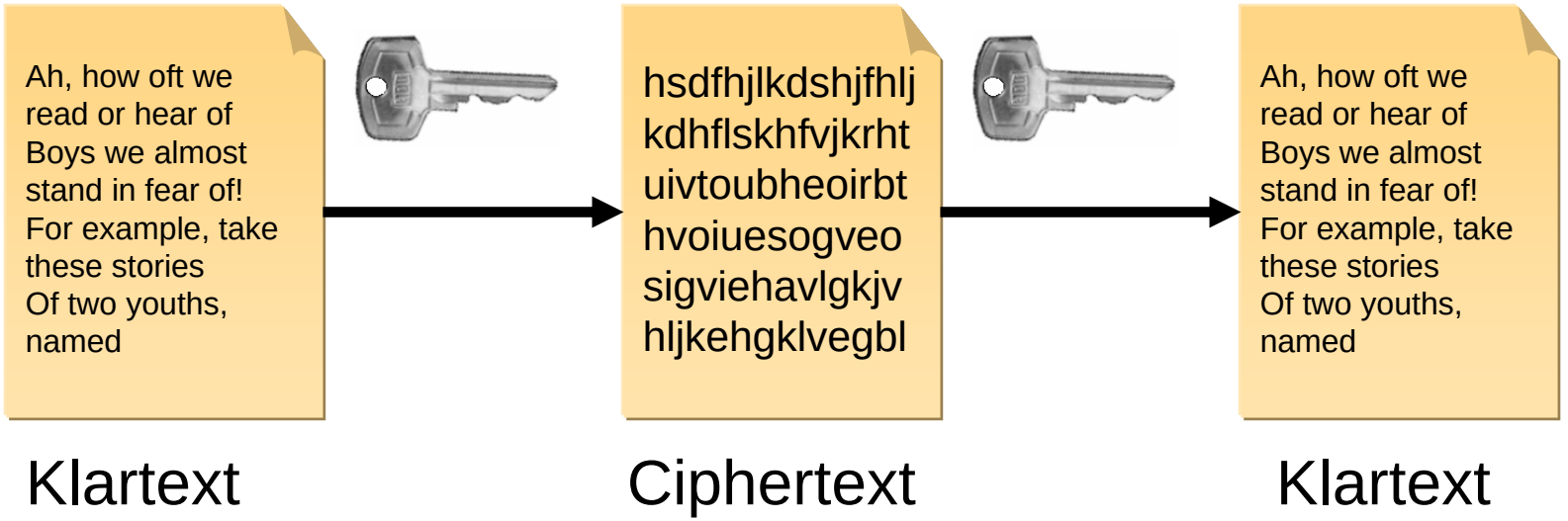
HA2 = md5 (GET+/secure-htm)

Response = md5(HA1+nonce+rquestcounter+cnonce+qop+HA2)

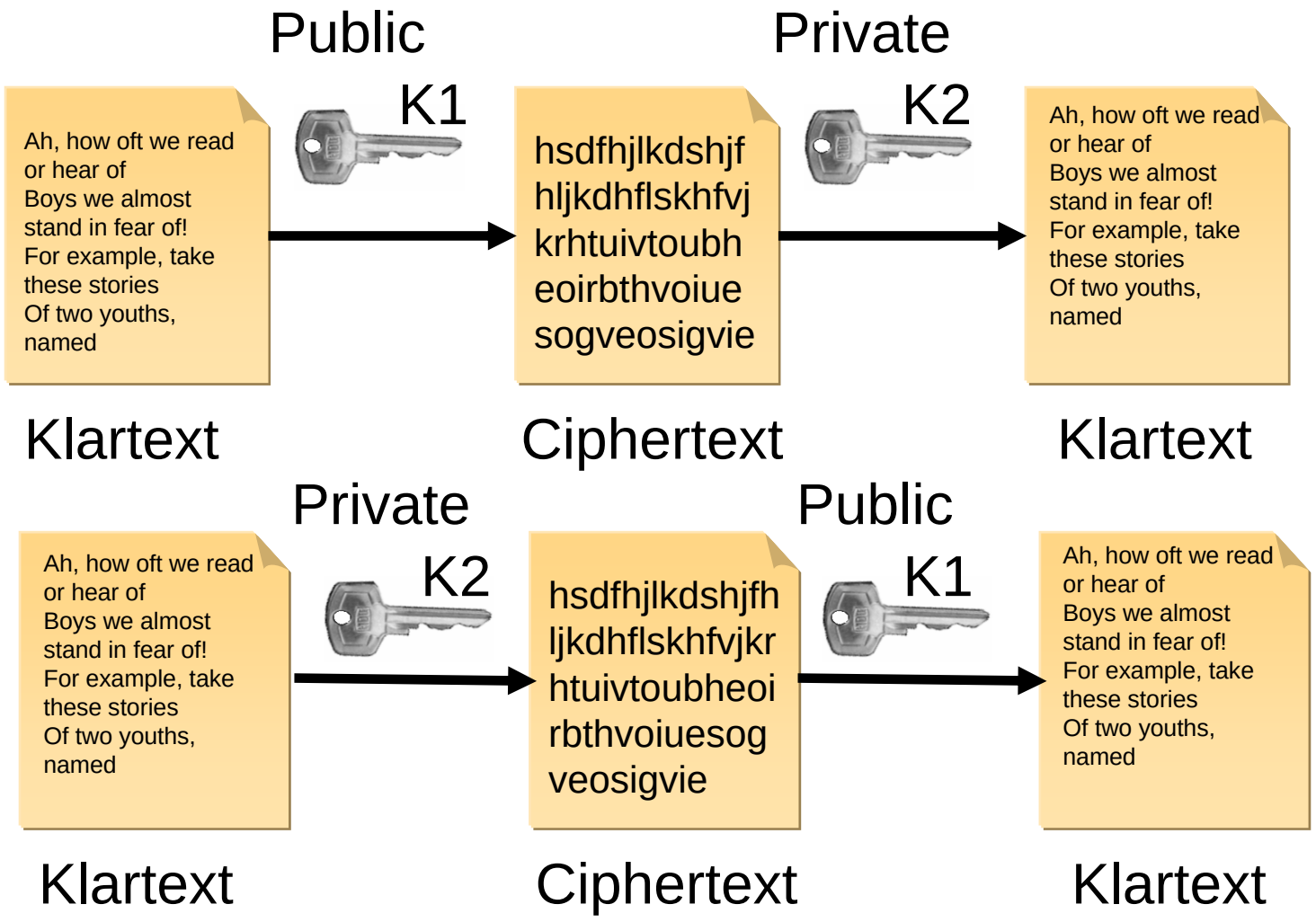
Cnounce = ClientNounce

qop = Quality of Protection

Symmetrische Schlüssel



Asymmetrische Schlüsel



Digital Signature Algorithm (DSA)

- US Regierungsstandard
- Erfunden von David W. Kravitz einem ehemaligen NSA Mitarbeiter
- Royalty free
- Basiert auf Primzahlen
- Arbeitet mit Public/Private Schlüsselpaar

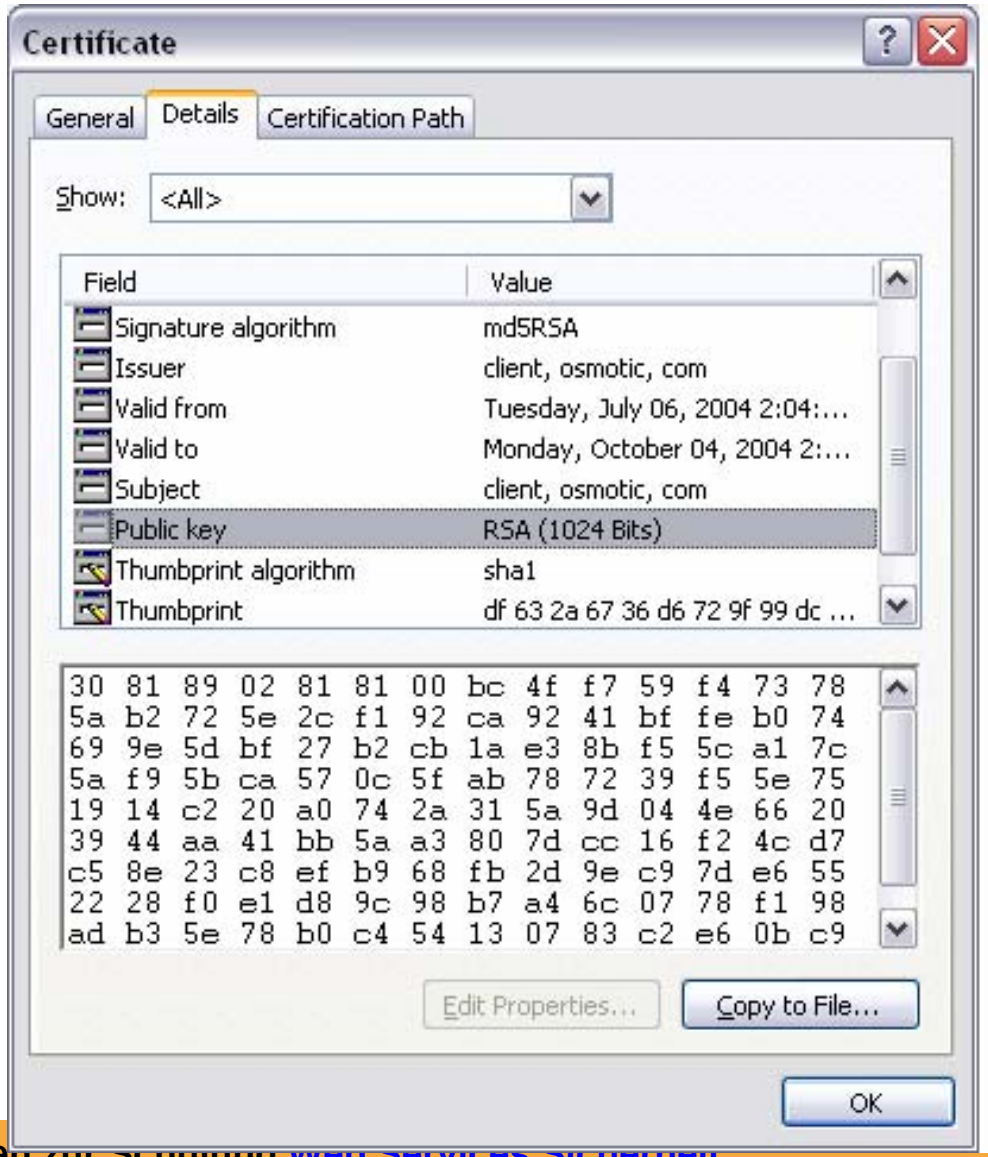
Blowfish

- Symmetrischer Block Chiper Algorithmus
- 1993 von Bruce Schneier
- Weit verbreitet, wird aber immer mehr von Twofish oder AES abgelöst
- Public Domain
- Schlüssellänge 32-448 Bits
- Einer der schnellsten Algorithmen bis auf den Wechsel von Schlüsseln
  - PC -> Geeignet für Passwörter
- Großer Footprint von ca. 4KByte
- Sicher bei kurzen Klartexten

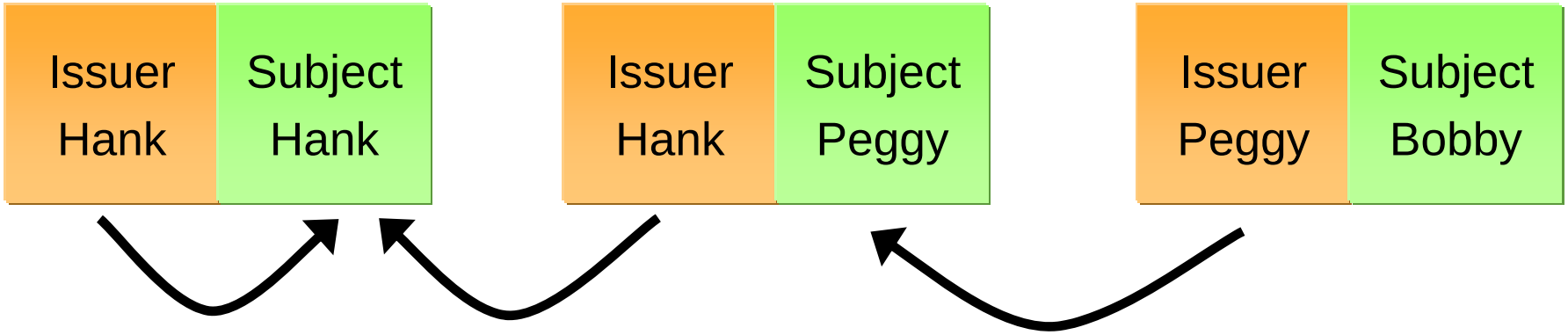
Inhalt X.509 Zertifikat

- Version
  - Version des Standards V1, V2 oder V3
- Serial Number
- Signature Algorithm Identifier
- Issuer Name
  - X.500 Distinguished Name
- Validity Period
- Subject Name
- Subject Public Key Information
  - Wert
  - Algorithm Identifier
  - Parameter

Certificate Details



Kette des Vertrauens



Struktur Keystore

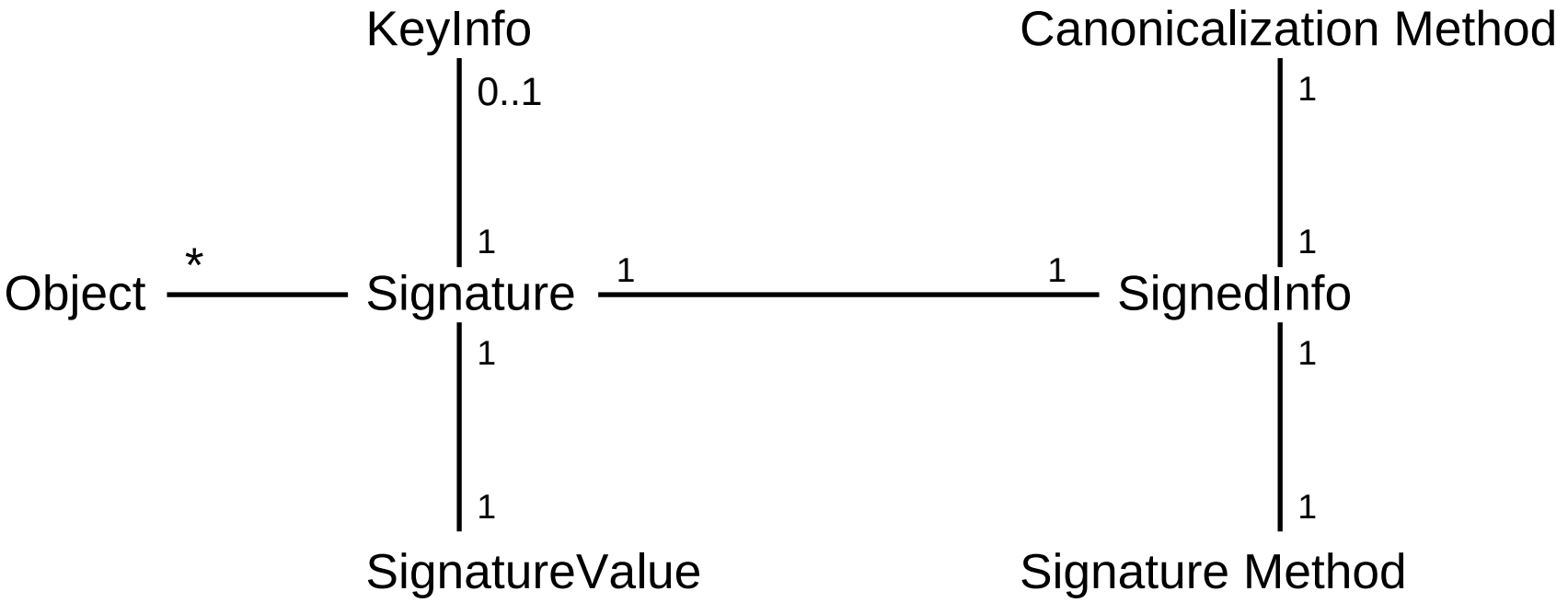
Erzeugen der Keystores

```
keytool -genkey -alias server -keyalg RSA
        -dname "CN=server, O=osmotic, C=com" -keypass simple
        -storepass simple -keystore server-store -storetype JKS

keytool -export -alias server -file server.cer
        -keystore server-store -storepass simple



keytool -import -noprompt -alias client -file client.cer
        -keystore server-store -storepass simple

keytool -import -noprompt -alias server -file server.cer
        -keystore client-store -storepass simple
```

Configuration in web.xml

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>AxisServlet</web-resource-name>
        <url-pattern>/services/BusTour</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>webservice</role-name>
    </auth-constraint>
</security-constraint>

<login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>AxisServlet</realm-name>
</login-config>
<security-role>
    <description>the webservice role</description>
    <role-name>webservice</role-name>
</security-role>
```

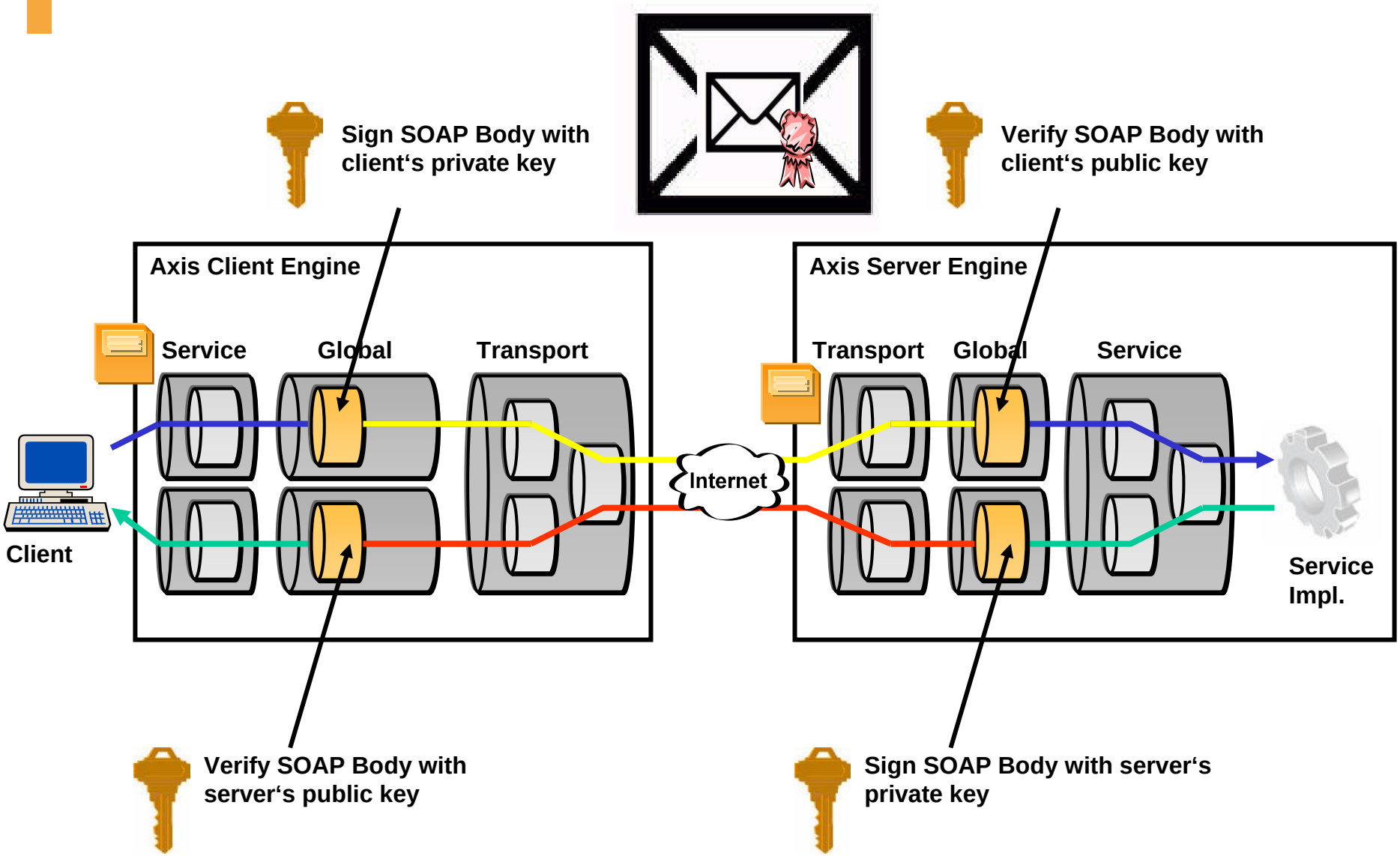
Struktur Signature Dokument



WS-I Basic Security Profile

- Draft (June 2004)
- Focuses on
  - HTTP over TLS
  - OASIS Web Services Security 1.0
    - **Username Token Profile**
    - **X.509 Certificate Token Profile**
    - **Planning: Kerberos Token Profile**
    - **Considering: SAML Token Profile and XRML Token Profile**

Transparent Signing and Encryption



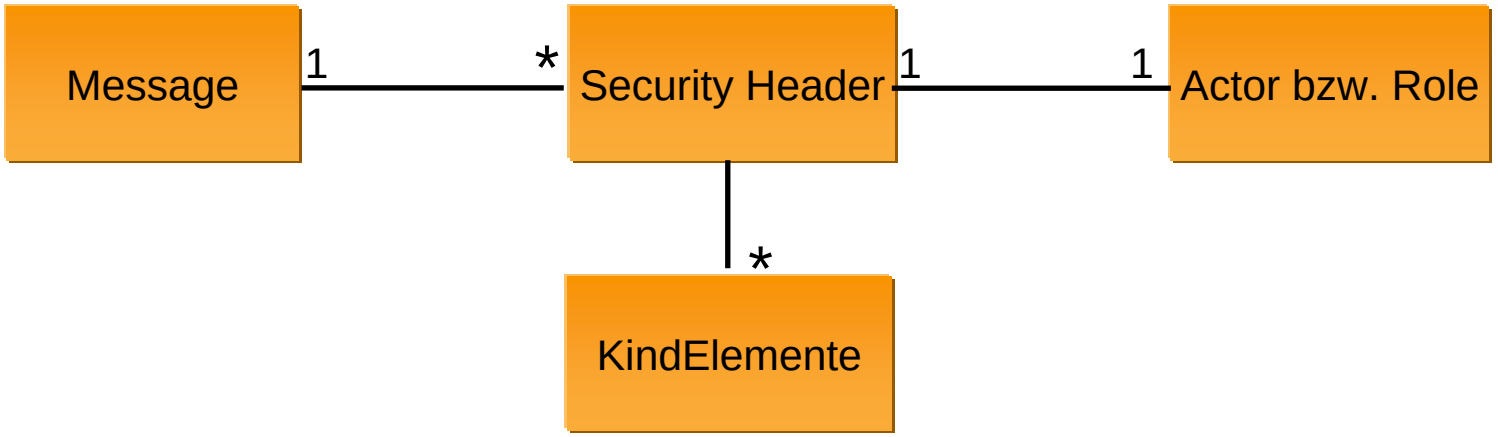
Signed Request (Abridged)

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
 xmlns:SOAP-SEC="http://schemas.xmlsoap.org/soap/security/2000-12“ ...>
 <soapenv:Header>
  <SOAP-SEC:Signature>
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
     <ds:CanonicalizationMethod
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
     <ds:SignatureMethod
      Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
     <ds:Reference URI="#Body">
      <ds:DigestMethod
       Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>2jmj7l5rSw0yVb/vlWAYkK/YBwk=</ds:DigestValue>
     </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>xlYchckeGheFa7IJioYbn9KSHg==</ds:SignatureValue>
    <ds:KeyInfo>[ X509 Certificate]</ds:KeyInfo>
   </ds:Signature>
  </SOAP-SEC:Signature>
 </soapenv:Header>
 <soapenv:Body>[Cut out]</soapenv:Body>
</soapenv:Envelope>
```

Security Header

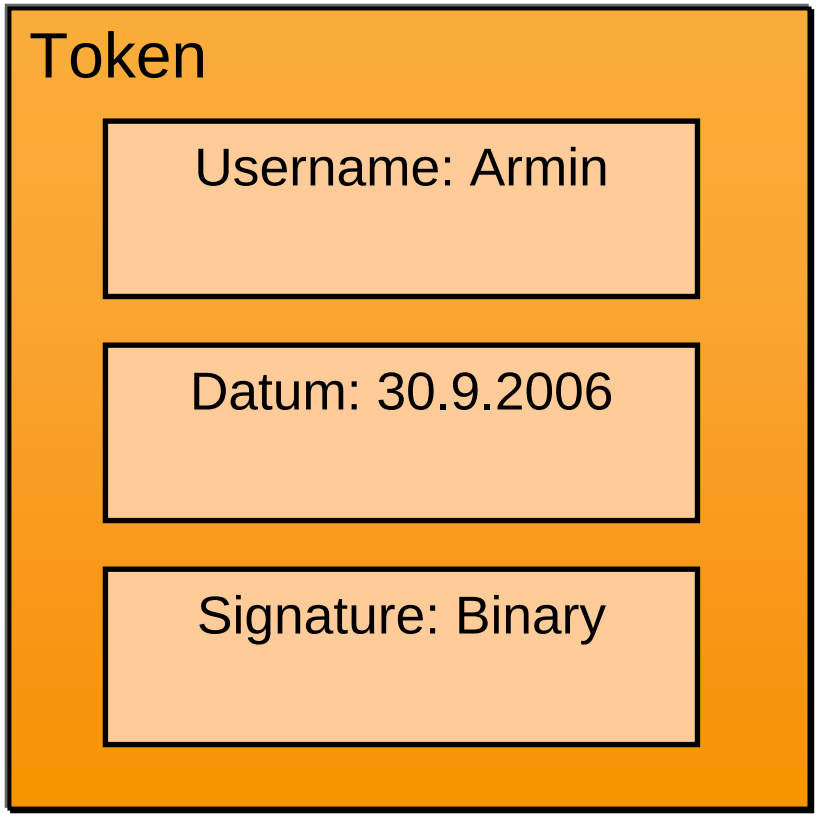
- Beschreibt die Verschlüsselungs- und Signierungsschritte, die der Sender angewandt hat.
- Kind Elemente sind geordnet.

```
<S11:Header>
    ...
    <wsse:Security S11:actor="..." S11:mustUnderstand="...">
        ...
    </wsse:Security>
    ...
</S11:Header>
```

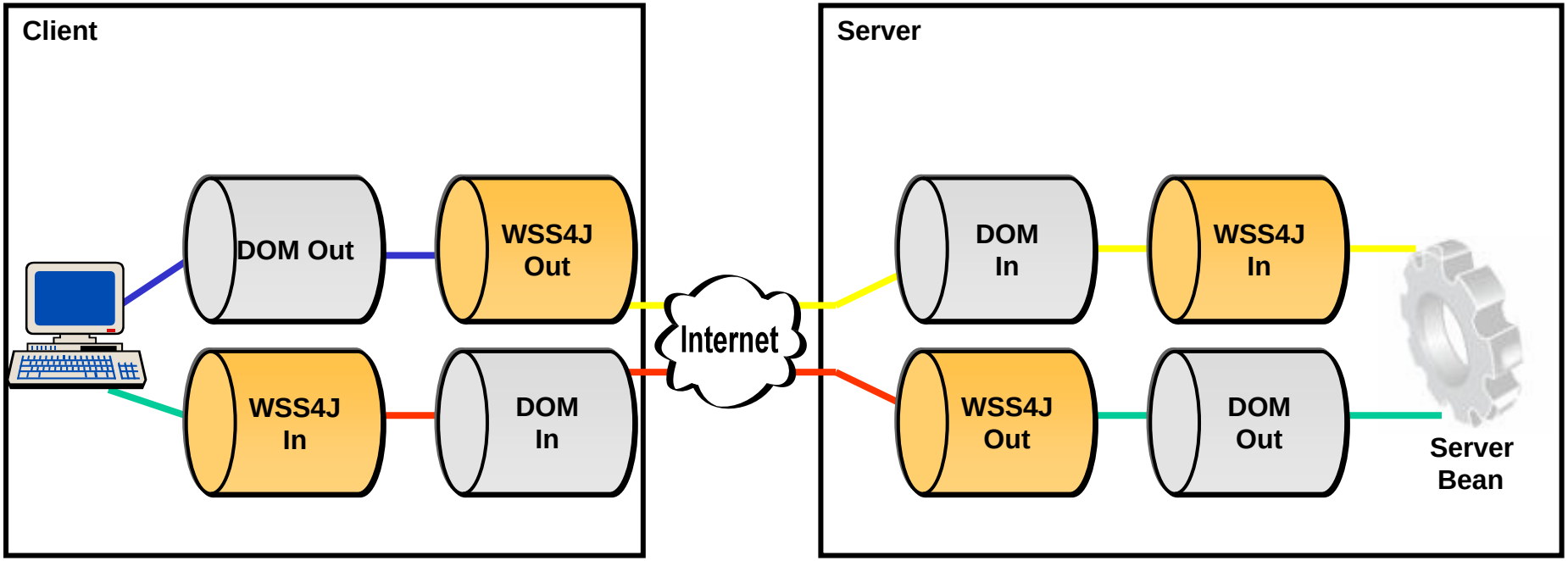


Token

- Ein oder mehrere Claims
- Kann signiert und verschlüsselt werden
- z.B. X509 Zertifikat (Signed Token)



WSS Handler in XFire



Request

Request with signed body

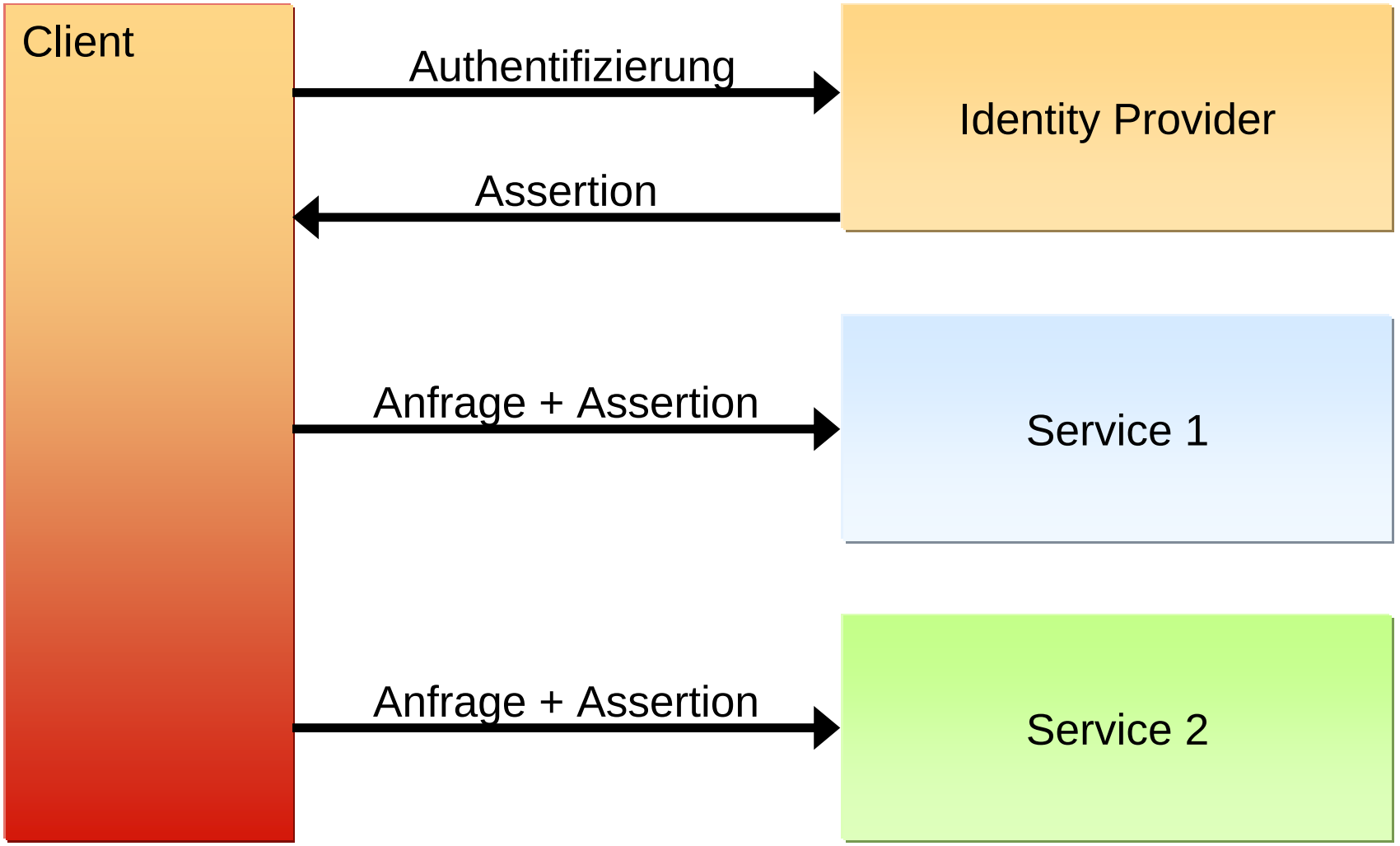
Response

Response with signed body

Apache WSS4J

- Java Bibliothek für WSS
  - SOAP Message Security 1.0 200401
  - Username Token Profile V. 1.0
  - X.509 Token Profile V 1.0
- Wird verwendet von:
  - Axis
  - XFire

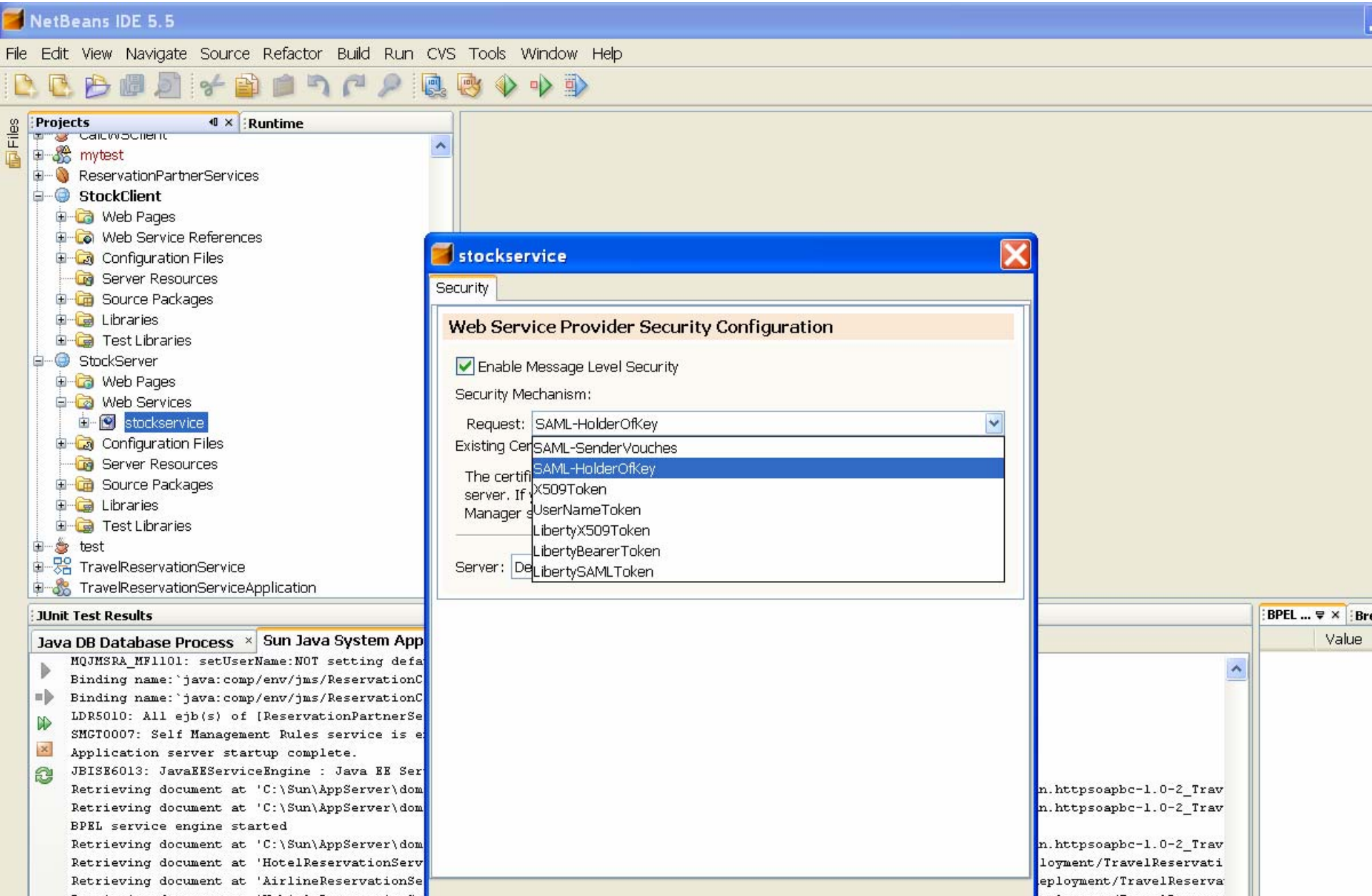
SSO



Assertion

- Aussage einer Authorität (Identity Provider) über einen Principal
- Inhalt
  - Herausgeber, Signatur, Subjekt, Bedingungen, Statements
- SAML Statements
  - Authentication
  - Attribute
  - Authorization Decision
- Wird von Identity zum Service Provider gesendet
- Können in WSS „eingewickelt„ werden

SAML Konfiguration in NetBeans



SAML Assertion (vereinfacht)

```
<env:Header>
  <wsse:Security>
    <saml:Assertion AssertionID="sfad86e51"
     IssueInstant="2007-01-15T16:47:19Z"
     Issuer="localhost">
      <saml:AuthenticationStatement
       AuthenticationInstant="2007-01-15T16:42:24Z"
       AuthenticationMethod="urn:com:sun:identity:Application">
        <saml:Subject>
          <saml:NameIdentifier>stockservice</saml:NameIdentifier>
          <saml:SubjectConfirmation>
            <saml:ConfirmationMethod>...:holder-of-key</saml:Confi
          </saml:SubjectConfirmation>
        </saml:Subject>
    </saml:AuthenticationStatement>
    <Signature><Reference URI="#sfad8 “/></Signature>
    </saml:Assertion>
  <wsu:TimeStamp>...</wsu:TimeStamp>
  </wsse:Security>
</env:Header>
```

SAML 2.0

- Wurde durch Funktionen von Liberty ID-FF und Shibboleth beeinflußt
- Pseudonyme für Privacy
- Identifier Management (für Pseudonyme)
- Metadata
- Encryption
- Attribute Profile
- Session Management
- Unterstützung für mobile Geräte
- Identity Provider Discovery