Sicherheitsaspekte in Service Orientierten Architekturen

Eike Falkenberg

Sommersemester 2006 Anwendungen I

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Agenda

- SOA?
- Web Services?
- Sicherheitsrisiko Web Services
- Web Services & Sicherheit
- Sichere SOAs durch sichere Web Services?

- SOA?
- Web Services?
- Sicherheitsrisiko Web Services
- Web Services & Sicherheit
- Sichere SOAs durch sichere Web Services?

Was ist eigentlich SOA?

Die Welt ohne SOA:

- Anwendungs Monolithen
- Durch permanente Weiterentwicklung totgepflegte Anwendungen
- Integrierung durch proprietäre Protokolle, Kommunikation durch schreiben in Dateien
- Änderungen in Geschäftsprozessen sind nur schwer umzusetzen

Was ist eigentlich SOA?

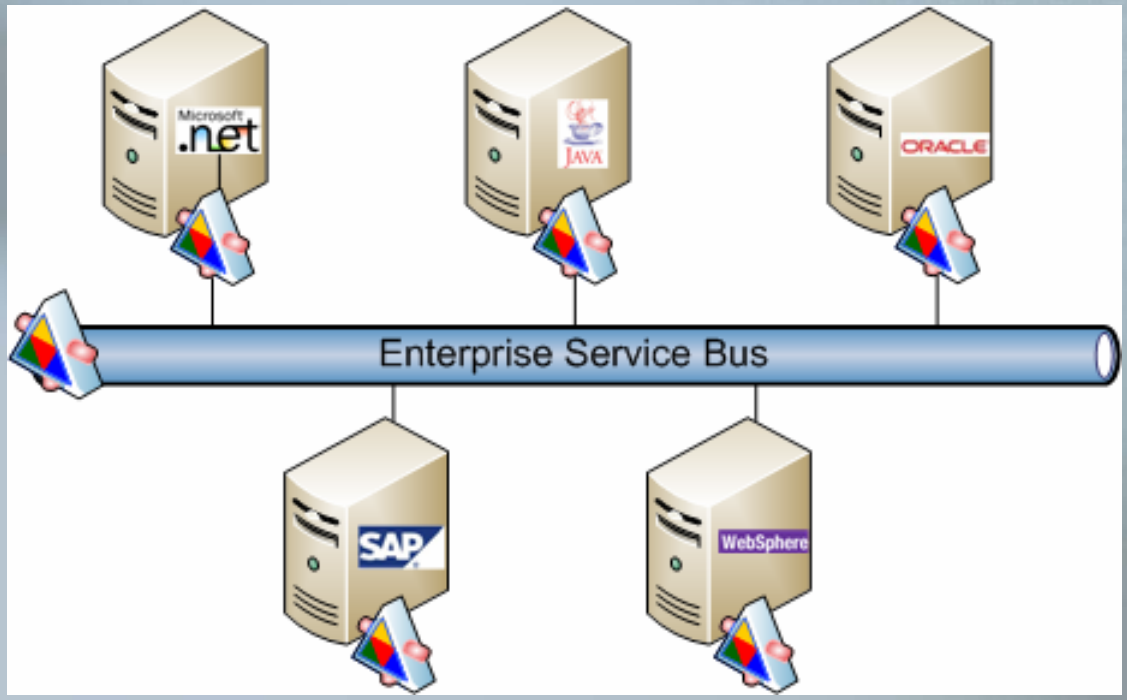
- Architekturpattern
- Fachspezifische Anwendungsteile werden in wiederverwendbaren Services gekapselt
- Lose Kopplung
- Einheitlicher Zugriff
- Anbindung von Legacy Systemen
- Technische Grundlage meist Web Services (sonst MOM)

Services

- Kapselung von Logik und Komplexität fachspezifischer Anwendungsteile
- Wiederverwendbar
- Flexible Abbildung von Geschäftsprozessen durch Process Choreography (PC)
- Services werden über einen Enterprise Service Bus (ESB) angesprochen

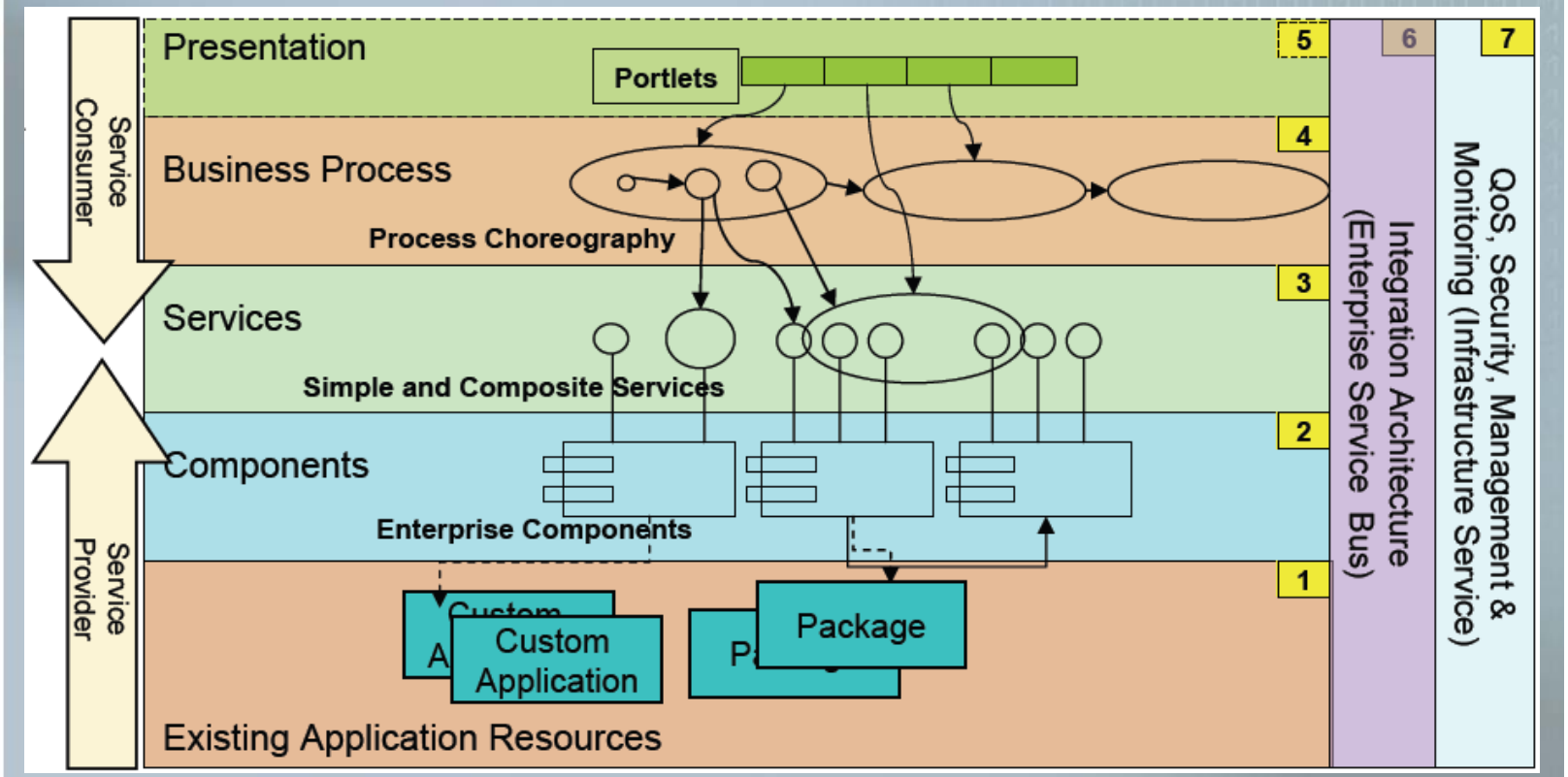
Was ist eigentlich SOA?

Enterprise Service Bus



Was ist eigentlich SOA?

Übersicht

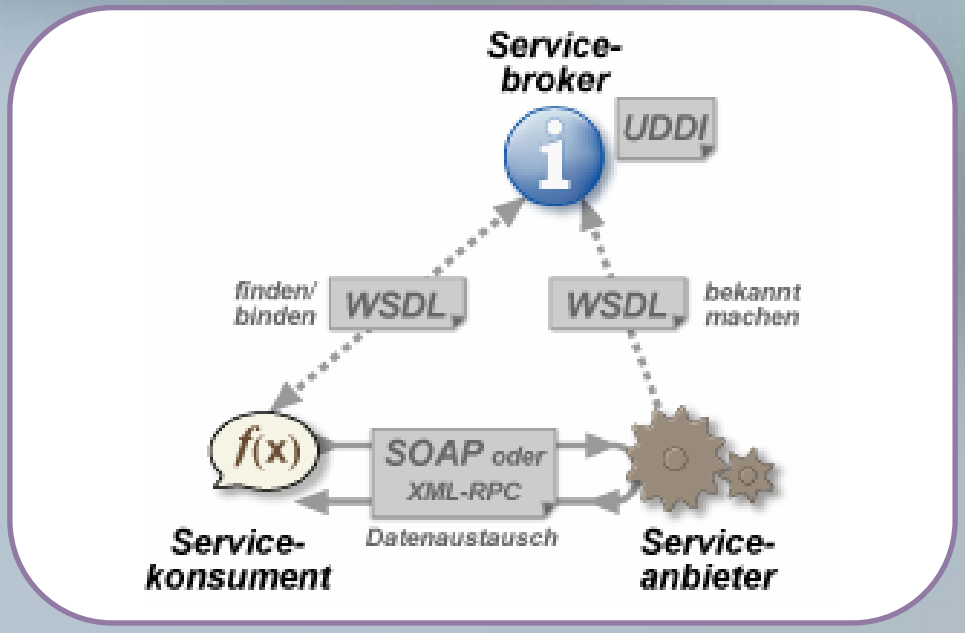


(aus:Olaf Zimmermann "Jumpstarting SOA Projects")

- SOA?
- Web Services?
- Sicherheitsrisiko Web Service
- Web Service Sicherheit
- Sichere SOAs durch sichere Web Services?

Was sind Web Services?

- Interoperable- und plattformunabhängige Kommunikation mittels XML Nachrichten (meist) via HTTP



- Was ist eigentlich SOA?
- Was sind eigentlich Web Services?
- Sicherheitsrisiko Web Service
- Web Service Sicherheit
- Sichere SOAs durch sichere Web Services?

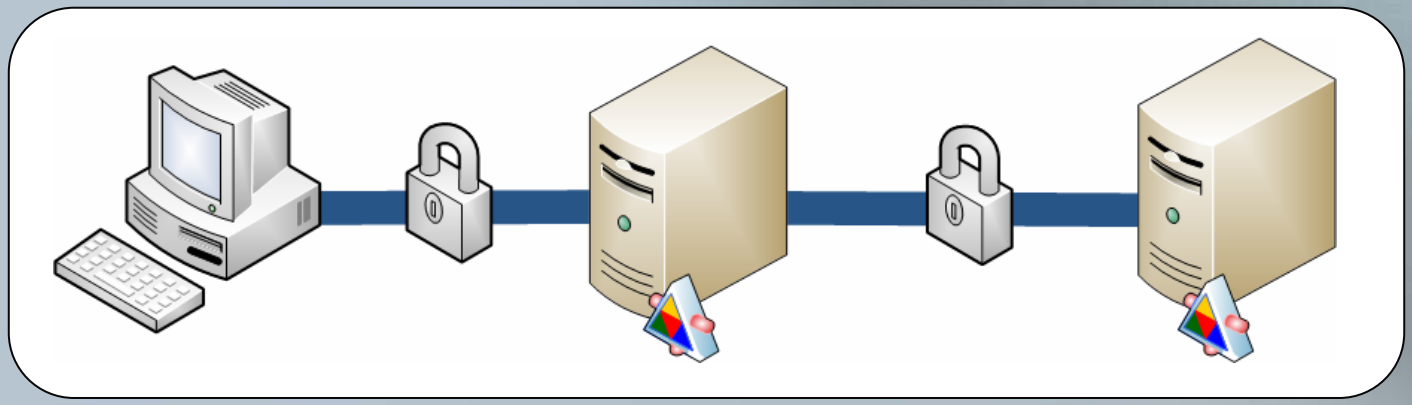
Sicherheitsrisiko Webservices

- Webservices sind zu offen
- Firewalls helfen kaum
- Integrität der Nachrichten kann nicht gewährleistet werden
- Authentifizierung und Autorisierung nicht einheitlich und somit Fehleranfällig

- SOA?
- Web Services?
- Sicherheitsrisiko Web Services
- Web Services & Sicherheit
- Sichere SOAs durch sichere Web Services?

Web Service Sicherheit

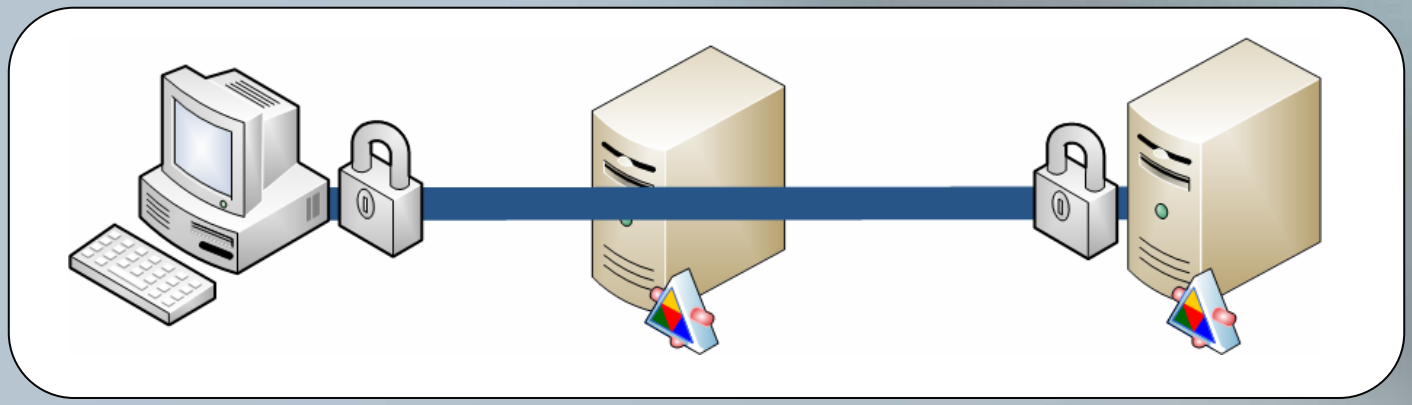
- Warum nicht einfach **SSL**?
  - fehlende Autorisierung
  - verschlüsselt komplette Nachricht
  - Point-to-Point unzureichend
  - Unsicherheitsfaktor Intermediaries



Web Service Sicherheit

WS-Security

- SOAP Header für
  - Autorisation
  - Verschlüsselung
  - Signatur



Web Service Sicherheit

WS-Security UsernameToken

```
<xs:element name="UsernameToken">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Username"/>
            <xs:element ref="Password" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

```
<wsse:UsernameToken>
    <wsse:Username>Eike</wsse:Username>
    <wsse:Password Type="wsse:PasswordText">Kennwort</wsse:Password>
</wsse:UsernameToken>
```

```
<wsse:UsernameToken>
    <wsse:Username>Eike</wsse:Username>
    <wsse:Password Type="wsse:PasswordDigest">
        KE6QugOpkPyT3EoOSEgT3OW4Keg=</wsse:Password>
    <wsse:Nonce>5uW4ABku/m6/S5rnE+L7vg==</wsse:Nonce>
    <wsu:Created xmlns:wsu=
        "http://schemas.xmlsoap.org/ws/2002/07/utility">
            2006-06-04T08:44:02Z
    </wsu:Created>
</wsse:UsernameToken>
```

Web Service Sicherheit

WS-Security BinarySecurityToken (X.509 Zertifikat, Kerberos Ticket)

```
<xs:element name="BinarySecurityToken">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="Id" type="xs:ID" />
                <xs:attribute name="ValueType" type="xs:QName" />
                <xs:attribute name="EncodingType" type="xs:QName" />
                <xs:anyAttribute namespace="##other"
                    processContents="strict" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
```

```
<wsse:BinarySecurityToken
    ValueType="wsse:X509v3"
    EncodingType="wsse:Base64Binary"
    Id="SecurityToken-f49bd662-59a0-401a-ab23-1aa12764184f"
>MIIHdjCCB...</wsse:BinarySecurityToken>
```

Web Service Sicherheit

WS-Security Signaturen

- XML Signature (XMLDsig)
- Integrität der Nachrichten
  - Nachricht wurde nicht verändert
- Authentifikation
  - Nachricht stammt wirklich von der angegeben Entität

Signierung mit Privatem Schlüssel (X.509), Sitzungsschlüssel (Kerberos) oder Passwort (UsernameToken)

Web Service Sicherheit

```
<wsse:Security
    xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/07/secext">

  <wsse:BinarySecurityToken id="Kerberosv5ST"
                            wsse:ValueType="wsse:Kerberosv5ST"
                            wsse:EncodingType="wsse:Base64Binary">
    MIIEZzCCA9CgAwIBAgIQEmtJZcOrqrKh5i...
  </wsse:BinarySecurityToken>

  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
      <ds:SignatureMethod Algorithm="...#DES-MD5"/>
      <ds:Reference>
        <ds:Transforms>
          <ds:Transform>Algorithm="http://...#MyTranform"/>
          <ds:Transform>Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="...#DES-MD5"/>
        <ds:DigestValue>AhdsgHu...</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>BL8jdfToEb11vXcMZNNjPOV...</ds:SignatureValue>
    <ds:KeyInfo>
      <wsse:SecurityTokenReference>
        <wsse:Reference URI="#Kerberosv5ST"/>
      </wsse:SecurityTokenReference>
    </ds:KeyInfo>
  </ds:Signature>

</wsse:Security>
```

Web Service Sicherheit

WS-Security Verschlüsselung

- Verhindert unerlaubten Zugriff auf Informationen
- XML Encryption

Web Service Sicherheit

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope  xmlns:soap=".." xmlns:xenc="..">
    <soap:Header xmlns:wsse=".." xmlns:wsu="..">
        <wsu:Timestamp>
            <wsu:Created wsu:Id="...">2006-06-04T09:26:15Z</wsu:Created>
            <wsu:Expires wsu:Id="...">2006-06-04T10:31:15Z</wsu:Expires>
        </wsu:Timestamp>
        <wsse:Security soap:mustUnderstand="1" >
            <xenc:ReferenceList>
                <xenc:DataReference URI="#EncryptedContent-f6f50b24-3458-41d3-aac4-390f476f2e51" />
            </xenc:ReferenceList>
        </wsse:Security>
    </soap:Header>
    <soap:Body>
        <xenc:EncryptedData
            Id="EncryptedContent-f6f50b24-3458-41d3-aac4-390f476f2e51"
            Type="http://www.w3.org/2001/04/xmlenc#Content">
            <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
            <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
                <KeyName>Symmetrischer Schlüssel</KeyName>
            </KeyInfo>
            <xenc:CipherData>
                <xenc:CipherValue>InmSSXQcBV5UiT...  Y7RVZQqnPpZYMg==</xenc:CipherValue>
            </xenc:CipherData>
        </xenc:EncryptedData>
    </soap:Body>
</soap:Envelope>
```

Triple DES

Verschlüsselte Daten

Web Service Sicherheit

WS-Security 1.1 (WS-*)

- WS-Trust
- WS-Secure-Conversation
- WS-SecurityPolicy
- (WS-Federation)

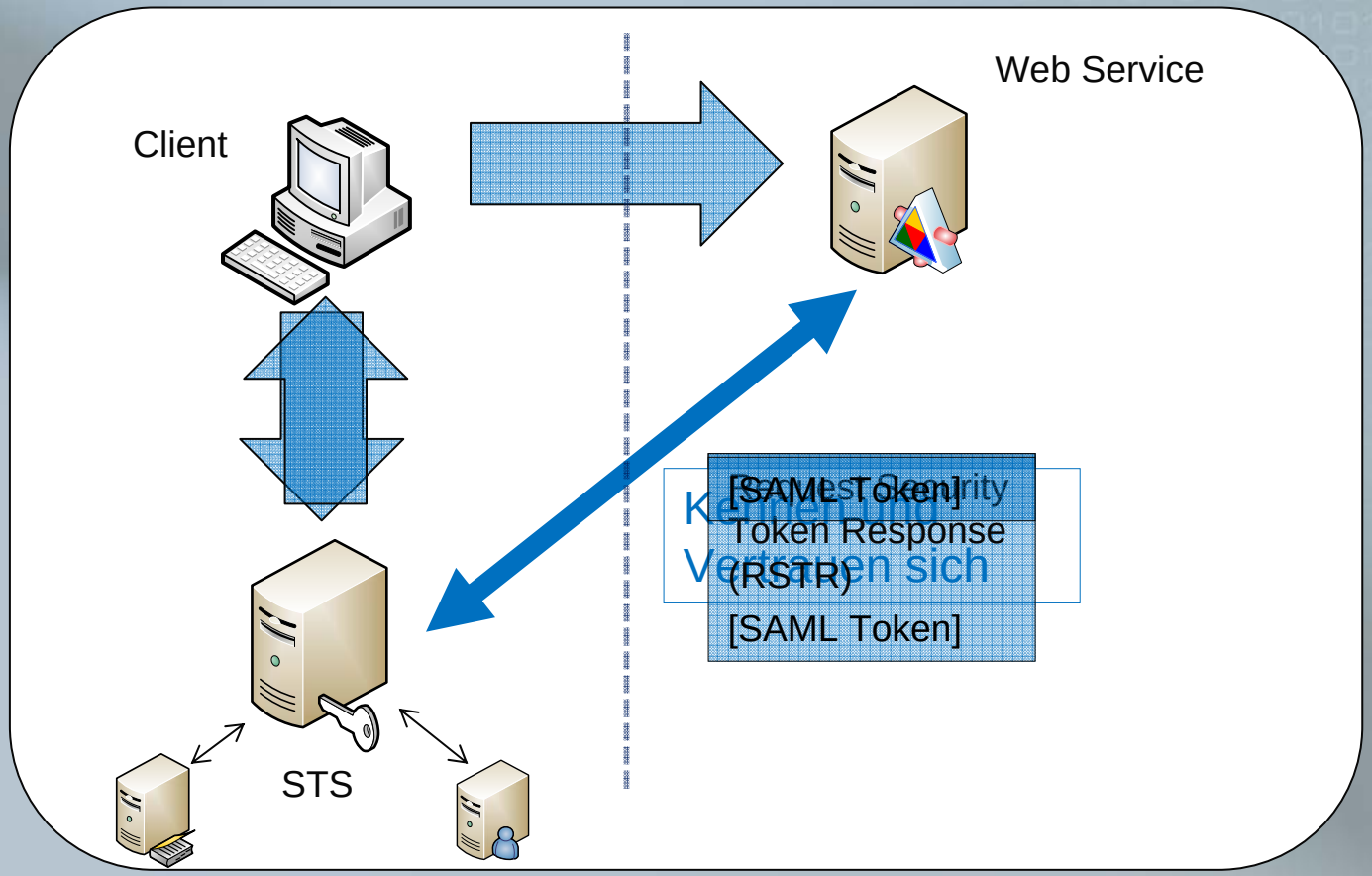
Web Service Sicherheit

WS-Trust

- Security Token Service (STS)
- Aufgabe des STS:
  - Ausgabe von Token
  - Prüfung von Token
  - Erneuerung von Token
  - Entwertung von Token
- Zentrale Stelle für die Verwaltung der Security Tokens

Web Service Sicherheit

Security Token Service



Web Service Sicherheit

WS-Secure-Conversation

- Sicherheitskontext für aufeinanderfolgende Nachrichten
- STS wird entlastet
- Conversation Partner leiten sich Folgeschlüssel aus dem Basisschlüssel ab

Web Service Sicherheit

WS-SecurityPolicy

- Deklarative Beschreibung der Sicherheitsanforderungen eines Webservices
  - Erforderliches Token-Format
  - Welchem STS vertraut der Service
- Mehrere Alternativen werden Angeboten, der Client sucht sich eine aus

Web Service Sicherheit

WS-SecurityPolicy

```
<wsp:Policy>
    <sp:SignedSupportingTokens>
        <wsp:Policy>
            <wsp:ExactlyOne>
                <wsp:All>
                    <sp:UsernameToken/>
                </wsp:All>
            </wsp:ExactlyOne>
        </wsp:Policy>
    </sp:SignedSupportingTokens>
    <sp:SignedParts>
        <sp:Body/>
    </sp:SignedParts>
</wsp:Policy>
```

Web Service Sicherheit

WS-Security in .NET:

- ASMX ist die Web Service Implementierung im .NET Framework
- Web Service Enhancements (WSE) ist eine Extension, die WS-* in .NET verfügbar macht (aktuell: WSE 3.0)
- Windows Communication Foundation (WCF, a.k.a. "Indigo") ist Microsofts nächster Schritt: Einheitliches Programmier Modell für WS-*, Messaging, Queuing, Transactions, ...

- SOA?
- Services?
- Sicherheitsrisiko Web Services
- Web Services & Sicherheit
- Sichere SOAs durch sichere Web Services?

Sichere SOA durch sichere WS?

- SOA Komponenten unterschiedlicher Hersteller erschweren die Integration von Sicherheit
- Viele Spezifikationen sind noch nicht in den Applikationen integriert
- Einzelne Webservices absichern langt nicht aus, ganzheitliches Sicherheitskonzept erforderlich (ein Master Thema?)

Weiterführende Links

SOA

- http://www.cio.com/archive/11504/soa.html
- http://de.bea.com/loesungen/soa/index.jsp
- http://www-306.ibm.com/software/info/openenvironment/soa/
- http://msdn.microsoft.com/architecture/soa/
- http://www.sap.com/platform/netweaver/index.epx
- http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2005/zimmermann/slides.pdf

WS-*

- http://www.oasis-open.org/committees/wss
- http://msdn.microsoft.com/webservices/
- http://ws.apache.org/wss4j/
- http://incubator.apache.org/tsik/
- http://www-128.ibm.com/developerworks/library/specification/ws-secure/
- http://www.w3.org/TR/xmldsig-core/

Vielen Dank
für die
Aufmerksamkeit!