Allgemeine Übersicht WS-Security

Alexander Grünke
Teleseminar: Web Services - Sommersemester 04
Betreuer: Jochen Dinger

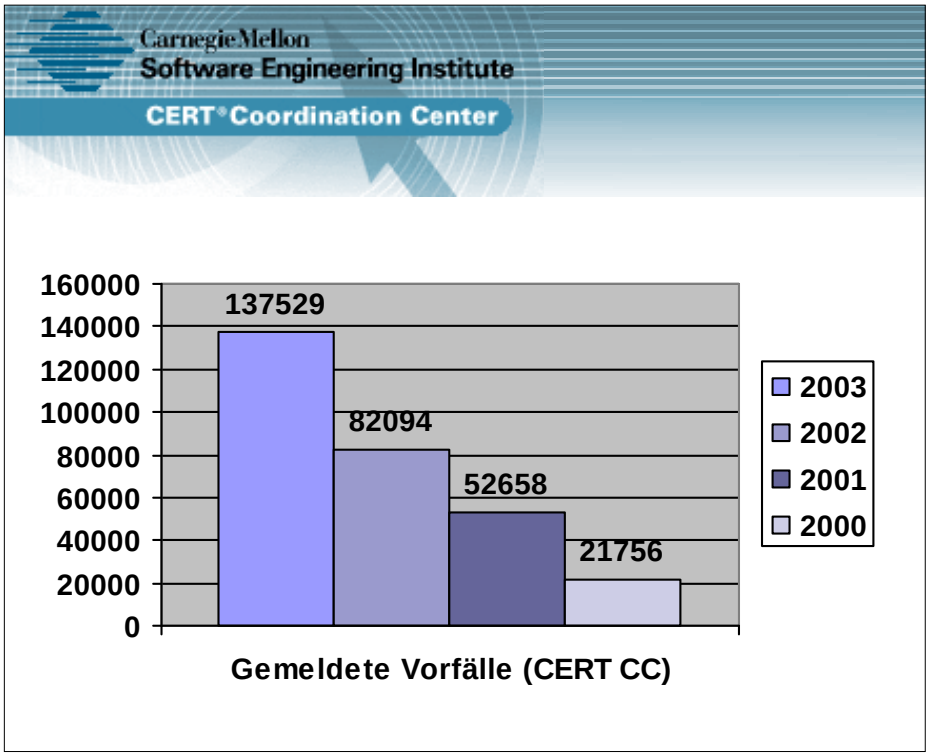
06.07.2004

Inhalt

- Einleitung und Motivation
- Sicherheitsanforderungen an Web-Services
  - Allgemeine Anforderungen
  - Spezielle Anforderungen
  - Anforderungen an Firewalls
- Existierende Technologien
  - SSL / TLS
  - XML Encryption
  - XML Signature
- Neue Technologien: WS-Security
- Zusammenfassung

Einleitung und Motivation

- Die Zahl der Sicherheitsvorfälle nimmt jährlich zu
- Web Services sind neue Zielgruppe für Angriffe
- Es gilt, bestehende Sicherheitskonzepte für Web-Services zu nutzen und neue zu finden, um Sicherheit zu gewährleisten



Sicherheitsanforderungen an Web-Services: Allgemeine Anforderungen

- Sicherheitsbausteine:
  - Vertraulichkeit
  - Integrität
  - Nicht-Abstreitbarkeit
  - Authentifizierung
  - Autorisierung
  - [Datenschutz und Verfügbarkeit]
- Für ein sicheres System müssen alle Bausteine bedacht werden
- Bzgl. Datensicherheit unterscheidet man generell zwei Zustände, in denen sich Daten befinden können:
  - Im Speicher
  - Auf dem Transportweg

Allgemeine Sicherheitsanforderungen: Vertraulichkeit (Confidentiality)

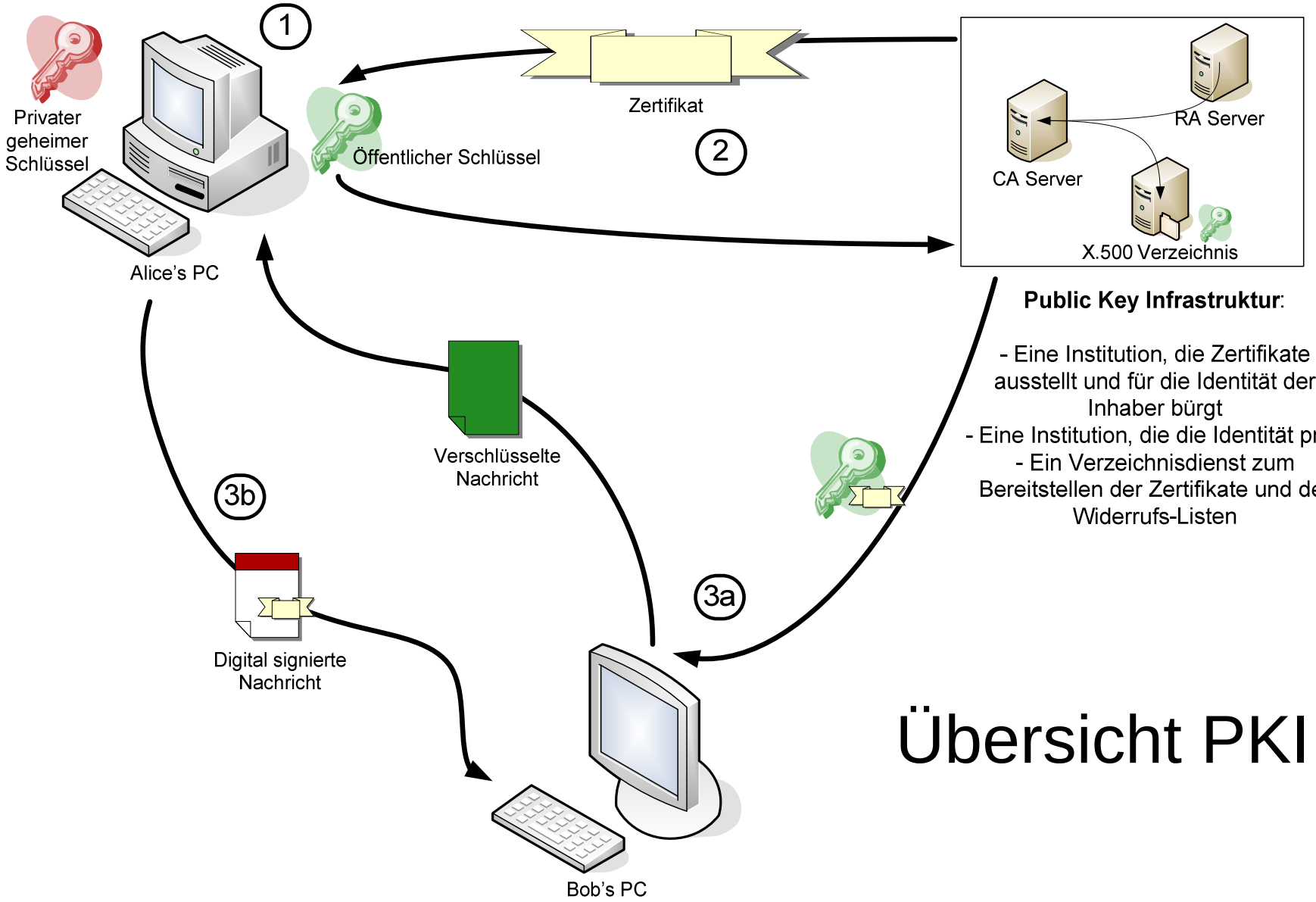
- Kein unbefugter Dritter soll Kommunikation mitlesen können
- Verschlüsselung
  - Heutige Praxis: Plaintext + bekannter Algorithmus + Schlüssel = Ciphertext
  - Symmetrische Algorithmen (Secret Key)
    - Zum Ver- und Entschlüsseln wird der gleiche geheime Schlüssel verwendet
    - schneller als asymmetrische Verfahren
    - DES, DES³, AES/Rijndael usw.
  - Asymmetrische Algorithmen (Public Key)
    - Zum Verschlüsseln wird ein öffentlicher Schlüssel verwendet, zum Entschlüsseln ein geheimer privater Schlüssel
    - RSA usw.

Allgemeine Sicherheitsanforderungen: Integrität (Integrity)

- Es soll erkennbar sein, wenn die Daten während des Transports manipuliert wurden
- Hashing
  - Hash-Funktion berechnet eindeutige Kurzfassung der Daten, d.h. jede Änderung der Daten führt zu anderem Ergebnis der Hash-Funktion
  - MD2, MD4, MD5, SHA
  - ABER: Hash-Wert könnte von Drittem an geänderte Nachricht angepasst werden
- Digitale Signaturen
  - RSA lässt sich auch ‚invers' verwenden, sprich man verschlüsselt mit dem geheimen privaten Schlüssel und entschlüsselt mit dem öffentlichen
  - Absender verschlüsselt Hash-Wert mit geheimem Schlüssel, jeder kann Hash-Wert mittels öffentlichem Schlüssel entschlüsseln also prüfen, ob Nachricht unverändert ist

Allgemeine Sicherheitsanforderungen: Nicht-Abstreitbarkeit (Non-Repudiation)

- Der Absender soll nicht behaupten können, die Nachricht NICHT gesendet zu haben
- Digitale Zertifikate
  - Persönliche Daten des Besitzers des geheimen Schlüssels (z.B. Name, Adresse etc.) werden zusammen mit dem öffentlichen Schlüssel, einer Seriennummer und einem Verfallsdatum in einem Digitalen Zertifikat gespeichert
  - X.509
- Public Key Infrastruktur
  - Digitale Zertifikate werden von Certificate Authorities (CA) ausgestellt, wobei Vorraussetzung ist, dass die Identität des zukünftigen Zertifikatsinhabers vorher von einer Registration Authority (RA) überprüft wurde
  - Zertifikate können dann in einem öffentlichen Verzeichnis abgelegt werden (z.B. auf X.500 basierend)
  - Zusammen bilden CAs, RAs und X.500 Verzeichnisse die sog. Public Key Infrastrukur (PKI)



Übersicht PKI

Allgemeine Sicherheitsanforderungen: Authentifizierung (Authentication)

- Die Identität des Kommunikationspartners soll eindeutig nachgewiesen werden
- PKI
  - Mögliches Szenario: A sendet B zufälligen Bit-String, B verschlüsselt mit privatem Schlüssel, sendet Chiffretext an A
- SmartCards
- Biometrie
- Username + Passwort

Allgemeine Sicherheitsanforderungen: Autorisierung (Authorization)

- Ein User soll nur das machen dürfen, wofür er explizit Rechte erhalten hat

- Rollenbasierte Zugriffskontrolle
  - Nutzer werden entsprechend ihrer Aufgaben Gruppen oder auch Rollen zugeordnet, die jeweils über bestimmte Rechte verfügen
  - Rollen und Gruppen sind stark an die realen Gegebenheiten innerhalb einer Organisation angelehnt

Allgemeine Sicherheitsanforderungen: Datenschutz (Privacy) und Verfügbarkeit (Availability)

- Den rechtlichen und sonstigen relevanten Datenschutzbestimmungen muss genügt werden

- Es soll sichergestellt sein, dass der Service jederzeit verfügbar ist

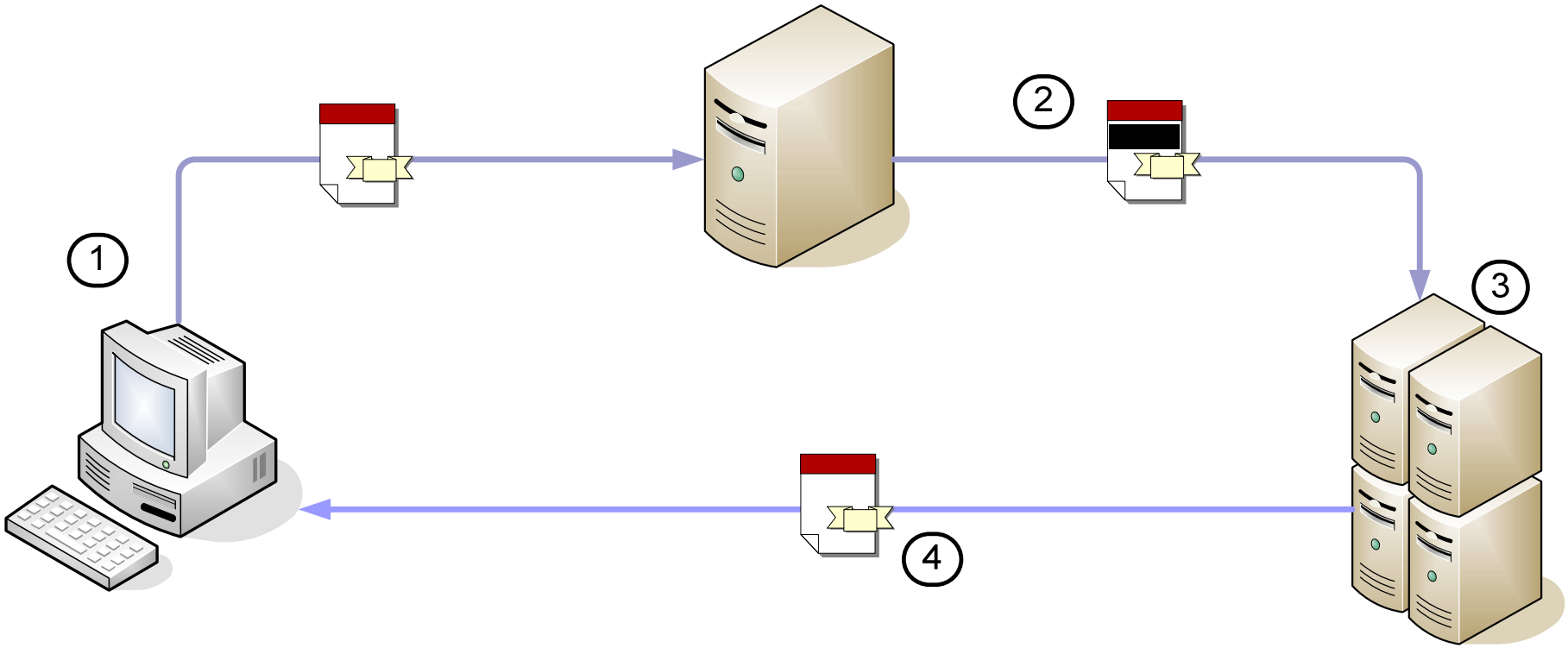
- Denial of Service (DoS) Attacken
  - Ziel ist es, alle Ressourcen eines Services zu verbrauchen, so dass er für reguläre Nutzer nicht mehr verfügbar ist
  - Weiterentwicklung: Distributed DoS (DDoS)
  - Viele Rechner werden ‚gehacked', DoS Tools installiert und somit eine sehr viel grössere Menge an Angreifern geschaffen

Sicherheitsanforderungen in WS: Spezielle Anforderungen

- Ende-zu-Ende Sicherheit
  - Sicherheit soll während gesamter Übertragung, auch über mehr als einen Web-Service, gewährleistet sein
  - Problem bei SSL: auf Zwischenknoten sind die Daten unverschlüsselt
- Spezielle Autorisierungsverfahren
  - Web-Service muss zwecks Autorisierung Informationen über End-Nutzer erhalten, auch wenn dieser nicht direkt mit Web-Service kommuniziert
  - Nutzer muss sich nicht selbst bei jedem Web-Service erneut ausweisen

Sicherheitsanforderungen in WS:

Beispiel



Sicherheitsanforderungen in WS: Anforderungen an Firewalls

- Typen:
  - Paket-Filternde Firewalls
  - Firewalls auf Applikations-Ebene

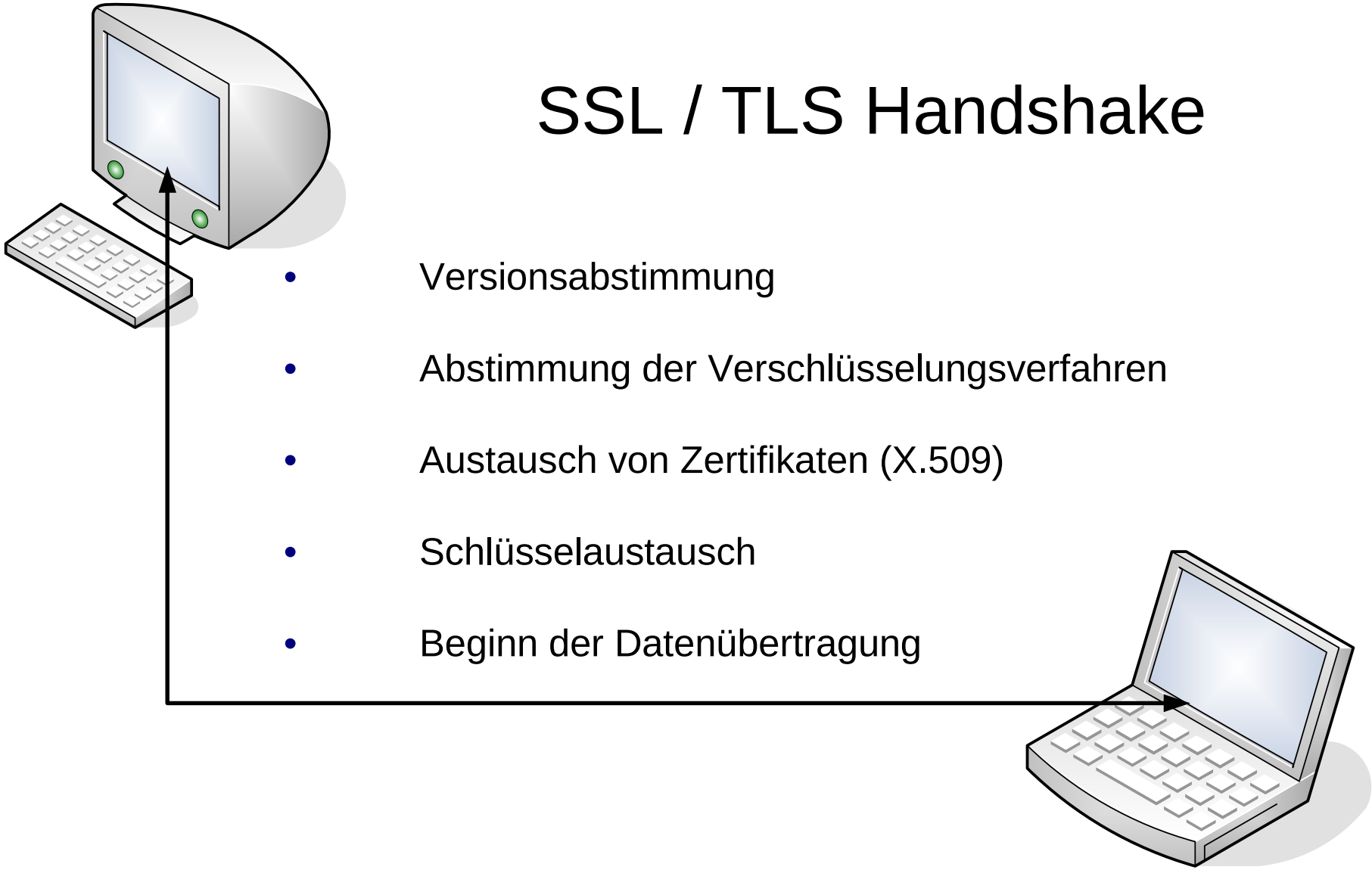
- Firewalls sollen SOAP Nachrichten erkennen und filtern
- Firewalls sollen erkennen, ob Nachrichten an Web-Service gerichtet sind, der für Absender erreichbar sein soll
- Firewalls sollen feststellen, ob Inhalt einer SOAP Nachricht syntaktisch korrekt ist
- Firewalls sollen vorher genannte Sicherheitsanforderungen prüfen bzw. ergänzen

Existierende Technologien: SSL / TLS

- SSL 2.0, Netscape, 1995
- SSL 3.0, Netscape, 1996
- TLS 1.0 auf SSL basierender Nachfolger, IETF (RFC 2246), 1999

- Protokolle gewährleisten innerhalb HTTP Punkt-zu-Punkt-Verbindung Vertraulichkeit, Integrität und einseitige oder beidseitige Authentifizierung

SSL / TLS Handshake



- Versionsabstimmung
- Abstimmung der Verschlüsselungsverfahren
- Austausch von Zertifikaten (X.509)
- Schlüsselaustausch
- Beginn der Datenübertragung

Existierende Technologien: XML Encryption (XMLEnc)

- W3C Recommendation, Dezember 2002

- Definiert XML-Syntax für verschlüsselte Daten

- Ermöglicht es, gezielt nur bestimmte Teile von XML-Dokumenten zu verschlüsseln

- KEINE neuen kryptographischen Algorithmen!

Existierende Technologien: XML Encryption: Beispiel

```
<?xml version='1.0'?>
<PaymentInfo
      xmlns='http://example.org/paymentv2'>

   <Name>John Smith<Name/>

   <CreditCard Limit='5,000' Currency='USD'>
      <Number>4019 2445 0277 5567</Number>
      <Issuer>Bank of the Internet</Issuer>
      <Expiration>04/02</Expiration>
   </CreditCard>

</PaymentInfo>
```

```
<?xml version='1.0'?>
<PaymentInfo
      xmlns='http://example.org/paymentv2'>

   <Name>John Smith<Name/>

   <EncryptedData
     Type='http://www.w3.org/2001/04/xmlenc#
       Element'
     xmlns='http://www.w3.org/2001/04/xmlenc#'
     >
       <CipherData>
          <CipherValue>A23B45C56
          </CipherValue>
       </CipherData>
   </EncryptedData>

</PaymentInfo>
```

Existierende Technologien: XML Signature (XMLSig)

- IETF / W3C Recommendation, Februar 2002
- Definiert XML-Syntax für digitale Signaturen
- Ermöglicht analog, gezielt nur bestimmte Teile eines XML-Dokuments zu signieren
- Canonicalization [= Normalisierung von XML-Dokumenten] nötig

Existierende Technologien

- Authentifizierung und Autorisierung
  - Security Assertion Markup Language (SAML)
  - XML Access Control Markup Language (XACML)
  - Microsoft Passport
  - Liberty Alliance Project
- PKI
  - XML Key Management Specification (XKMS)

Web Services Security

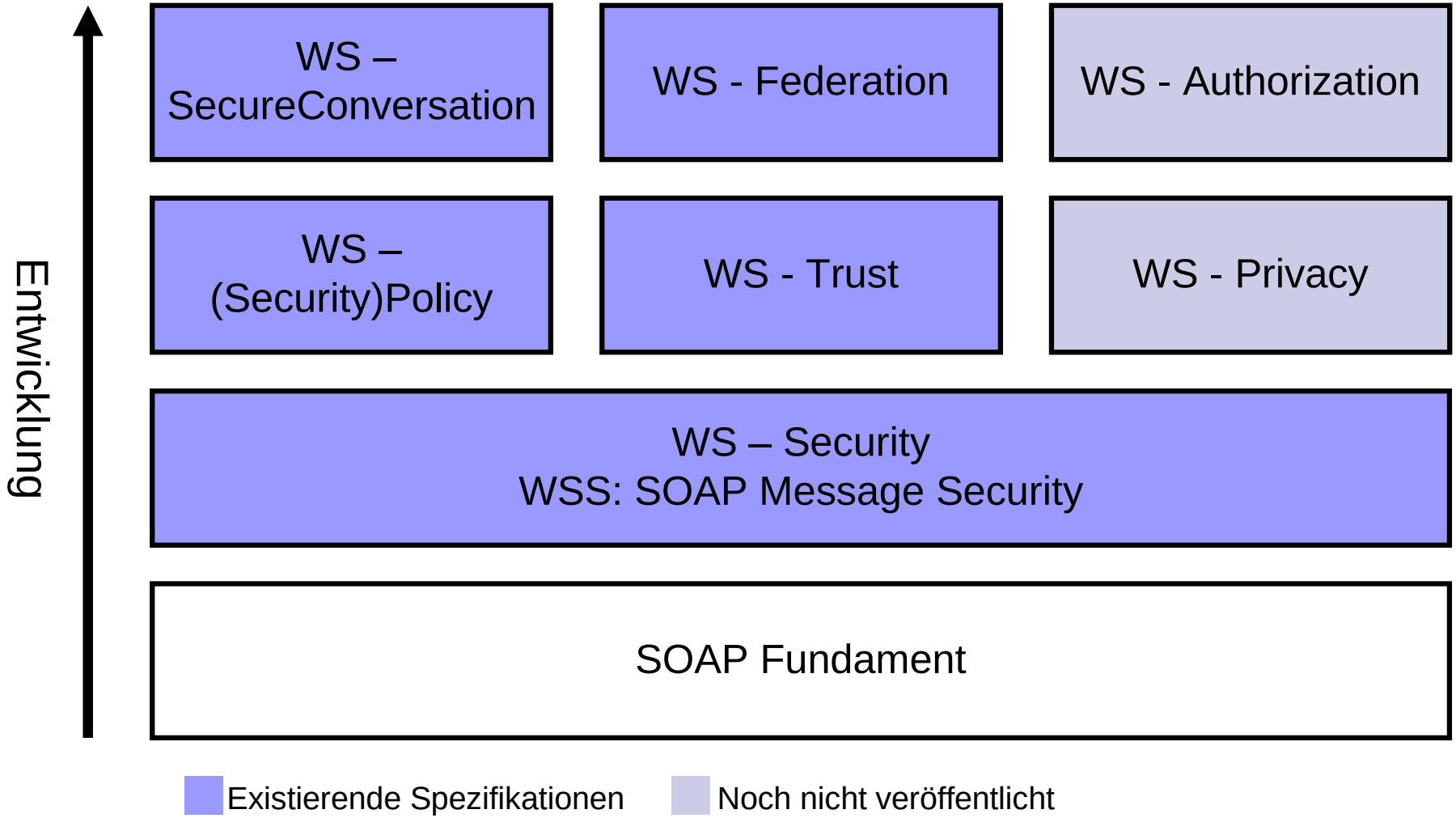
- In Entwicklung seit 2001, veröffentlicht von IBM und Microsoft im April 2002
- Juni 2002 zwecks Standardisierung zu Teilen an OASIS übergeben

- Reihe von Spezifikationen, welche Einbindung von Sicherheits-Elementen in SOAP-Nachrichten definieren

- Standards nutzen bestehende Technologien wie z.B. XML Encryption und Signature, X.509 usw.

- Definitionen sind sehr modular gehalten – es ist exakt definiert, wie Sicherheitstechniken eingebunden werden, aber sehr offen gehalten, was das für Sicherheitsverfahren sein sollen

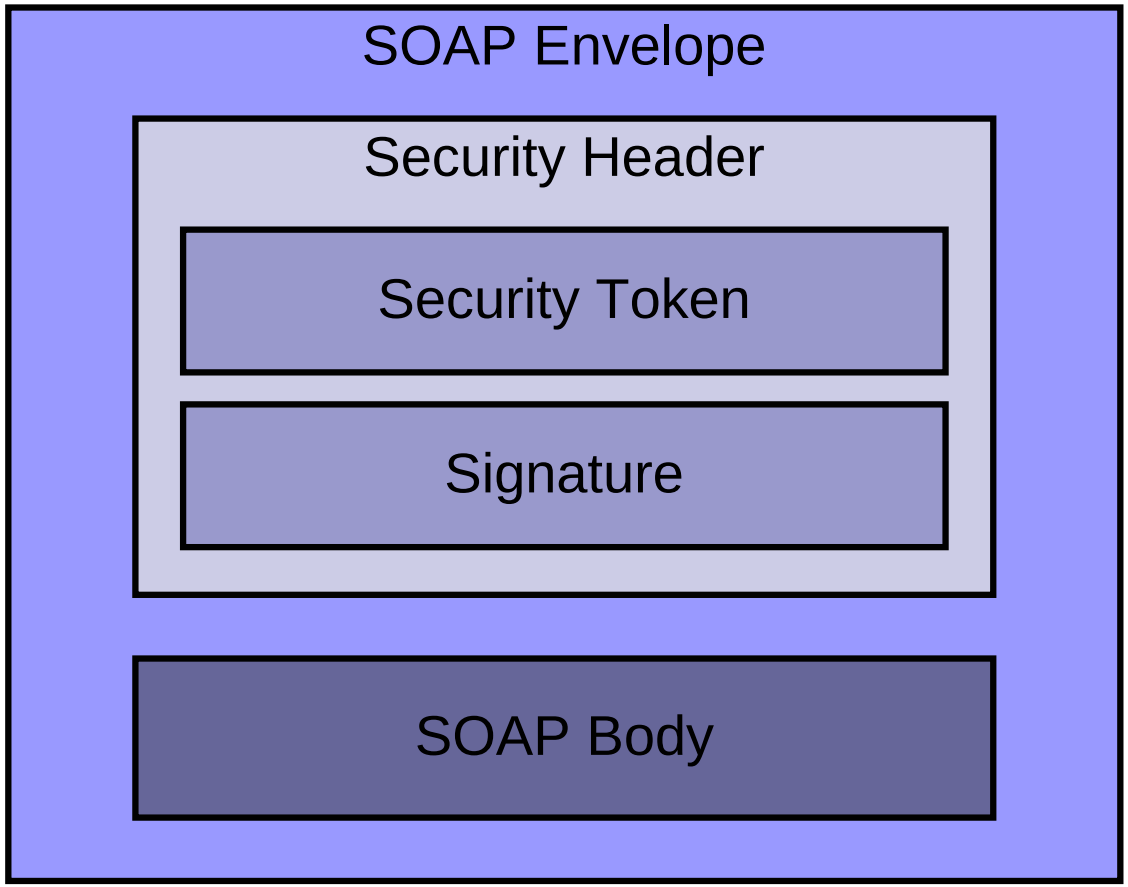
- Massiv durch Industrie vorangetrieben, mittlerweile unterstützt durch BEA Systems, Computer Associates, HP, IBM, Microsoft, Novell, SAP, Sun und andere


WS – SecureConversation
WS - Federation
WS - Authorization
WS – (Security)Policy
WS - Trust
WS - Privacy
WS – Security
WSS: SOAP Message Security
SOAP Fundament
Entwicklung
Existierende Spezifikationen
Noch nicht veröffentlicht

WS-Security:
WSS: SOAP Message Security

- WSS: SOAP Message Security 1.0, OASIS Standard, März 2004
- Beschreibt Sprache zur Definition eines SOAP-Protokolls für Austausch einzelner Nachrichten, das Vertraulichkeit und Integrität gewährleistet
- Unterstützt Vielzahl von Sicherheitstechnologien
- nutzt speziell XMLEnc und XMLSig
- Beschreibt generelle Syntax zur Einbindung von Sicherheits-Merkmalen (sog. Tokens; z.B. Name, Identität, Schlüssel...) in SOAP-Nachrichten
- Beschreibt Codierung digitaler Sicherheits-Merkmale
- Sub-Standards:
  - WSS: UsernameToken Profile 1.0
  - WSS: X.509 Certificate Token Profile 1.0

WS-Security: Schema



WS-Security: WS-Policy

- WS-SecurityPolicy 1.0, IBM/Microsoft/VeriSign/RSA, Dezember 2002
- WS-Policy 1.1, IBM/Microsoft/BEA/SAP, Mai 2003

- Definiert SOAP-Syntax-Erweiterung zur Beschreibung von Verfahrensweisen

- z.B. Beschreibung von Anforderungen, Präferenzen, Fähigkeiten (welche Authentifizierungsverfahren werden unterstützt, welche bevorzugt etc.)

- Komplett erweiterbar

WS-Security: WS-Trust

- WS-Trust 1.1, IBM/Microsoft/VeriSign/BEA/RSA u.a., Mai 2004
- Definiert SOAP-Syntax-Erweiterung für die Anforderung und Ausstellung von Sicherheits-Tokens und für die Vermittlung von Vertrauensverhältnissen
- Mögliche Technologien: X.509, Kerberos shared-secret tickets, Password Digests etc.
- Sehr flexibel, erweiterbar
- Security-Token-Service: verlangt gültiges Token, stellt daraufhin eigenes Token aus

WS-Security: WS-Federation

- WS-Federation 1.0, IBM/Microsoft/VeriSign/BEA/RSA, Juli 2003
- Definiert auf WS-SecureConversation, WS-Policy, WS-Trust und WSS: SOAP Message Security basierende Mechanismen um Zusammenschlüsse mehrere Vertrauenszonen in Verbund zu ermöglichen
- Identity-Provider: Authentifizierungs Service, Erweiterung zu Security-Token-Service
- Pseudonym-Service: Verwaltung verschiedender Aliase für verschiedene Zonen, Ergänzung zu Identity-Provider
- Möglichkeiten:
  - Single-Sign-On
  - Identity-Mapping

WS-Security: WS-Privacy

- Soll Zusammenspiel von WSS: SOAP Message Security, WS-Policy und WS-Trust definieren, um automatische Bekanntgabe/Einhaltung von Datenschutz-Richtlinien innerhalb von SOAP-Kommunikation zu ermöglichen

WS-Security: WS-SecureConversation

- WS-SecureConversation 1.1, IBM/Microsoft/VeriSign/BEA/RSA u.a., Mai 2004
- Definiert auf WS-Trust und WSS: SOAP Message Security basierende Erweiterungen um sichere Kommunikation über eine oder mehrere Nachrichten hinweg zu ermöglichen
- Aufbau eines Security-Kontexts, gemeinsame Nutzung eines Security-Kontexts, Erlangen von Schlüsseln aus bestehenden Security-Kontexten

WS-Security: WS-Authorization

- Soll definieren, wie Zugriffsrechte für Web-Services spezifiziert und verwaltet werden
- Soll flexibel und erweiterbar sein

WS-Security: Beispiel

```
<?xml version="1.0" encoding="utf-8"?>
<S11:Envelope xmlns:S11="..." xmlns:wsse="..." xmlns:wsu="..." xmlns:ds="...">
   <S11:Header>
      <wsse:Security xmlns:wsse="...">
         <xxx:CustomToken wsu:Id="MyID" xmlns:xxx="http://fabrikam123/token">
            FHUIORv...
         </xxx:CustomToken>
         <ds:Signature>
            <ds:SignedInfo>
               <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
               <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
               <ds:Reference URI="#MsgBody">
                  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <ds:DigestValue>LyLsF0Pi4wPU...</ds:DigestValue>
               </ds:Reference>
            </ds:SignedInfo>
```

WS-Security: Beispiel

```
            <ds:SignatureValue>DJbchm5gK...</ds:SignatureValue>
            <ds:KeyInfo>
               <wsse:SecurityTokenReference>
                  <wsse:Reference URI="#MyID"/>
               </wsse:SecurityTokenReference>
            </ds:KeyInfo>
         </ds:Signature>
      </wsse:Security>
   </S11:Header>
   <S11:Body wsu:Id="MsgBody">
      <tru:StockSymbol xmlns:tru="http://fabrikam123.com/payloads">
         QQQ
      </tru:StockSymbol>
   </S11:Body>
</S11:Envelope>
```

Produkte / Implementierungen

- Java: Apache WSS4J für Axis
- .NET: Microsoft Web Services Enhancements 2.0
- IBM WebSphere Application Server V5.0.2

- Reactivity XML Firewall
- DataPower XS40 XML Firewall
- ForumSystems XWall
- Westbridge Technology XML Message Server
- VeriSign Trust Gateway

Ausblick / Kommende Spezifikationen

- WSS: Rights Expression Language (REL) Token Profile, OASIS
- WSS: Security Assertion Markup Language (SAML) Token Profile, OASIS
- WSS: Kerberos Token Profile, OASIS
- WSS: Minimalist Token Profile, OASIS

Zusammenfassung

- Sicherheitsanforderungen von Web-Services gehen über allgemeine Sicherheitsanforderungen hinaus
- Bestehende Technologien werden in neue Standards integriert, z.B. XMLEnc, X.509 etc.
- Neue Standards sind modular aufgebaut
- Application-Level Firewalls werden nötig