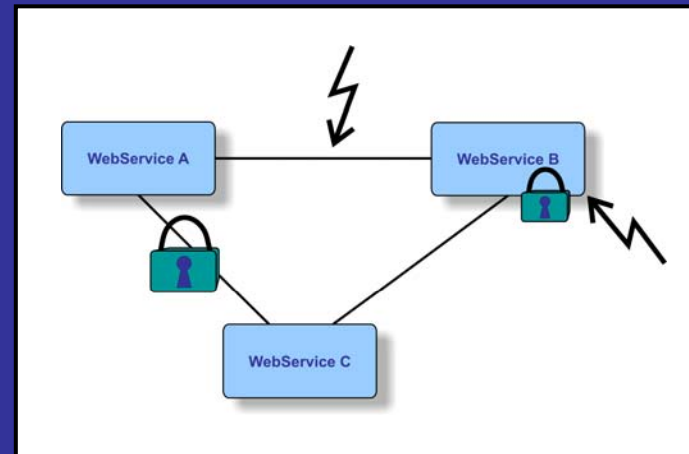


Secure WebServices (SWS)

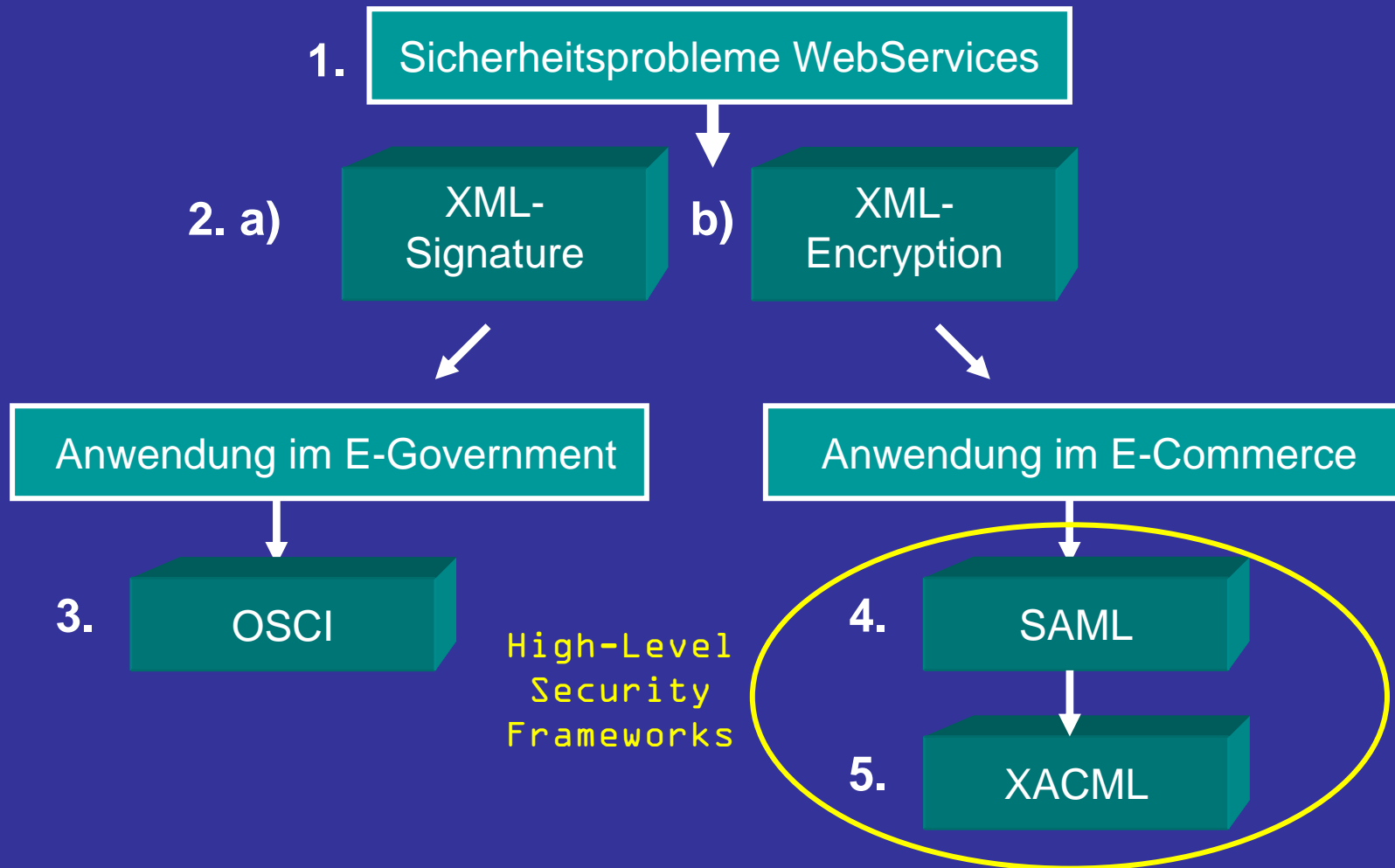
Verteilte
Datenbanksysteme
SS2005

Tobias Giese
Masterstudiengang Informatik
HS-Harz

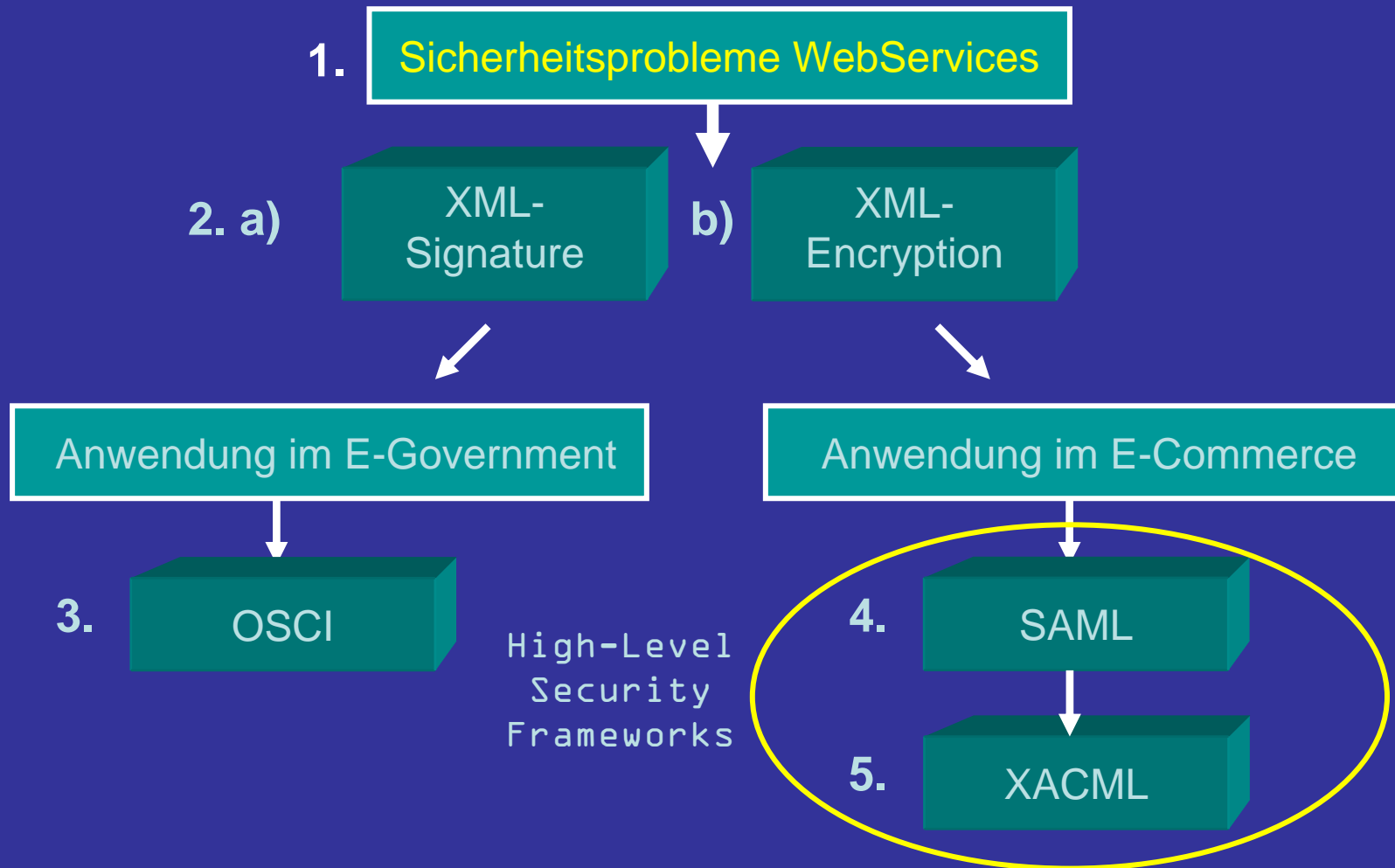
20.06.2005



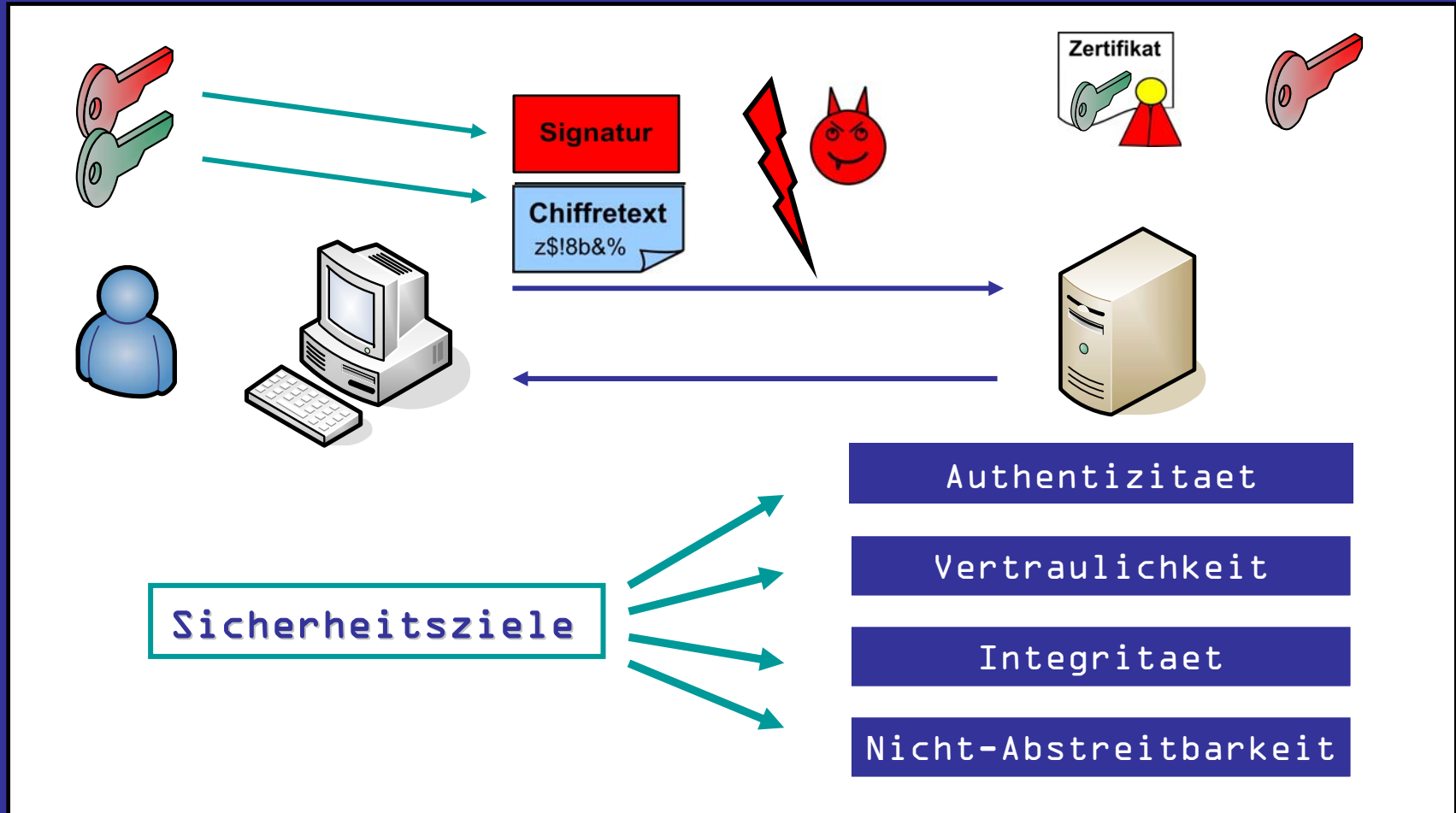
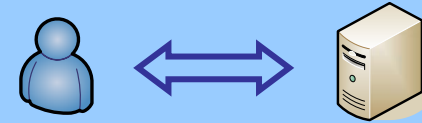
◆ Roadmap



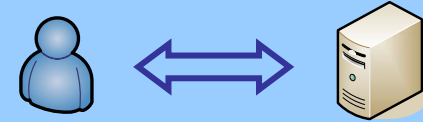
◆ Roadmap



◆ Sicherheitsziele



◆ Sicherheitsmassnahmen



Sicherheitsziele

Sicherheitsmassnahmen

Authentizitaet

Signatur

Vertraulichkeit

Verschluesselung

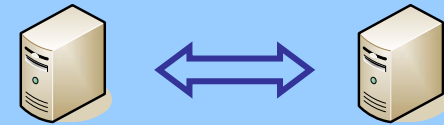
Integritaet

Signatur/Hashfunkt.

Nicht-Abstreitbarkeit

Signatur/Zeitstempel

◆ WebServices im Einsatz

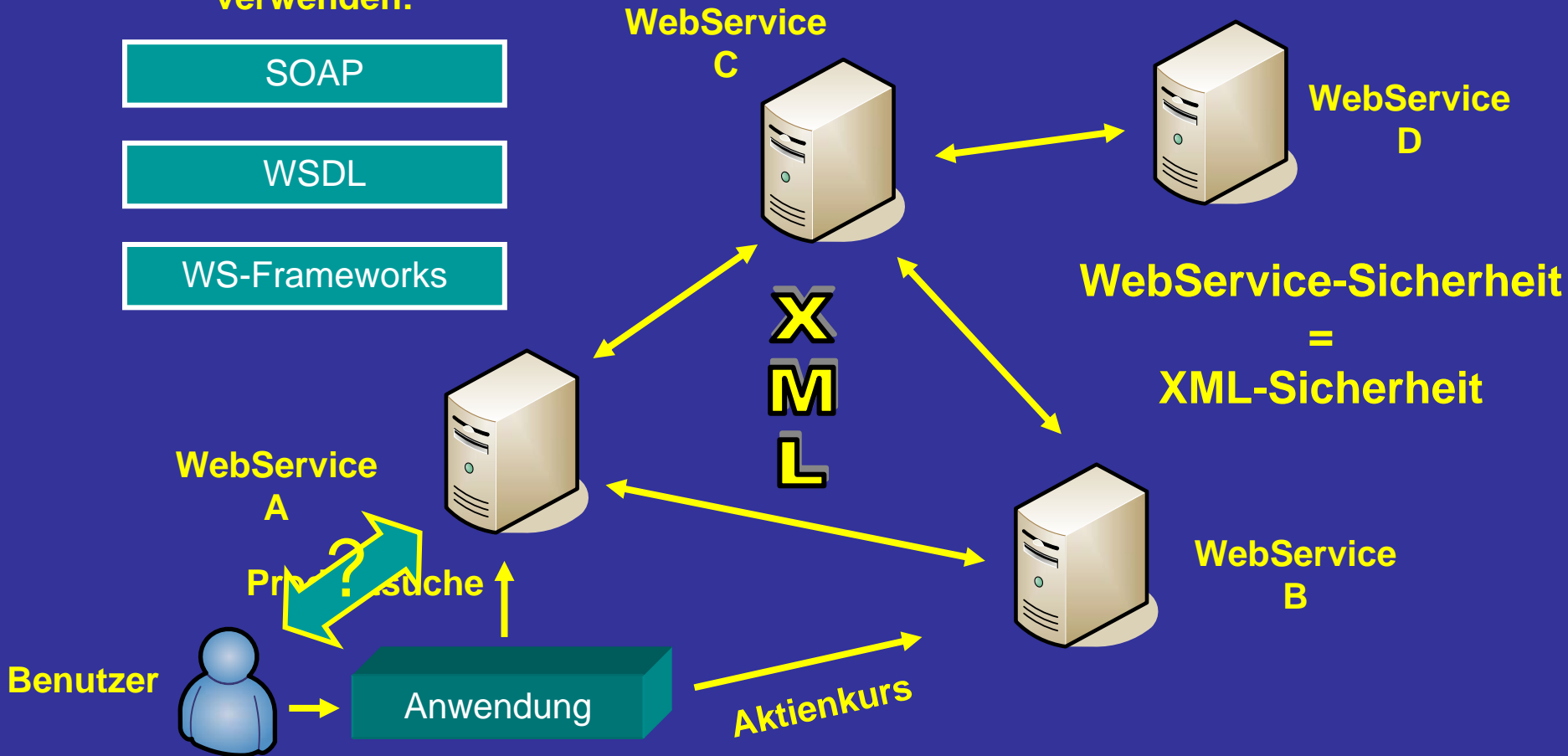


verwenden:

SOAP

WSDL

WS-Frameworks

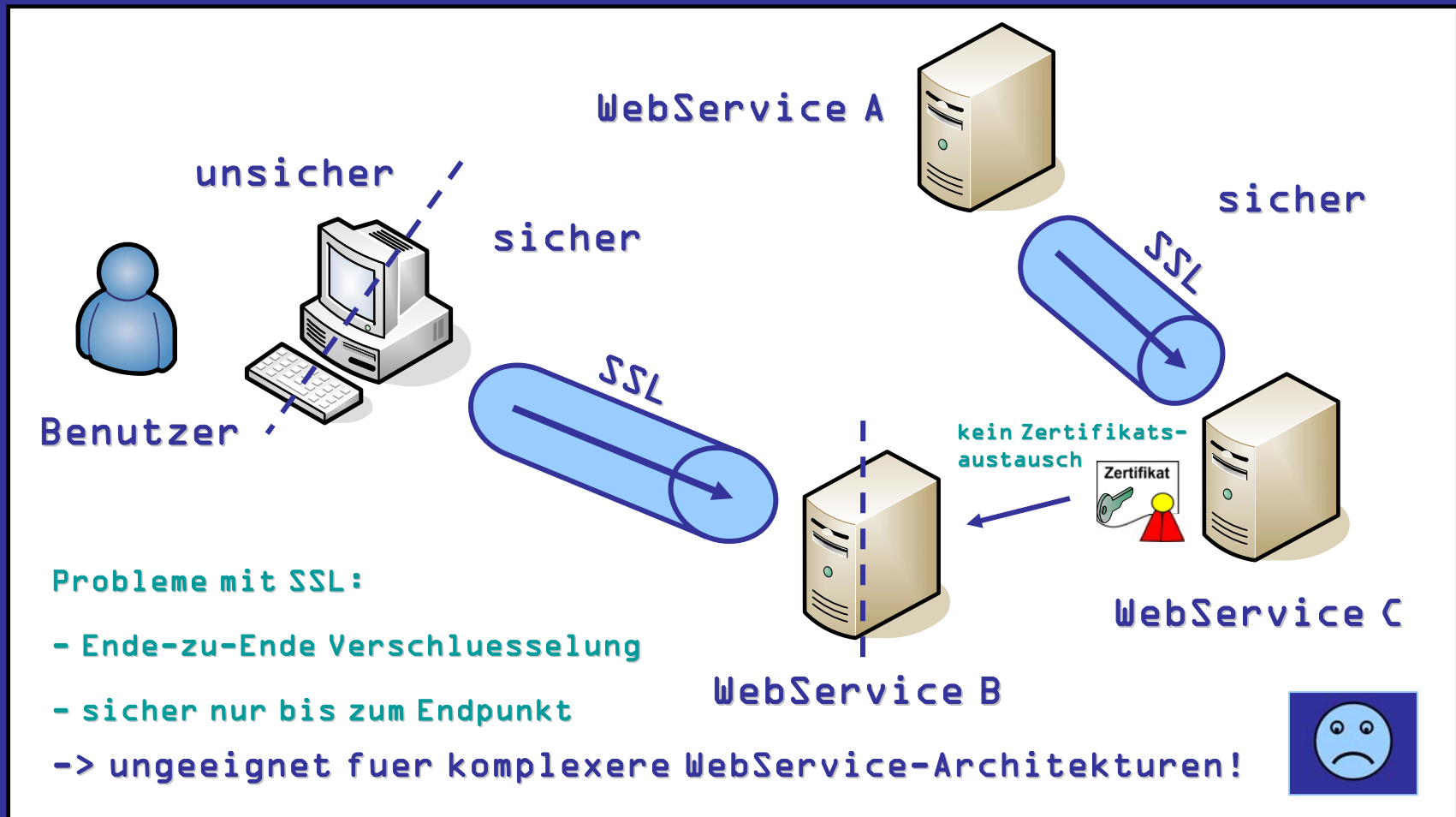


◆ Ausgangsbasis und Probleme

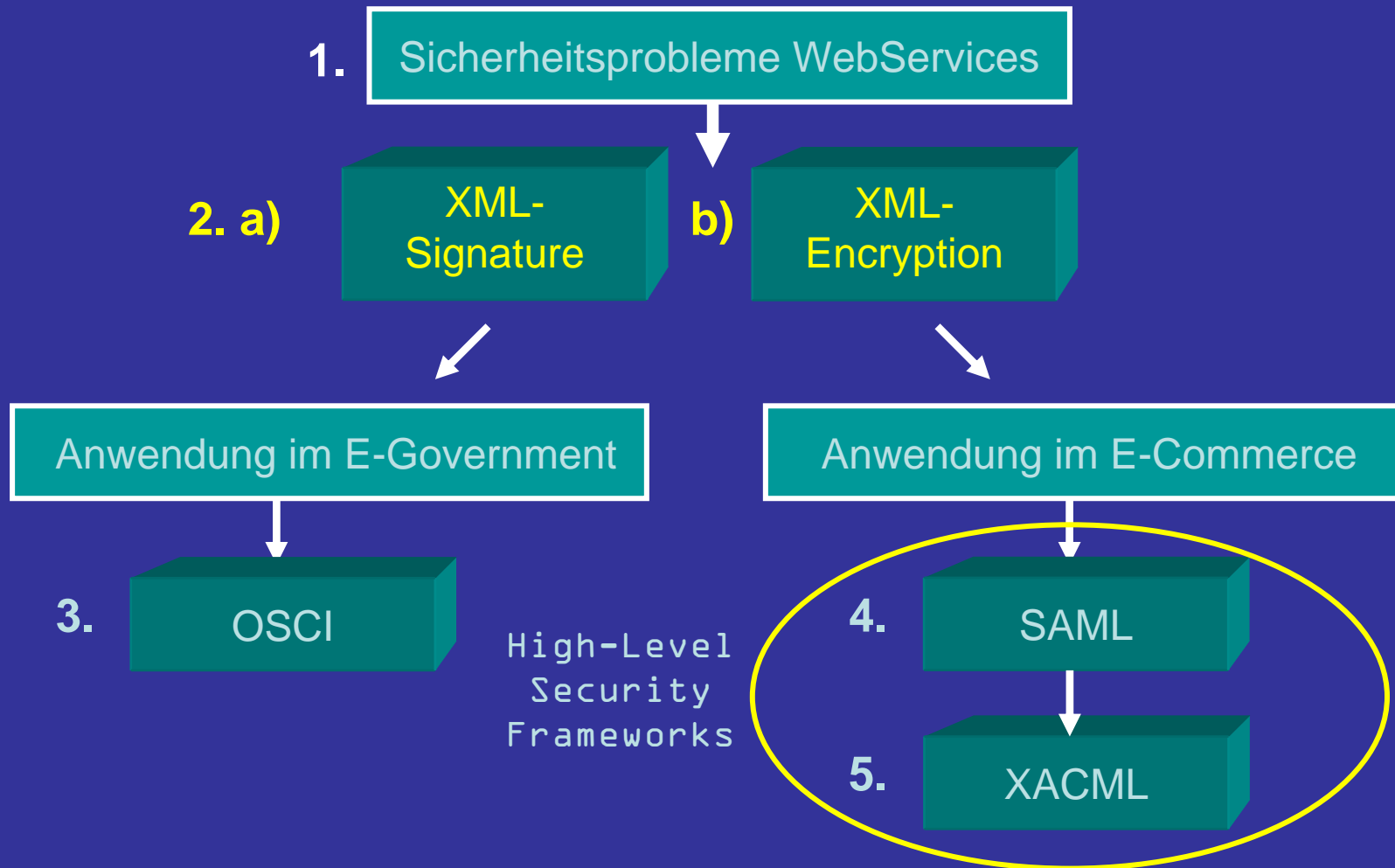
- **WebServices** gut geeignet für **verteilte Aufgaben**
 - entworfen für **Maschine-Maschine**-Kommunikation
 - **menschl.** Autor.-Komponente fällt weg (GUI / Zertifikate)
- **Spezifikation** der WebServices beinhaltet **nur** die zur **Kommunikation** notwendigen Formate & Infrastrukturen
 - **keine** Definition von **Si-Massnahmen / - Zielen**
 - Sicherheit hängt vom verwendeten Transportprotokoll ab
 - **kein Standard** für die **sichere** Kommunikation
- **Gegenmaßnahmen:** Neuer Standard der OASIS
 - SOAP-Erweiterungen
 - Nachrichten-Integrität
 - Nachrichten-Vertraulichkeit
 - Authentizität



◆ SSL – Sicherung auf Transportschicht



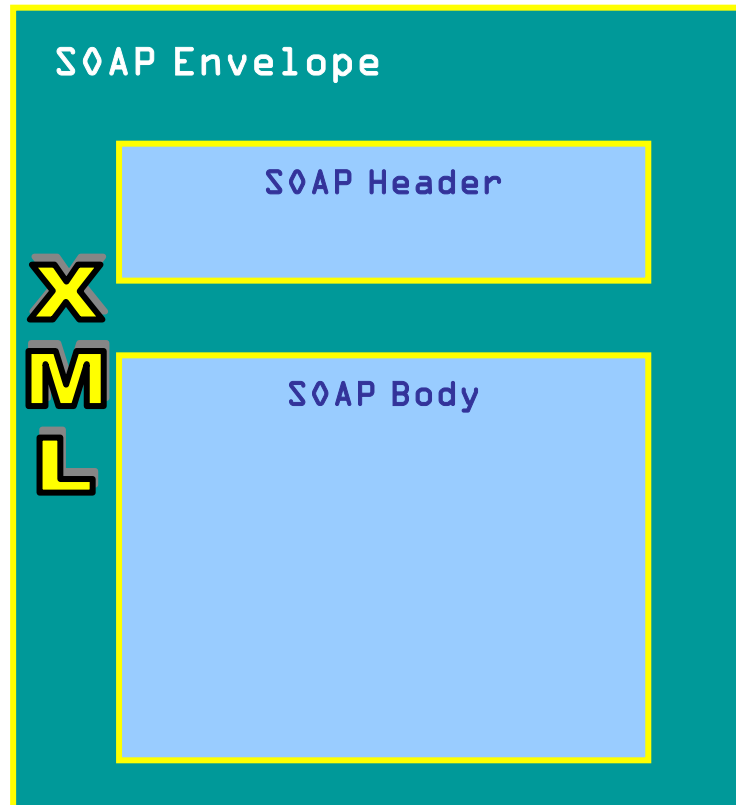
◆ Roadmap



Sicherheit vereinbart im Header

◆ SOAP over HTTP-Binding

HTTP

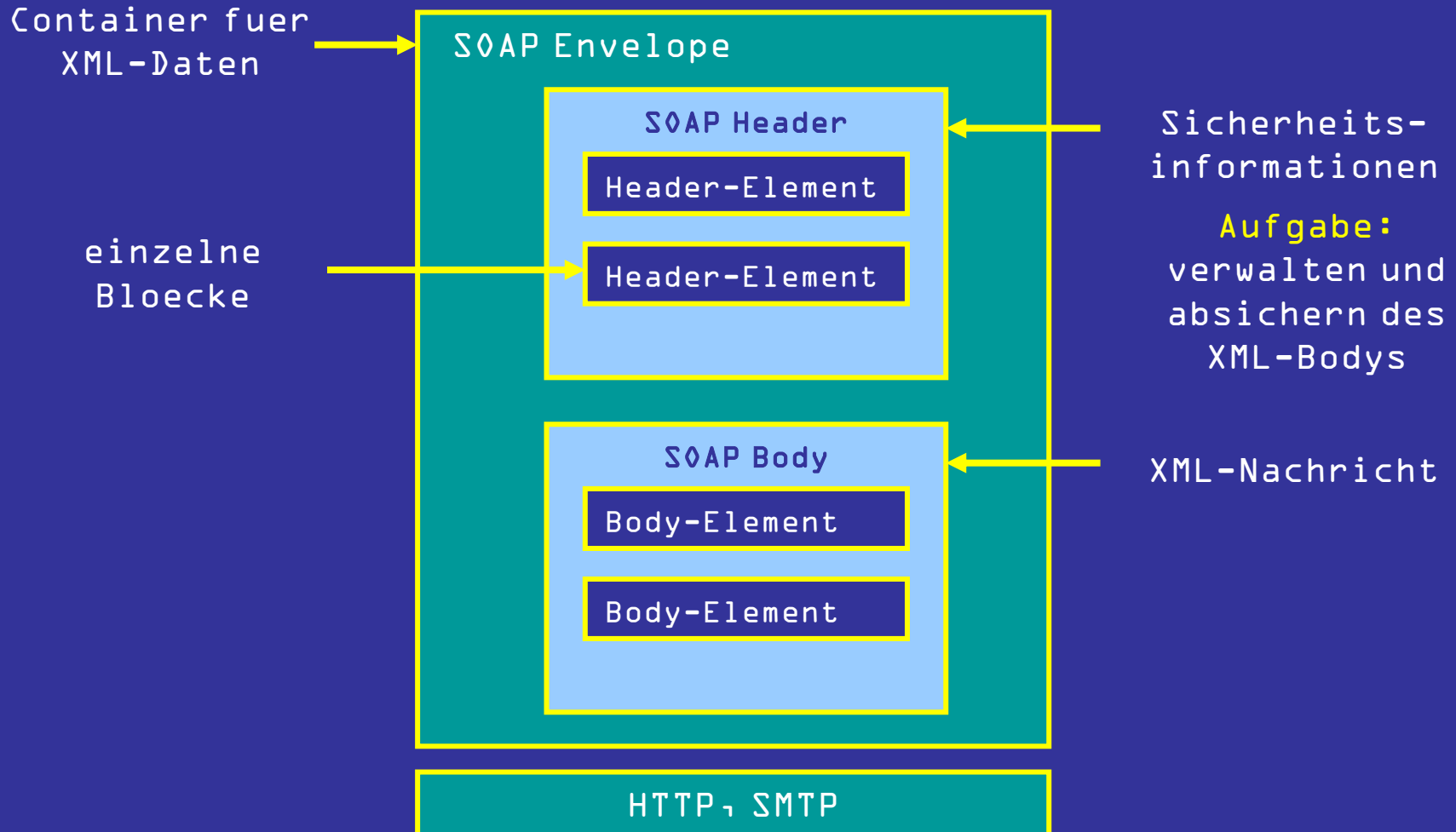


◆ XML-Signature

- 2002 vom **W3C** standardisiert
- Umsetzung der **digitalen Signatur**
- Sicherheitsziele:
 - **Integrität** der Nachricht
 - **Authentizität** des Absenders
- Signaturraum:
 - **kompletter XML-Baum** (XML-Dokument)
 - **einzelne Elemente** oder **Anlagen** (z.B. Bilder)
- **keine** Schlüsselmanagement-Aufgaben
- **Beschreibung** des Schlüssels wird mitgeschickt

Sicherheit vereinbart im Header

◆ SOAP-Nachricht – Aufbau

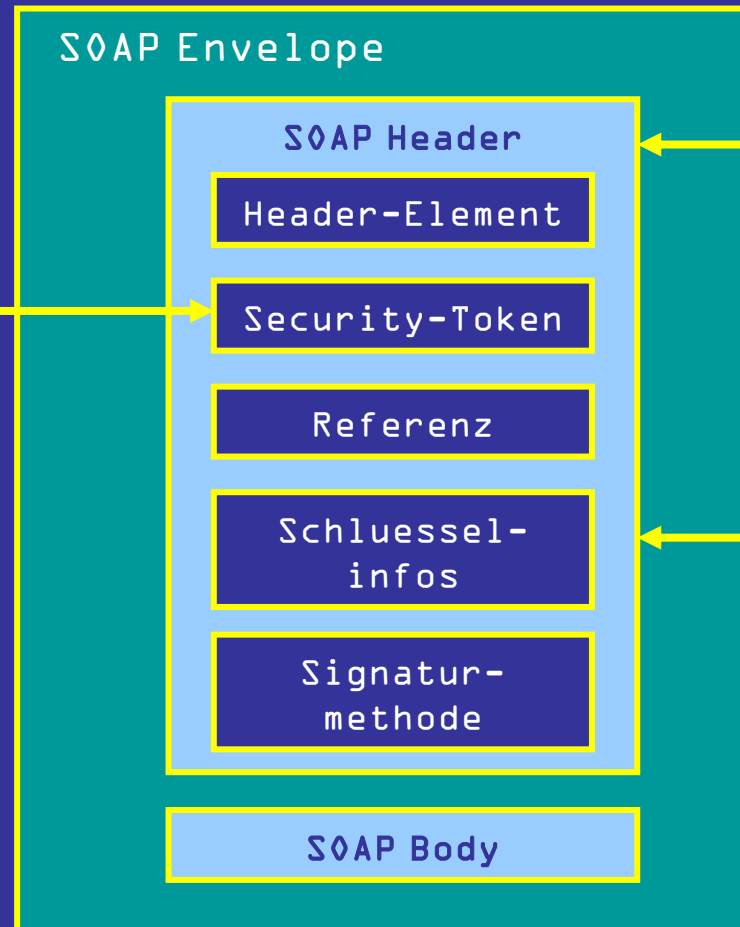


Sicherheit vereinbart im Header

◆ SOAP-Nachricht – Header

Hinzufuegen von
 Identitaet ->
 bereit fuer WS-
 Security z.B.:

- Schuessel
- Zertifikate
- Assertions
- Kerberos Tickets



Sicherheits-
 informationen

Aufgabe:

verwalten und
 absichern des
 XML-Bodys

XML-Nachricht

◆ Signierte Kauforder

```
<PurchaseOrder id="po1">  
  <SKU>12356</SKU>  
  <Quantity>12356</Quantity>  
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#2">  
    <SignedInfo>  
      <Reference URI="#po1" />  
    </SignedInfo>  
    <SignatureValue>5Ajz8Ds539iaP4sdA</SignatureValue>  
    <KeyInfo>  
    </KeyInfo>  
  </Signature>  
</PurchaseOrder>
```

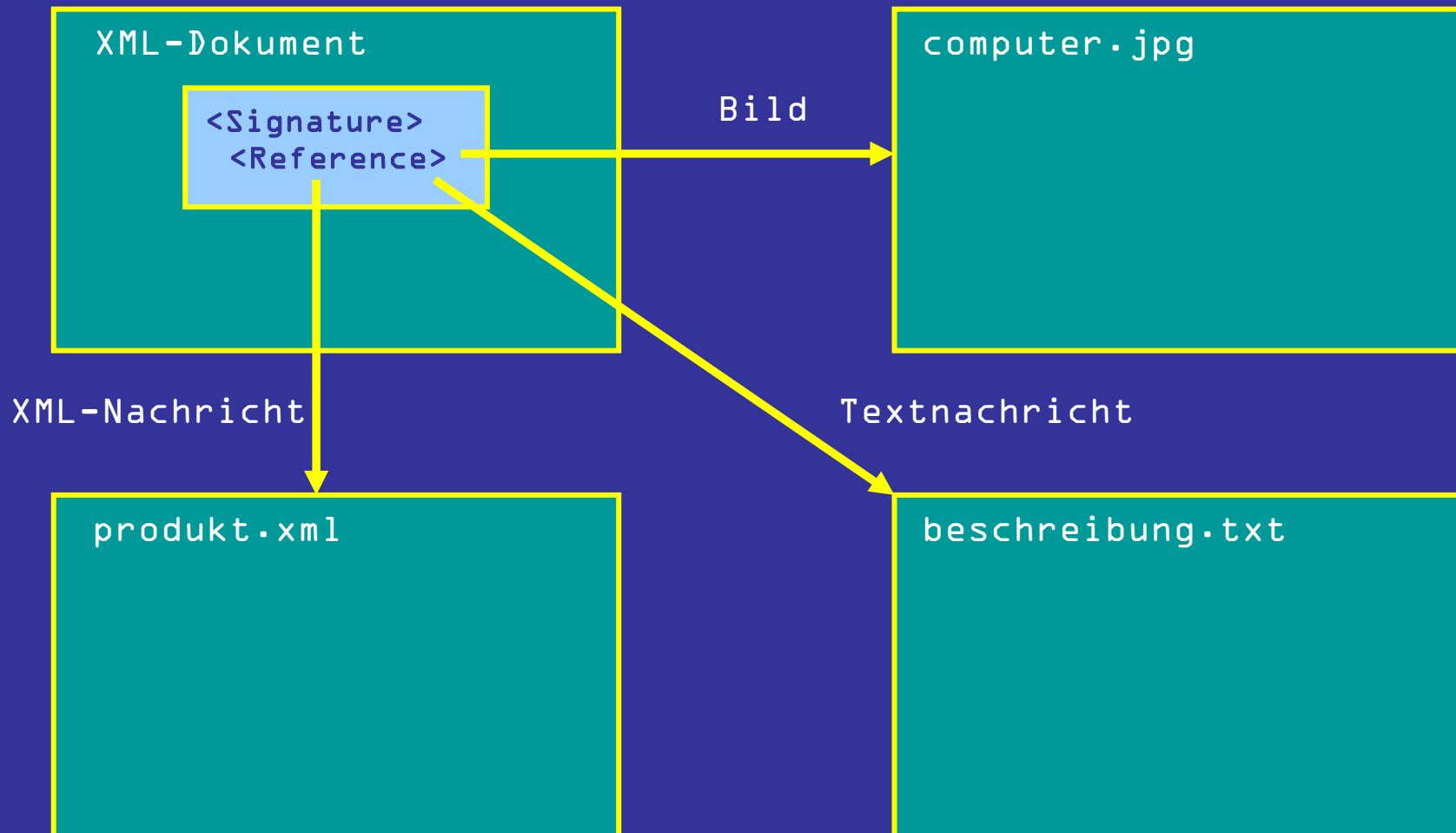
Referenz auf XML-Knoten,
XML-Dokument oder andere

Signatur

Schlüsselinfos
z.B.:öffentliches S. +
Zertifikat

Detached Signatur

◆ Signature-Referenzierung



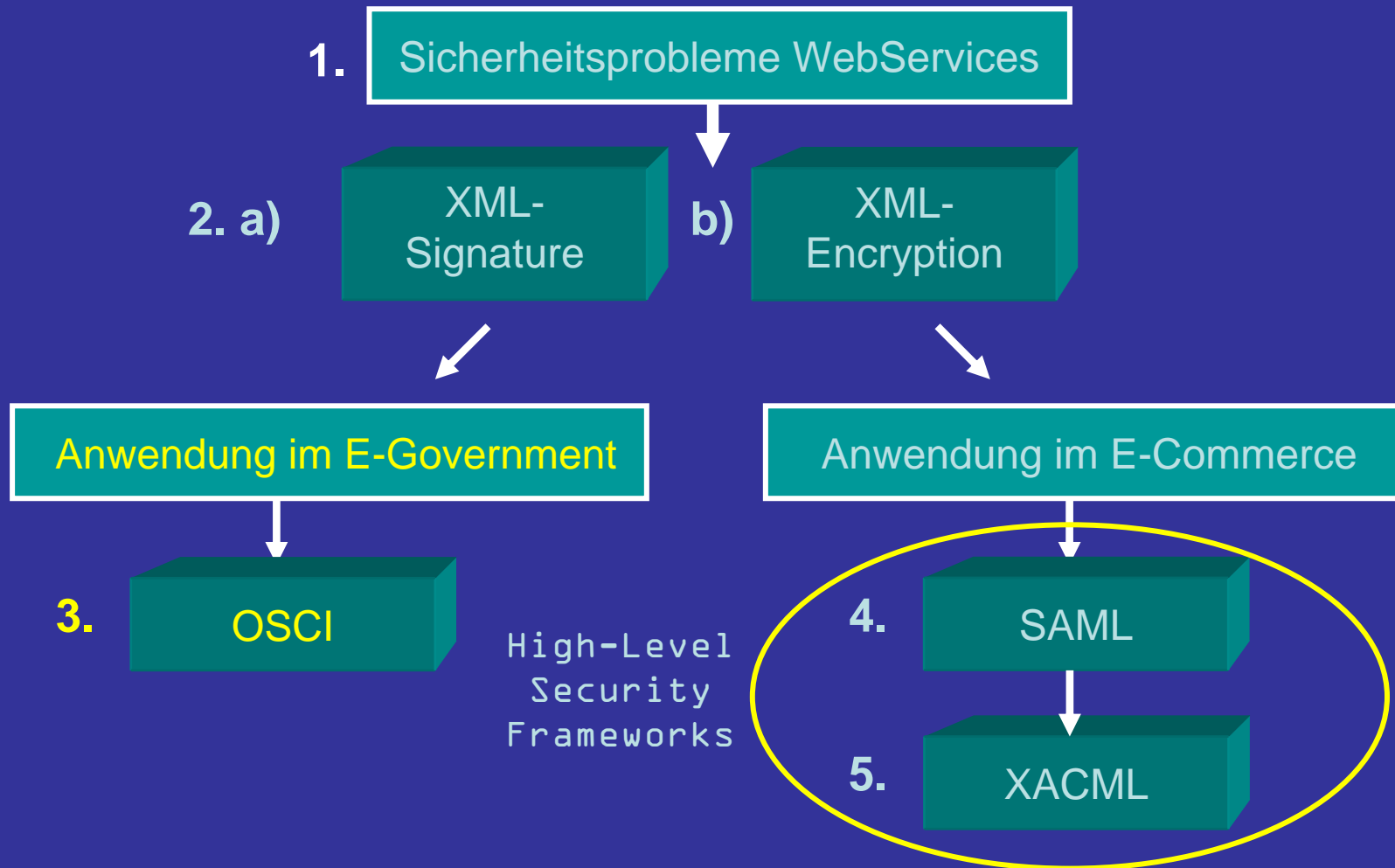
◆ XML-Encryption

- 2002 vom **W3C** standardisiert
- Syntax zur **Verschlüsselung**
- Sicherheitsziele:
 - **Vertraulichkeit** der Nachricht
- Verschlüsselungsraum:
 - **kompletter** XML-Baum (XML-Dokument)
 - **einzelne Elemente** oder **Anlagen** (z.B. Bilder)
- **keine** Schlüsselmanagement-Aufgaben
- Beschreibung des Schlüssels wird mitgeschickt
 - **PKI**: **öffentlicher** Schlüssel des **Empfängers**
 - **Hybridverfahren**: erst wird ein **Sitzungsschlüssel** ausgehandelt und dieser dann per **PKI** ausgetauscht

◆ Canonical XML

- Methode zur Generierung eines **einheitlichen XML-Dokumentes**
 - verhindert syntaktische Inkonsistenzen
 - schafft standardisierte Form für das Signieren und Verschlüsseln
- Einsatz von Prozessoren
- z.B. Entfernen von Whitespaces um **Hashfunktionen** anwenden zu können (bei XML-Signature)

◆ Roadmap



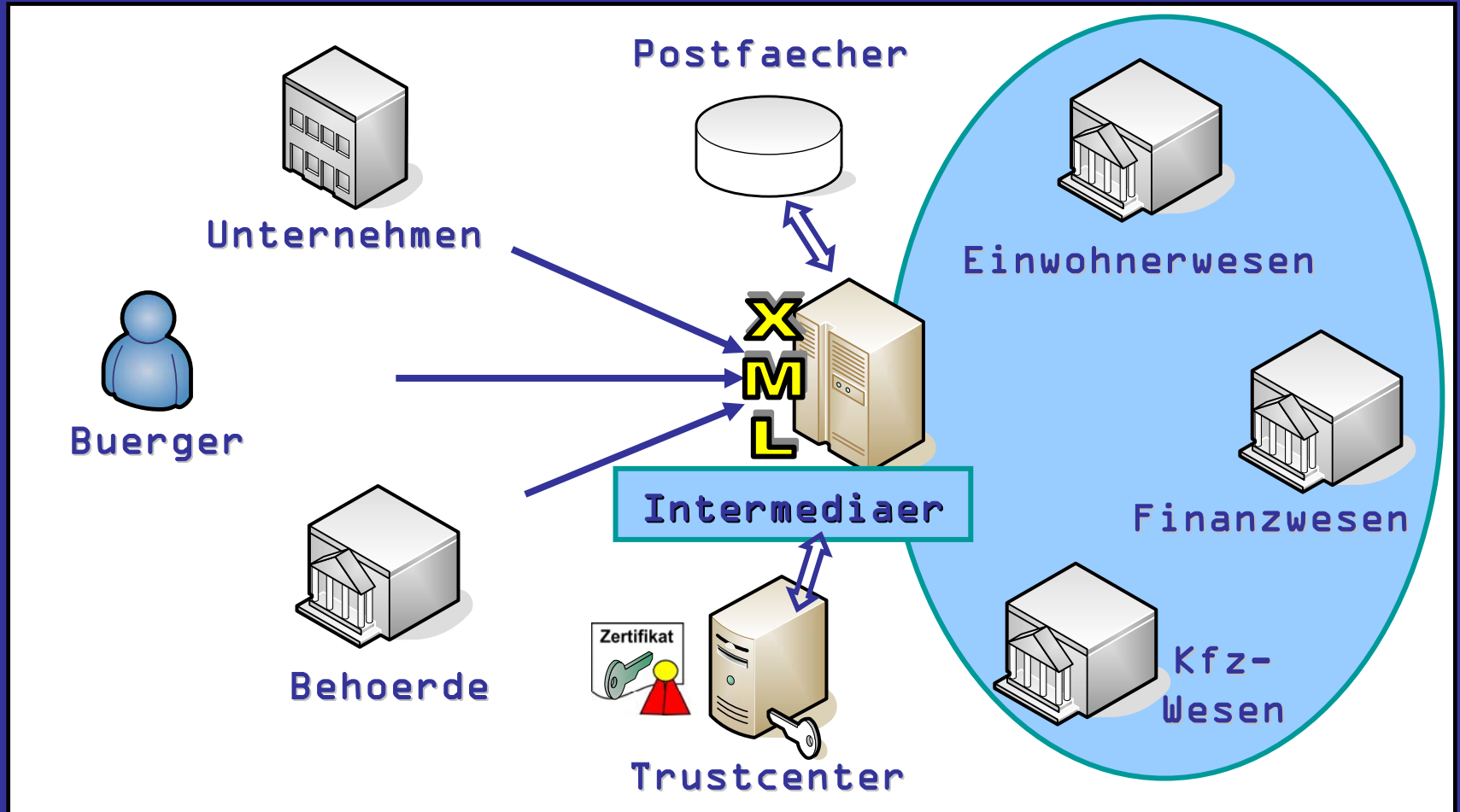
◆ Open Services Computer Interface (OSCI)

- Ziel des E-Government: mehr **Bürgernähe** und **Vertrauen** durch schnelle, **transparente** und **sichere** Kommunikation
- Sicherheitsinfrastruktur für
 - **Vertraulichkeit** (Ende-zu-Ende **Verschlüsselung** mit **XML-Encryption**)
 - **Authentizität** (elektronische Signatur mit **XML-Signature**)
 - **Integrität** (elektronische Signatur **XML-Signature**)
 - **Verbindlichkeit** (**Laufzettel** und **Quittungen** → beidseitig)
- Zentrale Rolle spielt der Intermediär
 - **Zwischenspeichern** von Nachrichten → **asynchrone** Kommunikation
 - **Fortschreiben** des Laufzettels **ohne** Zugriff auf Inhaltsdaten
 - erbringt **Mehrwertdienst** (Signaturprüfung)
- alle Sicherheitsmechanismen von OSCI sichergestellt
 - **keine** Si-Anforderungen an darunterliegende Transportprotokolle



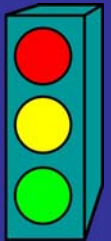
Intermediaer erbringt Mehrwertdienste

OSCI Infrastruktur

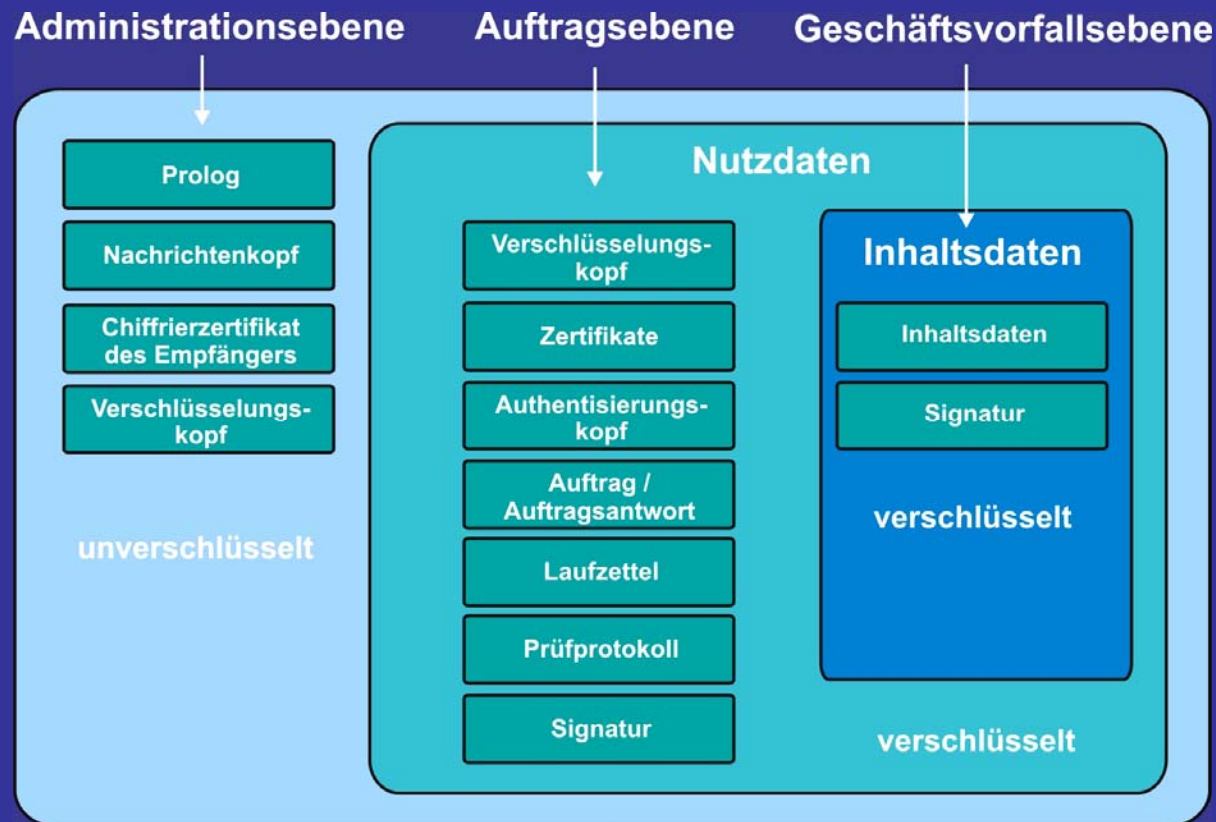


◆ OSCI - II

- medienbruchfreie Weiterverarbeitung der Daten → XML
- Nutzung von SOAP als Transportprotokoll
- Prinzip des doppelten Umschlags
- Intermediär == sicherer Mailserver
 - hält Postfächer für Empfänger, muss sich vorher authentifizieren
 - Postfach + Identität mit X.509-Zertifikat an K-Partner gebunden (SigG)
 - überträgt jede Nachricht (falsche Signatur, ohne Verschlüsselung usw.)
 - Empfänger entscheidet den weiteren Verwendungsweg
 - Laufzettel erweitert um Prüfprotokoll (Ergebnisse der Sig-Prüfungen)



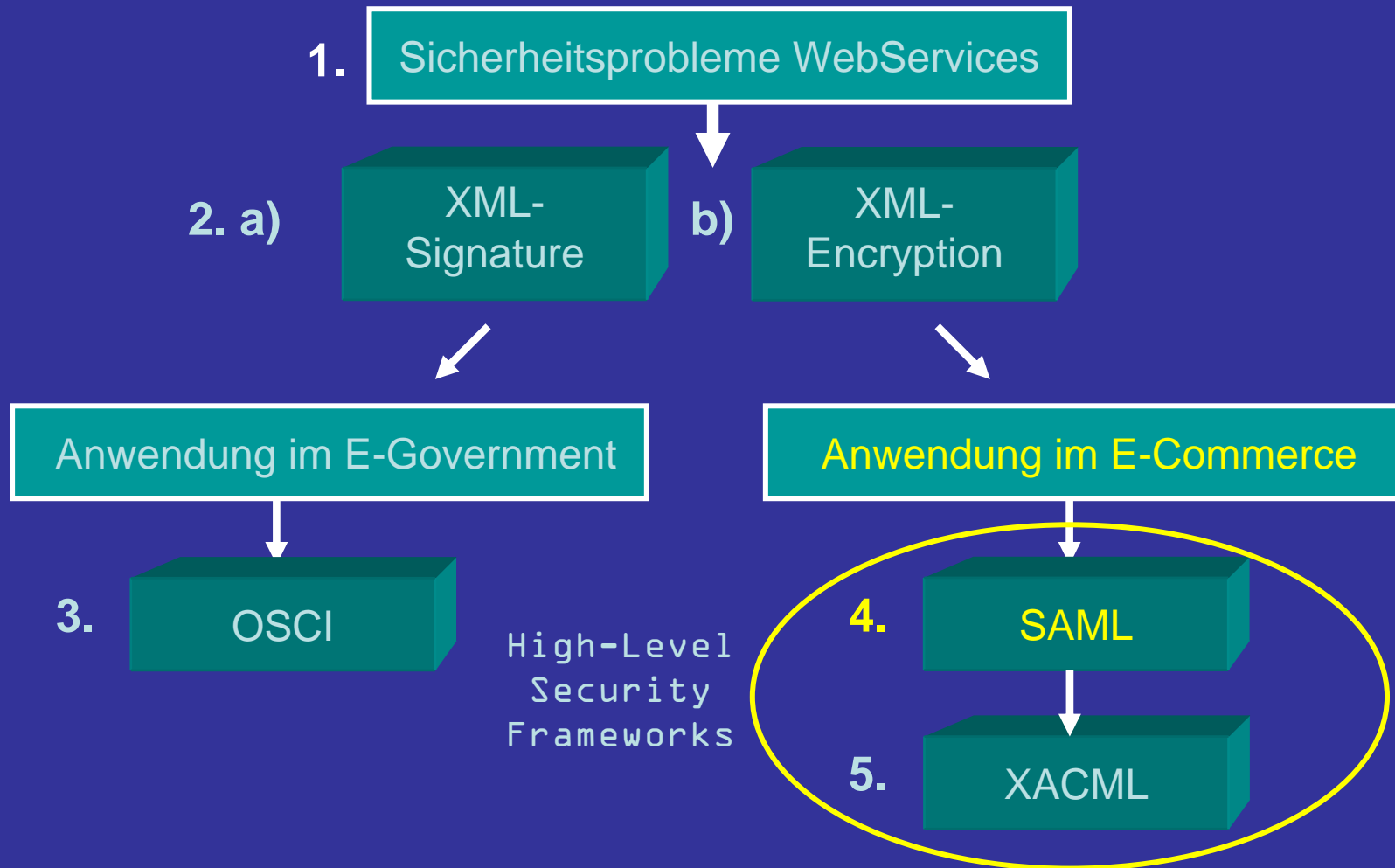
◆ OSCI – Nachrichtenstruktur



◆ OSCI – Ebenen



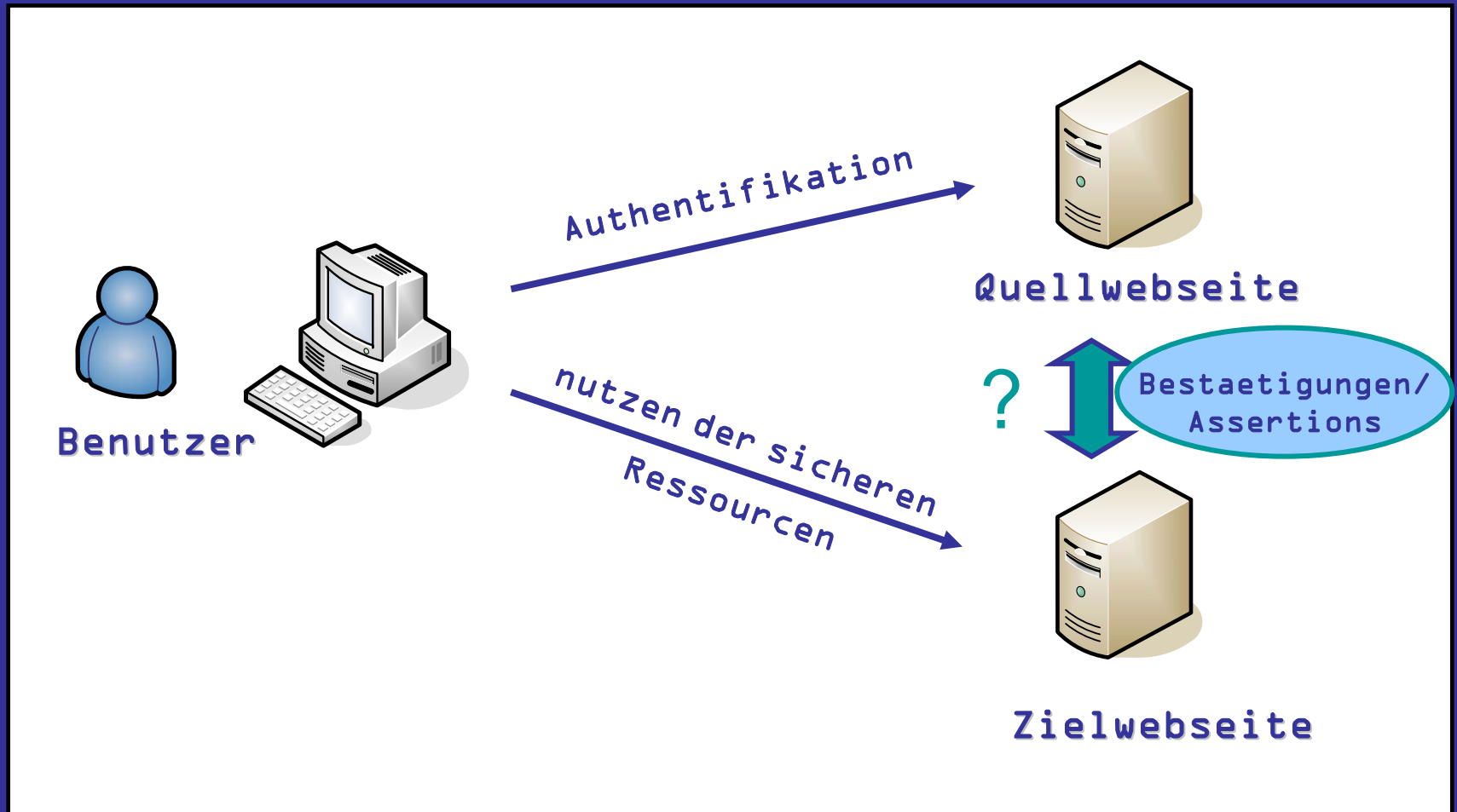
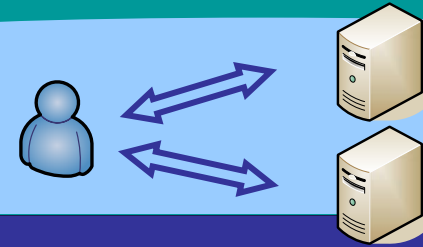
◆ Roadmap



◆ Security Assertion Markup Language (SAML)

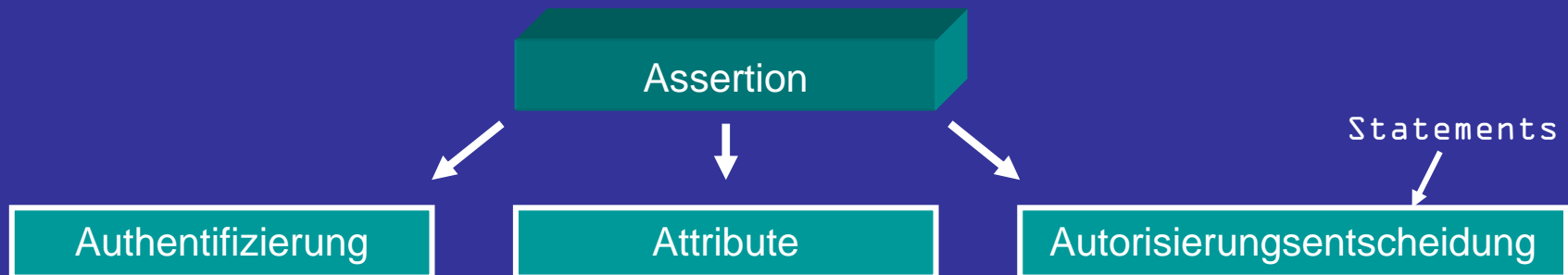
- **OASIS-Standard** (IBM, SAP, Nokia, Sun, kein MS)
- **XML-Framework** zum Austausch von **Sicherheitsinformationen**
- definiert **Dokumentstruktur** von Si-Infos zwischen Services
- erlaubt **Interaktion** zwischen sicheren **Services**
- benutzt **Sicherheitsbehauptungen** (assertions)
- stellt einen Schablone dar für den Entwurf von skalierbaren und förderierten Systeme mit Webinfrastruktur
- XML-basiertes **Request/Response-Protokoll**
- **Use-Cases:**
 1. Single Sign On
 2. Distributed Transactions
 3. Authorization Services

◆ Szenario – Single Sign On



◆ SAML Konzepte – Bestätigungen (Assertions)

- **Bestätigung:** Informationspaket das ein oder mehrere **Statements** enthält. Ein XML-Schema enthält das festgeschriebene Format.
 - sind ein **Fakt** über eine **Person** oder Programm



◆ SAML Bestätigung – Struktur

Authentifizierung

- **Authentifizierung:** eine Bestätigungsinstanz bestätigt das ein **Subjekt** zu einer bestimmten Zeit **authentifiziert** wurde („**Master Giz**“ hat sich an der Domain „**www.hs-harz.de**“ authentifiziert)

Attribute

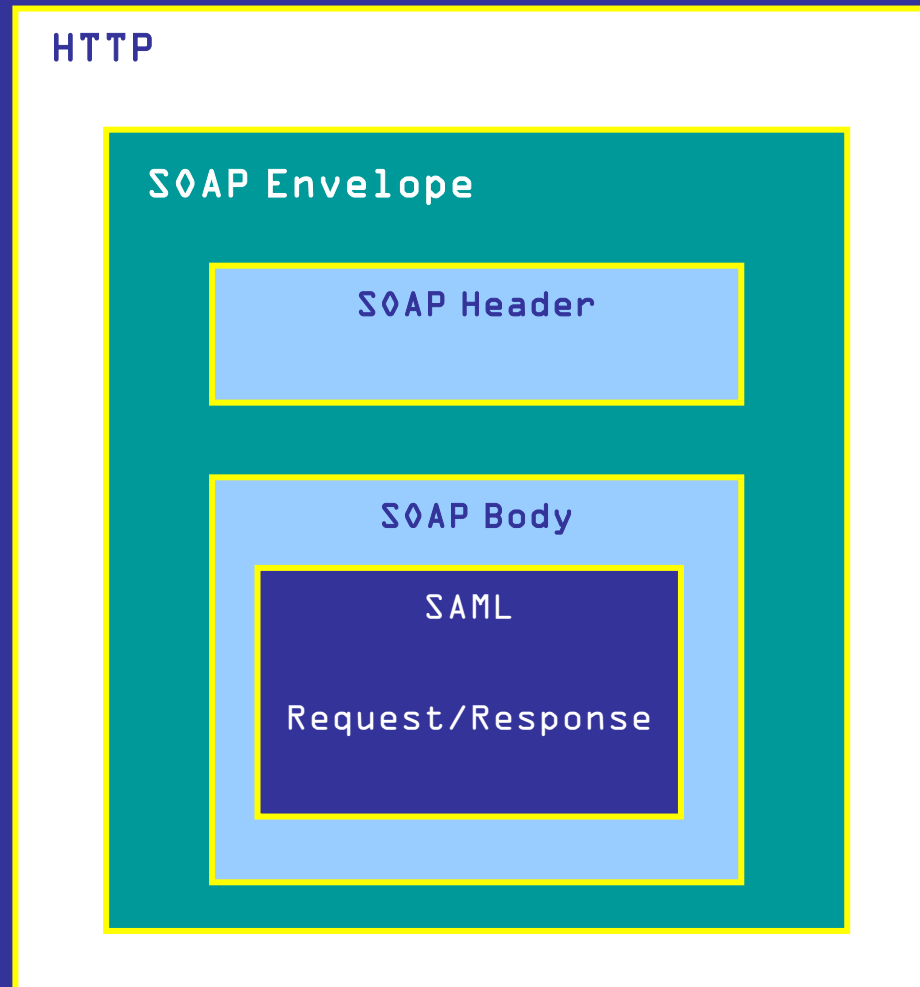
- **Attribute:** bestätigt das zu einem Subjekt eine bestimmte **Eigenschaft** gehört („**Master Giz**“ gehört zur „**Studentenschaft**“)

Autorisierungsentscheidung

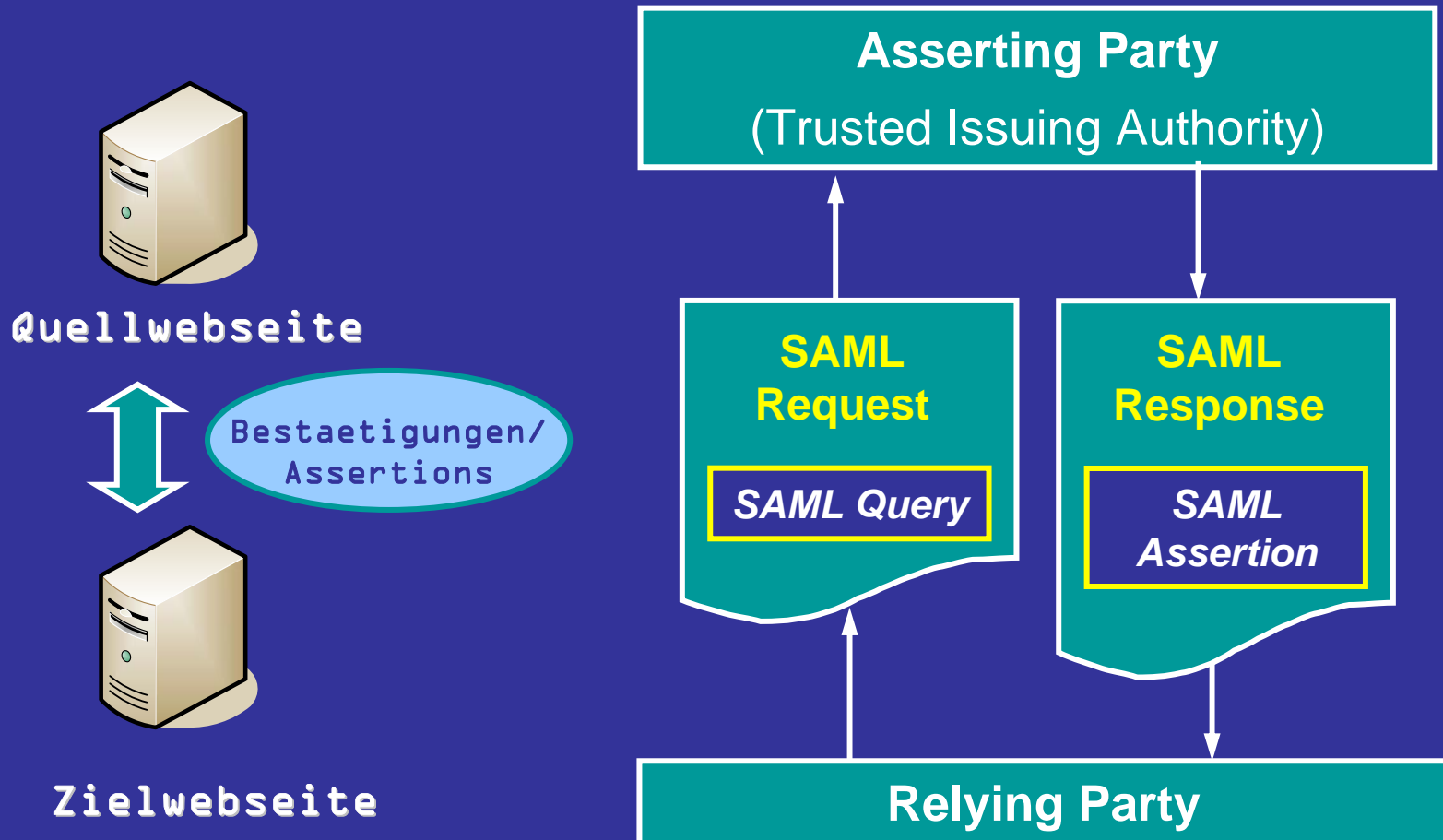
- **Autorisierungsentscheidung:** bestätigt den Zugriff eines Subjektes auf eine **Ressource** mit bestimmten Zugriffsrecht

Sicherheit vereinbart im Header

◆ SAML – SOAP over HTTP-Binding



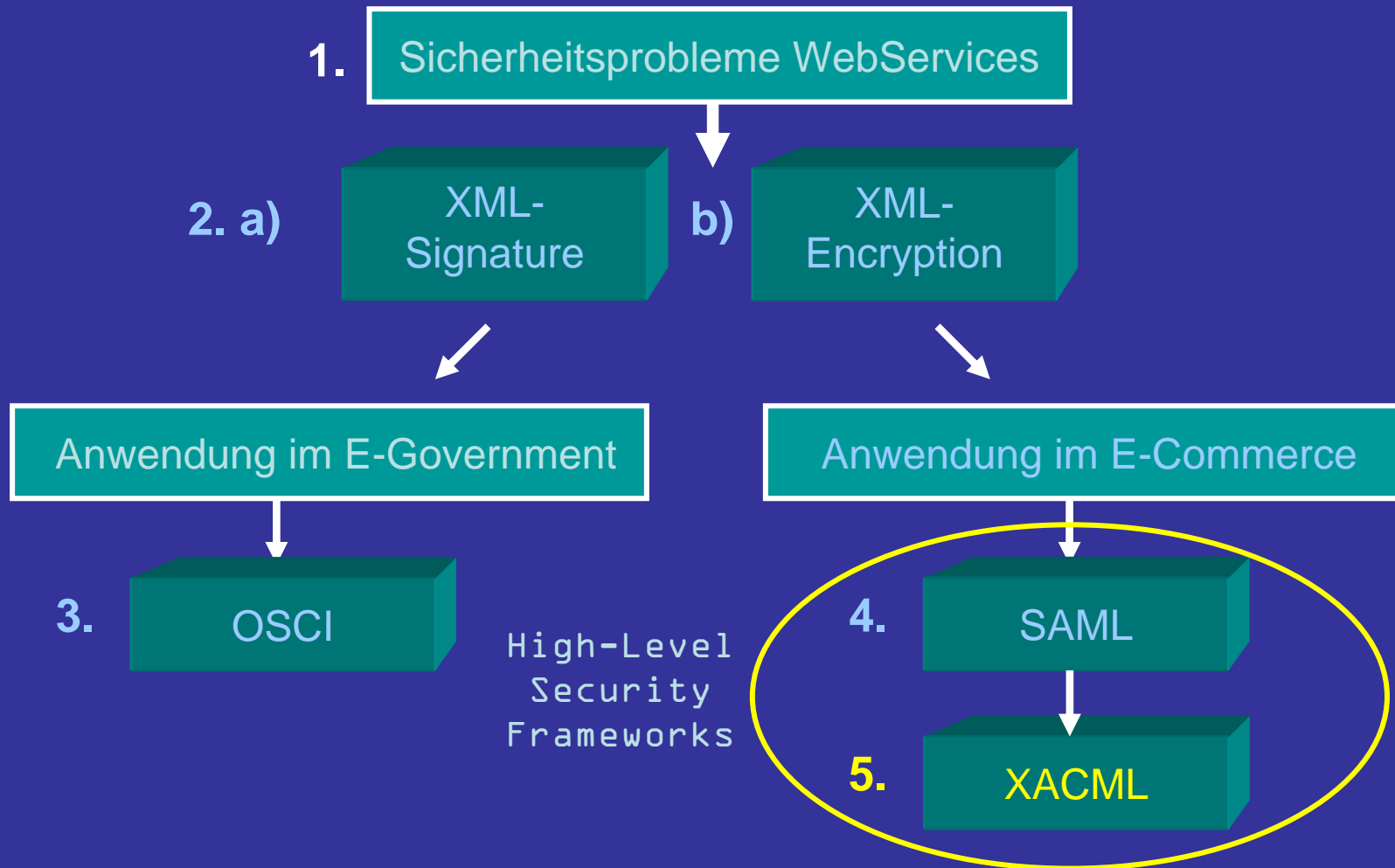
◆ SAML Protokoll um Bestätigung zu erhalten



◆ Beispiel für ein Authorization Statement

```
<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="hs-harz.de"
        Name="Master Giz" />
      <saml:ConfirmationMethod>
        http://...core-25/sender-vouches
      </saml:ConfirmationMethod>
    </saml:Subject>
  </saml:AuthenticationStatement>
</saml:Assertion>
```

◆ Roadmap

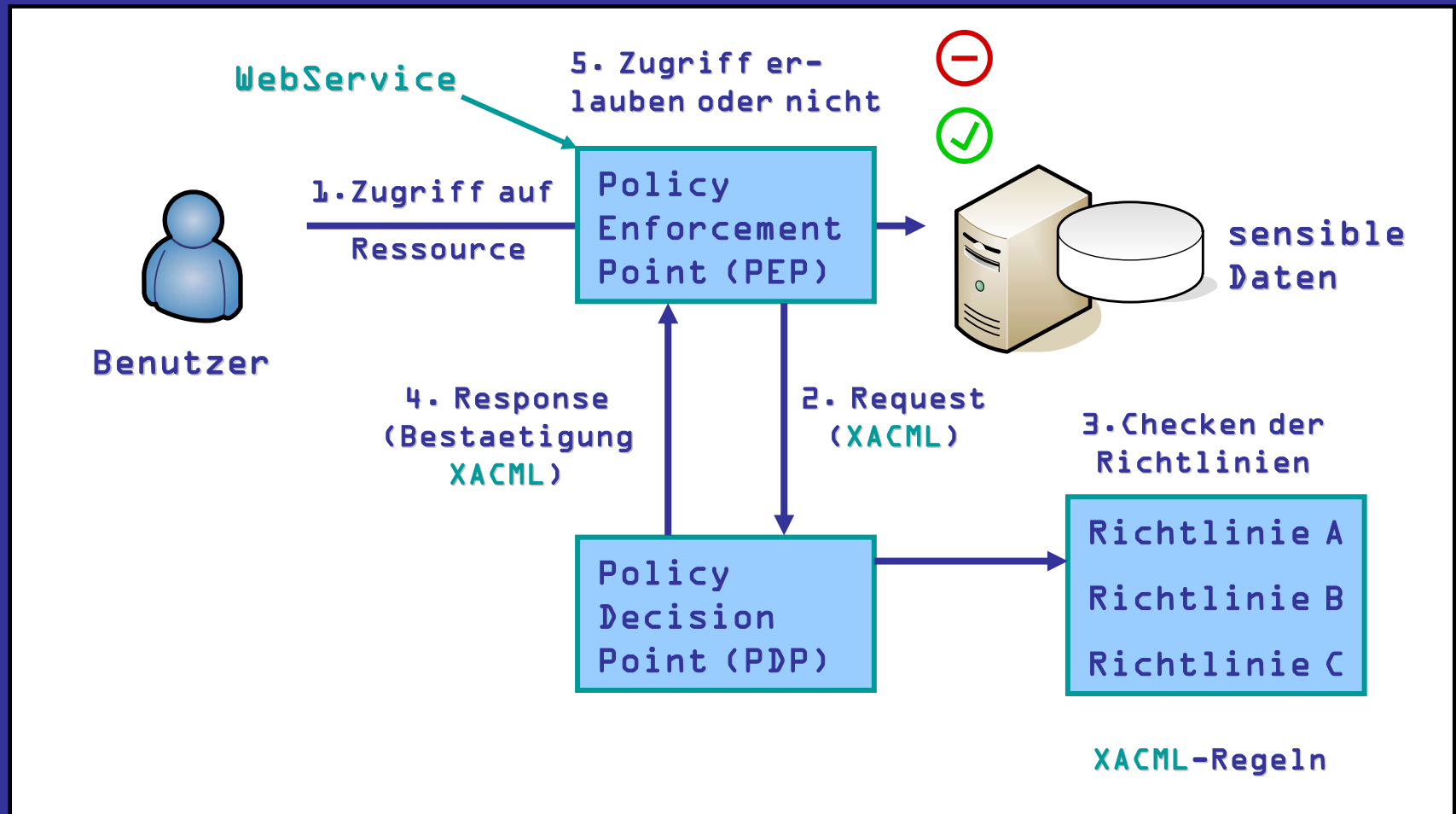


◆ eXtensible Access Control Markup Language (XACML)

- spezifiziert von der **OASIS** (Organization for the Advancement of Structured Information Standards)
- schützt **Inhalt** vor unautorisierten **Veränderungen** beim **Datenaustausch** zwischen Unternehmen
- Erweiterter XML-Syntax zur gemeinsamen Dokumentennutzung
 - **Attribute** und **Tags** beschreiben Inhalte
- unterstützt Firmen beim Erstellen und Ausrollen von **Autorisierungsrichtlinien** (authorization policies)
- Sun's XACML 1.0 Implementierung als Referenz

Zugriffsregeln definieren & abfragen

Zugriffskontrolle mit XACML



◆ Literaturverzeichnis

- *Rosenberg J. / Remy D. :*
Securing Web Services with WS-Security, SAMS Mai 2004
- **OASIS** - Organization for the Advancement of Structured Information Standards
<http://www.oasis-open.org/home/index.php> 10.06.2005

Fragen?



Antworten!

Vielen Dank für ihre Aufmerksamkeit!

Tobias Giese

20.06.2005

Secure WebServices

Kontakt:

tobias.giese@gmx.de



Unterlagen:

<http://www.tobias-giese.de/secure-webservices.pdf>