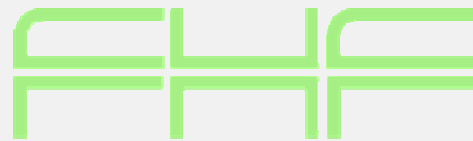


# Grundlagen, Standards und Einsatz sicherer Web Services

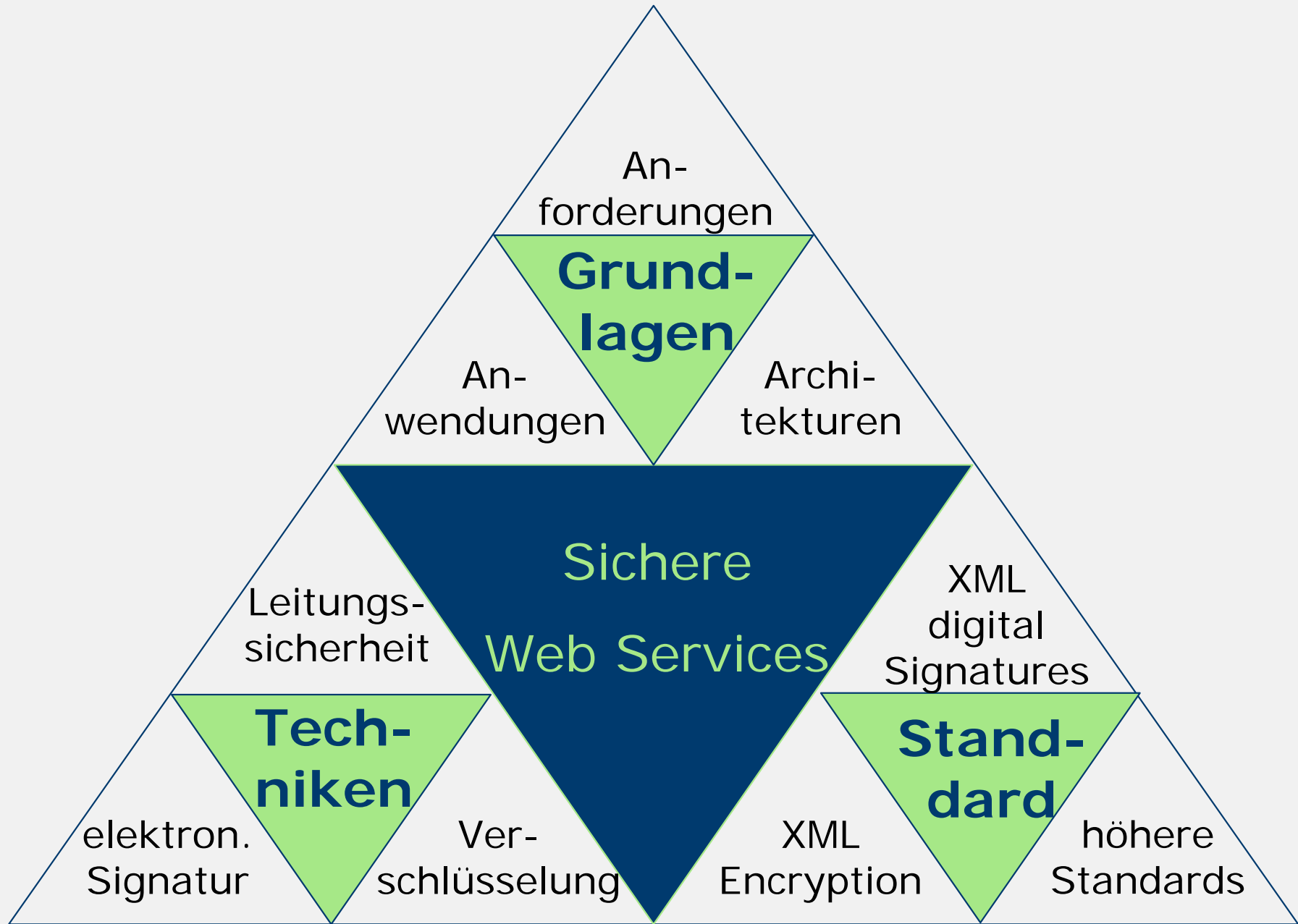


Prof. Mario Jeckle

Fachhochschule Furtwangen

[mario@jeckle.de](mailto:mario@jeckle.de)

<http://www.jeckle.de>



# Grundlagen: Anforderungen

- **Vertraulichkeit** (confidentiality)  
Schutz der Daten vor dem (lesenden) Zugriff unbefugter Dritter
- **Berechtigung** (authorization)  
Gewährleistet Befugnis des Anforderers zur Nutzung des Diensten
- **(Daten-)Konsistenz** (data integrity)  
Verlangt modifikationsfreies Eintreffen der versandten Daten
- **Glaubwürdigkeit des Ursprungs**  
(message origin authentication)  
Garantiert, daß eine Nachricht willentlich durch einen Sender erstellt wurde
- **Verbindlichkeit** (non-repudiation)  
Stellt sicher, daß der Sender die Autorenschaft nicht leugnen kann

# Grundlagen: Anforderungen

## SOAP Envelope

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <e:CallerID
      xmlns:e="http://example.org/callerID"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <e:reference>uuid:af271da6-3ef6-436d-86da-a0333d0535f0</e:reference>
      <e:dateAndTime>2003-10-20T07:48:00.000+01:00</e:dateAndTime>
    </e:CallerID>
  </env:Header>
  <env:Body>
    <m:message xmlns:m="http://example.org/message">
      Das Pferd frisst keinen Gurkensalat
    </m:message>
  </env:Body>
</env:Envelope>
```

H  
E  
A  
D  
E  
RB  
O  
D  
Y

# Grundlagen: Anwendungen

- **Vertraulichkeit** (confidentiality)
  - Schutzwürdige Daten im Sinne des BDSG
  - (Staats-)Geheimnisse, Patientendaten, Gehaltsdaten
- **Berechtigung** (authorization)
  - Zugriff auf Systeme und Daten
  - Zugangsberechtigungen
- **(Daten-)Konsistenz** (data integrity)
  - Geschäftsverkehr: Verträge, Angebote, Bestellungen
  - Langzeitarchivierung, Dokumentation
- **Glaubwürdigkeit des Ursprungs**  
(message origin authentication)
  - Geschäftsverkehr
  - Jegliche vertrauenswürdige Kommunikation (-> SPAM)
- **Verbindlichkeit** (non-repudiation)
  - Bestellungen
  - Gerichtsfeste Dokumente

# Grundlagen: Anwendungen

- Grundsätzlich: Einsatz kryptographischer Techniken
  - Verschlüsselung
  - davon abgeleitet: Digitale Signatur
- Prinzip: Erweiterung oder Modifikation der zu übertragenden Daten um gewünschten Anforderungen zu genügen
- Technik: Verschiedene mathematische Verfahren
  - Schlüsselaustausch nach Diffie-Hellman
  - Rivest-Shamir-Adleman (RSA)
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard/Rijndael
  - IDEA
  - Twofish
  - ...

# Grundlagen: Anwendungen

## Verschlüsselung:

- **Symmetrisch:** Sender und Empfänger besitzen den selben Schlüssel
- **Asymmetrisch:** Sender und Empfänger besitzen verschiedene Schlüssel

## Vorgehen (Verschlüsselung):

### 1. (unchiffrierter) Klartext

Das Pferd frisst keinen Gurkensalat

### 2. Geheimer Schlüssel

SECRETKEYWORD

### 3. Chiffre (=verschlüsselter Klartext)

Veu Gjxbh dnwjl ogzrxx Ksnyvqkenrx

# Grundlagen: Anwendungen

## Verschlüsselung:

- **Symmetrisch:** Sender und Empfänger besitzen den selben Schlüssel
- **Asymmetrisch:** Sender und Empfänger besitzen verschiedene Schlüssel

## Vorgehen (Entschlüsselung):

### 1. Chiffre (=verschlüsselter Klartext)

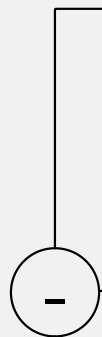
V e u G j x b h ...  
21 4 20 6 9 23 27 7 ...

### 2. Geheimer Schlüssel

S E C R E T K E Y W O R D  
18 4 2 17 4 19 10 4 24 22 14 17 4

### 3. (unchiffrierter) Klartext

D a s P f e r d ...  
3 0 18 15 5 4 17 3 ...





# Grundlagen: Anwendungen

## Verschlüsselung:

- Symmetrisch: Sender und Empfänger besitzen den selben Schlüssel
- **Asymmetrisch:**  
Sender und Empfänger besitzen verschiedene Schlüssel

## Vorgehen (Verschlüsselung):

### 1. (unchiffrierter) Klartext

Das Pferd frisst keinen Gurkensalat

### 2. Geheimer *Verschlüsselungsschlüssel*

YOUDONTKNOW

### 3. Chiffre (= verschlüsselter Klartext)

Bom Strkn sfeggn nsvgoa Uqppyqgnekq

## Prinzip:

$$D+Y=B$$

$$a+0=o$$

...

# Grundlagen: Anwendungen

## Verschlüsselung:

- Symmetrisch: Sender und Empfänger besitzen den selben Schlüssel
- **Asymmetrisch:**  
Sender und Empfänger besitzen verschiedene Schlüssel

## Vorgehen (Entschlüsselung):

### 1. Chiffre (= verschlüsselter Klartext)

Bom Strkn sfeggn nsvgoa Uqppyqgnekq

### 2. Geheimer *Entschlüsselungsschlüssel*

CMGWMMNHQNM

### 3. (unchiffrierter) Klartext

Das Pferd frisst keinen Gurkensalat

## Prinzip:

$$B+C=D$$

$$o+M=a$$

...

# Grundlagen: Anwendungen

## Verschlüsselung:

- Symmetrisch: Sender und Empfänger besitzen den selben Schlüssel

- **Asymmetrisch:**

Sender und Empfänger besitzen verschiedene Schlüssel

- Die beiden Schlüssel stehen in einem mathematischen Zusammenhang
- Die richtige Wahl des Verfahrens der Schlüsselerzeugung verhindert in der Praxis die Berechnung des einen Schlüssels aus dem anderen
- Zusammenhang im Beispiel:  
Einzelne Schlüsselkomponenten geben immer 26 und führen damit einen vollständigen Alphabetchlauf herbei.
- Bekannteste Anwendung asymmetrischer Verschlüsselung:
  - Pretty Good Privacy
  - Digitale Signaturen

# Grundlagen: Anwendungen

## Verschlüsselung:

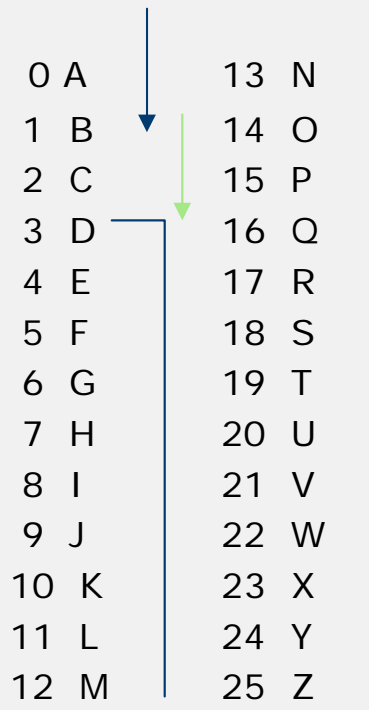
- Symmetrisch: Sender und Empfänger besitzen den selben Schlüssel
- **Asymmetrisch:**  
Sender und Empfänger besitzen verschiedene Schlüssel
- Zusammenhang im Beispiel:  
Einzelne Schlüsselkomponenten geben immer 26 und führen damit einen vollständigen Alphabetdurchlauf herbei.

Schlüssel 1:	Y	O	U	D	O	N	T	...
	24	14	20	3	14	13	19	...
Schlüssel 2:	C	M	G	W	M	N	H	...
	2	12	6	22	12	13	7	...
	<hr/>							...
	26	26	26	26	26	26	26	...

# Grundlagen: Anwendungen

## Verschlüsselung:

- Symmetrisch: Sender und Empfänger besitzen den selben Schlüssel
- **Asymmetrisch:**  
 Sender und Empfänger besitzen verschiedene Schlüssel
- Zusammenhang im Beispiel:  
 Einzelne Schlüsselkomponenten geben immer 26 und führen damit einen vollständigen Alphabetdurchlauf herbei.



D	a	s	Pferd	...
3	0	18	...	
Y	O	U	DONT	...
24	14	20	...	
B	o	m	Strkn	...
1	14	12	...	
C	M	G	W	...
2	12	6	22	...
D	a	s	Pferd	...
3	0	18	...	

Ver-  
schlüsselung

Ent-  
schlüsselung

# Grundlagen: Anwendungen

## Verschlüsselung:

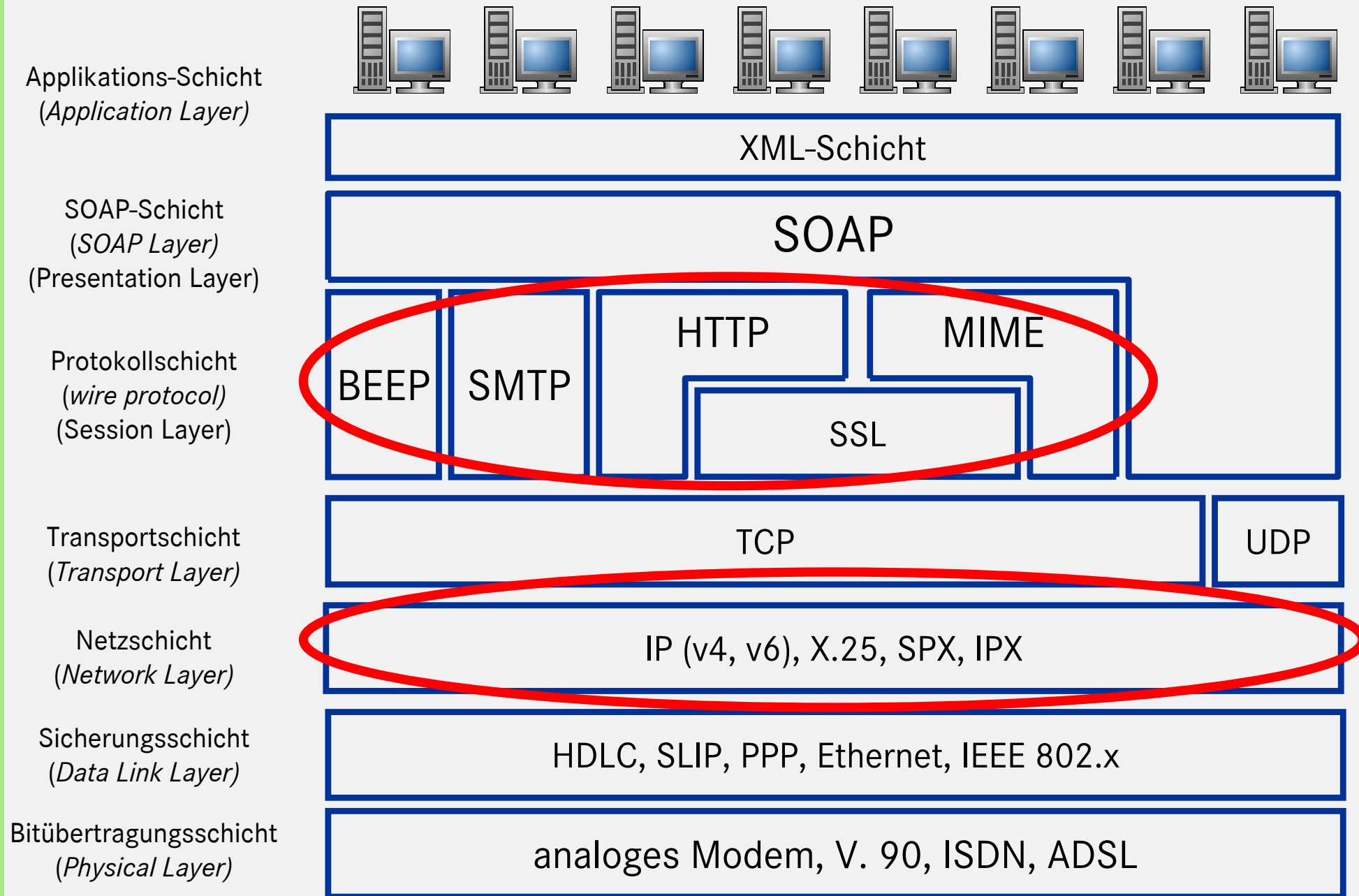
- Symmetrisch: Sender und Empfänger besitzen den selben Schlüssel

- **Asymmetrisch:**

Sender und Empfänger besitzen verschiedene Schlüssel

- Die beiden Schlüssel stehen in einem mathematischen Zusammenhang
- Die richtige Wahl des Verfahrens der Schlüsselerzeugung verhindert in der Praxis die Berechnung des einen Schlüssels aus dem anderen
- Zusammenhang im Beispiel:  
Einzelne Schlüsselkomponenten geben immer 26 und führen damit einen vollständigen Alphabetchlauf herbei.
- Bekannteste Anwendung asymmetrischer Verschlüsselung:
  - Pretty Good Privacy
  - Digitale Signaturen

# Techniken: Leitungssicherheit



# Techniken: Leitungssicherheit

- Bekannte Techniken zur Leitungssicherung
  - SSL/TLS
    - Sichert HTTP-Schicht
    - Erfüllt alle gesteckten Sicherheitsziele
  - IPSEC
    - Sichert IP-Schicht
    - Erfüllt alle gesteckten Sicherheitsziele



# Techniken: Leitungssicherheit

Einsatz von Leitungssicherheit für Web Services kaum geeignet ...

- SSL sichert nur Punkt zu Punkt Verbindungen
- Nur für bestimmte Übertragungsprotokolle einsetzbar
- Asynchrone Aufrufe nicht unterstützt
- Gesamte übertragene Daten werden gesichert

⇒ Intermediäre nicht mehr nutzbar

⇒ SOAP-Header nicht mehr auswertbar

# Techniken: Verschlüsselung

- Ziel:
  - Vertraulichkeitsschutz
- Ablauf:
  - Sender bearbeitet zu übertragende Daten so, daß sie ausschließlich für den intendierten Adressaten lesbar sind
  - Übertragung der so chiffrierten Daten
  - Empfänger entschlüsselt Daten

# Techniken: Verschlüsselung

```

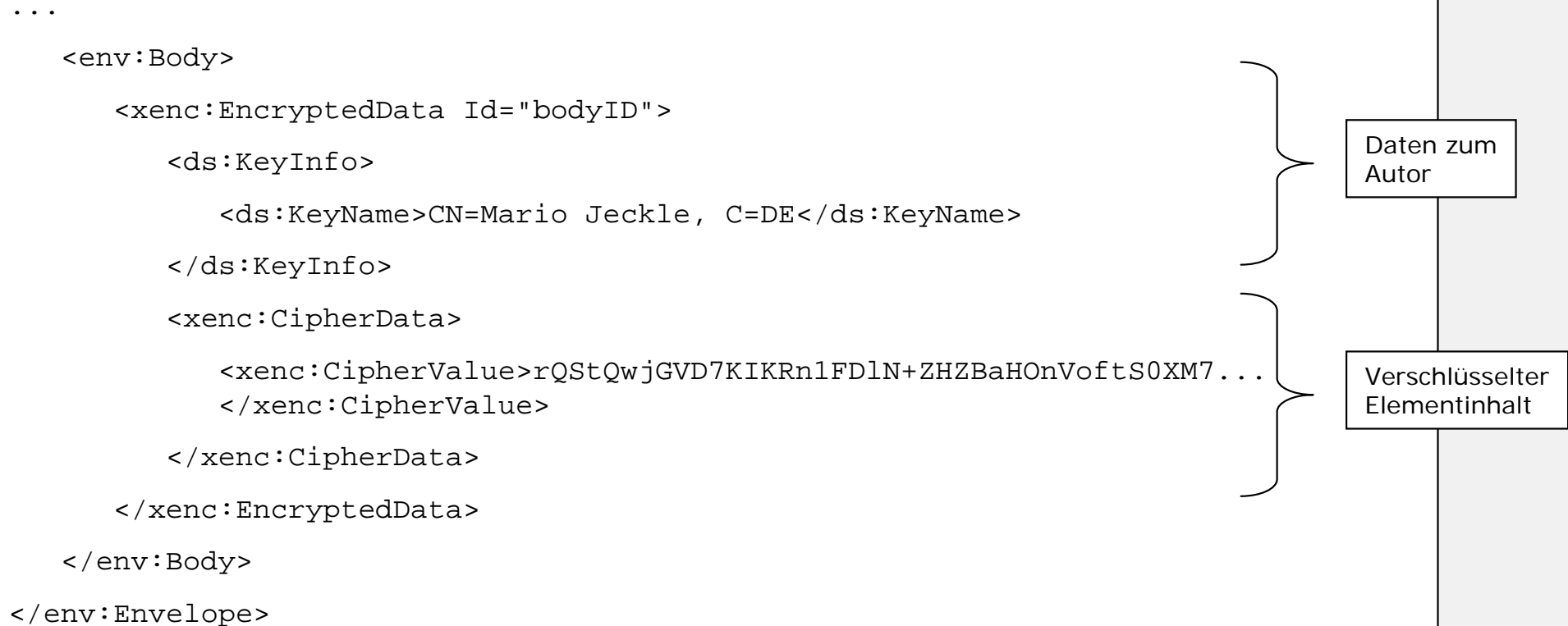
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope,"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
  <env:Header>
    <wsse:Security>
      <xenc:ReferenceList>
        <xenc:DataReference URI="#bodyID" />
      </xenc:ReferenceList>
    </wsse:Security>
    <e:CallerID
      xmlns:e="http://example.org/callerID"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <e:reference>uuid:af271da6-3ef6-436d-86da-a0333d0535f0</e:reference>
      <e:dateAndTime>2003-10-20T07:48:00.000+01:00</e:dateAndTime>
    </e:CallerID>
  </env:Header>
  ...
  
```

Verwaltungsdaten  
XML-Namensräume

XML-Security  
Verwaltungsdaten  
Referenz auf zu  
signierendes Element

Unsensible Daten  
verbleiben im  
Klartext

# Techniken: Verschlüsselung



# Techniken: Verschlüsselung

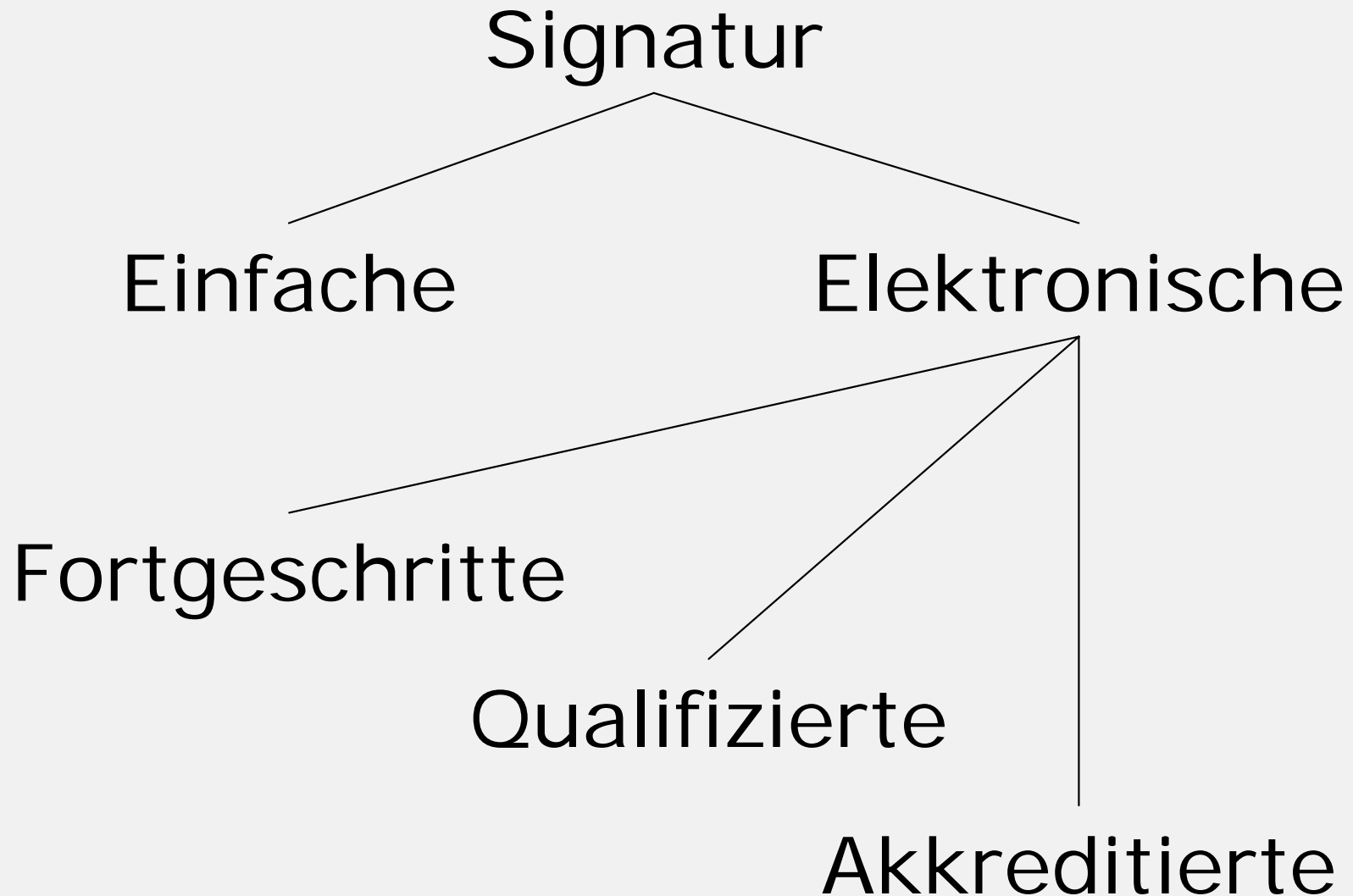
## Zusammenfassung

- Erreichung des gesteckten Ziels:  
Schutz der Vertraulichkeit
- Kryptographische Veränderung des  
Nutzeninhaltes  
(Bedarfsgesteuerte Anwendung möglich)
- Übertragung von verschlüsseltem Dokument  
und beschreibenden Metadaten
- Durch bestehende Umsetzungen  
(teilweise sogar als Open-Source verfügbar)  
leicht in existierende XML-Lösungen  
integrierbar

# Techniken: Elektronische Unterschrift

- Ziele:
  - Aufdeckung potentieller Datenverfälschung
  - Unbestreitbare Autorenschaft
  - Rechtliche Verbindlichkeit
- Ablauf:
  - Sender „unterschreibt“ zu übertragende Daten
  - Übertragung von Daten und Unterschrift
  - Empfänger prüft Unterschrift

# Techniken: Elektronische Unterschrift



# Elektronische Unterschrift: Gesetzliche Anforderungen

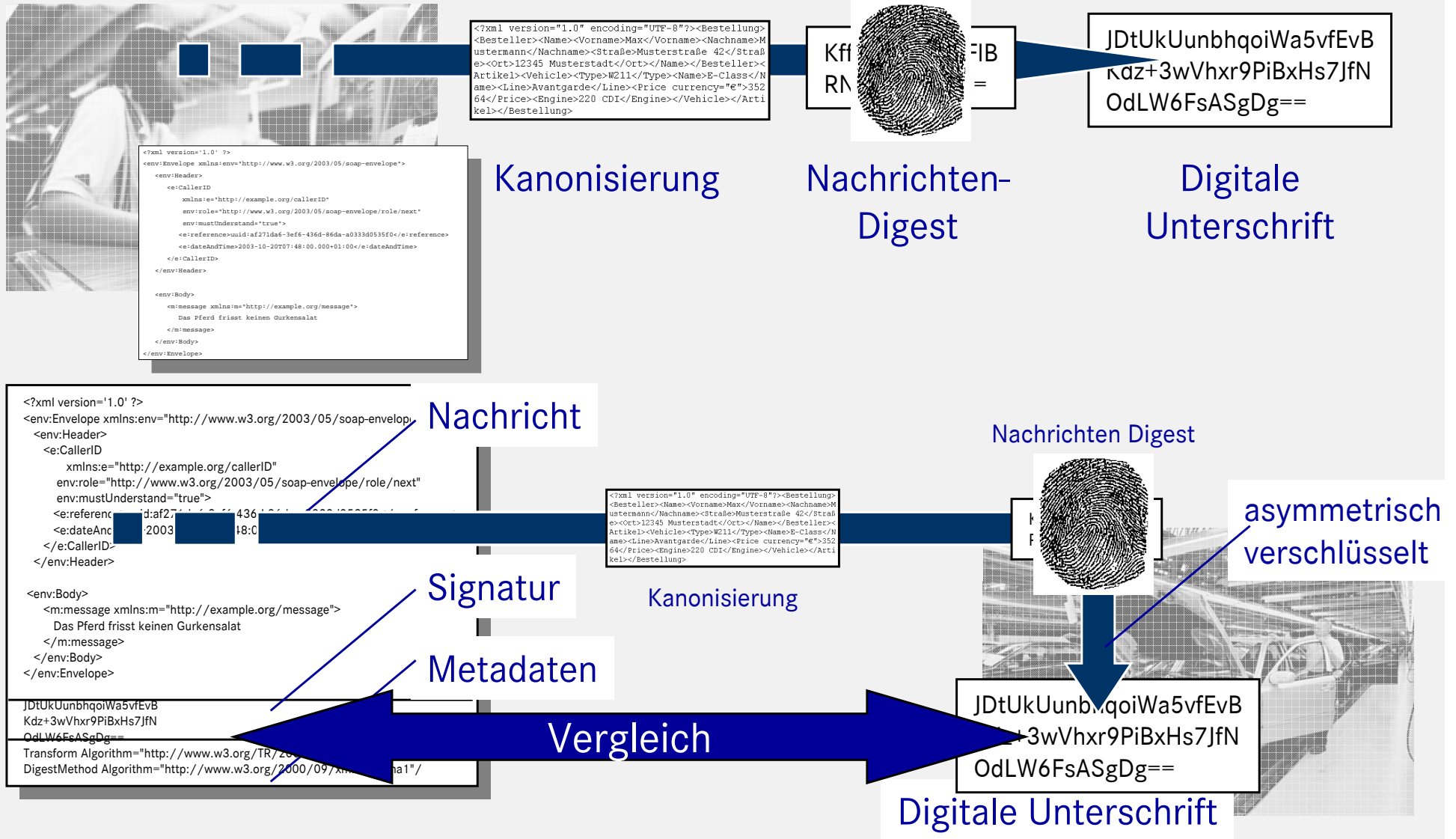
- „Sichere“ kryptographische Verfahren
- „Einmaliger“ (d.h. vertauschungssicherer) Schlüssel
- Zuverlässige Bindung von Schlüssel und Inhaber (PIN, biometrische Verfahren)
- Zuverlässige Prüfung (Zertifikatsprüfung)
- Neusignierung bereits signierter Inhalte
- Einsatz geeigneter technischer Komponenten
- Einweg Hashfunktion
- Behördlich genehmigte Zertifizierungsstellen
- Wissentlich und willentliche Signatur



# Elektronische Unterschrift: Gesetzliche Anforderungen

- Elektronische Signatur
  - § 2 Nr. 1 SigG
  - Keine Sicherheitsanforderungen
- Fortgeschrittene elektronische Signatur
  - § 2 Nr. 2 SigG
  - Ausschließliche Zuordnung an Unterzeichner
  - Identifizierung des Unterzeichners
  - Erzeugung unter alleiniger Kontrolle des Unterzeichners
  - Erkennbarkeit nachträglicher Veränderungen
  - Beispiel: Pretty good Privacy
- Qualifizierte elektronische Signatur
  - § 2 Nr. 3 SigG
  - Beruhen auf gültigem qualifiziertem Zertifikat
  - Inhaltliche Anforderungen an Zertifikat
  - Staatliche Aufsicht aber genehmigungsfrei
  - Haftung
- Akkreditierte elektronische Signatur
  - § 15 SigG
  - Wie qualifizierte Signatur
  - Vorabprüfung durch Zertifizierungsdienstanbieter
  - Gesonderte Prüfung der technischen Komponenten
  - RegTP stellt Wurzelzertifikat aus
  - Beispiel: Signtrust

# Techniken: Elektronische Unterschrift



# Standards: XML-Verschlüsselung

- XML Encryption Syntax and Processing (W3C Recommendation seit 2002-12-10)
- Web Service-Security (OASIS)  
SOAP Message Security (Standard 2004-04-06)
- Kennzeichen
  - Operieren ausschließlich auf der Applikationsebene und sind daher netzwerkseitig transparent
  - bilden ein Rahmenwerk
  - sind um Algorithmen erweiterbar
  - sind interoperabel (Signatur nach Verschlüsselung möglich und getestet)
  - XML Encryption gestatten auch Super-Encryption (d.h. Verschlüsselung bereits verschlüsselter Inhalte)
  - lassen Infrastruktur (z.B. PKI RFC 2459) außer Acht

# Standards: XML Signatur

- XML Digital Signatures  
(W3C Recommendation/IETF RFC 3072 seit 2002-02-12)
- Web Service-Security (OASIS)  
SOAP Message Security  
(Standard 2004-04-06)
- Kennzeichen
  - Operieren ausschließlich auf der Applikationsebene und sind daher netzwerkseitig transparent
  - bilden ein Rahmenwerk
  - sind um Algorithmen erweiterbar
  - sind interoperabel  
(Signatur nach Verschlüsselung möglich und getestet)
  - lassen Infrastruktur (z.B. PKI RFC 2459) außer Acht

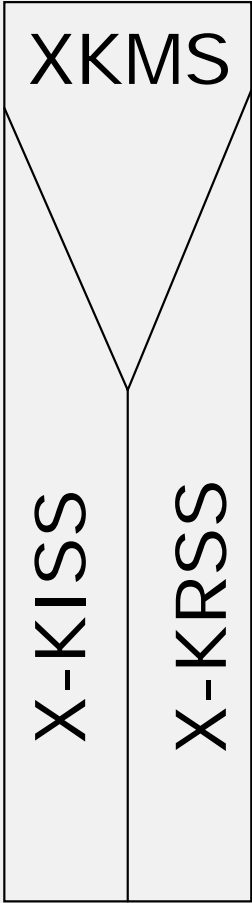
# Standards: höhere Standards

Extensible Access Control Markup Language  
(XACML)

Security Assertions Markup Language  
(SAML)

XML Digital Signatures

XML Encryption



# Standards: höhere Standards

- W3C:
  - XML Key Management (XKMS)  
Zugriffsprotokoll auf Schlüsselmanagementsdienst
- OASIS-Anwendungsstandards
  - Extensible Access Control Markup Language (XACML)  
Richtlinien (*policies*) für den Datenzugriff
  - Security Assertions Markup Language (SAML)  
Darstellung und Austausch von Authentisierungsdaten.

# Sichere Web Services: Erfahrungen

## .NET-Erfahrungen

- Bestehende API instabil und durch Service Pack geändert.
- Anfänglich keine Interoperabilität mit Java erzielbar.

# Sichere Web Services: Erfahrungen

## Anwenderakzeptanz und -toleranz

- Sicherheit ist ein sozio-technisches Problem.
- Sicherheit wird gefordert und ihr Vorhandensein generell begrüßt.
- Realisierte Lösung erfordert Identifikation durch tastaturbasierte Paßworteingabe.
- Manuelle Authentisierung und Autorisierung jedes entfernten Funktionsaufrufs nicht praktikabel.
- Sinnvollerweise Kopplung an Systemlogin (*single-sign-on*).



# Sichere Web Services: Erfahrungen

## Standards nutzen – Standards setzen

- Standard-basiertheit unbedingt sinnvoll.
- Plattformübergreifende Standards ermöglichen Einsatz der Technik überhaupt erst.
- Implementierungen noch in Kinderschuhen und daher mit Kinderkrankheiten behaftet.
- Konkrete Einsatzkonfiguration des Standards nicht standardisiert ausdrückbar.  
=> Erschwert Kontrahierung signifikant.  
*XML DSig Profile* könnte Lösung sein.

# XML DSig Profile

- Inhalt:
  - Genutzte Algorithmen per QName.
  - Verwendete Transformationen.
  - ...
- Anwendung:
  - Sinnvollerweise Ablage in UDDI, gemeinsam mit WSDL-formulierter Schnittstellenbeschreibung des Dienstes.
  - Ermöglicht gesicherten Erstkontakt.
- Existierender Ansatz:
  - *Web Service Policy*

# XML DSig Profile

## Web Service Policy

Ein Beispiel:

```
<wsp:Policy xmlns:wsse="..." xmlns:wsp="...">
  <wsp:ExactlyOne>
    <wsse:SecurityToken wsp:Usage="wsp:Required" wsp:Preference="1">
      <wsse:TokenType>wsse:Kerberosv5TGT</wsse:TokenType>
    </wsse:SecurityToken>
    <wsp:All wsp:Preference="100">
      <wsse:SecurityToken>
        <wsse:TokenType>wsse:X509v3</wsse:TokenType>
      </wsse:SecurityToken>
      <wsse:Integrity>
        <wsse:Algorithm Type="wsse:AlgEncryption"
          URI="http://www.w3.org/2001/04/xmlenc#3des-cbc" />
      </wsse:Integrity>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

# Sichere Web Services Empfehlungen

- Schulung und Sensibilisierung der Stakeholder.
- Bereitstellung der notwendigen Applikationsschnittstellen
- Identifikation der organisatorischen Ansatzpunkte (Bereiche, Prozesse, Datenflüsse)
- Festlegung des notwendigen Sicherheitsgrades
- Auswahl geeigneter Verfahren für Signatur und Verschlüsselung
- Prozeßimplementierung  
Schlüssel- und Zertifikatsverwaltung
- Faustregel: Mindestens fortgeschrittene elektronische Signatur – Notwendigkeit von Verschlüsselung prüfen.

# Sichere Web Services

## Empfehlungen

- Gleichzeitige Realisierung von Sicherheitsmechanismen auf verschiedenen Kommunikationsschichten ist kein Widerspruch
- SSL zwar weit verbreitet aber inadäquat für serviceorientierte Verarbeitung bzw. in der technischen Realisierung zuweilen lückenhaft  
=> Einsatz von TLS und/oder S-HTTP
- Sollte Existenz der Kommunikation bereits ein sicherheitsrelevantes Datum darstellen  
=> Einsatz von Netzsicherheitsmaßnahmen wie IPSec oder VPN

