



Semantic Web

Vorlesung

Dr. Harald Sack

Hasso-Plattner-Institut für Softwaresystemtechnik

Universität Potsdam

Wintersemester 2008/09



<http://sw0809.blogspot.com/>

Blog zur Vorlesung: <http://sw0809.blogspot.com/>

Die nichtkommerzielle Vervielfältigung, Verbreitung und Bearbeitung dieser Folien ist zulässig
(Lizenzbestimmungen CC-BY-NC).

Semantic Web - Vorlesungsinhalt

2



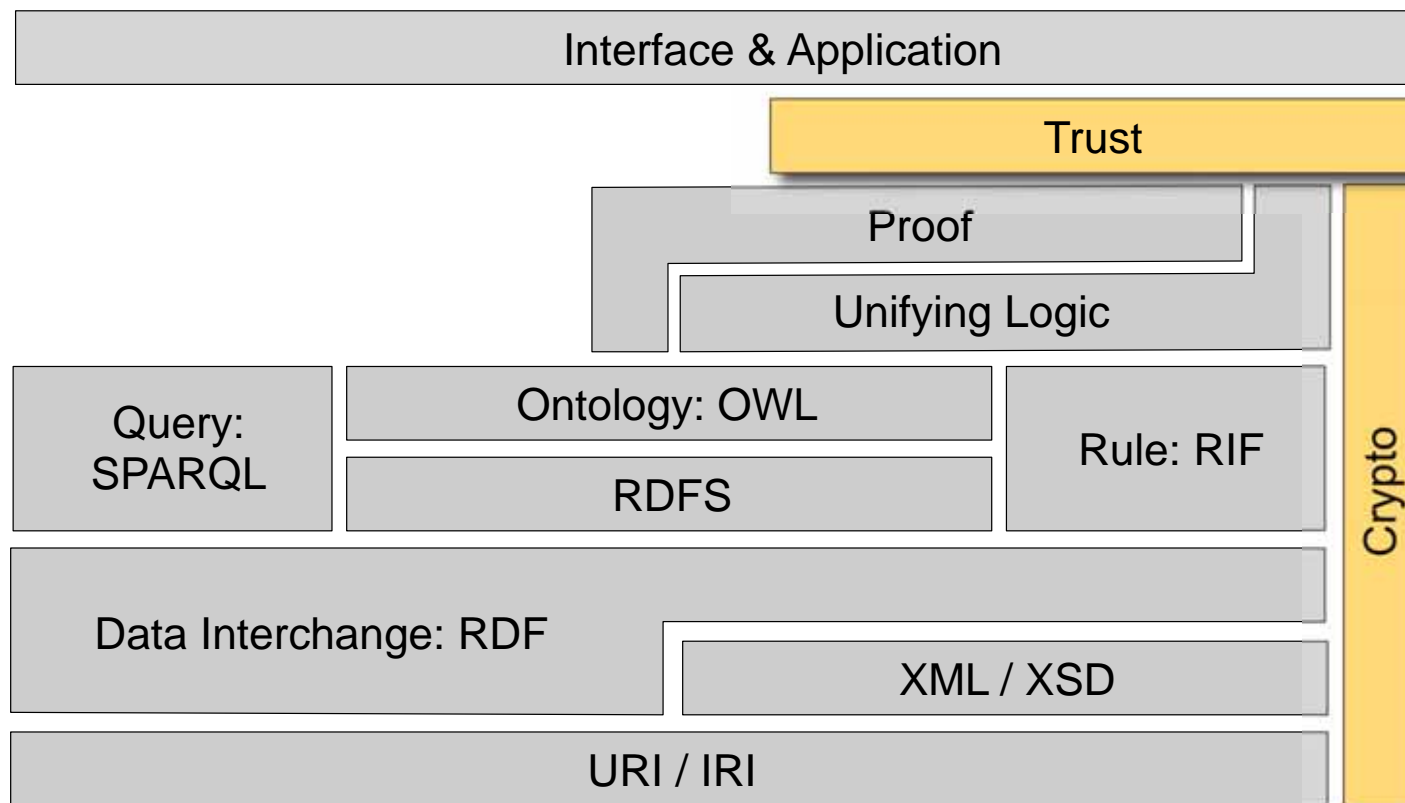
12.02.2009 – Vorlesung Nr. 13

1. Einführung
2. Die Sprachen des Semantic Web
3. Wissensrepräsentation
4. Ontology Engineering
5. **Web of Trust**

5. Web of Trust

3

Semantic Web Architektur



5. Web of Trust



„Unter Vertrauen wird die Annahme verstanden, dass Entwicklungen einen positiven oder erwarteten Verlauf nehmen...“

-- Wikipedia

5. Web of Trust

5.1. Motivation

5.2. Kryptografische Grundlagen

5.3. XML Encryption

5.4. XML Signature

5.5. Voting-, Rating- und Reputationsysteme

5. Web of Trust

5.0. Motivation

Online-Auktionen und Vertrauen



5. Web of Trust

5.0. Motivation

Vertrauen - ja oder nein?



5. Web of Trust

5.0. Motivation

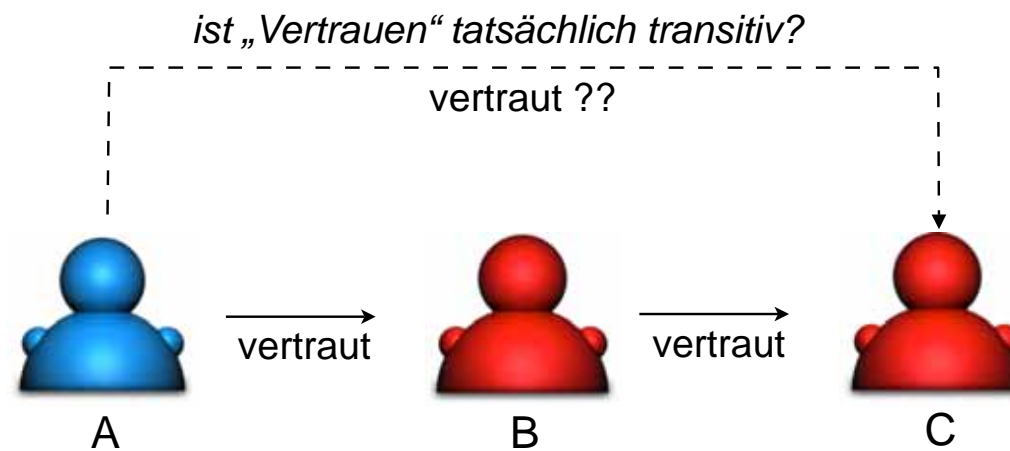
Vertrauen - ja oder nein?



5. Web of Trust

5.0. Motivation

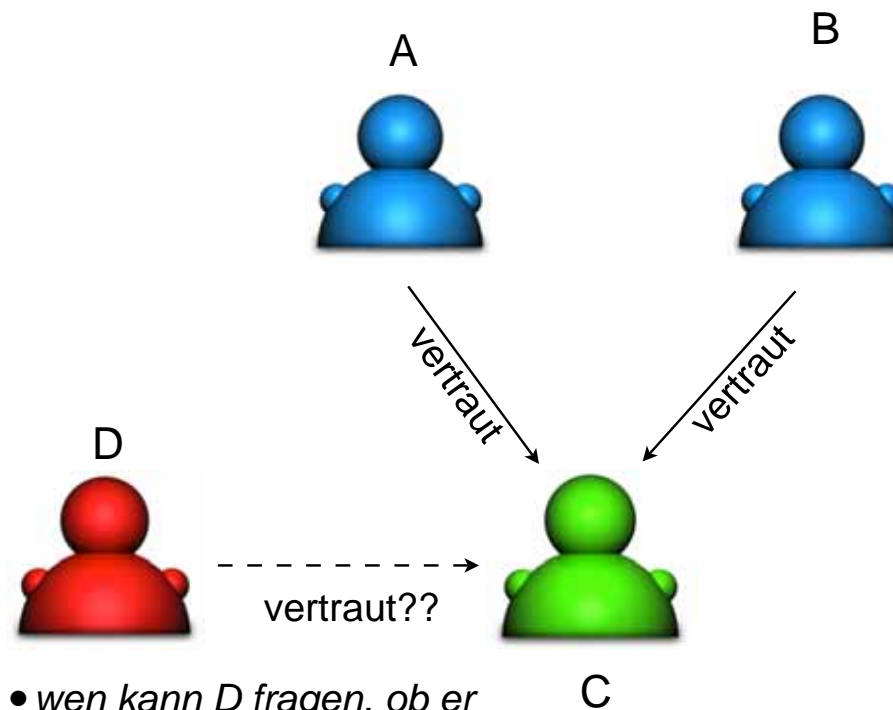
Vertrauen - ja oder nein?



5. Web of Trust

5.0. Motivation

Vertrauen - ja oder nein?



- *wen kann D fragen, ob er tatsächlich C vertrauen kann ?*

5. Web of Trust

5.1. Motivation

5.2. Kryptografische Grundlagen

5.3. XML Encryption

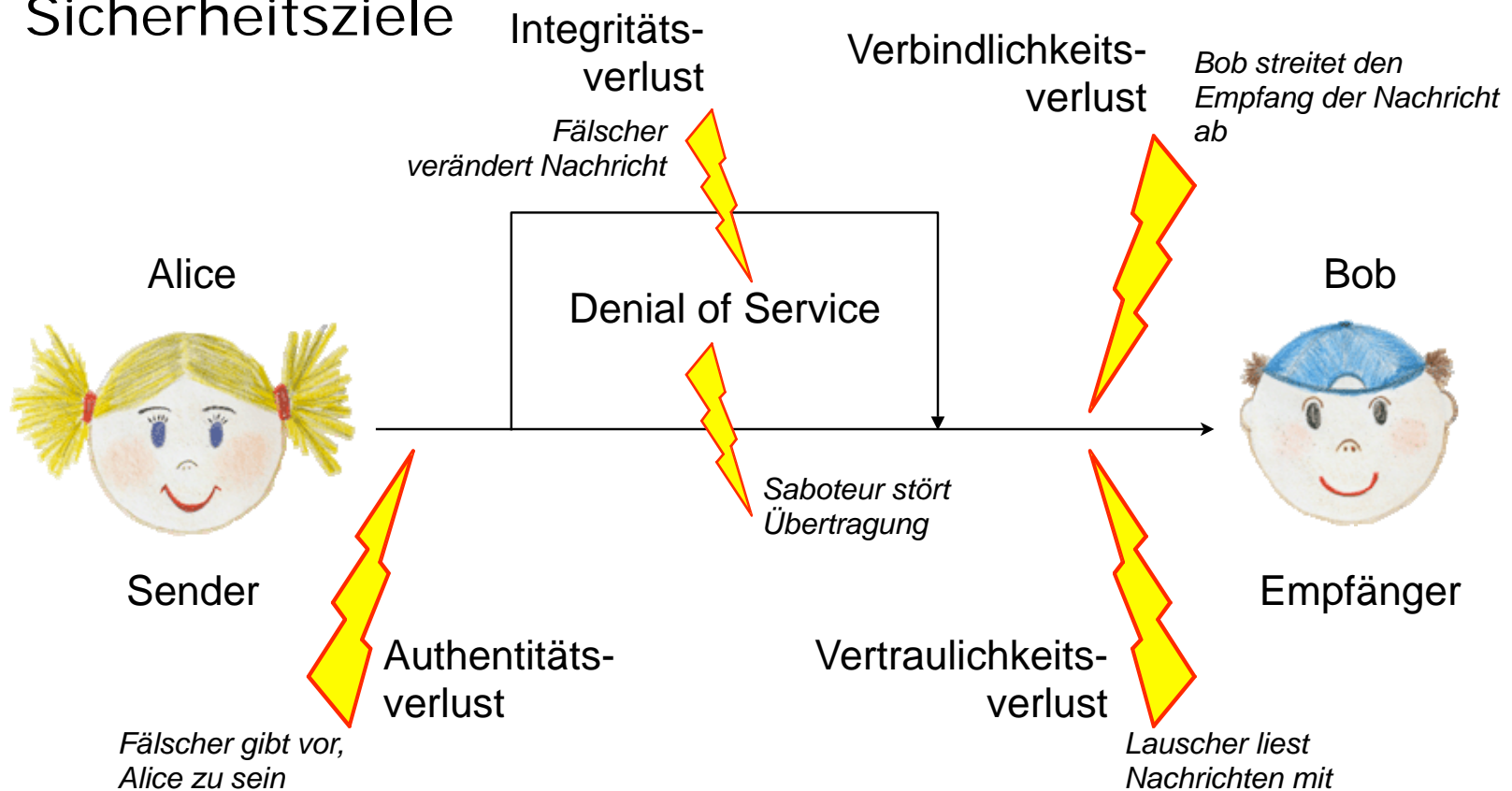
5.4. XML Signature

5.5. Voting-, Rating- und Reputationsysteme

5. Web of Trust

5.2. Kryptografische Grundlagen

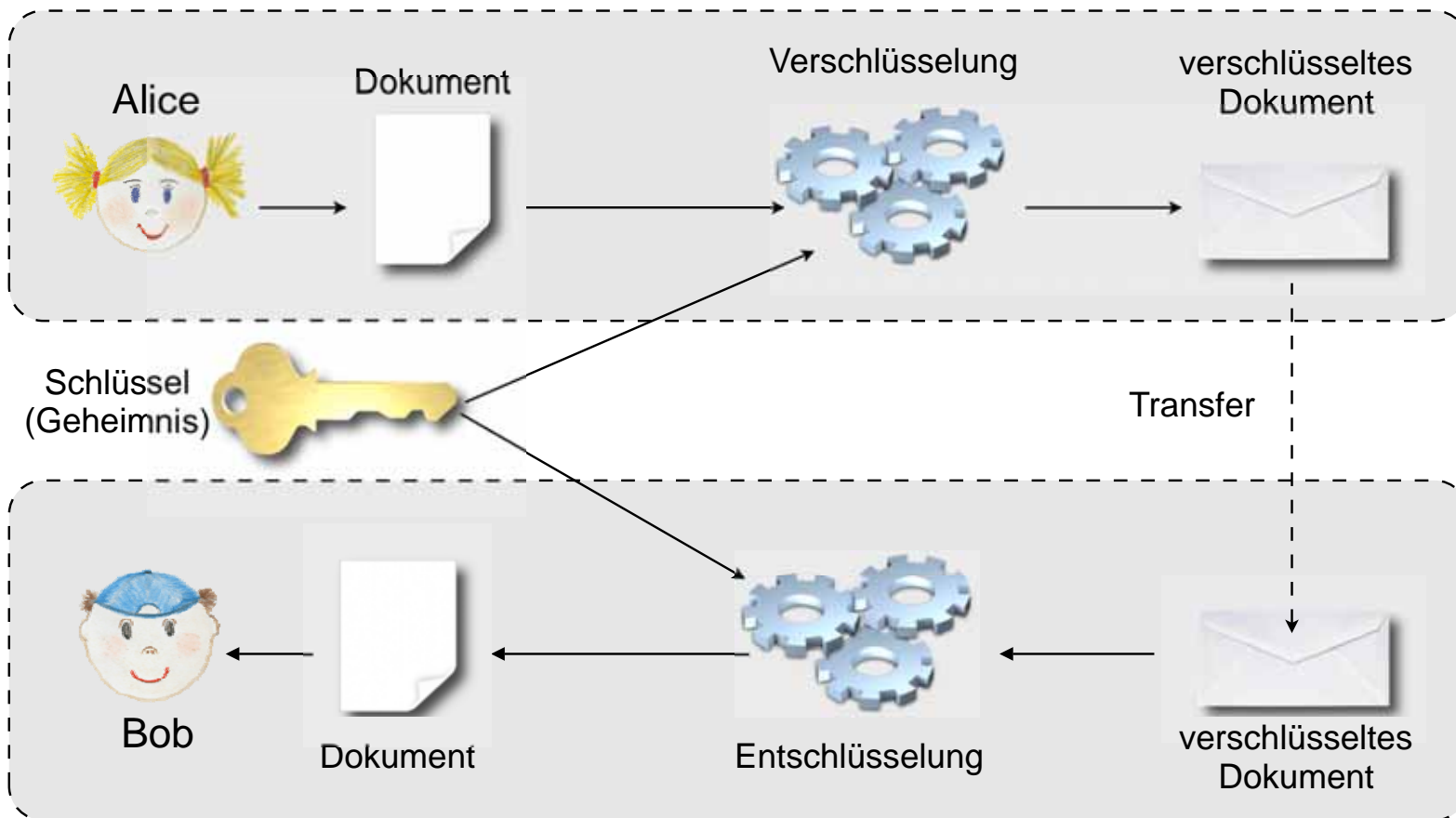
Sicherheitsziele



5. Web of Trust

5.2. Kryptografische Grundlagen

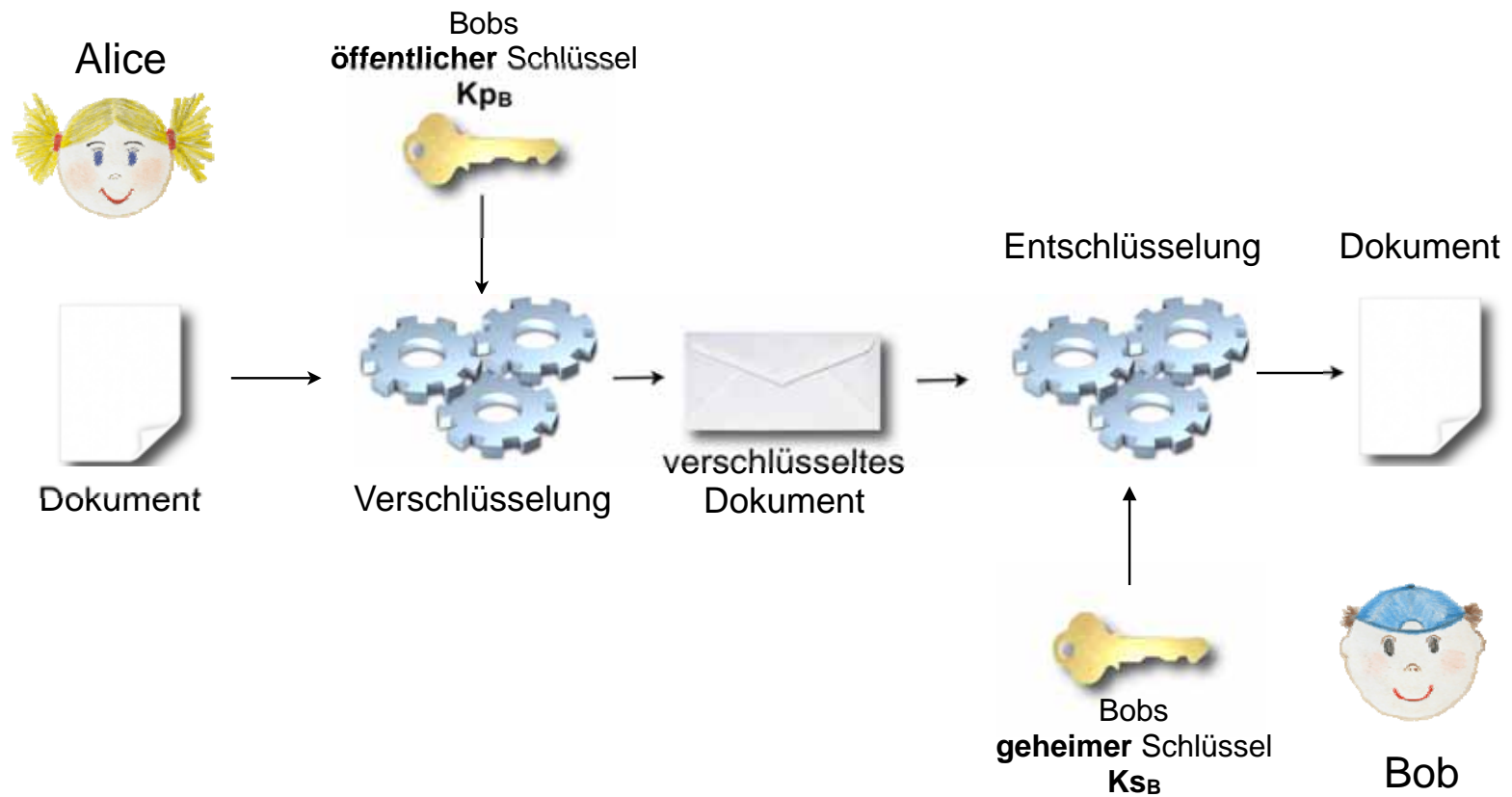
Symmetrische Verschlüsselungsverfahren



5. Web of Trust

5.2. Kryptografische Grundlagen

Asymmetrische Verschlüsselungsverfahren

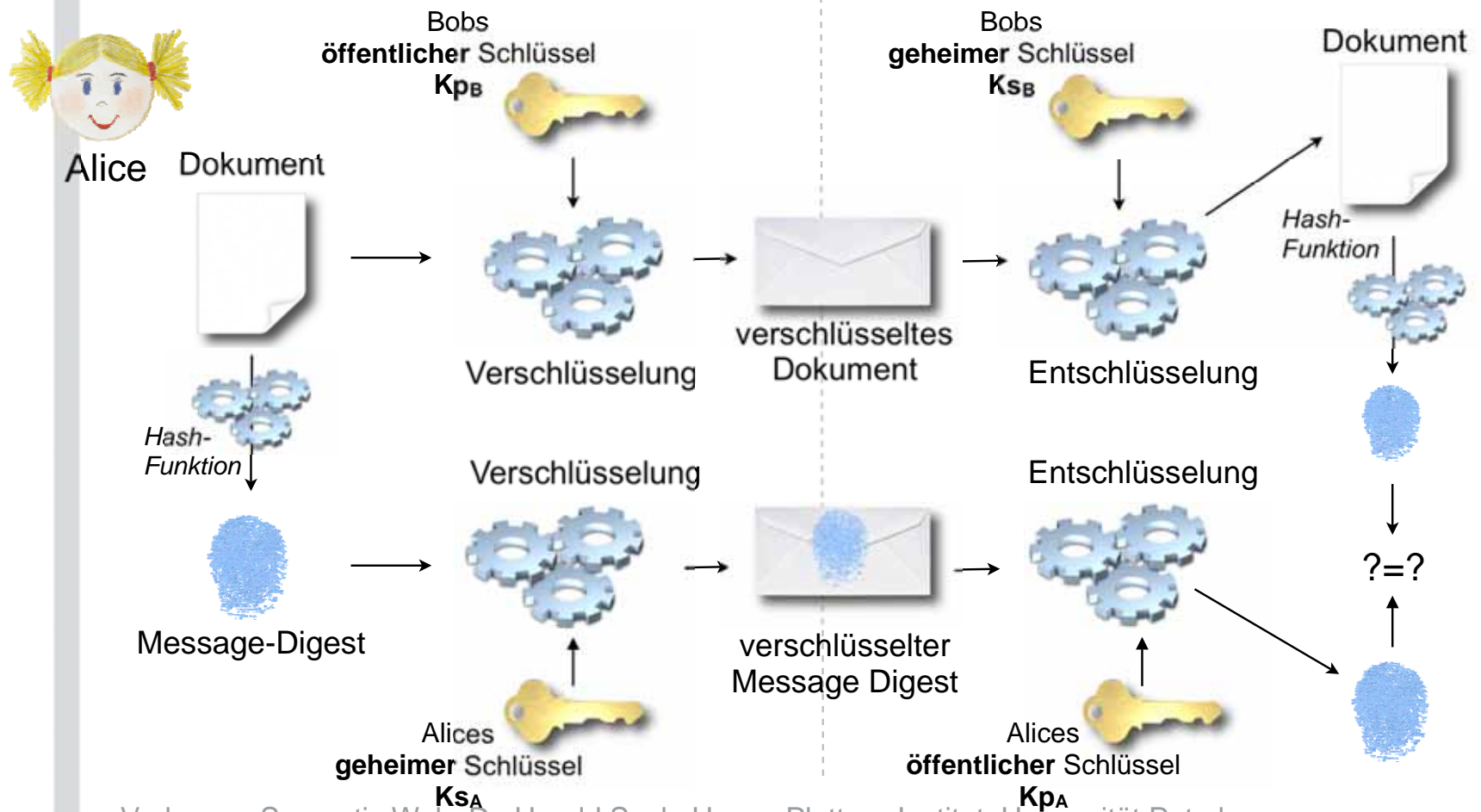


5. Web of Trust

5.2. Kryptografische Grundlagen

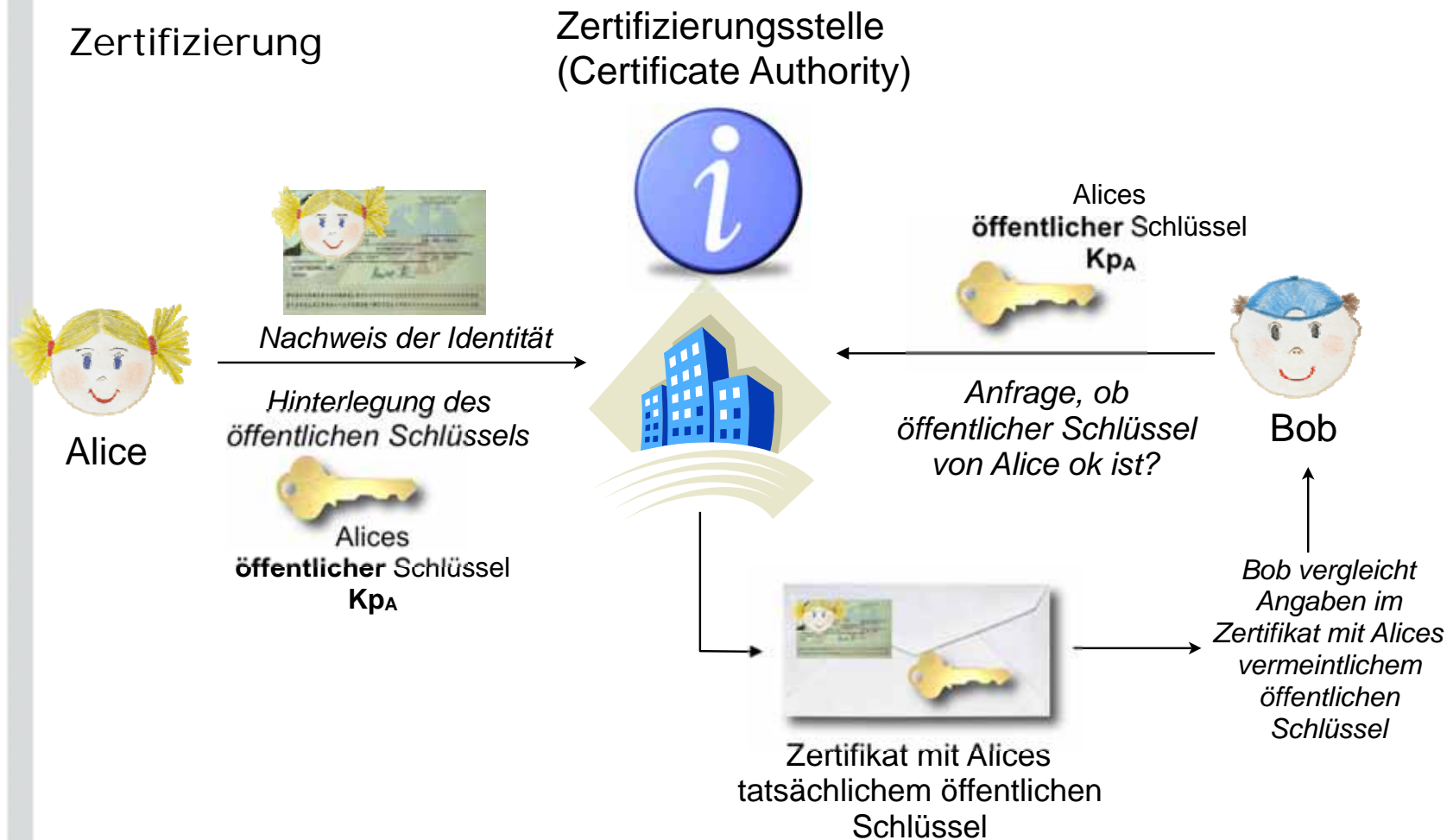


Digitale Signaturen



5. Web of Trust

5.2. Kryptografische Grundlagen



5. Web of Trust

5.1. Motivation

5.2. Kryptografische Grundlagen

5.3. XML Encryption

5.4. XML Signature

5.5. Voting-, Rating- und Reputationsysteme

Semantic Web und Kryptografie

- Auf welche Weise kann Kryptografie das Semantic Web unterstützen?
 - **Geheimhaltung / Vertraulichkeit**
durch unterschiedliche Verschlüsselungstechniken
 - **Authentifikation / Autorisierung** und **Integrität**
durch asymmetrische Verschlüsselungsverfahren
und digitale Signaturen von Semantic Web
Inhalten



XMLEncryption und XMLSignature

- **Problem:**
herkömmliche Verschlüsselungsverfahren erlauben nur die explizite Verschlüsselung von einzelnen Ressourcen als Ganzes
- **Situation 1:**
 - Eine Ontologie, die in einer OWL-Datei vorliegt, kann nur über die Verschlüsselung der gesamten Datei gesichert werden
 - Unterschiedliche Nutzer sollen aber - je nach Berechtigung - unterschiedliche Teile dieser Datei lesen können
- **Situation 2:**
 - Eine Semantic Web Ressource setzt sich aus Teilen zusammen, die aus unterschiedlichen Quellen stammen
 - Sind alle Teile der Ressource gleichermaßen vertrauenswürdig?

XMLEncryption (XML-Enc)

- W3C-Recommentation, 2002
- erlaubt Verschlüsselung von XML-basierten Dokumenten auf unterschiedlichen Granularitätstufen
 - komplettes XML-Dokument
 - komplettes XML-Element inkl. Name, Inhalt und Kind-Elemente
 - Inhalt eines XML-Elements
 - Textinhalt eines XML-Elements ohne Kind-Elemente
- Unterschiedliche verwendete Verschlüsselungsverfahren sind dabei via URI-Angabe frei wählbar

```
<EncryptedData type="uri">  
  verschlüsselter Inhalt  
</EncryptedData>
```

XMLEncryption Granularität

- Beispiel
 - enthält schützenswerte Informationen über Kreditkartendaten

```
<?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>Example Bank</Issuer>  
      <Expiration>04/02</Expiration>  
    </CreditCard>  
  </PaymentInfo>
```

XMLEncryption Granularität

- Beispiel
 - enthält schützenswerte Informationen über Kreditkartendaten

```
<?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>
```

- Verschlüsselung des kompletten <CreditCard>-Elements

XMLEncryption Granularität

- Beispiel
 - enthält schützenswerte Informationen über Kreditkartendaten

```
<?xml version='1.0'?>
  <PaymentInfo xmlns='http://example.org/paymentv2'>
    <Name>John Smith</Name>
    <CreditCard Limit='5,000' Currency='USD'>
      <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
        Type='http://www.w3.org/2001/04/xmlenc#Content'>
        <CipherData>
          <CipherValue>A23B45C56</CipherValue>
        </CipherData>
      </EncryptedData>
    </CreditCard>
  </PaymentInfo>
```

- Verschlüsselung des Inhalts des <CreditCard>-Elements

XMLEncryption Granularität

- Beispiel

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/paymentv2'>
  <Name>John Smith</Name>
  <CreditCard Limit='5,000' Currency='USD'>
    <Number>
      <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
        Type='http://www.w3.org/2001/04/xmlenc#Content'>
        <CipherData>
          <CipherValue>A23B45C56</CipherValue>
        </CipherData>
      </EncryptedData>
    </Number>
    <Issuer>Example Bank</Issuer>
    <Expiration>04/02</Expiration>
  </CreditCard>
</PaymentInfo>
```

- Verschlüsselung von Textinhalt im <CreditCard>-Element

EncryptedData Element

- Ergebnis der Verschlüsselung ist `<EncryptedData>` Element (abgeleitet von `<EncryptedType>` Element)
- `<EncryptedData>` Element ersetzt ursprüngliches Element
- `<EncryptedData>` Element kann wieder verschlüsselt werden
- Über diese „Superencryption“ sind hierarchische Zugriffsrechte definierbar

```
<EncryptedData Id? Type? MimeType? Encoding?>  
  <EncryptionMethod />  
  <KeyInfo />  
  <CipherData />  
  <EncryptionProperties>  
</EncryptedData>
```

EncryptedType Element

```
<complexType name='EncryptedType' abstract='true'>  
  <sequence>  
    <element name='EncryptionMethod'  
      type='xenc:EncryptionMethodType'  
      minOccurs='0' />  
    <element ref='ds:KeyInfo' minOccurs='0' />  
    <element ref='xenc:CipherData' />  
    <element ref='xenc:EncryptionProperties' minOccurs='0' />  
  </sequence>  
  <attribute name='Id' type='ID' use='optional' />  
  <attribute name='Type' type='anyURI' use='optional' />  
  <attribute name='MimeType' type='string' use='optional' />  
  <attribute name='Encoding' type='anyURI' use='optional' />  
</complexType>
```

EncryptedKey

- ebenfalls abgeleitet von abstraktem Typ `<EncryptedType>`
- dient der Schlüsselübertragung
- tritt als eigenständiges XML-Element oder als Kindelement von `KeyInfo` auf
- kann Zeiger auf Daten enthalten, die mit diesem Schlüssel verschlüsselt wurden (`ReferenceList`)

```
<EncryptedKey Id? Type? MimeType? Encoding?>  
  <EncryptionMethod />  
  <KeyInfo />  
  <CipherData />  
  <EncryptionProperties>  
</EncryptedKey>
```

EncryptionMethod

- beschreibt das zur Verschlüsselung verwendete Verfahren
- Verfahren wird über URI in <Algorithm> spezifiziert

```
<complexType name='EncryptionMethodType' mixed='true'>  
  <sequence>  
    <element name='KeySize' minOccurs='0' type='xenc:KeySizeType'/>  
    <element name='OAEPparams' minOccurs='0' type='base64Binary'/>  
    <any namespace='##other' minOccurs='0' maxOccurs='unbounded'/>  
  </sequence>  
  <attribute name='Algorithm' type='anyURI' use='required'/>  
</complexType>
```

CypherData

- enthält verschlüsselte Daten
 - direkt im Dokument als Base64 kodierte Zeichenkette über `<CipherValue>`
 - als URI Referenz `<CipherReference>`

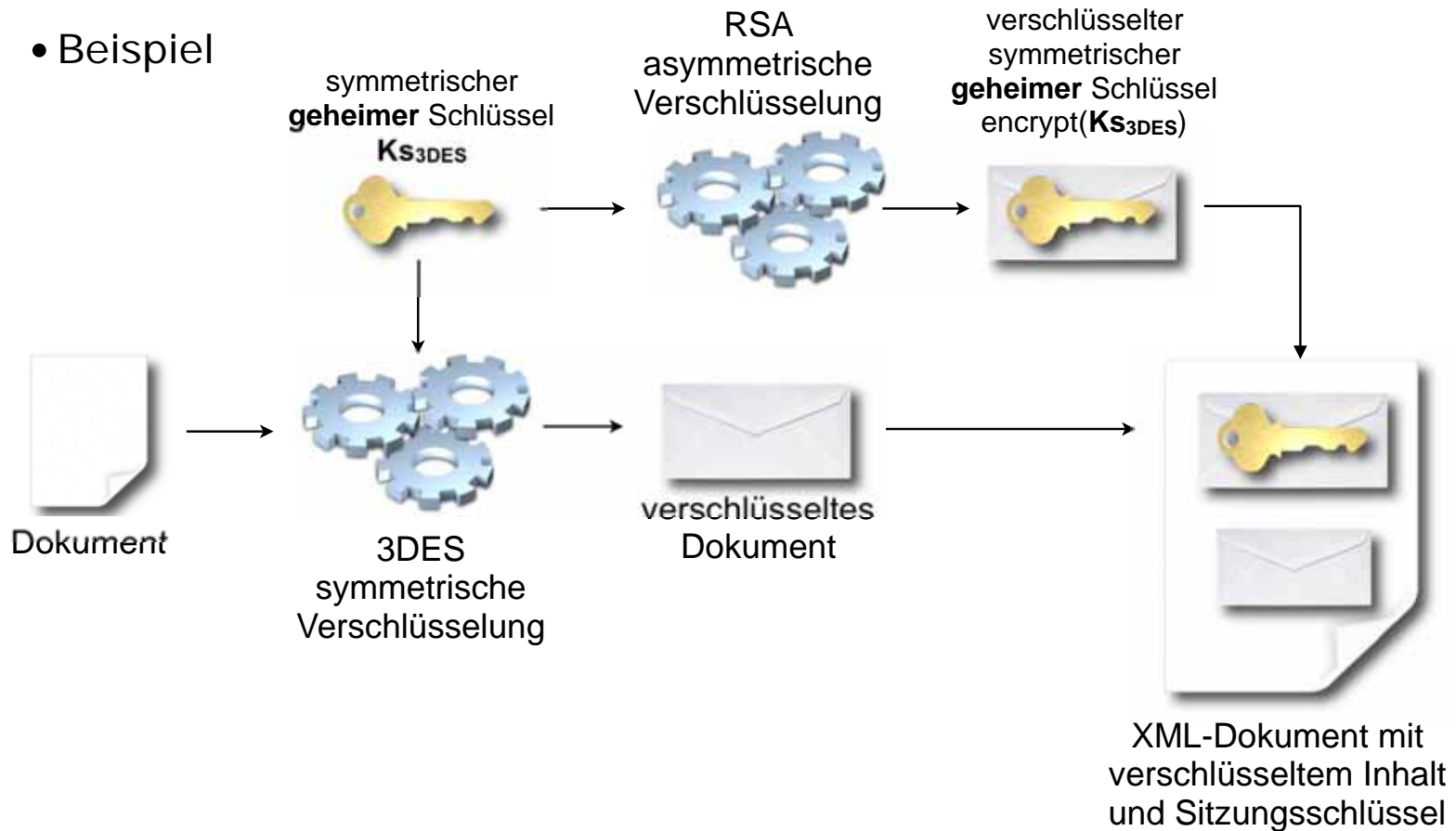
```
<element name='CipherData' type='xenc:CipherDataType'/>
<complexType name='CipherDataType'>
  <choice>
    <element name='CipherValue' type='base64Binary'/>
    <element ref='xenc:CipherReference'/>
  </choice>
</complexType>
```

5. Web of Trust

5.3. XMLEncryption

XMLEncryption

- Beispiel



5. Web of Trust

5.3. XML Encryption

```
<EncryptedData Id="xy" type="http://www.w3.org/2001/04/xmlenc#Element"
  xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc"/>
  <KeyInfo xmlns:ds="http://www.w3.org/2001/04/xmldsig#">
    <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
      <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2001/04/xmldsig#">
        <ds:KeyName>HPI symmetric 3DES-Key</KeyName>
      </ds:KeyInfo>
      <CipherData>
        <CipherValue>afds45765EOZHKJFNSDK8ohkh8...</CipherValue>
      </CipherData>
    </EncryptedKey>
  </KeyInfo>
  <CipherData>
    <CipherValue>SDHFVUIK8ohkh847gzT454Kjghfjkjh56...</CipherValue>
  </CipherData>
  ...
</EncryptedData>
```

5. Web of Trust

5.1. Motivation

5.2. Kryptografische Grundlagen

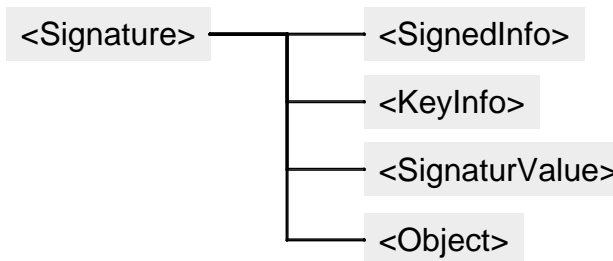
5.3. XML Encryption

5.4. XML Signature

5.5. Voting-, Rating- und Reputationsysteme

XMLSignature

- W3C Recommendation, 2002 (älter als XML Encryption)
- digitale Signatur für XML-Dokumente
- zur Sicherung von Authentizität, Integrität und Verbindlichkeit



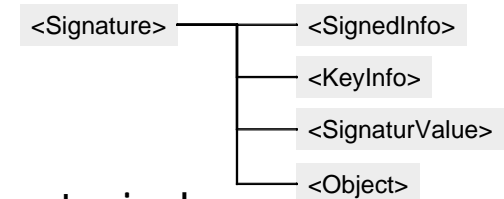
- unterscheide
 - **Enveloped Signature**
Signatur ist in das unterschriebene Objekt mit eingebettet
 - **Enveloping Signature**
unterschriebenes Objekt ist in die Signatur eingebettet
 - **Detached Signature**
unterschriebenes Objekt wird lediglich über URI referenziert

5. Web of Trust

5.4. XMLSignature

<Signature> Element

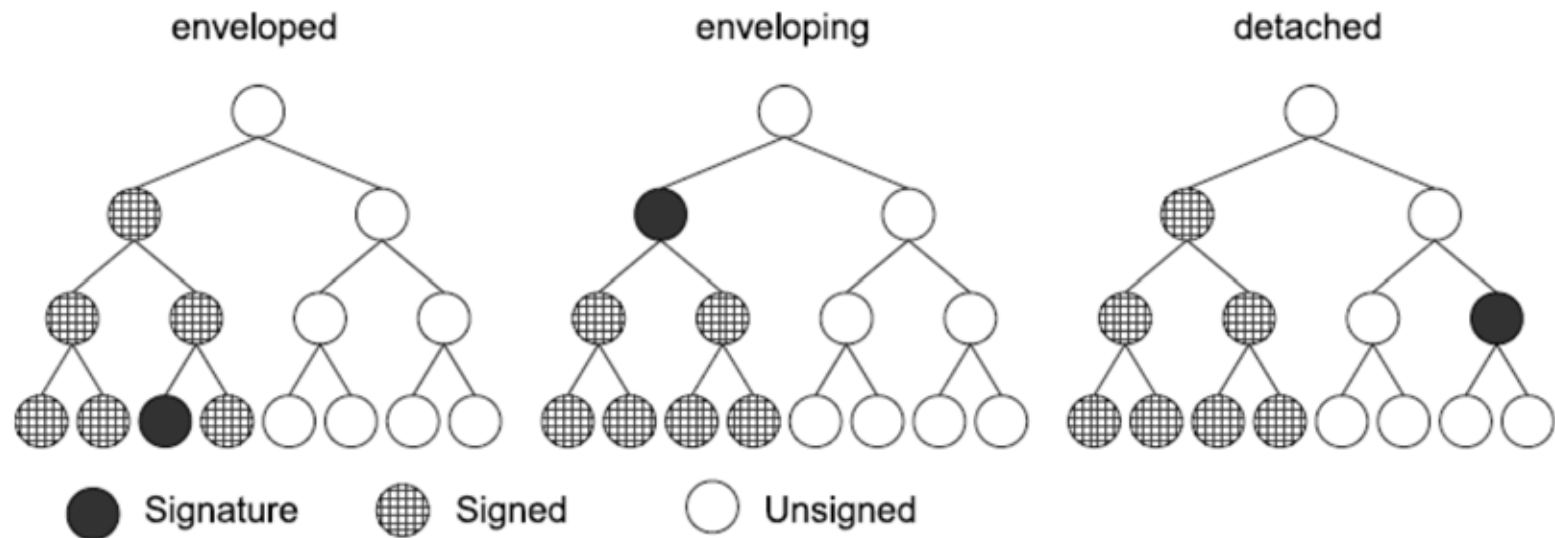
- Wurzelement von XML Signature
- <SignedInfo> und <SignatureValue> sind obligatorisch
- <KeyInfo> und <Object> sind optional
- **<SignedInfo>**
 - enthält Kanonisierungs- und Signaturverfahren
 - enthält die Referenzen (Reference) auf die signierten Daten
- **<SignatureValue>**
 - enthält Signatur (Base64) des <SignedInfo> Elements
- **<KeyInfo>**
 - notwendig zur Validierung der Signatur beim Empfänger
 - enthält Informationen über Zertifikate, Schlüssel, Schlüsselnamen oder Public-Key-Management Daten
 - zeigt auf den öffentlichen Schlüssel oder enthält diesen



5. Web of Trust

5.4. XMLSignature

XMLSignature - Einbettung



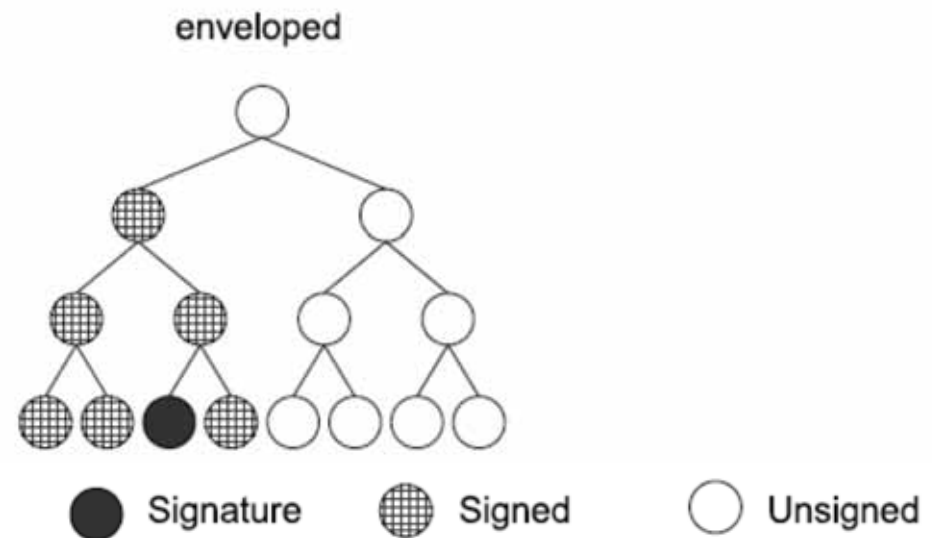
5. Web of Trust

5.4. XMLSignature

XMLSignature - Einbettung

- Enveloped Signature
Signatur ist in das unterschriebene Objekt mit eingebettet

```
<dokumentPart Id="dok">  
  <inhalt> ... </inhalt>  
  <Signature>  
    ...  
    <Reference URI="dok"/>  
    ...  
  </Signature>  
</dokumentPart>
```



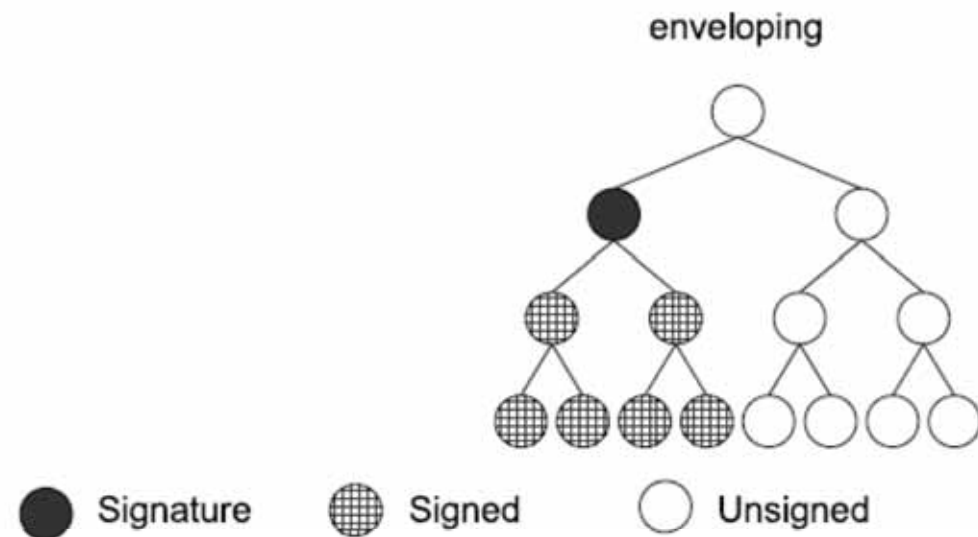
5. Web of Trust

5.4. XMLSignature

XMLSignature - Einbettung

- Enveloping Signature
unterschiedenes Objekt ist in die Signatur eingebettet

```
<Signature>  
...  
<Reference URI="obj"/>  
...  
<Object Id="obj">  
  <dokumentPart>  
  ...  
  </dokumentPart>  
</Object>  
</Signature>
```



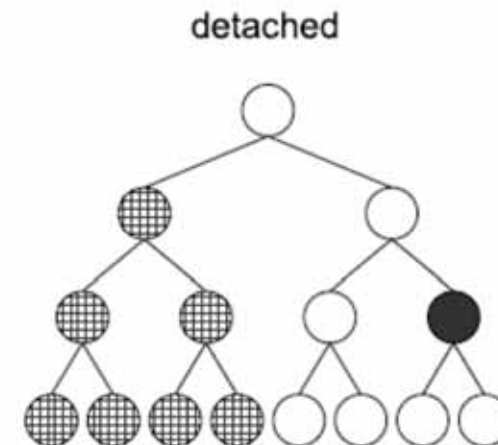
5. Web of Trust

5.4. XMLSignature

XMLSignature - Einbettung

- Detached Signature
unterschiedenes Objekt wird lediglich über URI referenziert

```
<dokumentPart>  
  <Signature>  
    ...  
    <Reference URI="http://..."/>  
    ...  
  </Signature>  
</dokumentPart>
```



● Signature ◐ Signed ○ Unsigned

5. Web of Trust

5.4. XMLSignature

XMLSignature

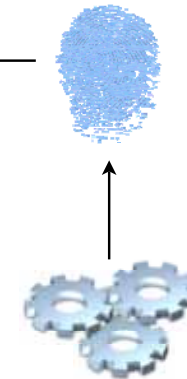
- SOAP Nachricht
- Digitale Signatur im SOAP-Header, die den Inhalt des SOAP-Body bzw. einzelne Teile davon signiert

SOAP Header

```
<Signature>  
<Reference URI="SEC"/>  
...  
</Signature>
```

SOAP Body

```
<Object ID="SEC" ...>  
...  
</Object>
```



5. Web of Trust

5.4. XMLSignature

XMLSignature - Ablauf

- Unterscheide **Generierung** und **Validierung** der Signatur
- Semantisch gleiche Dokumente können auf unterschiedliche syntaktische Weise beschrieben werden
- Daher **Kanonisierung** notwendig, d.h. Elemente werden vorher in eine Normalform gebracht
- Zudem können **Transformationen** definiert werden, d.h. Elemente werden in anderes Format umgewandelt
- Nach Kanonisierung und ev. notwendiger Transformationen kann dann die Signatur generiert werden

5. Web of Trust

5.4. XMLSignature

XMLSignature - Generierung

1. Erzeugen der Referenzen (für jedes zu signierende Element)

- Anwenden von Transformationen
- Darüber Fingerabdruck berechnen
- Erzeugen des <Reference Elements>
(Identifikation, Transformationen, MDC, Fingerabdruck)

```
<Reference URI="http://hpi-web.de/test01.htm">  
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmlsig#sha1" />  
  <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>  
</Reference>  
<Reference  
  URI="http://www.w3.org/TR/2000/WD-xmlsig-core-20000228/signature-example.xml">  
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmlsig#sha1"/>  
  <DigestValue>UrXLDLBIta6skoV5/A8Q38GEw44=</DigestValue>  
</Reference>
```

5. Web of Trust

5.4. XMLSignature

XMLSignature - Generierung

2. Erzeugen der Signatur

- Erzeugen des <SignedInfo> Elements (mit obigen Reference Elementen)

```
<SignedInfo Id="foobar">  
  <CanonicalizationMethod  
    Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>  
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />  
  <Reference URI="http://hpi-web.de/test01.htm">  
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />  
    <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>  
  </Reference>  
  <Reference  
    URI="http://www.w3.org/TR/2000/WD-xmldsig-core-20000228/signature-example.xml">  
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>  
    <DigestValue>UrXLDLBlta6skoV5/A8Q38GEw44=</DigestValue>  
  </Reference>  
</SignedInfo>
```

5. Web of Trust

5.4. XMLSignature

XMLSignature - Generierung

2. Erzeugen der Signatur

- Kanonisieren des <SignedInfo> Elements, darüber Fingerabdruck berechnen und diesen verschlüsseln

```
<SignatureValue>MC0E LE=</SignatureValue>  
  
<KeyInfo>  
  <X509Data>  
    <X509SubjectName>  
      CN=Harald Sack,O=HPI, ST=POTSDAM,C=GE  
    </X509SubjectName>  
    <X509Certificate>MIID5jCCA0+gA...IVN</X509Certificate>  
  </X509Data>  
</KeyInfo>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo Id="foobar">
    <CanonicalizationMethod
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
    <Reference URI="http://hpi-web.de/test01.htm">
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
    </Reference>
    <Reference
      URI="http://www.w3.org/TR/2000/WD-xmldsig-core-20000228/signature-example.xml">
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>UrXLDLBlta6skoV5/A8Q38GEw44=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>MC0E LE=</SignatureValue>
  <KeyInfo>
    <X509Data>
      <X509SubjectName>
        CN=Harald Sack,O=HPI, ST=POTSDAM,C=GE
      </X509SubjectName>
      <X509Certificate>MIID5jCCA0+gA...IVN</X509Certificate>
    </X509Data>
  </KeyInfo>
</Signature>

```

5. Web of Trust

5.4. XMLSignature



XMLSignature - Validierung

1. Validieren der Referenzen (SignedInfo kanonisiert)

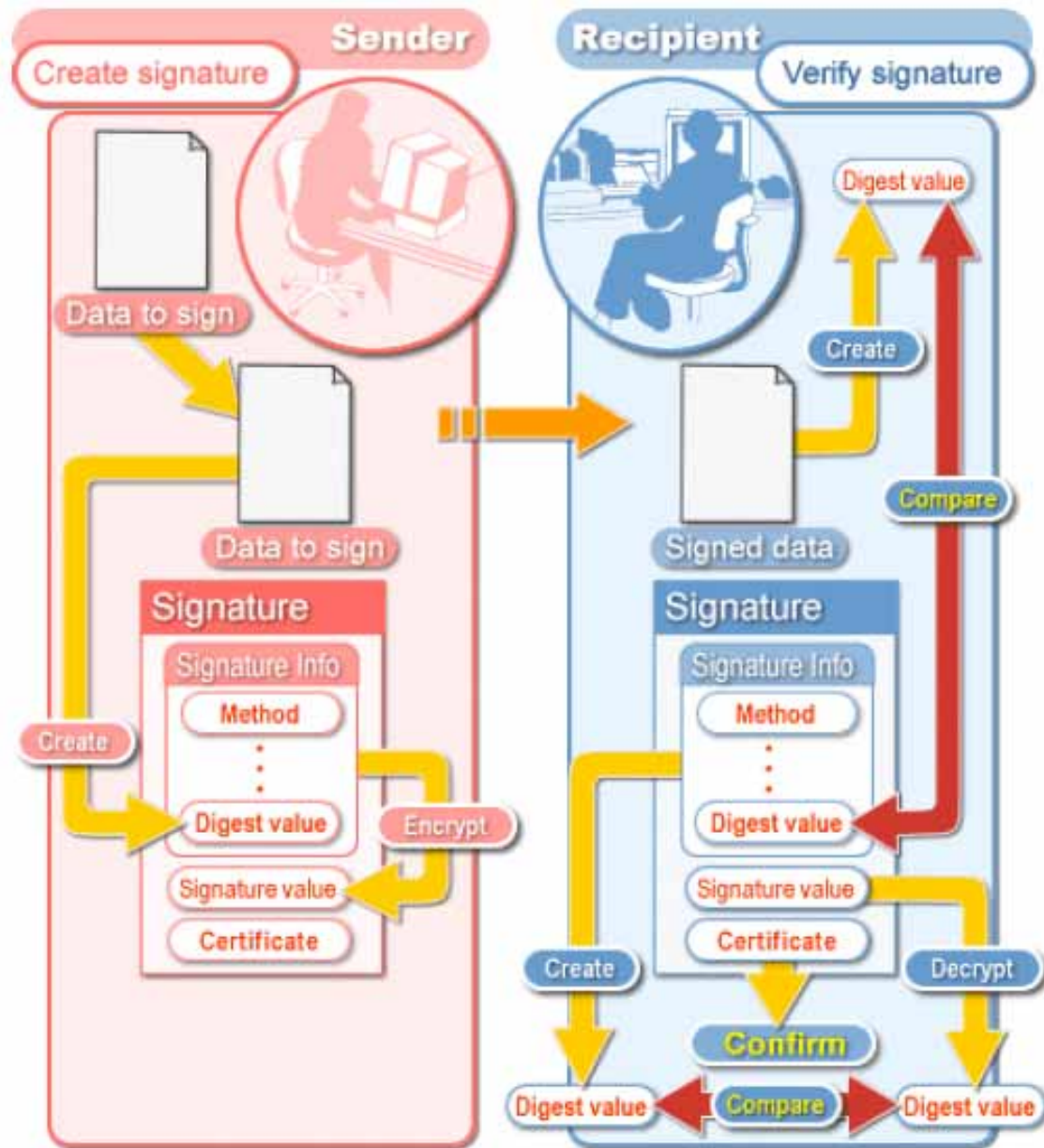
- Ermittlung der Daten (URI) und (falls nötig) Transformationen anwenden
- Fingerabdruck laut `<DigestMethod>` berechnen
- Diesen mit Fingerabdruck aus `<DigestValue>` vergleichen

2. Validieren der Signatur

- Ermittlung des Schlüssels aus `<KeyInfo>` oder anderer Quelle
- Ermittlung der kanonisierten Form der `<SignatureMethod>`,
- Berechnen des Fingerabdrucks über `<SignedInfo>`,
- Entschlüsselung des Fingerabdrucks aus `<SignatureValue>` und Vergleich der beiden

5. Web of Trust
5.4. XMLSignature

XMLSignature



5. Web of Trust

5.4. XMLSignature

XMLSignature - Beispiel

```
<Signature Id="MyFirstSignature,,  
  xmlns="http://www.w3.org/2000/09/xmldsig#">  
  <SignedInfo>  
    <CanonicalizationMethod Algorithm="http://www.w3.org/xml-c14n"/>  
    <SignatureMethod Algorithm="http://www.w3.org/rsa-sha1"/>  
    <Reference URI="http://www.foo.org/dsci1297.jpg">  
      <DigestMethod Algorithm="http://www.w3.org/sha1"/>  
      <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>  
    </Reference>  
  </SignedInfo>  
  <SignatureValue> MC0CFFrVLtRIka4455 </SignatureValue>  
  <KeyInfo>  
    <X509Data>  
      <X509SubjectName> DN=Max Mustermann </X509SubjectName>  
    </X509Data>  
  </KeyInfo>  
</Signature>
```

5. Web of Trust

5.1. Motivation

5.2. Kryptografische Grundlagen

5.3. XML Encryption

5.4. XML Signature

5.5. Voting-, Rating- und Reputationsysteme

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Vertrauen

Vertrauen ist der individuelle Glaube an die positive Entwicklung von Ereignissen, meist im zwischenmenschlich-interaktiven Bereich, gebunden an die eigenen Wertvorstellungen und Erfahrungen....

Aus **systemtheoretischer Sicht**: Vertrauen ist ein „*Mechanismus zur Reduktion sozialer Komplexität*“.

Dort wo die rationale Abwägung von Informationen (aufgrund **unüberschaubarer Komplexität**, wegen **Zeitmangels** zur Auswertung oder des **gänzlichen Fehlens von Informationen** überhaupt) nicht möglich ist, befähigt Vertrauen dennoch zu einer auf Intuition gestützten Entscheidung...

-- aus wikipedia.de

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Reputation

Reputation bezeichnet die gesellschaftliche Wertschätzung einem Menschen oder einer Gruppe gegenüber....

In der **Marketingtheorie**: Summe von Einzelerwartungen und -erfahrungen über Vertrauenswürdigkeit und Kompetenz eines Anbieters

Nach chinesischer Denkweise hat jeder Mensch ein Gesicht. Sein Gesicht wird ihm durch soziale Anerkennung gegeben oder durch Missachtung entzogen. Das Gesicht eines anderen zu wahren, heißt in erster Linie, seine Schwachstellen nicht bloßzulegen. Wer Ansehen gibt, gewinnt damit zugleich selbst an Ansehen. Wer einem Anderen das Gesicht nimmt, hat damit seines auch verloren.

-- aus wikipedia.de

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Web of Trust

- Das Semantic Web ist (wie das Internet) ein **Open Publishing System**
 - Jeder kann publizieren
 - Keine Qualitätskontrolle
 - Keine zentrale Kontroll-Instanz

→ Erfordernis Informationen bezüglich Qualität/Verlässlichkeit zu filtern.
- Grundlage dieser Filterung sind Voting-/Rating-/ und Reputations-systeme.
- Aufgaben eines Open Rating Systems:
 - **Aggregation** von unterschiedlichsten Bewertungen aus unterschiedlichen Quellen zu einem einheitlichen **Ranking**
 - **Meta-Ranking** – Bewertung der Qualität der Bewertungen

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Meta-Ranking

- einfaches mathematisches Modell

- Objekte

$$O := \{O_1, O_2, O_3, \dots\}$$

- Agenten

$$A := \{A_1, A_2, A_3, \dots\}$$

- Bewertungen von Objekten

$$D := \{d_1, d_2, d_3, \dots\}$$

- Bewertungen von Agenten

$$T := \{t_1, t_2, t_3, \dots\}$$

- Partielle Funktion zur
Bewertung von Objekten

$$R: A \times O \rightarrow D$$

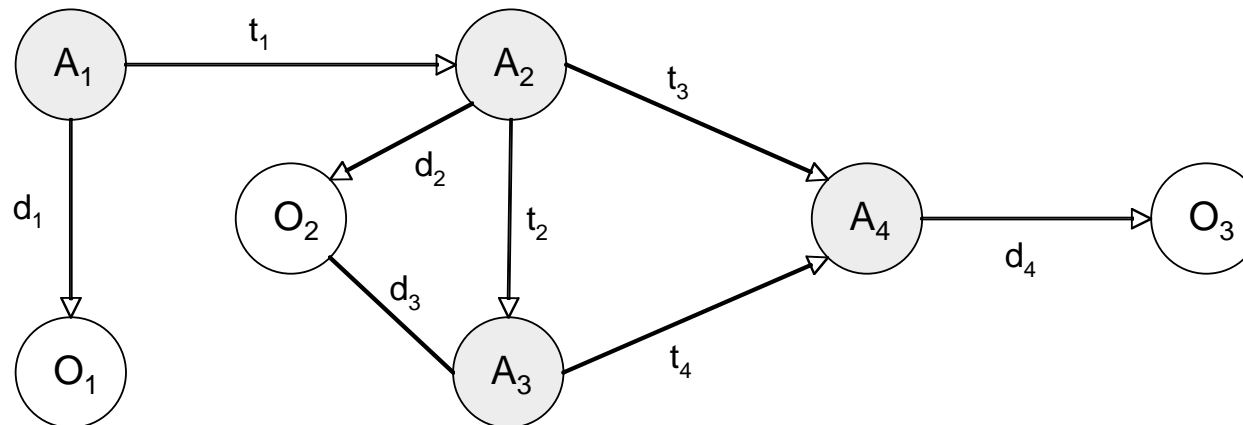
- Partielle Funktion zur
Bewertung von Agenten

$$W: A \times A \rightarrow T$$

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Meta-Ranking



5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme



Meta-Ranking

- **Rating-Problem**

- Vervollständige die Matrix R mit Hilfe der Matrix W .
- Wenn ein Agent ein Objekt nicht selber direkt bewertet hat, dann wird unter Ausnutzung von Vertrauensbeziehungen der Agenten untereinander berechnet, wie er dieses Objekt bewerten würde.

- **Ranking-Problem**

- Ordne eine Menge von Objekten aus der Sicht eines Agenten in die Reihenfolge gemäß ihrer Bewertung.
- In den meisten Fällen reicht es, bezüglich der Reihenfolge die ersten N Objekte aufzulisten, daher Top- N -Ranking-Problem.

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Beispiel Ebay

- Agenten werden von anderen Agenten hinsichtlich der Qualität eines Geschäftsablaufs bewertet
- Ebay Bewertungen: {negativ, neutral, positiv}
- Globales Rating wird einfach aus Anteil der positiven Bewertungen errechnet

[← Zurück zur vorherigen Suche](#) [Startseite](#) > [Gemeinschaft](#) > [Bewertungsportal](#) > **Bewertungsprofil**

Bewertungsprofil: [redacted] (5325 ★) [mich](#) 

Bewertungsprofil: 5325
Positive Bewertungen: 97,4%




Mitglieder, die mich positiv bewertet haben: 5463

Mitglieder, die mich negativ bewertet haben: 147

Alle positiven Bewertungen: 7402

[Weitere Informationen](#) zur Bedeutung dieser Zahlen.

Aktuelle Bewertungen:

	Letzter Monat	Letzte 6 Monate	Letzte 12 Monate
 positiv	490	2619	4520
 neutral	6	35	68
 negativ	15	37	100

Zurückgezogene Gebote (in den letzten 6 Monaten): 0

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme



Modifizierter Google PageRank Algorithmus

- Googles patentiertes Verfahren zur Ermittlung des globalen Rankings einer WebPage über deren Relevanz
- **Relevanzgewichtung**
 - Google unterscheidet „wichtige“ von „unwichtigen“ Dokumenten
 - „Wichtig“ bedeutet:
 1. ein Dokument ist um so „wichtiger“, je mehr andere Dokumente auf dieses Dokument via Links verweisen
 2. ein Dokument, auf das ein „wichtiges“ Dokument via Link verweist, ist selbst „wichtig“
 3. je mehr Links ein Dokument auf andere Dokumente enthält, desto „unwichtiger“ ist ein einzelner Link

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Google PageRank Algorithmus

- aus 1-3 lässt sich eine Formel zur Berechnung der „Wichtigkeit“ (PageRank, PR) eines Dokuments gewinnen:
- sei $PR(A)$ der zu ermittelnde PageRank des Dokuments A
- sei n die Anzahl der auf A verweisenden Links
- seien $T_1 \dots T_n$ Dokumente, die einen Link auf A enthalten
- seien $PR(T_1) \dots PR(T_n)$ die PageRanks der Dokumente $T_1 \dots T_n$
- sei $c(T_i)$ die Anzahl der ausgehenden Links von Dokument T_i
- sei d ein Dämpfungsfaktor ($0 < d < 1$)

$$PR(A) = (1 - d) + d \left(\sum_{i=1}^n \frac{PR(T_i)}{c(T_i)} \right)$$

Berechnung wird iterativ durchgeführt, bis sich ein stabiler Zustand (Fixpunkt) ergibt.

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Modifizierter PageRank Algorithmus

- Übertragung auf Web of Trust:
 - *Annahme*: Bewertungen nehmen nur Wert positiv an.
 - Es entsteht Graph $G := R_P \cup W_T$ mit Agenten und Objekten als Knoten und Kanten, wenn Agent das Objekt **positiv** bewertet hat bzw. Agent anderem Agenten **vertraut**.
 - PageRank auf G angewendet ergibt AORank für Agenten und Objekte.
- Anpassung, um negative Bewertungen zu berücksichtigen:
 - B_i = Menge von Agenten, die Objekt O_i negativ bewertet haben.
 - N_v = Anteil negativer Bewertungen, die ein Agent v abgegeben hat.

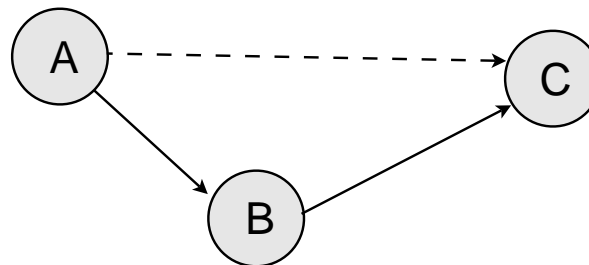
$$\text{ModifiedPageRank}(O_i) = \text{AORank}(O_i) - \sum_{v \in B_i} \frac{\text{AORank}(O_i)}{N_v}$$

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Reputationsnetzwerke

- Lokales Ranking berücksichtigt Kontext
(es wird immer nur die Reputation betrachtet, die ein Agent B in den Augen von Agent A genießt)
- Eine Kante von Agent A zu Agent B bezeichnet die **(direkte) Reputation**, die B aus Sicht von A genießt.
- Besteht zwischen den Agenten A und C keine Kante, kann eine **(indirekte) Reputation** über einen eventuell bestehenden Pfad von A nach B berechnet werden.

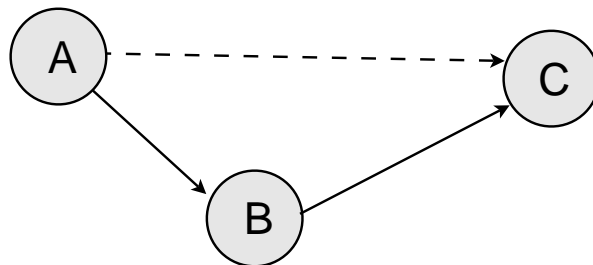


5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Reputationsnetzwerke

- Wie bestimme ich Reputation (A, C)?
- A = Quelle, C = Senke
- A fragt jeden Nachbarknoten n_i , der eine gute Reputation besitzt, rekursiv nach Reputation (n_i, C)?
- A akkumuliert die zurückgegebenen Reputationswerte (Durchschnittswert) und rundet diesen Wert kaufmännisch.



Unterscheiden gute und böse Knoten.

- Gute Knoten haben gute Reputation.
- Böse Knoten haben schlechte Reputation.
- Böse Knoten bewerten die Reputation aller Nachbarknoten falsch.

→ Hier können problematische Situationen entstehen...

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Vertrauensstrategien

- Wie können Agenten miteinander interagieren?
- Grundvoraussetzung ist Vertrauen.
- Vertrauen bildet sich durch Informationsbeschaffung und Informationsauswertung.
- Risiko sollte minimiert werden.
- Interaktionsmöglichkeiten/Nutzen sollte maximiert werden.

Strategie:

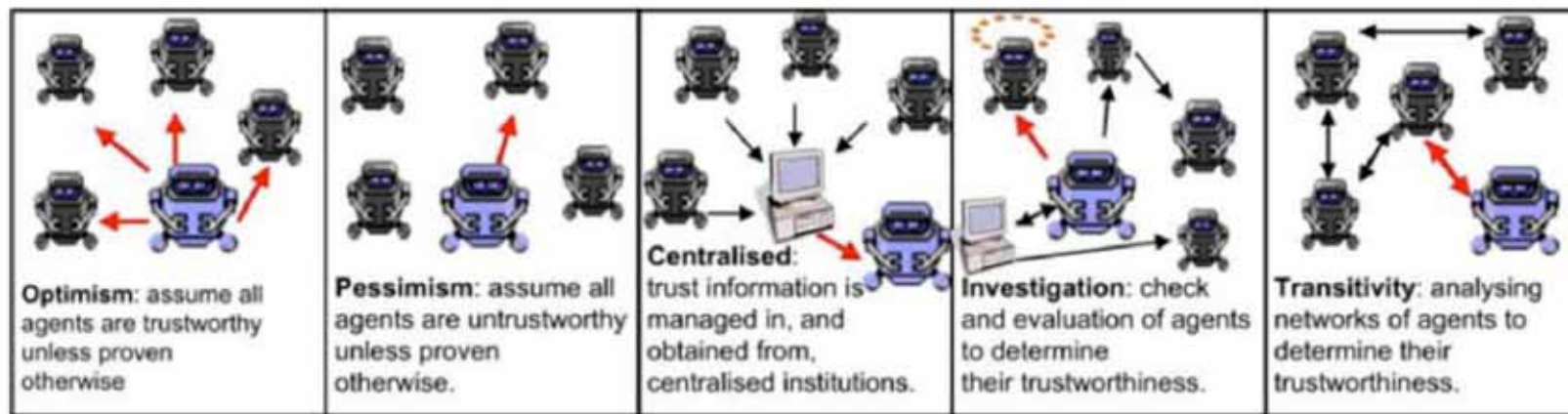
Die Strategie spiegelt die grundlegende Einstellung eines Agenten gegenüber einem anderen Agenten in einem System unter Unsicherheit wieder. Bei der Wahl der Strategie werden die relativen Kosten und der Nutzen der Interaktion mit dem entsprechenden Agenten berücksichtigt. Kombinationen und Wechseln der Strategien sind möglich.

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Vertrauensstrategien

- Optimistische Strategie
- Pessimistische Strategie
- Zentrale Strategie
- Recherche-Strategie
- Transitive Strategie



5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Vertrauensstrategien

- **Optimistische Strategie**
 - Vertraue per Default jedem Agenten.
 - Entziehe Vertrauen, wenn Gründe vorliegen.
 - Anwendbar, wenn
 - Nutzen der Kooperation sehr groß.
 - Kosten/Risiko des Betrugs sehr gering.
 - schnelle Initialisierung des Netzwerks notwendig.
- **Pessimistische Strategie**
 - Misstraue per Default jedem Agenten.
 - Schenke Vertrauen, wenn Gründe dafür vorliegen.
 - Anwendbar, wenn
 - Es nicht vorteilhaft ist, jemanden zu vertrauen.
 - Kosten/Risiko des Betrugs hoch.



Optimism: assume all agents are trustworthy unless proven otherwise



Pessimism: assume all agents are untrustworthy unless proven otherwise.

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Vertrauensstrategien

- **Zentrale Strategie**

- Zentrale Autorität sammelt Informationen über Agenten (Benutzerfeedback).
- Zentrale Autorität zertifiziert Agenten.
- Entscheidung über Ver-/Misstrauen fällt die zentrale Autorität.
- Vorteil: Agent muss nur noch der Autorität vertrauen.
- Bsp.: Ebay

- **Recherche-Strategie**

- Agenten erlangen Vertrauen ineinander, indem sie schrittweise unter Aufsicht einer dritten Instanz, der sie vertrauen, eine Art Vertrag aushandeln.
- Agenten agieren autonom und treffen Entscheidungen selbst.

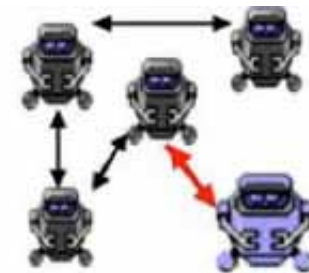


5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Vertrauensstrategien

- **Transitive Strategie**
 - Basiert auf Bewertungen des Vertrauens der Benutzer/Agenten untereinander.
 - Problem:
 - Vertrauen ist nur näherungsweise transitiv.
 - Vertrauen wird meist nicht im Kontext betrachtet.



Transitivity: analysing networks of agents to determine their trustworthiness.

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme



Vertrauensstrategien

- Welche Strategie soll gewählt werden?
 - Agenten sollen unter **Unsicherheit** miteinander agieren.
 - **Risiko** sollte minimiert werden.
 - Interaktionsmöglichkeiten/**Nutzen** sollte maximiert werden.
 - Zeit, bis gewisses Vertrauen erreicht (berechnet) wurde, sollte Zeitschranke nicht überschreiten.

- Risiko lässt sich in Form von anfallenden Kosten quantifizieren
 - Operationale Kosten
 - Opportunitätskosten
 - Ausfallkosten

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme



Vertrauensstrategien

• Operationale Kosten

- Kosten die zur Realisierung (Durchführung) der Strategie anfallen
- Proportional zur Komplexität der Strategie (Ressourcenbedarf)
- **Gering:** Optimismus, Pessimismus (einfache Tests)
- **Hoch:** Zentralisiert (Economie of Scale), Transitiv (hohe Nebenkosten), Recherchierend

• Opportunitätskosten

- Entstehen, wenn man Möglichkeiten, den maximalen Nutzen zu erzielen, nicht wahrnimmt.
- **Gering:** Optimismus
- **Mittel:** Zentralisiert (Anzahl zertifizierter Agenten), Recherchierend (Agent hat beschränkte Ressourcen)
- **Hoch:** Pessimismus, Transitiv (Netzwerkgröße)

5. Web of Trust

5.5. Voting-, Rating- und Reputationsysteme

Vertrauensstrategien

- **Ausfallkosten**

- Fallen an, wenn sich Agent anders als behauptet verhält.
- **Gering:** Pessimismus, Recherchierend
- **Mittel:** Transitiv (Netzwerkgröße, selbstregulierend)
- **Hoch:** Optimismus,
Zentralisiert (Misstrauen in Autorität)

5. Web of Trust

5.1. Motivation

5.2. Kryptografische Grundlagen

5.3. XML Encryption

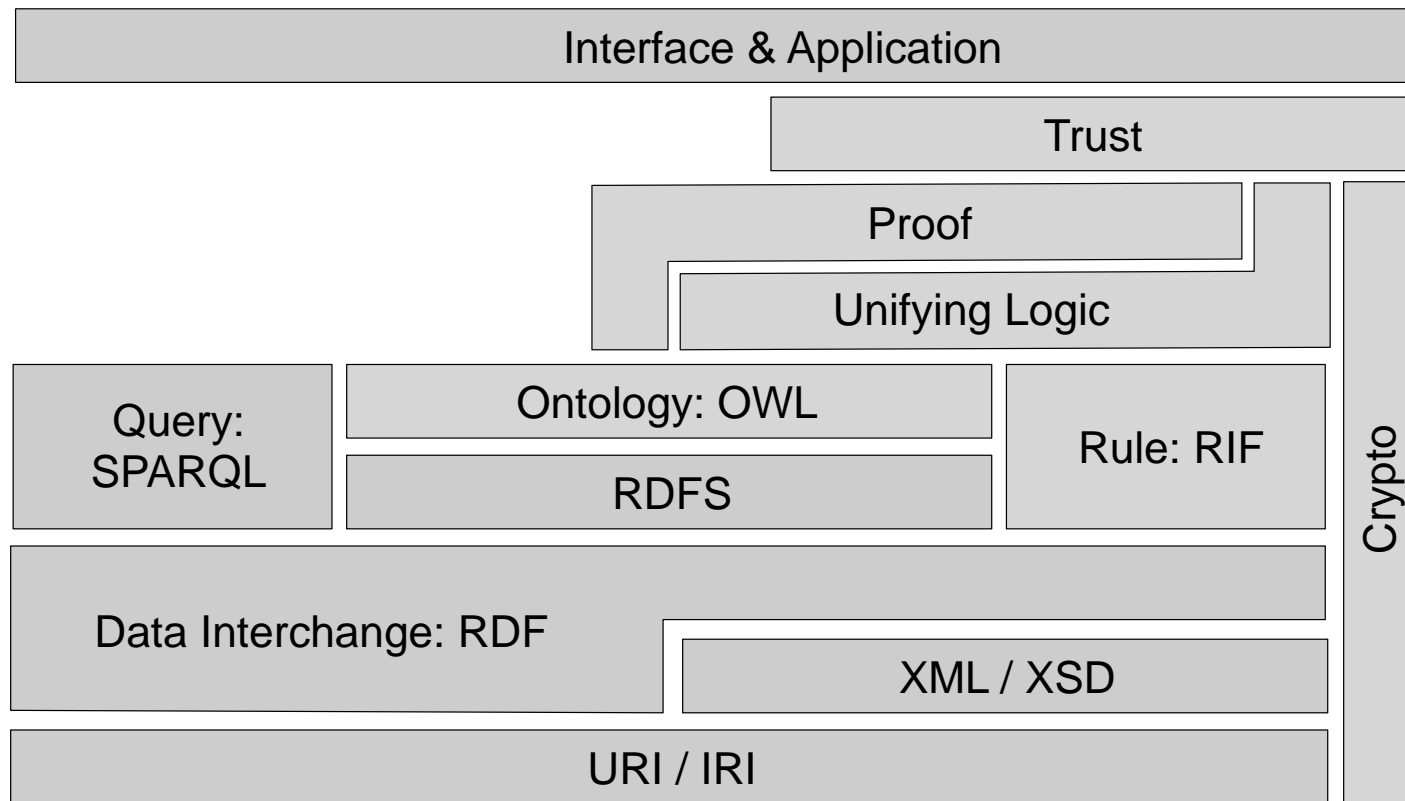
5.4. XML Signature

5.5. Voting-, Rating- und Reputationsysteme

5. Web of Trust

70

Semantic Web Architektur



Semantic Web - Vorlesungsinhalt

71

1. Einführung
2. Die Sprachen des Semantic Web
3. Wissensrepräsentation
4. Ontology Engineering
5. Web of Trust

5. Web of Trust

72

Literatur



- Blog
<http://sw0809.blogspot.com/>
- Materialien-Webseite
http://www.hpi.uni-potsdam.de/meinel/teaching/semantic_web_ws08090.html



- bibsonomy - Bookmarks
<http://www.bibsonomy.org/user/lysander07/sw0809-13>