

# XML-Signatur

Frank Markus Abbühl

`frank.abbuehl@fhm.edu`

Fachhochschule München  
Kryptologie, Prof. Köhler

# Warum?

*When you try to unify two opposing forces by creating a third alternative, you just end up with three opposing forces.*

– Joel Spolsky

# Warum?

*When you try to unify two opposing forces by creating a third alternative, you just end up with three opposing forces.*  
– Joel Spolsky

```
<?xml version="1.1" encoding="UTF-8"?>
<PaymentInfo xmlns="http://www.example.com/paymentv2">
  <Name>John Smith</Name>
  <CreditCard Id="smithcard" Limit="5000">
    <Number>8765 1755 4711</Number>
    <Issuer>Example Bank</Issuer>
    <Expiration>07/2004</Expiration>
  </CreditCard>
</PaymentInfo>
```

# Warum?

*When you try to unify two opposing forces by creating a third alternative, you just end up with three opposing forces.*  
– Joel Spolsky

```
<?xml version="1.1" encoding="UTF-8"?>
<PaymentInfo xmlns="http://www.example.com/paymentv2">
  <Name>John Smith</Name>
  <CreditCard Id="smithcard" Limit="5000">
    <Number>8765 1755 4711</Number>
    <Issuer>Example Bank</Issuer>
    <Expiration>07/2004</Expiration>
  </CreditCard>
  <Signature URI="#smithcard">
    E71h260mCduj9rgFnUmOcw4Uu7Y=
  </Signature>
</PaymentInfo>
```

# Anforderungen

W3C-Empfehlung: XML-Signature Syntax and Processing

- Einfachheit, Flexibilität, Robustheit
- Beliebige Daten
- Beliebige Algorithmen

Keine Vorgaben über

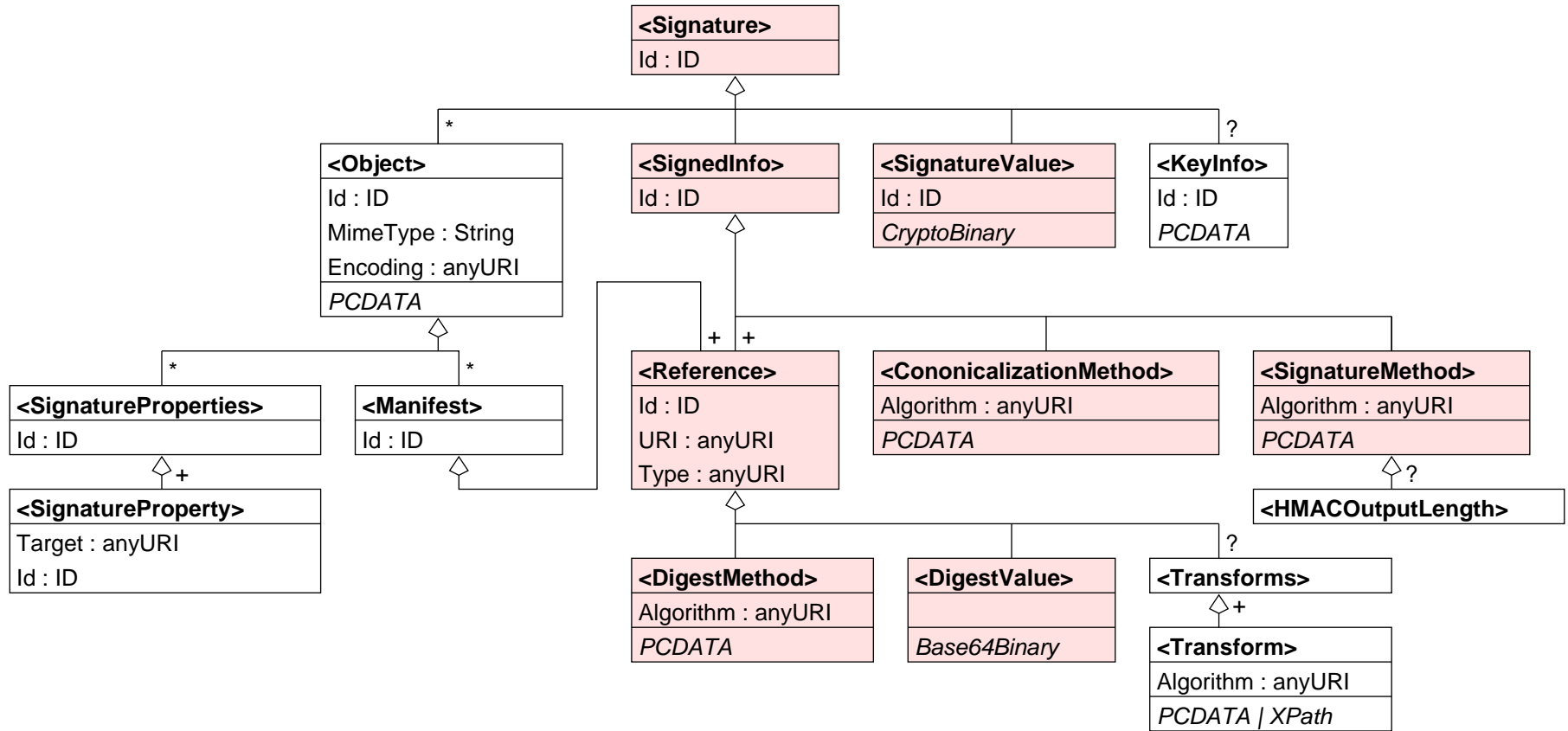
- Interpretation der Signatur
- Public Key Infrastruktur
- Vertrauensbeziehungen

# Syntax

Die XML-Signatur ist ein XML-Element. Es besteht aus:

- Signatur-Wert
- Manifest mit Referenzen auf Datenobjekte
- Schlüsselinformationen (optional)
- Eingebettete Objekte (optional)

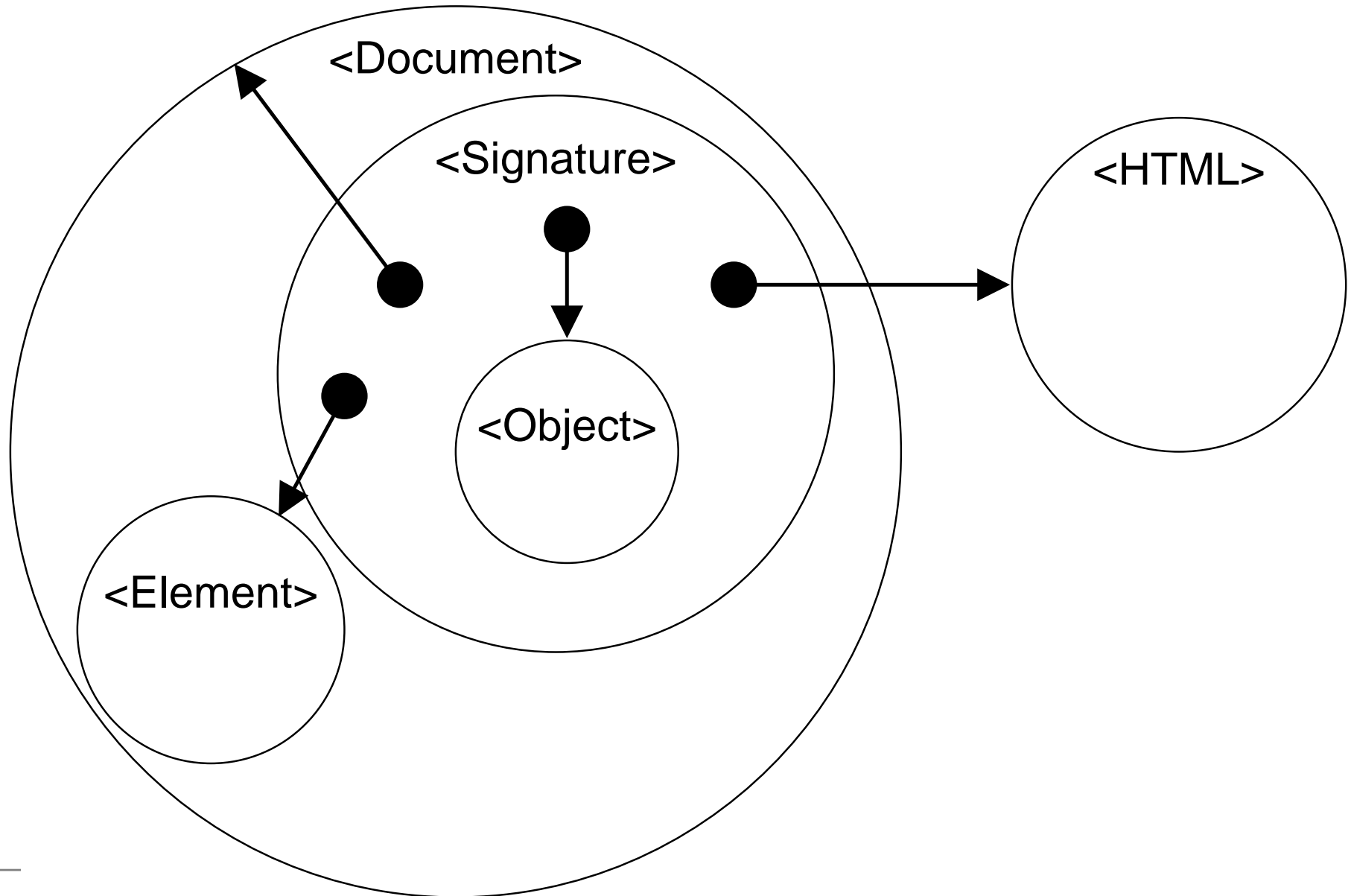
# Schema



# Beispiel

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
    <ds:SignatureMethod
      Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
    <ds:Reference URI="payment.xml">
      <ds:DigestMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <ds:DigestValue>E71h26OmCduj9rgFnUmOcw4Uu7Y=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>
    U3+JLn7CsHvFjx+sbbft7ESDxCBtsZ7V6F7mveA8/7idl9wioAyKig==
  </ds:SignatureValue>
</ds:Signature>
```

# Referenzen



# Erzeugung

## 1. Referenz-Elemente

- Transformation der Daten
- Berechnung des Hash-Werts
- Konstruktion des `<Reference>` Elements

## 2. Signatur

- Kanonisierung des `<SignedInfo>` Elements
- Berechnung des Hash-Werts
- Unterzeichnen mit dem Signatur-Schlüssel

# Validierung

## 1. Integrität der Referenzen

- Transformation der Daten
- Berechnung des Hash-Werts
- Vergleich mit dem `<DigestValue>`

## 2. Authentizität der Signatur

- Berechnung des Hash-Werts
- Prüfen mit dem Verifikations-Schlüssel
- Vergleich mit dem `<SignatureValue>`

# Sicherheit

Der Programmierer muss...

- Algorithmen auswählen
- Bedeutung der Signatur festlegen
- Dokumentformate oder Protokolle gestalten

Man beachte:

- Nur was signiert ist, ist auch sicher
- Nur was man sieht, soll man unterzeichnen

# Implementierungen

Hersteller	Name	Sprachen	Lizenz
Aleksey	XML Security Library	C	MIT (OSS)
Apache	XML Security	Java, C++	Apache (OSS)
Clarios	XML Signature/Enc.	Java, C++	royalty free
Entrust	Security Toolkit	Java	k.A.
IAIK	XML Signature Library	Java	800 €
IBM	XML Security Suite	Java	1000 \$
Microsoft	SignedXml	.NET	n.a.
NEC	XMLDSIG	Java	public domain
NeuClear	XMLSig	Java	GPL (OSS)
Phaos	Phaos XML	Java	k.A.
RSA Security	BSAFE Cert-J	Java	k.A.
SETCCE	XSign	COM/ActiveX	1940 €
Uni Pisa	Gapxse	Java	LGPL (OSS)
Verisign	XML Signature	Java	k.A.

# Standard-Dschungel

- XML Digital Signature (XMLDSIG)
- XML Encryption (XMLENC)
- XML Key Management (XKMS)
- Security Assertion Markup Language (SAML)
- XML Access Control Markup Language (XACML)
- Web Services Security (WSS)
- Platform for Privacy Preferences (P3P)
- eXtensible Rights Markup Language 2.0 (XrML)