
eXtensible Markup Language – Grundlagen

Christian Weber

c_weber@informatik.uni-kl.de

Überblick

- Einleitung – Was ist XML ?
- XML-Spezifikation
- XML-Verarbeitungsmodell

Was ist XML ?

1. XML ist eine Methode zur Speicherung strukturierter Daten in Textform

- Speicherungs- und Austauschformat
- Daten von Textauszeichnungen umgeben
- maschinenlesbar
- plattformunabhängig
- nicht alle Daten in XML speicherbar

Was ist XML ?

2. XML sieht aus wie HTML, ist aber kein HTML

- HTML (HyperText Markup Language)
- keine feste Bedeutung der Tags und Attribute

Was ist XML ?

3. XML ist Text, aber nicht zum lesen

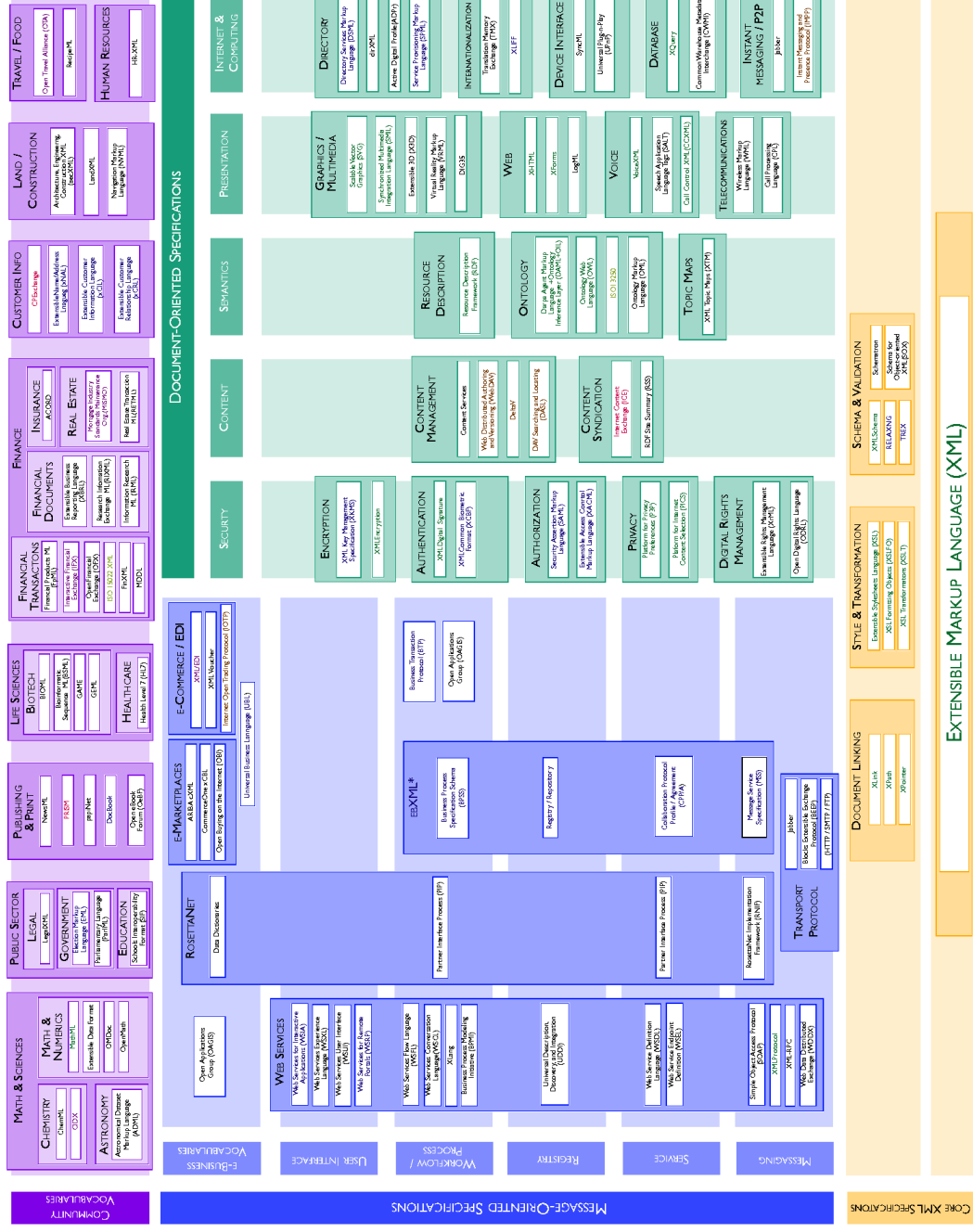
- Fehler sind auffindbar
- Format lange Zeit lesbar

4. XML ist ausführlich

- Dateien größer als in binären Formaten
- manuell Editierbar
- Dateigröße nicht mehr so wichtig
 - Festplattenkapazität günstiger
 - Komprimierung

Was ist XML ?

5. XML ist eine Familie von Technologien



Was ist XML ?

6. XML ist neu, aber nicht so neu

- 1969 Generalized Markup Language (GML)
- 1980 ANSI Entwurf einer standardisierten GML (SGML)
- 1986 SGML ISO-Standard: ISO 8879
- 1993 Entwurf von HTML
- 1994 Gründung des World Wide Web Consortium (W3C)
- 1996 Entwurf von XML
- 1998 Empfehlung von XML durch W3C
- 2000 XML-Namespaces
- 2001 XML-Schema

Was ist XML ?

7. XML ist modular

- XML Dokumente sind kombinierbar
- Definition von Namenräumen

Was ist XML ?

8. XML ist lizenzfrei, plattformunabhängig und gut unterstützt

- freie Werkzeuge für fast alle Plattformen
- nicht anbieterabhängig

XML-Spezifikation

- XML-Syntax
- XML-Dokumentstruktur
- Dokumenttypdefinitionen (DTD)
- Gültigkeit und Wohlgeformtheit von XML-Dokumenten
- XML-Namensräume
- XML-Schema

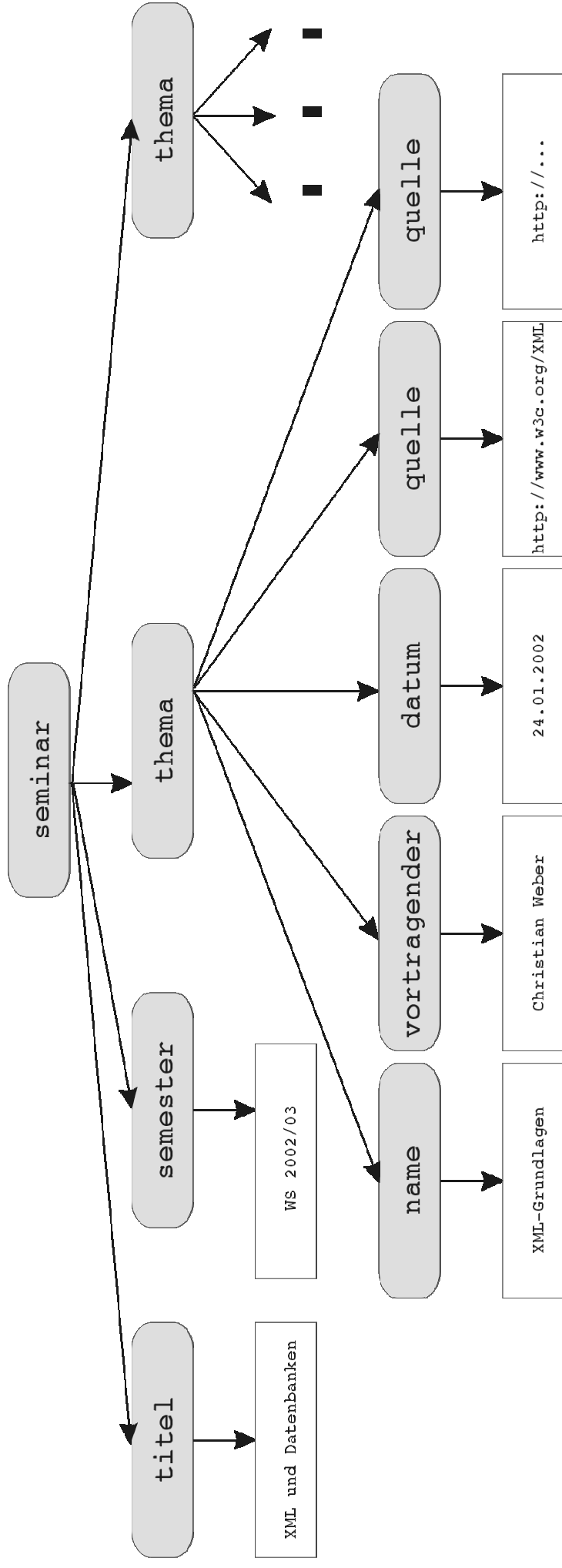
- **Elemente:**
 - `<titel>XML-Grundlagen</titel>`
 - `<titel></titel>`
 - `<titel/>`
- **Attribute:**
 - `<seminar semester="WS 2002/03">`
 - ...
- **gültige Zeichen in XML Namen:**
 - alphanumerische Zeichen
 - Unterstrich, Bindestrich, Punkt
 - Doppelpunkt (-> Namensräume)

Beispieldokument

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<seminar>
  <titel>XML und Datenbanken</titel>
  <semester>WS 2002/03</semester>
  <thema>
    <name>XML-Grundlagen</name>
    <vortragender>Christian Weber</vortragender>
    <datum>24.01.2003</datum>
    <quelle>http://www.w3c.org/XML/</quelle>
    <quelle>http://www.w3c.org/Style/XSL</quelle>
  </thema>
  <thema>
    <name>XML Verarbeitungsmodelle und Language Bindings</name>
    <vortragender>Christian Müller</vortragender>
    <quelle>http://www.w3c.org/XML</quelle>
  </thema>
</seminar>
```

Beispieldokument

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
<seminar semester="WS 2002/03">
  <titel>XML und Datenbanken</titel>
  <thema>
    <name>XML-Grundlagen</name>
    <vortragender>Christian Weber</vortragender>
    <datum>24.01.2003</datum>
    <quelle>http://www.w3c.org/XML/</quelle>
    <quelle>http://www.w3c.org/Style/XSL</quelle>
  </thema>
  <thema>
    <name>XML Verarbeitungsmodelle und Language Bindings</name>
    <vortragender>Christian Müller</vortragender>
    <quelle>http://www.w3c.org/XML/</quelle>
  </thema>
</seminar>
```



Kopf eines XML Dokuments

- XML Deklaration:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
```

- Deklaration einer Dokumenttyp-Definition:

```
<!DOCTYPE seminar SYSTEM  
"http://wwwdbis.informatik.uni-kl.de/seminar.dtd">
```

- Pfad im lokalen Dateisystem als Uniform Resource Identifier (URI)

Nicht erlaubte Zeichen / Kommentare

- **Entities:**
 - < <
 - > >
 - & &
 - " "
 - ' '
- **CDATA:**
 - <![CDATA[]]>
- **Kommentare:**
 - <!-- -->

Dokumenttyp-Definition (DTD)

- Grammatik für spezielle XML Sprache
- Auflistung aller erlaubter Elemente, Attribute im entsprechenden Kontext
- Definition von Entities
- Kennzeichnung der Häufigkeit der Elemente durch Suffixe:
 - ? kein oder ein Element
 - * kein oder beliebig viele Elemente
 - + ein oder mehrere Elemente
- alles muss explizit erlaubt werden

DTD Beispiele 1

Beispiel 1:

```
<!ELEMENT seminar (titel, thema*) >
<!ATTLIST seminar semester CDATA #REQUIRED>
<!ELEMENT titel (#PCDATA) >
<!ELEMENT thema (name, vortragender, datum?, quelle+) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT thema (#PCDATA) >
<!ELEMENT vortragender (#PCDATA) >
<!ATTLIST vortragender matr CDATA #IMPLIED>
<!ELEMENT datum (#PCDATA) >
<!ELEMENT quelle (#PCDATA) >
```

Beispiel 2:

```
<!ELEMENT vortragender (vorname, nachname, (telefon | email)) >
<!ELEMENT vorname (#PCDATA) >
<!ELEMENT nachname (#PCDATA) >
<!ELEMENT telefon (#PCDATA) >
<!ELEMENT email (#PCDATA) >
```

DTD Beispiele 2

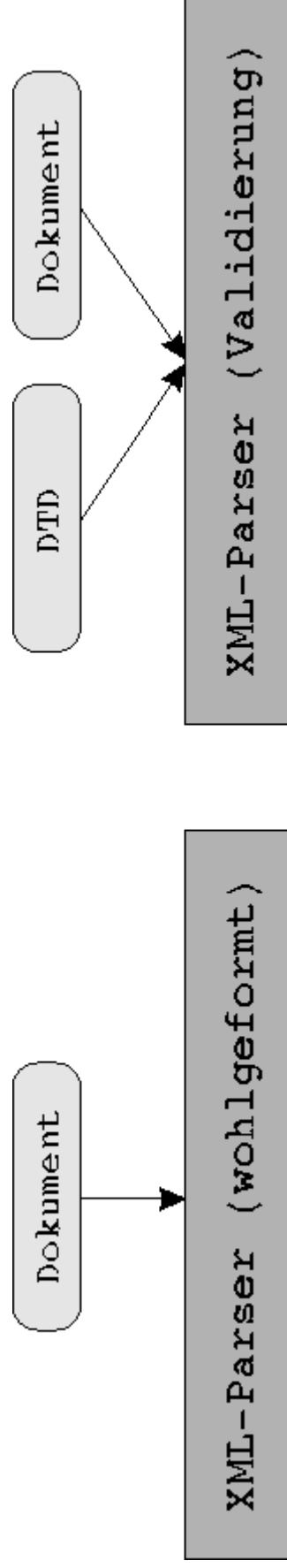
Beispiel3:

```
<?xml version="1.0" encoding="ISO-8859-1">
<!DOCTYPE lehrangebot [
  <!ELEMENT lehrangebot (vorlesung*, seminar*) >
  <!ELEMENT vorlesung (titel, semester, zusammenfassung) >
  <!ELEMENT seminar (titel, semester) >
  <!ELEMENT titel (#PCDATA) >
  <!ELEMENT semester (#PCDATA) >
  <!ELEMENT zusammenfassung (#PCDATA) >
  <!ENTITY sem "WS 2002/03">
  <!ENTITY dbaw SYSTEM "http://wwwdbis.informatik.uni-kl.de/DBAW.txt">
  <!ENTITY wws SYSTEM "http://wwwdbis.informatik.uni-kl.de/WWS.txt">
]>
<lehrangebot>
  <vorlesung>
    <titel>Datenbankanwendung</titel>
    <semester>&sem;</semester>
    <zusammenfassung>&dbaw;</zusammenfassung>
  </vorlesung>
  <vorlesung>
    <titel>Workflow und Web Services</titel>
    <semester>&sem;</semester>
    <zusammenfassung>&wws;</zusammenfassung>
  </vorlesung>
</lehrangebot>
```

Gültigkeit und Wohlgeformtheit von XML-Dokumenten

Regeln für die Wohlgeformtheit:

- XML-Deklaration muss am Anfang des Dokuments stehen
- Existenz genau eines Wurzelelements
- jedes Start-Tag besitzt auch ein End-Tag
- Elemente dürfen sich nicht überschneiden
- Werte von Attributen müssen in Anführungszeichen stehen
- Attribute eines Elementes müssen unterschiedliche Namen haben
- Ersetzung bestimmter Zeichen durch Entities



XML-Namesräume

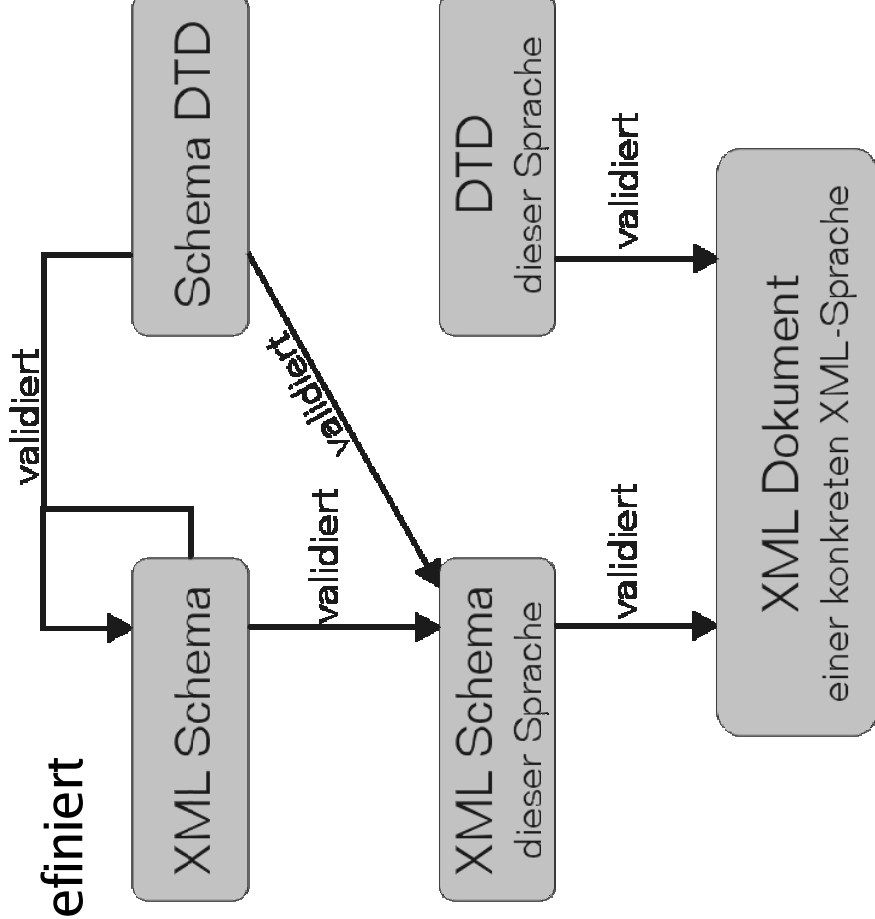
- Auflösung von Nameskonflikten
- Bindung von Elementen und Attributen an URI's
- Verwendung von Präfix statt URI's
- Trennung von Präfix und XML-Name durch einen Doppelpunkt
- Präfixe die mit `XML` beginnen, sind reserviert

Beispiel XML-Namesräume

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes">
<lehrveranstaltungen>
  <vor:vorlesung xmlns:vor="http://www.dbis.informatik.uni-kl.de/vorlesung">
    <vor:titel>Datenbankanwendung</vor:titel>
    <vor:thema>Inhalt und Motivation</vor:thema>
    <vor:thema>Anforderungen und Beschreibungsmodelle</vor:thema>
    <vor:thema>Logischer DB-Entwurf</vor:thema>
  </vor:vorlesung>
  <sem:seminar xmlns:sem="http://www.dbis.informatik.uni-kl.de/seminar" sem:semester="WS 2002/03">
    <sem:titel>XML und Datenbanken</sem:titel>
    <sem:thema>
      <sem:name>XML-Grundlagen</sem:name>
      <sem:vortragender matnr="123456">Christian Weber</sem:vortragender>
      <sem:datum>24.01.2002</sem:datum>
      <sem:quelle>http://www.w3.org/XML/</sem:quelle>
      <sem:quelle>http://www.w3.org/TR/1998/REC-xml-19980210</sem:quelle>
    </sem:thema>
    <sem:thema>
      <sem:name>XML-Verarbeitungsmodelle und Language Bindings</sem:name>
      <sem:vortragender>Christian Müller</sem:vortragender>
      <sem:quelle>http://www.w3.org/XML/</sem:quelle>
    </sem:thema>
  </sem:seminar>
</lehrveranstaltungen>
```

XML-Schema

- XML-Schema-Dokument ist selbst ein XML-Dokument
- Definiert wie DTD die Elemente und Attribute die vorkommen dürfen
- Elementen und Attributen wird ein Datentyp zugewiesen
 - 44 einfache Datentypen vordefiniert
- Definition eigener Datentypen
 - Wertebereiche



Beispiel XML-Schema Teil 1

```
<?xml version="1.0" encoding="iso8859-15"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="matType">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="100000">
        <xs:maxExclusive value="1000000">
          </xs:restriction>
        </xs:simpleType>
      </xs:restriction>
    </xs:simpleType>
  <xs:complexType name="vortragenderType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="matNr" type="matType" use="implied"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

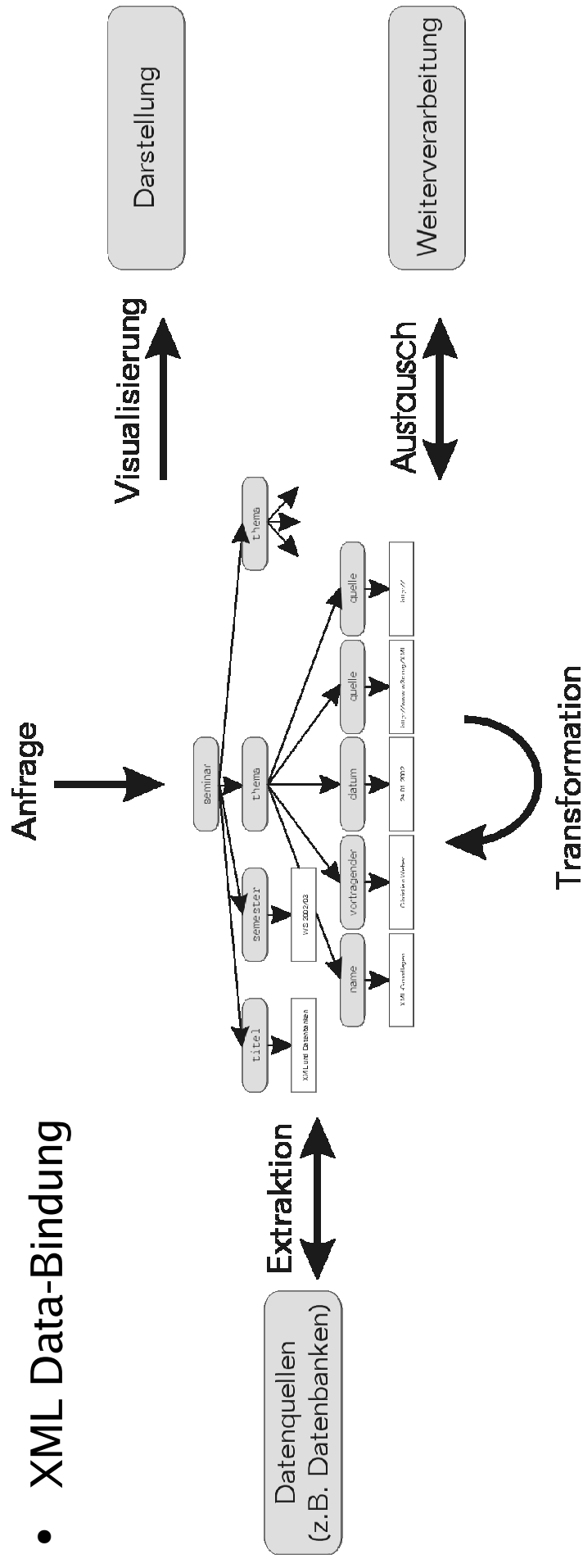
Beispiel XML Schema Teil 2

```
<xs:complexType name="themaType">
  <xs:sequence>
    <xs:element name="name" type="xs:string"
      minOccurs="1" maxOccurs="1" />
    <xs:element name="vortragender" type="vortragenderType"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="datum" type="xs:date"
      minOccurs="0" maxOccurs="1" />
    <xs:element name="quelle" type="xs:anyURI"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:element name="seminar">
<xs:complexType>
  <xs:sequence>
    <xs:element name="titel" type="xs:string"
      minOccurs="1" />
    <xs:element name="thema" type="themaType"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

XML-Verarbeitungsmodell

- Parsen von XML-Dokumenten
- XML-Anfragesprachen
- XML & Datenbanken
- Darstellung und Transformation von XML-Dokumenten
- XML WebServices
- XML Data-Bindung



SAX und DOM

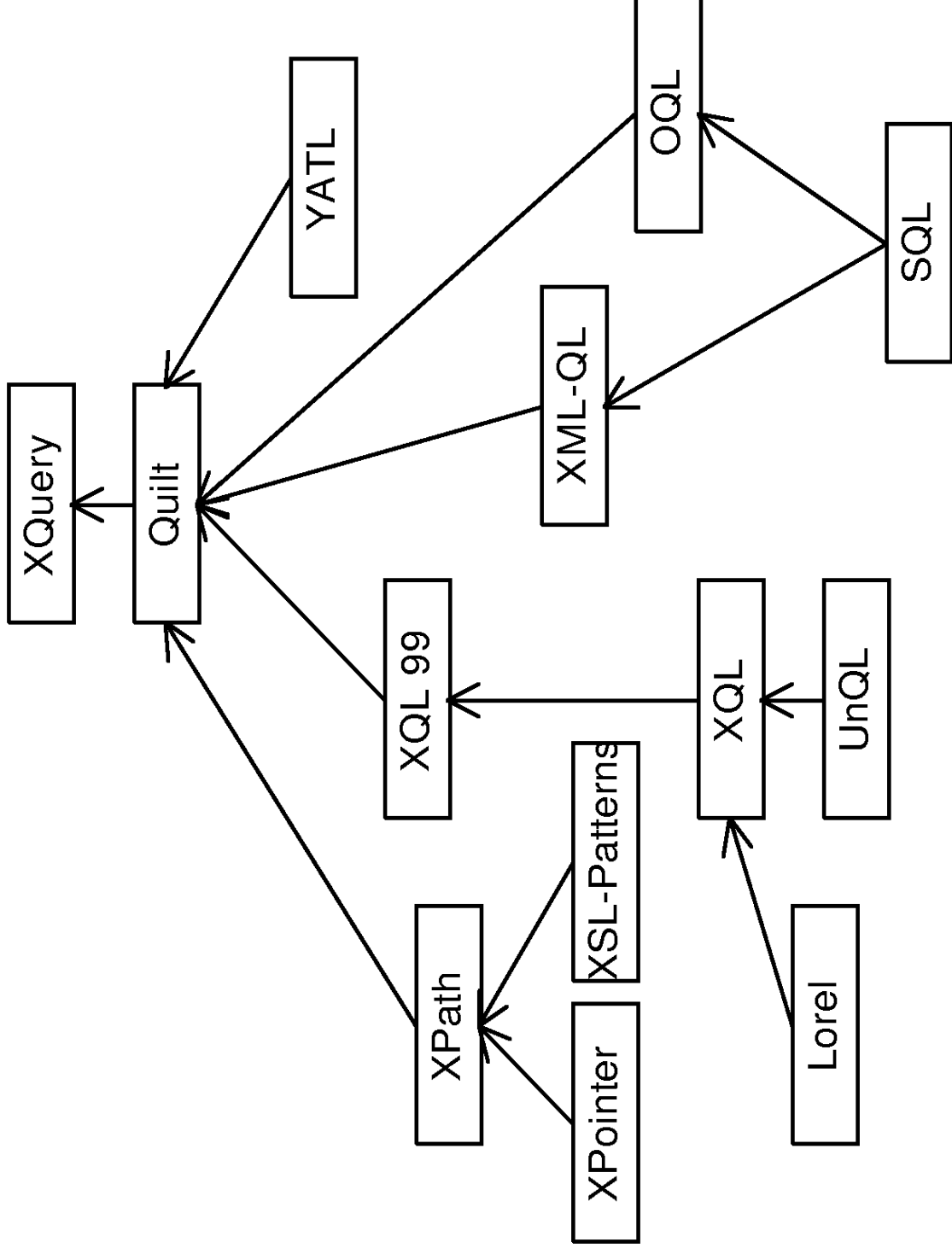
Simple API for XML:

- eventbasiert
- Anwender muss Eventhandler implementieren
- nur Lesezugriff
- schnell
- auf für große Dokumente geeignet

Document Object Model:

- Baumstruktur im Speicher
- Methoden zur Navigation und Manipulation
- auch Schreibzugriff
- langsam

XML-Anfragesprachen



XPath Beispiele

XPath-Ausdruck zur Lokalisierung aller Vortragender:

```
/seminar/thema/vortragender
```

Ergebnis:

```
<vortragender>Christian Weber</vortragender>  
<vortragender>Christian Müller</vortragender>
```

XPath-Ausdruck zur Lokalisierung aller direkten Kindknoten des aktuellen

Knotens:

```
/seminar/child::*
```

Ergebnisknotenmenge:

```
{titel, thema, thema}
```

Vergleich von XQuery und SQL

<i>XQuery</i>	<i>SQL</i>
FOR \$k IN /seminar RETURN \$k	SELECT * FROM seminar
FOR \$k IN /seminar/thema WHERE count (\$k/quelle) > 10 RETURN \$k/vortragender	SELECT vortragender FROM seminare GROUP BY vortragender HAVING COUNT (quelle) > 10

Probleme:

- XML-Dokumente enthalten beliebig viele ineinander verschachtelte Elemente
- Elemente können sich innerhalb eines Elements beliebig oft wiederholen
- Elemente müssen nicht immer die gleichen Kinderelemente enthalten
- In XML-Dokumenten ist die Reihenfolge der Elemente ein fester Bestandteil der Dokumentstruktur

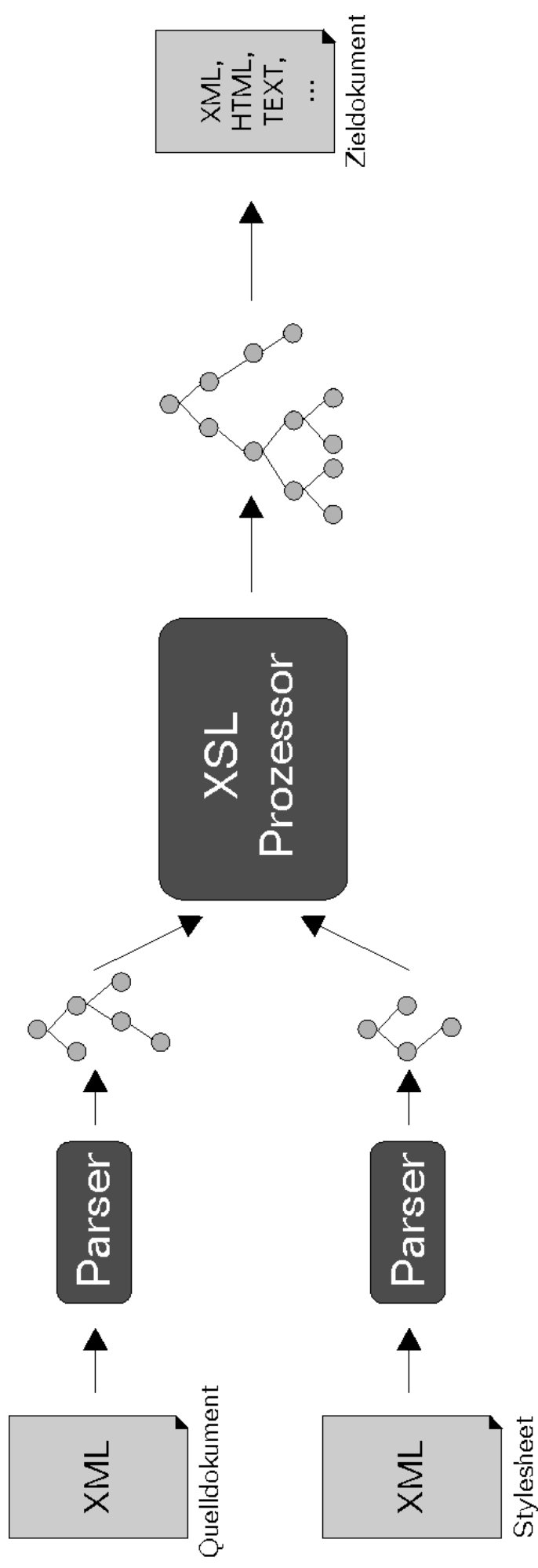
XML und objektorientierte Datenbanken

- Speicherung von hierarchisch strukturierten Daten möglich
- optimiert auf Speicherung von objektorientierten Programmen
- nicht effizient bei großen Datenmengen
- keine Unterstützung von XPath (--> XML Datenbanken)

Darstellung und Transformation von XML-Dokumenten

eXtensible Stylesheet Language (XSL)

- eXtensible Stylesheet Language Transformation (XSLT)
- eXtensible Stylesheet Language Formatting Objects (XSL-FO)



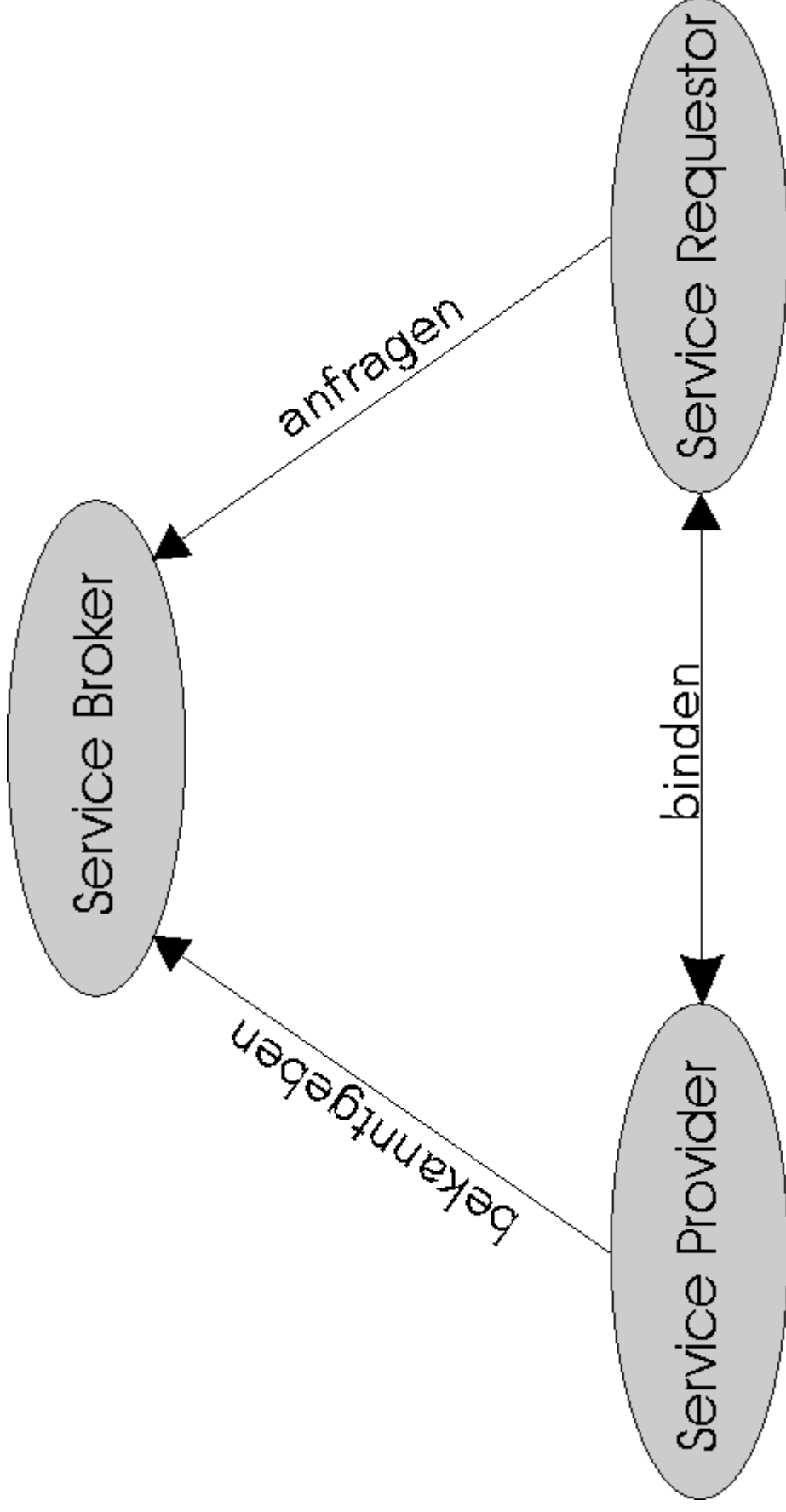
Beispiel XSLT Stylesheet

```
<?xml version="1.0" encoding="ISO-8859-1">
<xsl:stylesheet version="1.0" xmlns:xsl="http://www...">

<xsl:template match="/">
  <html><body>
    <h1>Titel:<xsl:value-of select="titel"/></h1>
    <xsl:apply-templates/>
  </body></html>
</xsl:template>

<xsl:template match="thema">
  <p>
Thema:<xsl:value-of select="name"/><br>
Person:<xsl:value-of select="vortragender"/>
  </p>
</xsl:template>

</xsl:stylesheet>
```

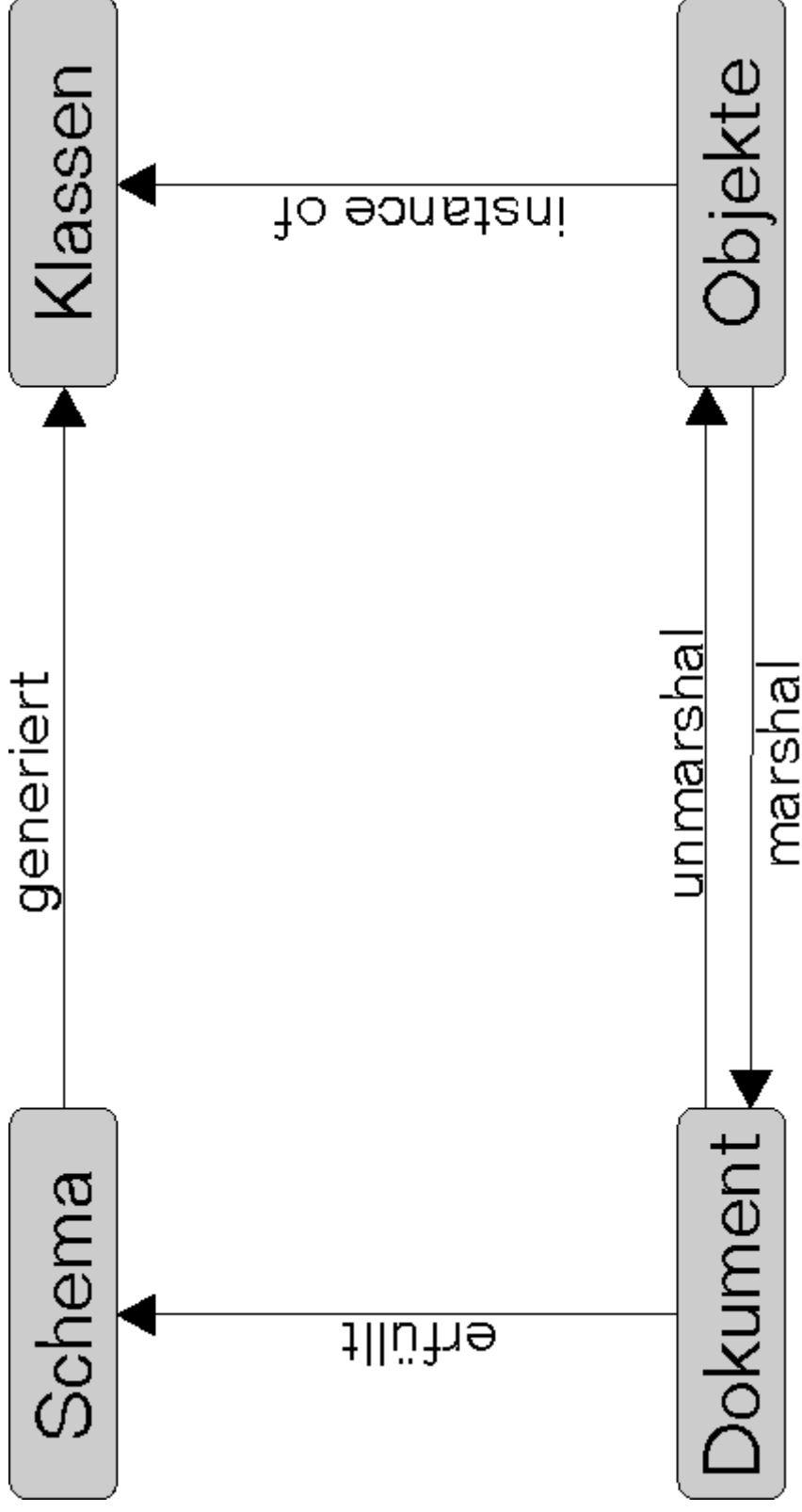


- Simple Object Access Protocol (SOAP)
- Universal Description, Discovery and Integration (UDDI)
- WebService Description Language (WSDL)

XML-Data-Binding

- Verbindung von XML und Hochsprache
 - keine Sammlung von Zugriffsroutinen (vgl. SAX, DOM)
 - spezifische Speicherungs- und Verarbeitungsstrukturen
- zwei verschiedene Kategorien:
 - Konfiguration zur Entwicklungszeit
 - Mapping
 - Generieren von Klassen
 - dynamisch zur Laufzeit einsetzbar

XML-Data-Binding



Zusammenfassung und Ausblick

- Zusammenfassung:
 - grundlegender Aufbau von XML Dokumenten
 - Vorteile von XML als Speicherungs- und Austauschformat
 - XML-Verarbeitungsmodell
- Ausblick:
 - Anfrageverarbeitung bzw. -formulierung
 - Datenbank-orientierte Verarbeitung
 - Dokumenten-orientierte Verarbeitung
 - XML-basierte Integrationskonzepte