

# DAIMLERCHRYSLER

## **Das Technikgebiet XML: Herkunft, praktische Anwendung und Zukunft**

Mario Jeckle

DaimlerChrysler Forschungszentrum Ulm

[mario.jeckle@daimlerchrysler.com](mailto:mario.jeckle@daimlerchrysler.com)

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

[www.jeckle.de](http://www.jeckle.de)

## Gliederung

*Der XML Hype ...*

- I Entwicklung der W3C eXtensible Markup Language  
Standards und Sprachen  
Das XML-Universum
  
- II Die XML-Sprachfamilie und ihre Anwendung
  - Hypertext: XHTML
  - Datenbeschreibung: XML Schema
  - Metadatenaustausch und Sprachdesign: OMG's XMI
  - Präsentation und Transformation: XSL(T)
  - SOAP und die XML Protocols
  
- III XML – Die Zukunft des Web?

## XML everywhere ...

Das Datenformat [XML] erleichtert den Informationsaustausch zwischen vernetzten Computern

XML schickt sich an in die Fußstapfen von HTML zu treten

[c't 10/2000, p. 200]

Das XML-Format, [...] das richtige Werkzeug zur Herstellung eigener Webinhalte

[DER SPIEGEL, 2000-06-23]

Sinnliche Suchmaschine [...] existierende Systeme wie [...] XML

[DER SPIEGEL, 2000-06-07]

Nachfolger für ungeliebte Cookies [...] Enge Verbindung von Java mit Erweiterung des HTML -Standards XML, [...]

[DER SPIEGEL, 1999-10-05]

Alle Dokumente sind gleich

[SZ, 1999-02-16]

Die Extended Markup Language für eCommerce

[F.A.Z.]

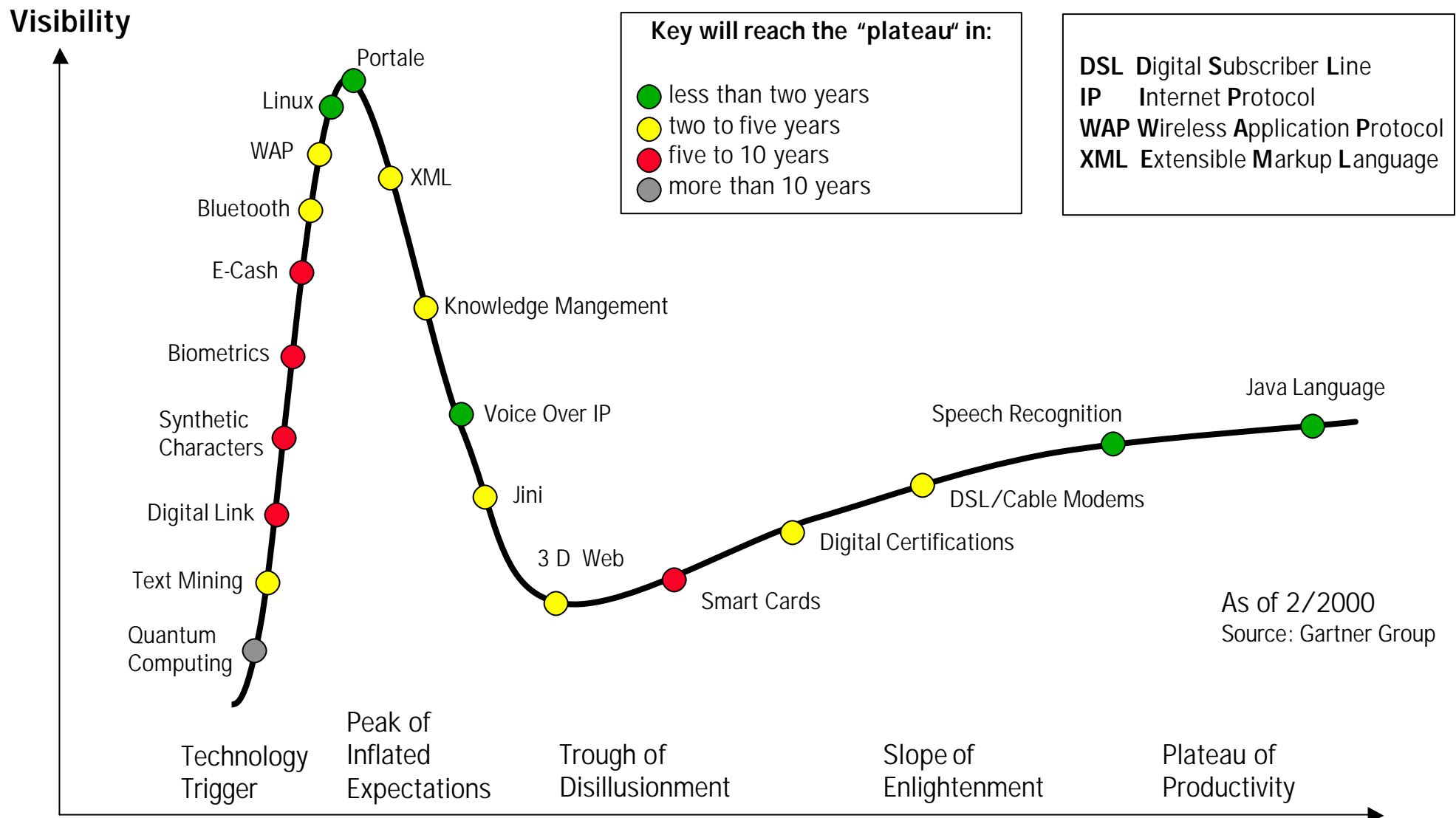
Sortieren statt Stottern  
Programmiersprache HTML stößt an ihre Grenzen  
XML ist kommender Code im Netz

[Süddeutsche Zeitung, 2000-01-11]

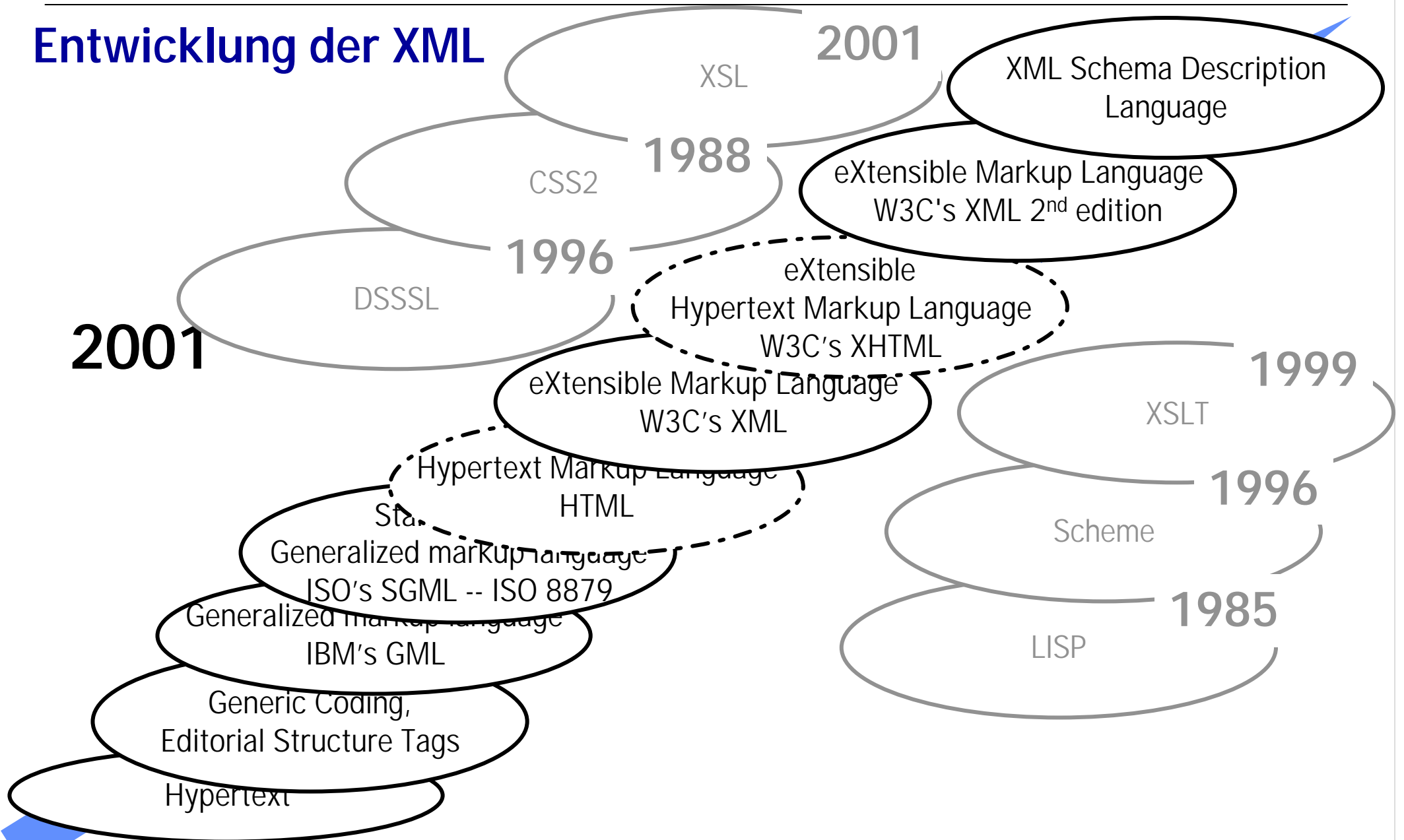
Ein digitales Esperanto für das Internet

[Die Welt, 2000-10-07]

# Gartner's Hype Life Cycle Model

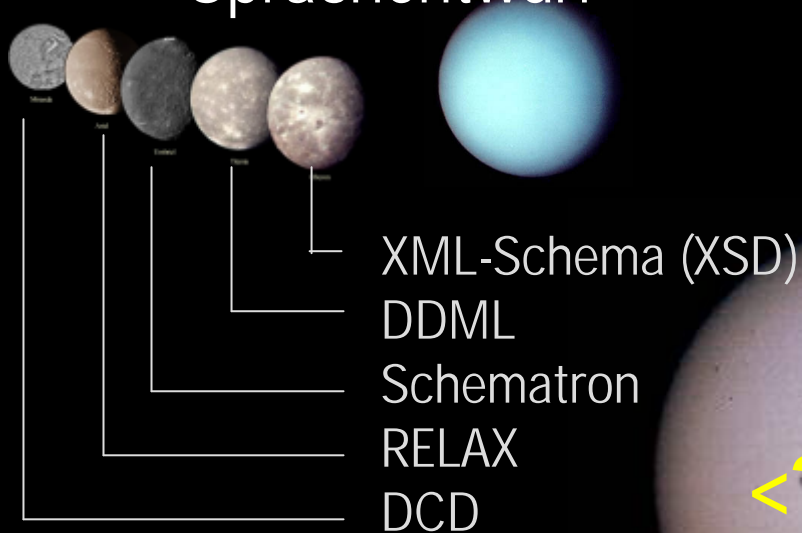


# Entwicklung der XML



# The XML-Univers

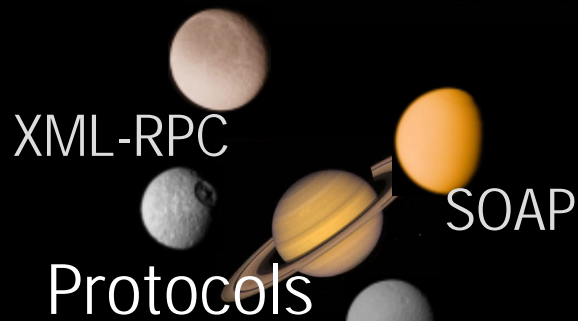
## Sprachentwurf



## Programmierung/ Schnittstellen



## Anwendungen



## Protocols

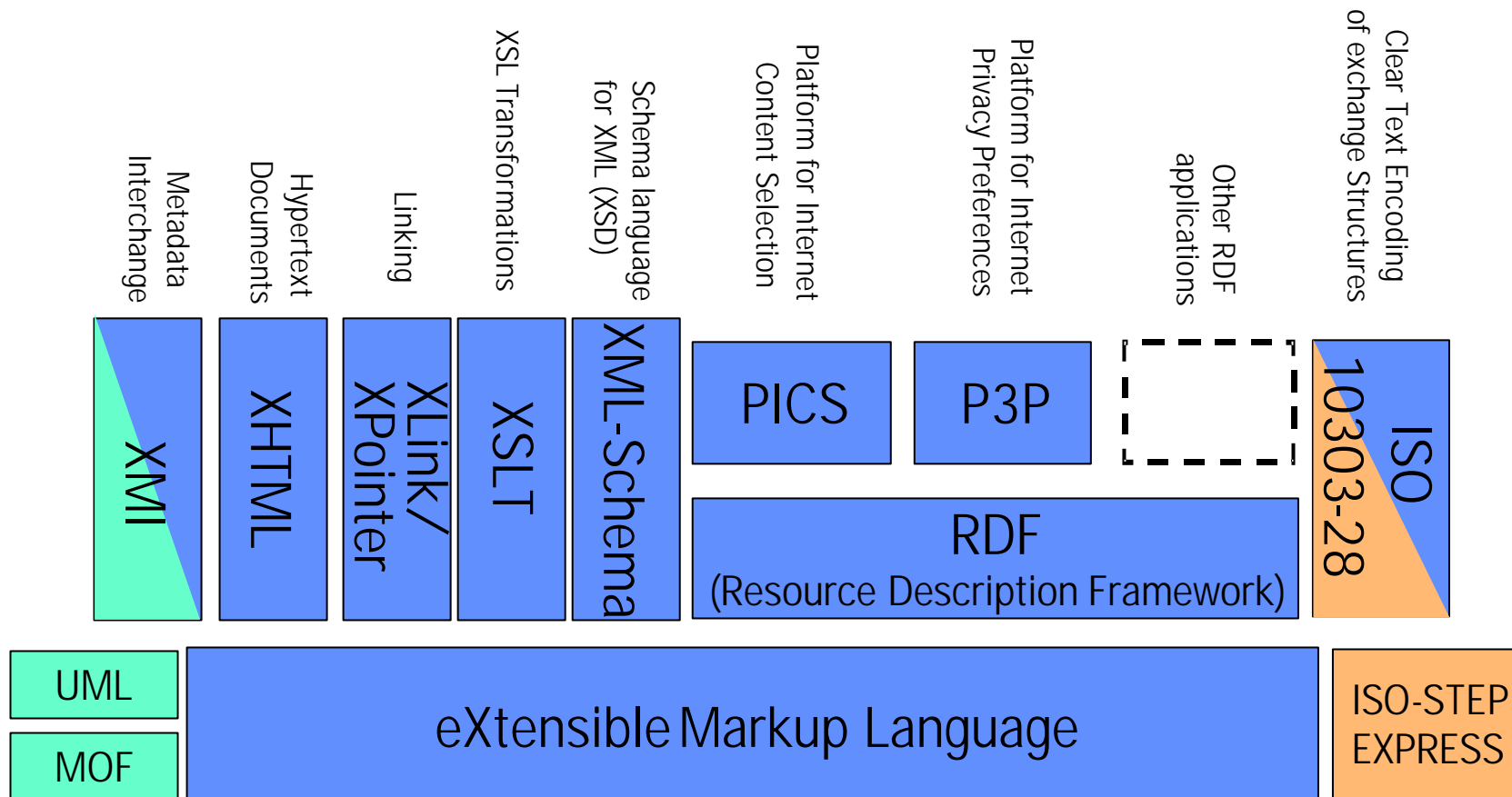
## Editoren

## werkzeugunterstützung

Stylesheet, Linkingprozessoren, ..

<?XML!>

# Die XML-Sprachfamilie



# HTML und XML



```

<html>
<head><title>jeckle.de</title></head>
<body bgcolor="#003366" topmargin="0" leftmargin="0" marginwidth="0" marginheight="0">
<table width="100%" border="0" cellspacing="0" cellpadding="0" height="65%" bgcolor="#f0f0f0">
<tr>
<td valign="bottom">&nbsp;</td>
<td colspan="2" valign="bottom" align="center">
<table width="800" border="0" cellspacing="0" cellpadding="0" align="center" height="315">
<tr bgcolor="#f0f0f0">
<td height="25">&nbsp;</td> <td height="25">&nbsp;</td>
<td height="25">&nbsp;</td> <td height="25">&nbsp;</td>
<td height="25">&nbsp;</td> <td height="25">&nbsp;</td>
<td height="25">&nbsp;</td> <td height="25">&nbsp;</td>
...

```



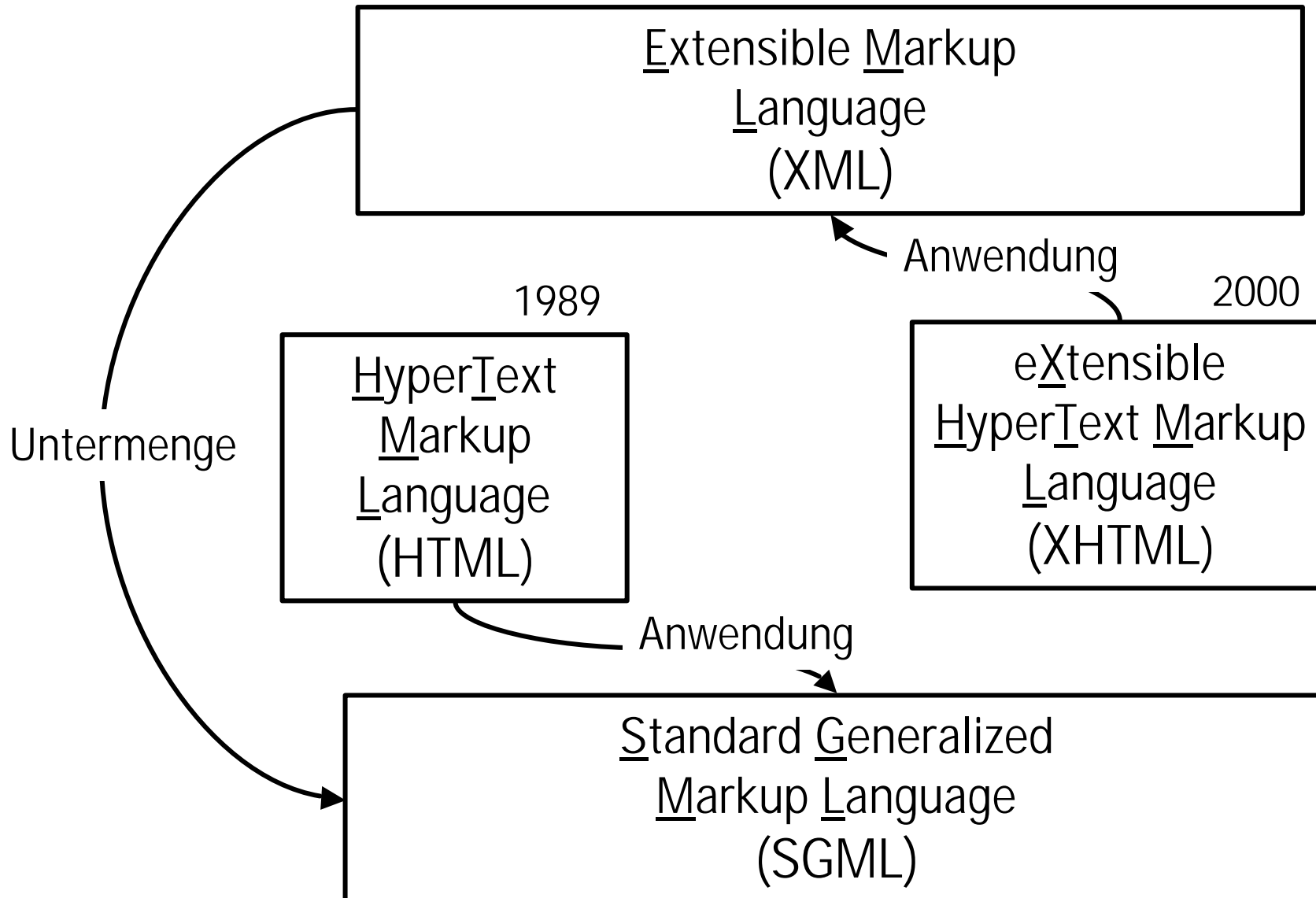
**<b>Tim Berners-Lee</b>**

Beginn Fettdruck (*bold*)

Ende Fettdruck

- HTML legt hauptsächlich das Präsentationsverhalten fest
- XML definiert die syntaktische Repräsentation des Information

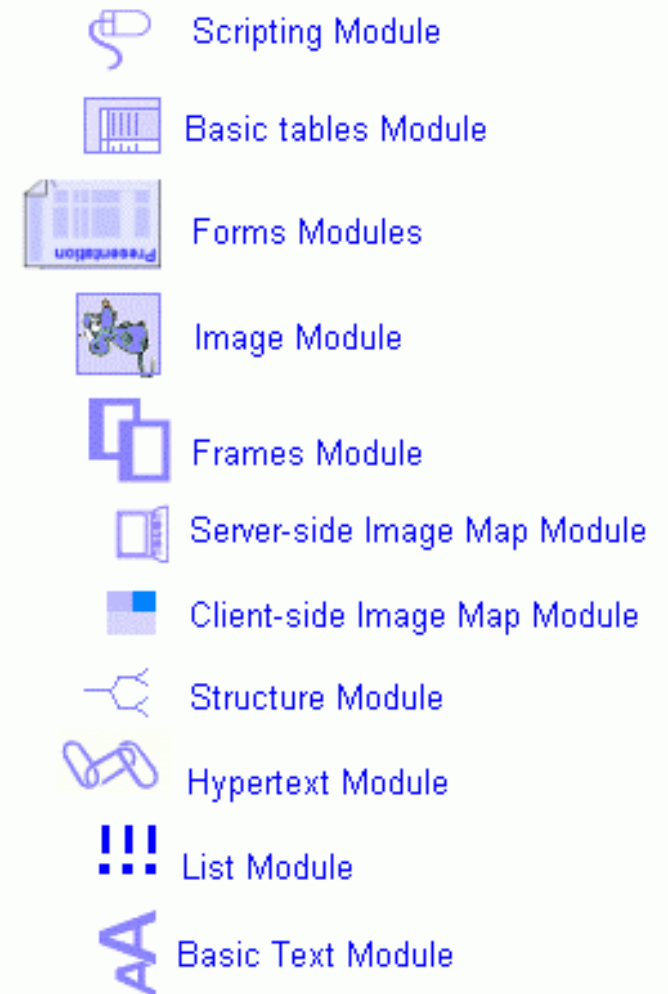
# SGML – HTML -- XML



## XHTML v1.1 und die Zukunft ...

- Modulares XHTML ist die funktionale Dekomposition von XHTML.
- Separierung der in XHTML adressierten verschiedenen Problembereiche, um diese
  - getrennt einsetzen zu können
  - erweitern zu können, ohne XHTML-Standard zu verletzen

XHTML 1.1 modules : some examples



## XML-Strukturprimitive

### Start Tag

<vortrag datum="2000-12-07">

Tag Name

Attribut

Attribut Name

Attribut Wert

Element

Das Technikgebiet XML: Herkunft,  
praktische Anwendung und Zukunft

<autor>Mario Jeckle</autor>

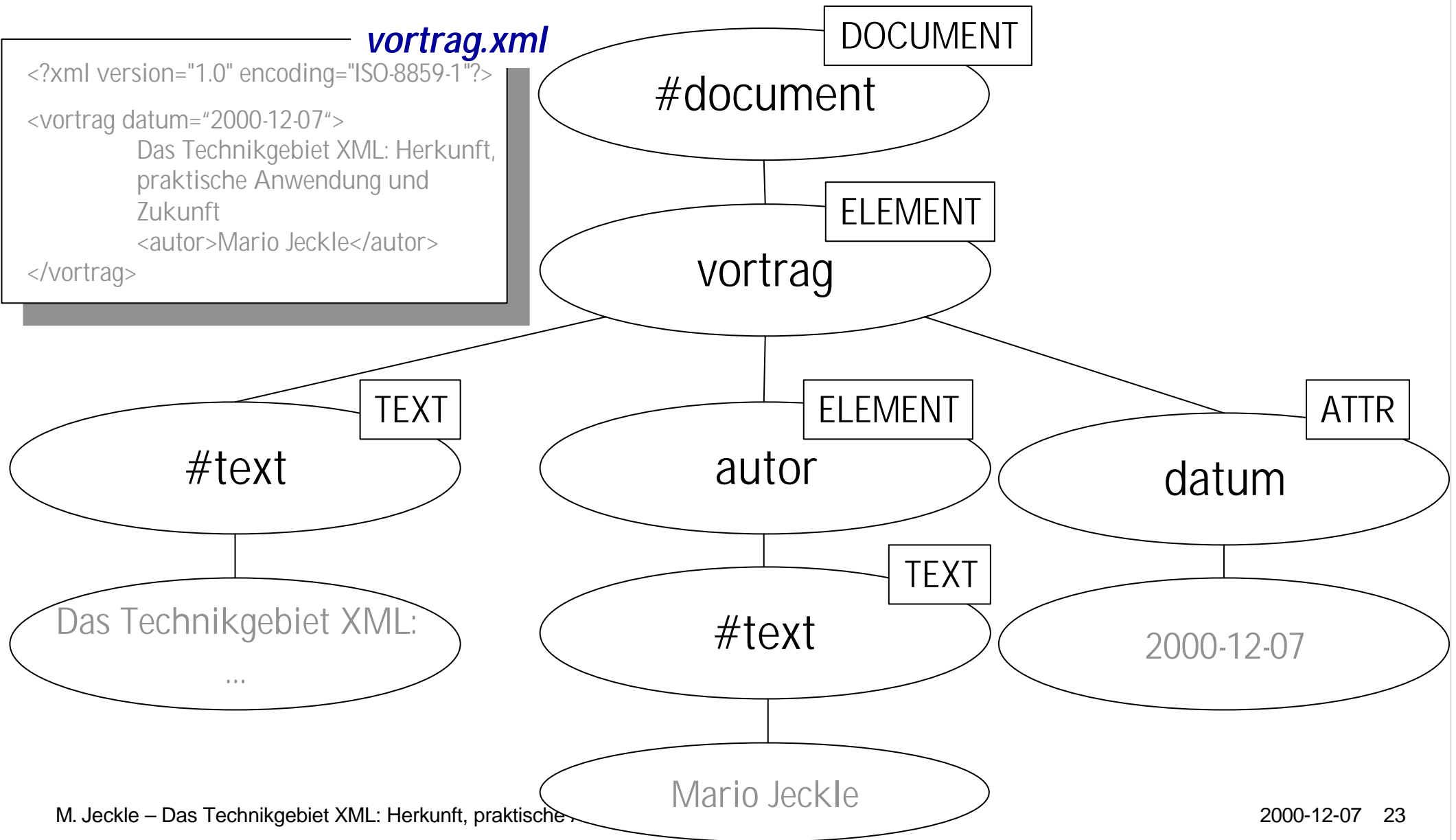
</vortrag>

End Tag

# XML-Dokumentstruktur

*vortrag.xml*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<vortrag datum="2000-12-07">
  Das Technikgebiet XML: Herkunft,
  praktische Anwendung und
  Zukunft
  <autor>Mario Jeckle</autor>
</vortrag>
```



## Die *Document Type Definition*

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

```
<vortrag datum="2000-12-07">
```

Das Technikgebiet XML: Herkunft,  
praktische Anwendung und Zukunft

```
<autor>Mario Jeckle</autor>
```

```
</vortrag>
```

```
<!ELEMENT vortrag (#PCDATA | autor)*>
```

```
<!ATTLIST vortrag  
    datum CDATA #IMPLIED>
```

```
<!ELEMENT autor (#PCDATA)*>
```

## Die *Document Type Definition*

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

```
<vortrag datum="2000-12-07">
```

```
  Das Technikgebiet XML: Herkunft,  
  praktische Anwendung und Zukunft
```

```
  <autor>Mario Jeckle</autor>
```

```
</vortrag>
```

```
<!ELEMENT vortrag (#PCDATA | autor)*>
```

```
<!ATTLIST vortrag  
  datum CDATA #IMPLIED>
```

```
<!ELEMENT autor (#PCDATA)*>
```

## Die *Document Type Definition*

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

```
<vortrag datum="2000-12-07">
```

```
  Das Technikgebiet XML: Herkunft,  
  praktische Anwendung und Zukunft
```

```
  <autor>Mario Jeckle</autor>
```

```
</vortrag>
```

```
<!ELEMENT vortrag (#PCDATA | autor)*>
```

```
<!ATTLIST vortrag  
  datum CDATA #IMPLIED>
```

```
<!ELEMENT autor (#PCDATA)*>
```

# Grenzen der DTD und notwendige Erweiterungen

## Structural schemas

Mächtigkeit annähernd ähnlich des bestehenden DTD-Mechanismus um Dokumentstruktur (Reihenfolge, Auftrittsvielfachheit von Elementen und Attribute) zu beschreiben.

Insbesondere sollen folgende Erweiterungen verwirklicht werden:

- Namespace Integration
- Definition von Einschränkungen für Elementinhalte
- Integration Strukturschema und primitive Datentypen
- Vererbung: DTD unterstützt nur *part-of*-Beziehungen
- Erweiterter Referenzierungsmechanismus (URI)
- Kommentare (*human* und *machine readable*)
- Nutzung der bestehenden XML Technologie

```
vortrag.xml  
<?xml version="1.0" encoding="ISO-8859-1"?>  
<vortrag datum="2000-12-07">  
  Das Technikgebiet XML: Herkunft,  
  praktische Anwendung und  
  Zukunft  
  <autor>Mario Jeckle</autor>  
</vortrag>
```

# Grenzen der DTD und notwendige Erweiterungen

## Primitive Datentypen ...als Ersatz für #PCDATA

```
vortrag.xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<vortrag datum="2000-12-07">
  Das Technikgebiet XML: Herkunft,
  praktische Anwendung und
  Zukunft
  <autor>Mario Jeckle</autor>
</vortrag>
```

- „klassische“ atomare Datentypen, ergänzt um SQL-artige, wie *integer*, *date*.
- Programmiersprachen-übliche (typischerweise Java-artige) *build-in types*
- uninterpretierte Binärstrukturen
- (durch Anwender) erweiterbares Typsystem
- lexikalische Definitionen
- Mengentypen
- Einschränkungen an Typen (Domänenrestriktion)
- *Web-enabled* (URIs, base64 encoding, etc.)

## Existierende Ansätze

### Erweiterungen des bestehenden (SGML-/XML-)DTD-Mechanismus

- Data Types for DTD (DT4DTD)

### Wissensbeschreibung

- Document Content Description for XML (DCD)  
(*RDF basierte Weiterentwicklung von XML-Data*)

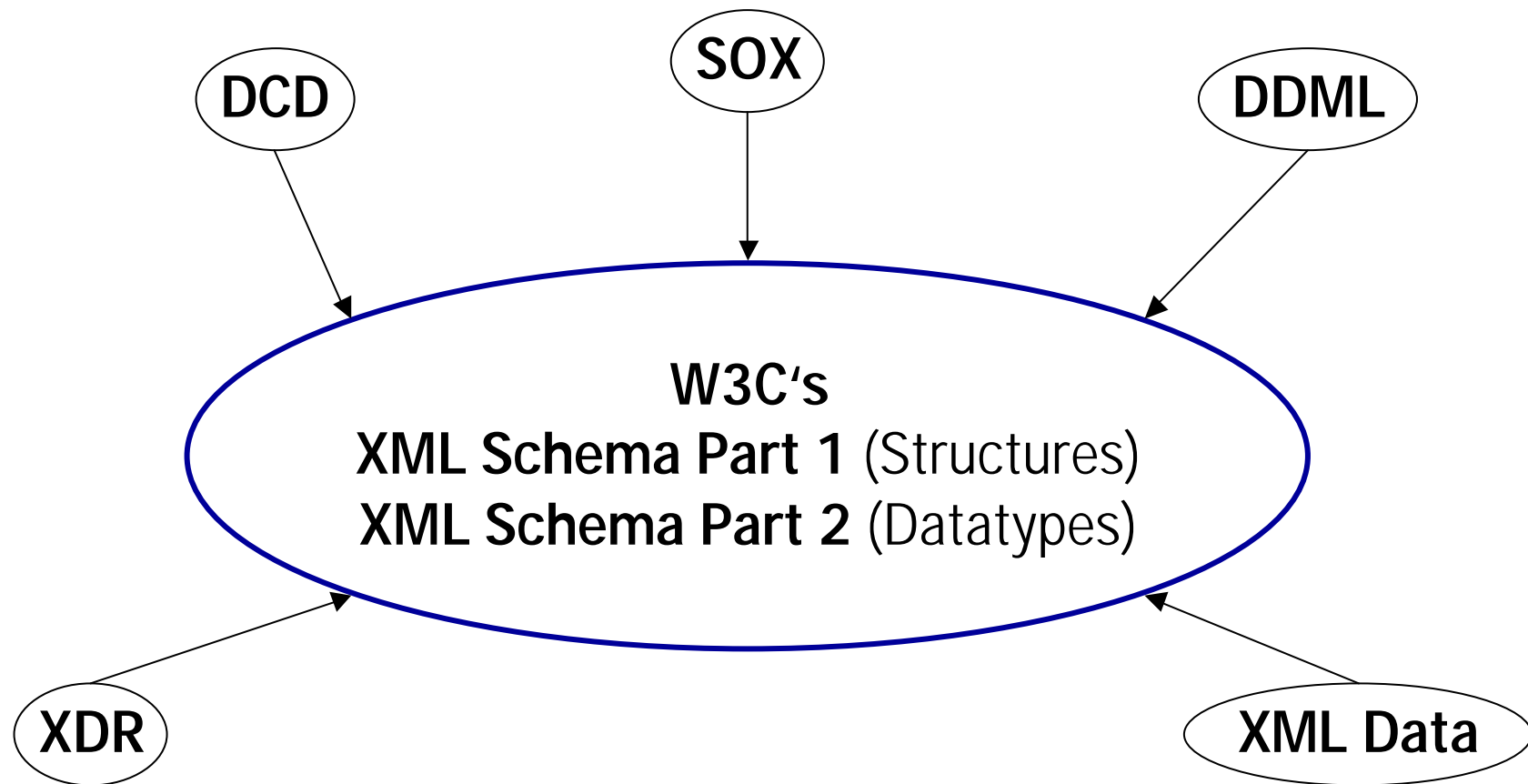
### Inspiziert durch XML-API-Entwicklung

- Schema for Object oriented XML (SOX)

### XML-Sprachen zur Inhaltsbeschreibung

- Document Definition Markup Language/XSchema (DDML)
- Schematron (XSLT-basierte Auswertung der Dokumentstruktur)
- XML-Data/XML-Data Reduced (XDR) (*erster Ansatz noch vor Verabschiedung XML 1.0*)
- Document Structure Description (DSD)
- Regular Language description for XML (RELAX)
- **W3C's XML-Schema**

## W3C's XML Schema



W3C's XML-Schema konsolidiert und integriert die wesentlichen existierenden Ansätze und entwickelt sie fort.

## Inhaltliche Charakteristika W3C's XML Schema

Gegenüber DTD ...

- stark erweiterte build-in Typen
  - Basistypen angelehnt an ISO 11404, SQL und Java
  - Datums- und Zeittypen angelehnt an ISO 8601
- Erweiterbares Typsystem
  - Typextension und -restriktion
  - lexikalische Definition
- Typbeziehungen (äquivalente Typen)
- „echte“ XML-Sprache
- Explizite Beziehungskardinalität
- Kontextabhängige Eindeutigkeit (Schlüssel und Referenzen)
- Elemente gleichen Namens mit verschiedenen Inhalten (Polymorphie)

## Schema Part 1 (Structures) -- *complex types*

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

```
<vortrag datum="2000-12-07">
```

Das Technikgebiet XML: Herkunft,  
praktische Anwendung und Zukunft

```
<autor>Mario Jeckle</autor>
```

```
</vortrag>
```

```
<!ELEMENT vortrag (#PCDATA | autor)*>
```

```
<!ATTLIST vortrag
    datum CDATA #IMPLIED>
```

```
<!ELEMENT autor (#PCDATA)*>
```

```
<xsd:schema xmlns:xsd = "http://www.w3.org/2000/10/XMLSchema">
```

```
<xsd:element name = "autor" type = "xsd:string"/>
```

```
<xsd:element name = "vortrag">
```

```
<xsd:complexType mixed = "true">
```

```
<xsd:choice minOccurs = "1" maxOccurs = "unbounded">
```

```
<xsd:element ref = "autor"/>
```

```
</xsd:choice>
```

```
<xsd:attribute name = "datum" use = "optional" type = "xsd:date"/>
```

```
</xsd:complexType>
```

```
</xsd:element>
```

```
</xsd:schema>
```

## Schema Part 1 (Structures) -- *elements*

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

```
<vortrag datum="2000-12-07">
```

Das Technikgebiet XML: Herkunft,  
praktische Anwendung und Zukunft

```
<autor>Mario Jeckle</autor>
```

```
</vortrag>
```

```
<!ELEMENT vortrag (#PCDATA | autor)*>
```

```
<!ATTLIST vortrag
    datum CDATA #IMPLIED>
```

```
<!ELEMENT autor (#PCDATA)*>
```

```
<xsd:schema xmlns:xsd = "http://www.w3.org/2000/10/XMLSchema">
```

```
<xsd:element name = "autor" type = "xsd:string"/>
```

```
<xsd:element name = "vortrag">
```

```
<xsd:complexType mixed = "true">
```

```
<xsd:choice minOccurs = "1" maxOccurs = "unbounded">
```

```
<xsd:element ref = "autor"/>
```

```
</xsd:choice>
```

```
<xsd:attribute name = "datum" use = "optional" type = "xsd:date"/>
```

```
</xsd:complexType>
```

```
</xsd:element>
```

```
</xsd:schema>
```

## Schema Part 1 (Structures) -- *attributes*

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

```
<vortrag datum="2000-12-07">
```

Das Technikgebiet XML: Herkunft,  
praktische Anwendung und Zukunft

```
<autor>Mario Jeckle</autor>
```

```
</vortrag>
```

```
<!ELEMENT vortrag (#PCDATA | autor)*>
```

```
<!ATTLIST vortrag
    datum CDATA #IMPLIED>
```

```
<!ELEMENT autor (#PCDATA)*>
```

```
<xsd:schema xmlns:xsd = "http://www.w3.org/2000/10/XMLSchema">
```

```
<xsd:element name = "autor" type = "xsd:string"/>
```

```
<xsd:element name = "vortrag">
```

```
<xsd:complexType mixed = "true">
```

```
<xsd:choice minOccurs = "1" maxOccurs = "unbounded">
```

```
<xsd:element ref = "autor"/>
```

```
</xsd:choice>
```

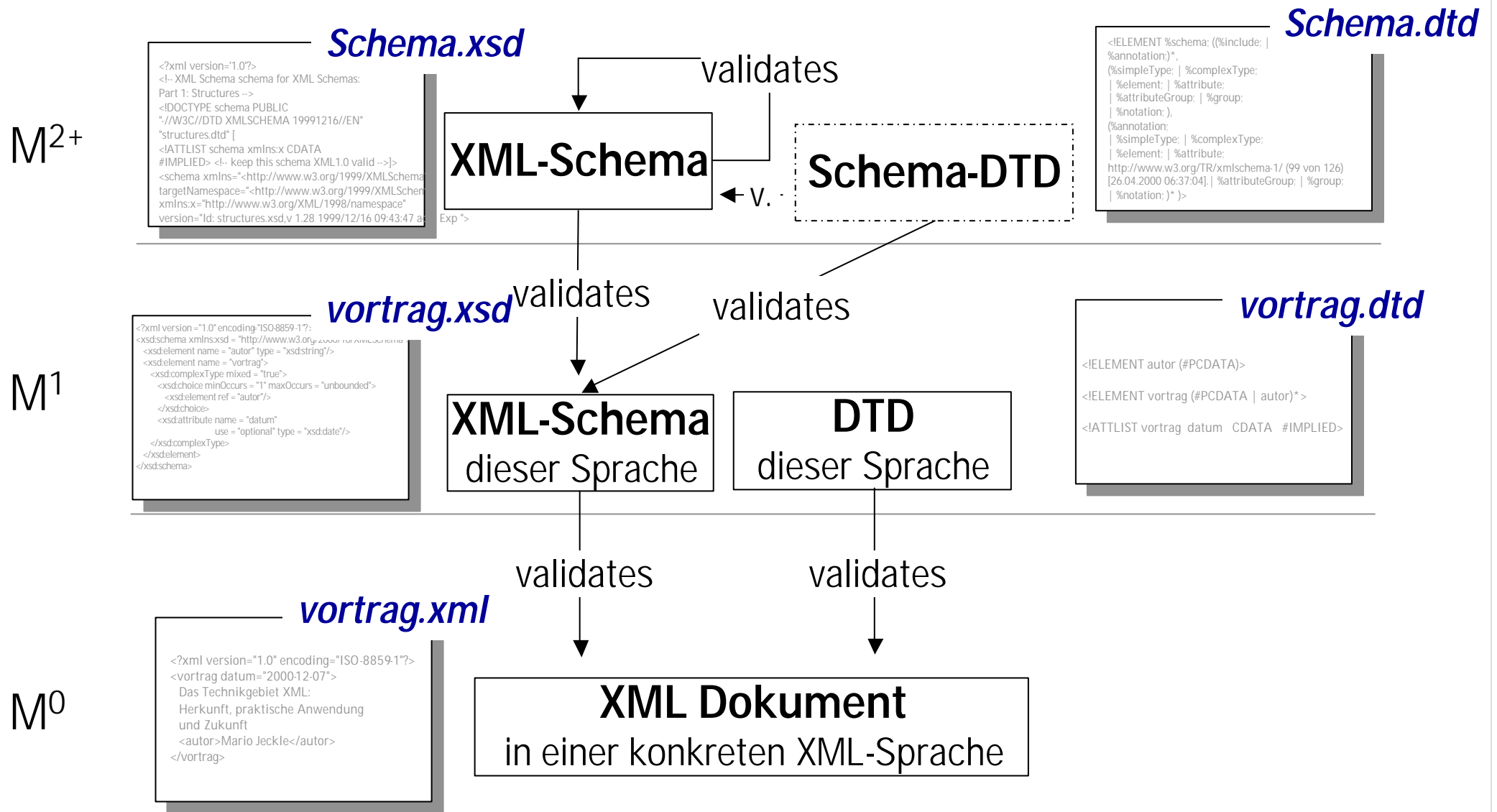
```
<xsd:attribute name = "datum" use = "optional" type = "xsd:date"/>
```

```
</xsd:complexType>
```

```
</xsd:element>
```

```
</xsd:schema>
```

# W3C's XML Schema Technologie -- Metamodellierung



# Entwurf von XML Sprachen – XML Schema

## Schema Part 2 – Datentypen – *build in datatypes*

- ID
- IDREF
- IDREFS
- ENTITY
- ENTITIES
- NMTOKEN
- NMTOKENS
- Name
- QName
- NCName
- NOTATION
- integer
- nonPositiveInteger
- negativeInteger
- long
- int
- short
- byte
- nonNegativeInteger
- unsignedLong
- unsignedInt
- unsignedShort
- unsignedByte
- positiveInteger
- boolean
- float
- double
- decimal
- string
- date
- time
- timeInstant
- timePeriod
- month
- year
- century
- recurringDate
- recurringDay
- timeDuration
- recurringDuration
- binary
- uriReference
- language

# Entwurf von XML Sprachen – XML Schema

## Schema Part 2 – Datentypen – *build in datatypes*

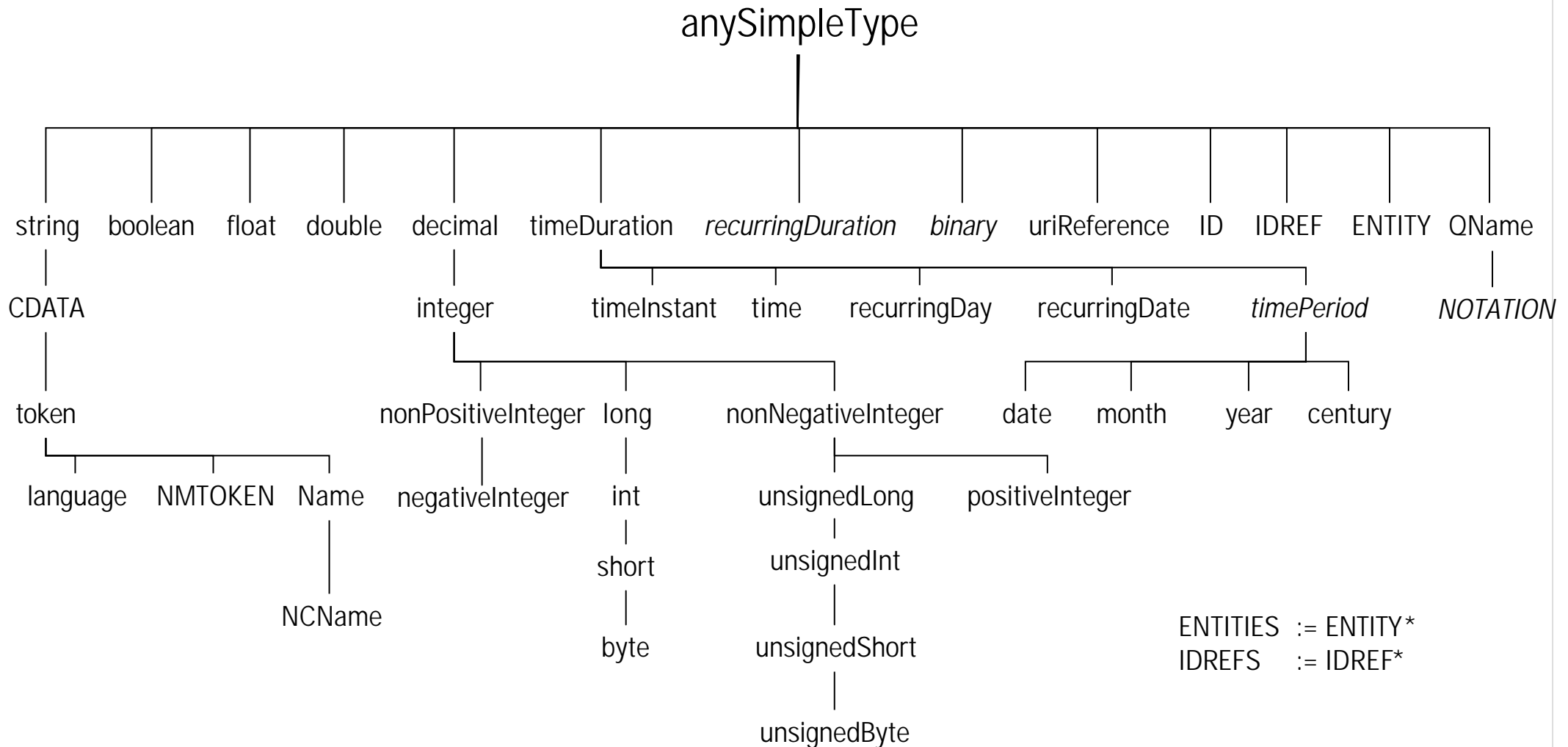
- integer -1, 0, 7683, +555
- nonPositiveInteger {..., -2, -1, 0}
- negativeInteger {..., -3, -2, -1}
- long  $-2^{63} \leq \text{long} \leq 2^{63}-1$
- int  $-2^{31} \leq \text{int} \leq 2^{31}-1$
- short  $-2^{15} \leq \text{short} \leq 2^{15}-1$
- byte  $-2^7 \leq \text{byte} \leq 2^7-1$
- nonNegativeInteger {0, 1, 2, ...}
- positiveInteger {1, 2, 3, ...}
- unsignedLong  $0 \leq \text{unsignedLong} \leq 2^{64}-1$
- unsignedInt  $0 \leq \text{unsignedInt} \leq 2^{32}-1$
- unsignedShort  $0 \leq \text{unsignedShort} \leq 2^{16}-1$
- unsignedByte  $0 \leq \text{unsignedByte} \leq 2^8-1$
- boolean {true, 1, false, 0}
- float 32-Bit Fließkommazahl gemäß IEEE 754-1985 -1E4, 12.64E8, 12e-2, INF
- double 64-Bit Fließkommazahl gemäß IEEE 754-1985
- decimal -12, 8, 3.14151592, +1.0

## Entwurf von XML Sprachen – XML Schema

### Schema Part 2 – Datentypen – *build in datatypes*

• string	ISO 10646 und Unicode	"hello world"
• date	ISO 8601	2000-10-00
• time	ISO 8601	17:00:00+1:00
• timeInstant	ISO 8601	2000-12-07T17:00:00+1:00
• timePeriod	ISO 8601	P7M
• month	ISO 8601	P1M
• year	ISO 8601	P1Y
• century	ISO 8601	19
• recurringDate	ISO 8601	PT24H
• recurringDay	ISO 8601	P1M2D
• timeDuration	ISO 8601	POY0M0DT1H30M
• recurringDuration	<i>abstrakter Typ; nur verwendbar durch Ableitung und Weiterspezifikation</i>	
• binary	<i>abstrakter Typ; nur verwandbar durch Ableitung und encoding-Spezifikation</i>	
• uriReference	IETF RFC2396	<a href="http://www.jeckle.de">http://www.jeckle.de</a>
• language	IETF RFC1766	de-de, en-uk, x-klingon

# Schema Part 2: Datatypes



## Schema Part 2: Datatypes

Ausschnitt aus XML Schema for Schemas (CR 2000-10-24):

*First the buildin primitive datatypes. These definitions are for information only, the real buildin definitions are magic. Note in particular there is no type named 'anySimpleType'.*

*The primitives should really be derived from no type at all, and anySimpleType should be derived as a union of all the primives.*

```
<xsd:simpleType name="decimal">
  <restriction base="anySimpleType" />
</xsd:simpleType>
<xsd:simpleType name="integer">
  <restriction base="decimal" />
</xsd:simpleType>
<xsd:simpleType name="nonPositiveInteger">
  <restriction base="integer" />
</xsd:simpleType>
```

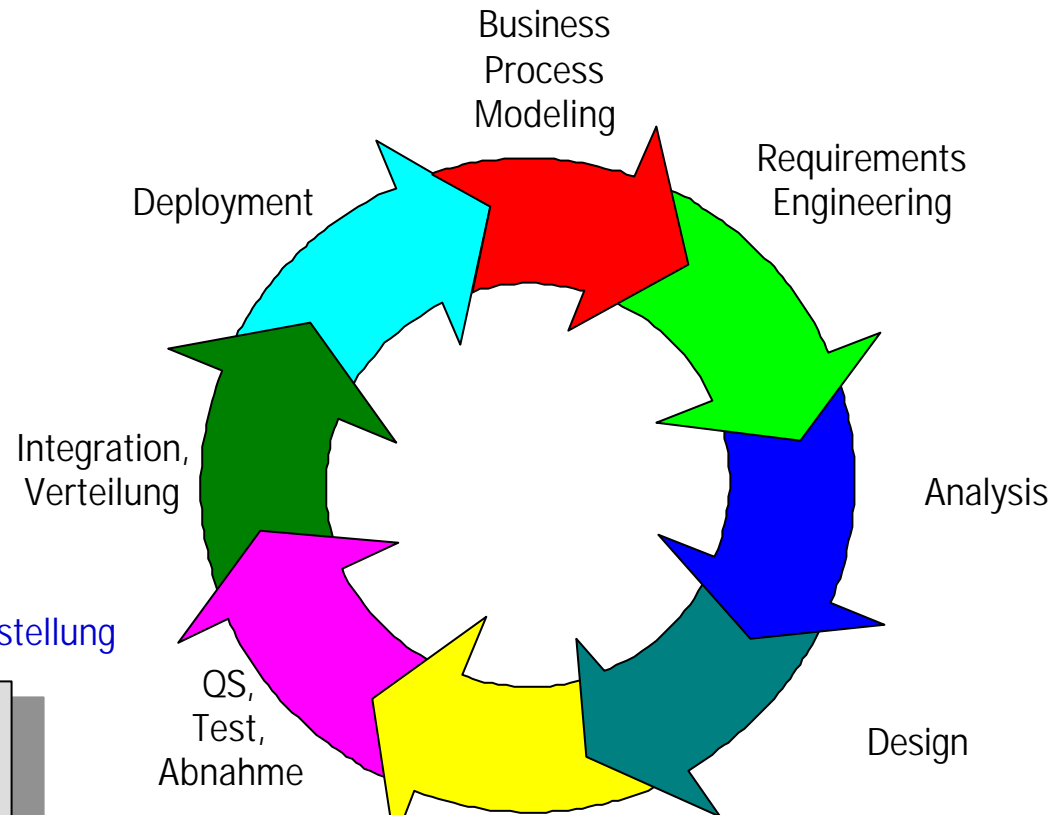
## Schema resumé

- Status: *candidate recommendation* (CR 2000-10-24)  
Letzter Status vor der Verabschiedung als *recommendation*
- Der zukünftige Standard zur XML-Sprachdefinition
- Werkzeugunterstützung verfügbar
  - Schema Editoren
  - Parser
- Umstellung von DTD-Repositories auf Schema zu beobachten
- Sprach*generierung* statt (manueller) Sprach*erstellung*
- XML-Schema ist keine Datenmodellierungssprache
- XML-Schema ist eine XML-Sprache
- Jedes Schema ist ein (schema) valid XML-Dokument

# XML-Sprachgewinnung

informale Formulierung

Ein Überweisungsvorgang setzt sich aus der Bonitätsprüfung des Auftraggebers vor Abbuchung vom Auftraggeberkonto, und Gutschrift auf dem Empfängerkonto zusammen.



formale Darstellung

```
context Person inv:
self.wife->notEmpty implies self.wife.
age >= 18 and
self.husband->notEmpty implies
self.husband.age >= 18
context Company inv:
self.employee->size <= 50
```

Programmcode

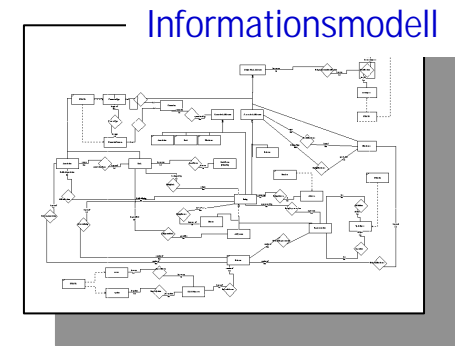
```
import java.io.FileInputStream;
import java.io.IOException;

public class TischRechner
{
    private FileInputStream fis = new FileInputStream("fileDescr.txt");
    private byte nextOperand; //0+, 1-, 2*, 3-/

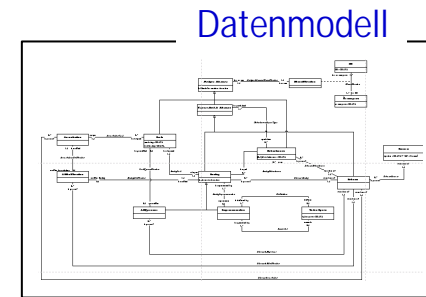
    private char getCharacterFromStdInput()
    {
        char readCharacter='0';
        try
        {
            readCharacter = (char) fis.read();
        }
        catch (Exception e)
        {
            // ...
        }
    }
}
```

Coding/  
Customizing

Analysis

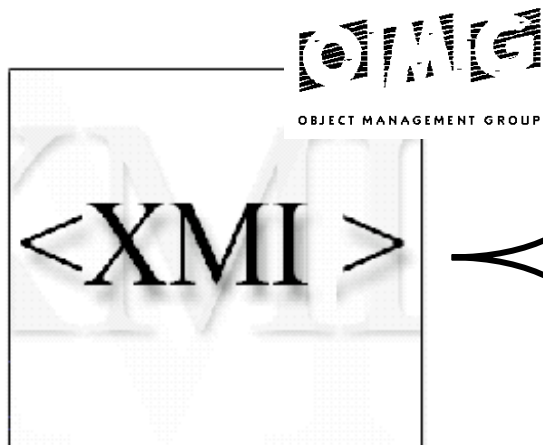


Design



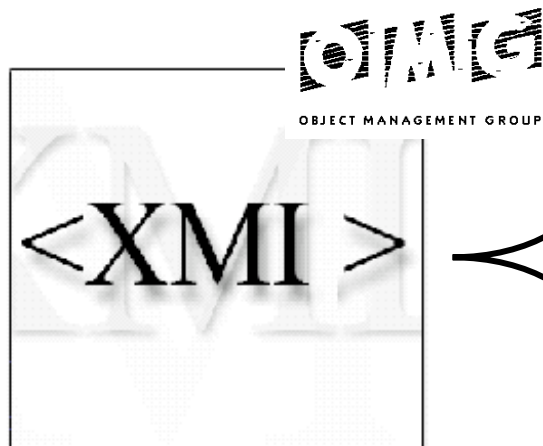
## XML-Sprachgewinnung

### Transfer von...



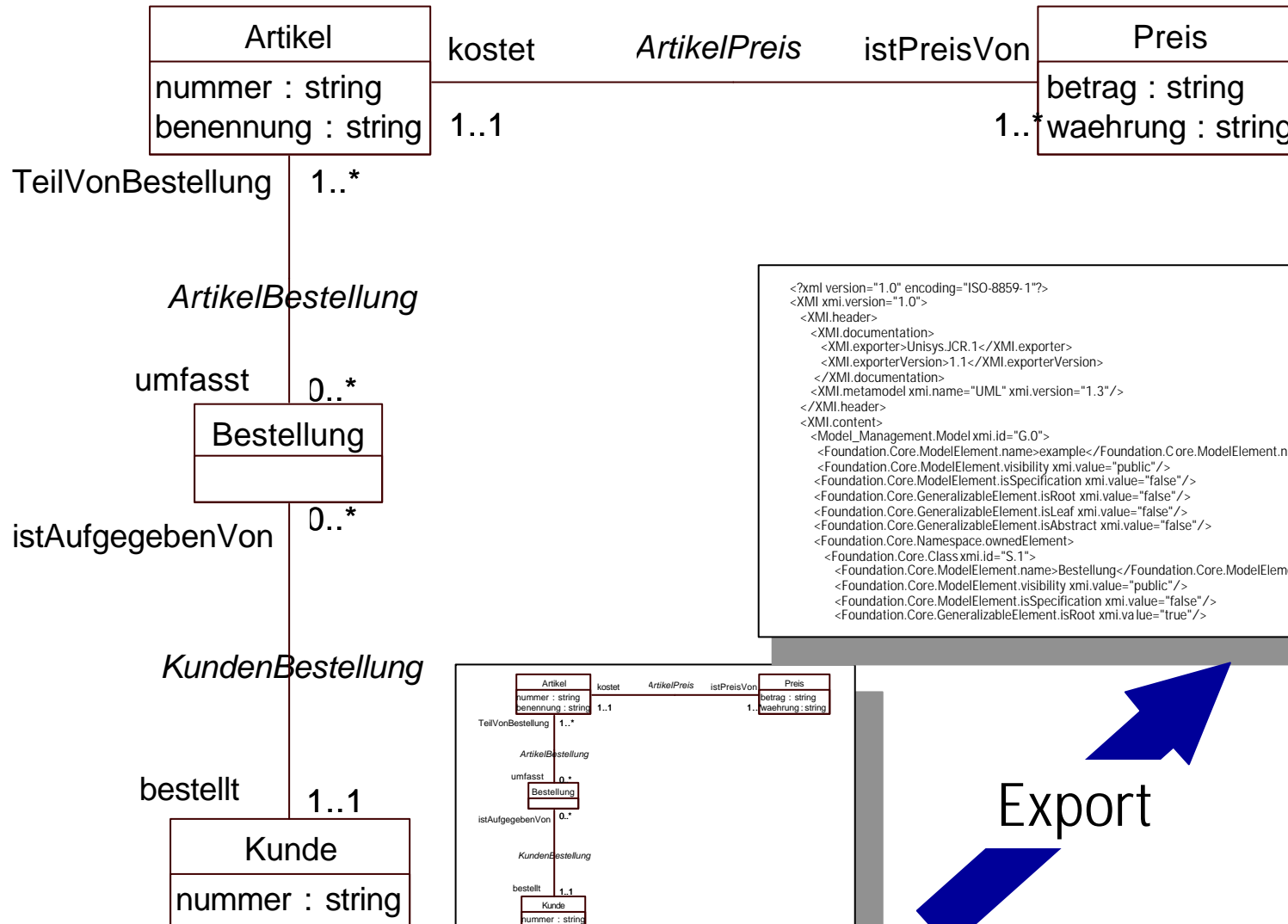
- Meta-Metamodell-Ausprägungen  
(vollständige Modellierungssprachen)
- Metamodell-Ausprägungen  
(vollständige Datenmodelle)
- Modell-Ausprägungen  
(vollständige Datenmodell-Ausprägungen/  
Instanzen)

## XML-Sprachgewinnung



- XML-DTD für MOF-Modelle  
XMI[MOF]
- XML-DTD für UML-Modelle  
XMI[UML]
- *Generation principles:*  
Beschreibung der Herleitung von  
XML-DTDs aus UML-basierten  
Modellen

# XML-Sprachgewinnung -- Beispiel

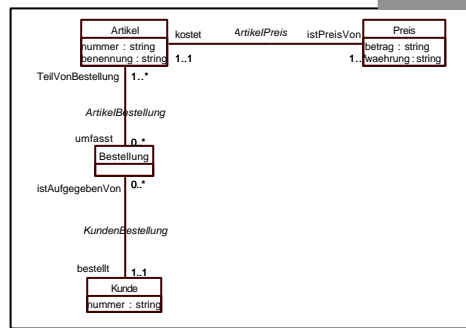


```

<ELEMENT example (( Bestellung | Artikel | Preis | Kunde )*) >
<IATTLIST model
  %XML.element.att;
  %XML.link.att;>
<ELEMENT Bestellung ( XML.extension* , Bestellung.bestellt?,
  estellung.TeilVonBestellung*)? >
<IATTLIST Bestellung
  %XML.element.att;
  %XML.link.att;>
<ELEMENT Artikel ( Artikel.benennung?, Artikel.nummer?, XML.extension* ,
  Artikel.istPreisVon* , Artikel.umfasst*)? >
<IATTLIST Artikel
  %XML.element.att;
  %XML.link.att;>
    
```

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<XML xml.version="1.0">
<XML.header>
  <XML.documentation>
    <XML.exporter>Unisys.JCR.1</XML.exporter>
    <XML.exporterVersion>1.1</XML.exporterVersion>
  </XML.documentation>
  <XML.metamodel xml.name="UML" xml.version="1.3"/>
</XML.header>
<XML.content>
  <Model_Management.Model xml.id="G.0">
    <Foundation.Core.ModelElement.name>example</Foundation.Core.ModelElement.name>
    <Foundation.Core.ModelElement.visibility xml.value="public"/>
    <Foundation.Core.ModelElement.isSpecification xml.value="false"/>
    <Foundation.Core.GeneralizableElement.isRoot xml.value="false"/>
    <Foundation.Core.GeneralizableElement.isLeaf xml.value="false"/>
    <Foundation.Core.GeneralizableElement.isAbstract xml.value="false"/>
    <Foundation.Core.Namespace.ownedElement>
      <Foundation.Core.Class xml.id="S.1">
        <Foundation.Core.ModelElement.name>Bestellung</Foundation.Core.ModelElement.name>
        <Foundation.Core.ModelElement.visibility xml.value="public"/>
        <Foundation.Core.ModelElement.isSpecification xml.value="false"/>
        <Foundation.Core.GeneralizableElement.isRoot xml.value="true"/>
    
```



Generierung

Export

## Sprachtransformation: XSLT

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bestellung>
  <artikel>
    <nummer>4711</nummer>
    <benennung>Wusch Superfein</benennung>
    <preis>
      <betrag>195,99</betrag>
      <waehrung>DEM</waehrung>
    </preis>
    <kunde nummer="X-363-23"/>
  </artikel>
</bestellung>
```

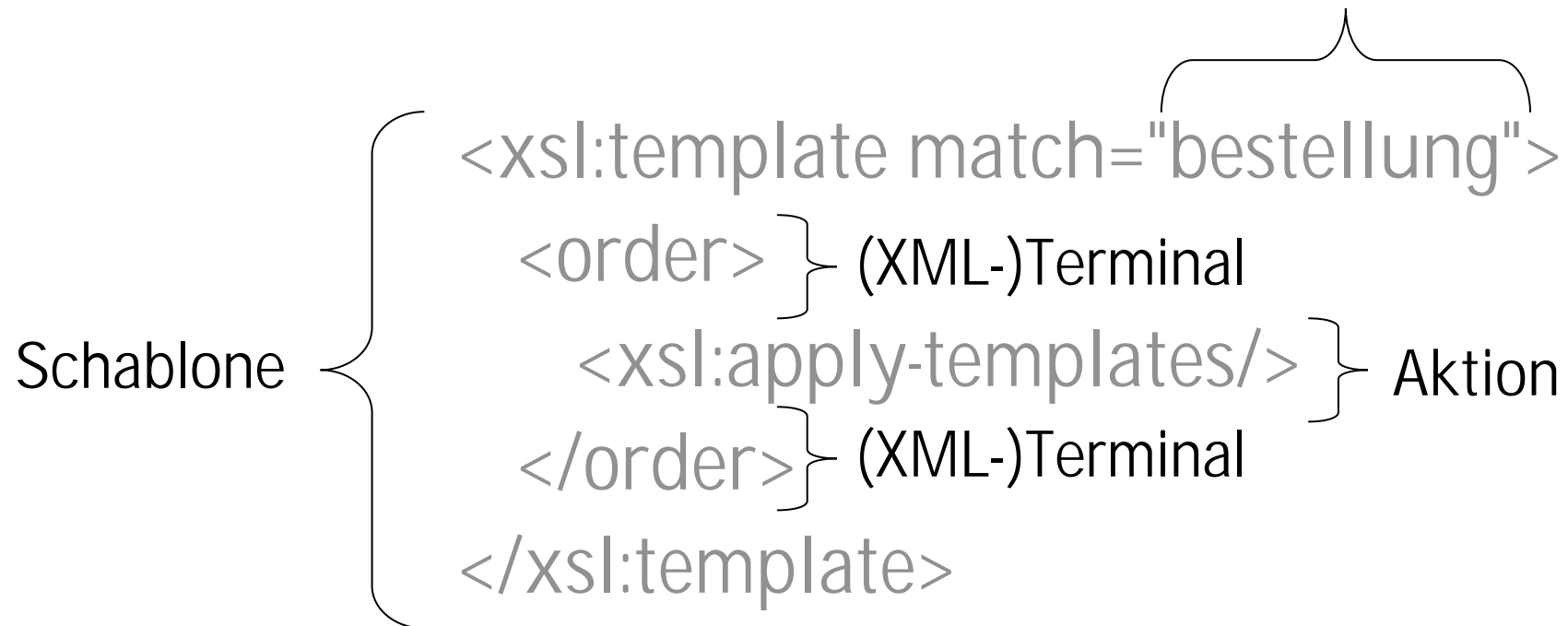
```
<?xml version="1.0" encoding="UTF-8"?>
<order>
  <customer>
    <custID>X-363-23</custID>
  </customer>
  <itemlist>
    <item>
      <itemNO>4711</itemNO>
      <identification>Wusch Superfein</identification>
      <price currency="DEM">195,99</price>
    </item>
  </itemlist>
</order>
```



Transformation

## Sprachtransformation: XSLT

(Such-)Muster als XPath-Ausdruck



# Sprachtransformation: XSLT

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bestellung>
  <artikel>
    <nummer>4711</nummer>
    <benennung>Wusch Superfein</benennung>
    <preis>
      <betrag>195,99</betrag>
      <waehrung>DEM</waehrung>
    </preis>
    <kunde nummer="X-363-23"/>
  </artikel>
</bestellung>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" extension-element-prefixes="">
  <xsl:output method="xml" encoding="ISO-8859-1" omit-xml-declaration="no" indent="yes" />

  <xsl:template match="bestellung">
    <order>
      <xsl:apply-templates select="//kunde"/>
    </order>
  </xsl:template>

  <xsl:template match="kunde">
    <customer>
      <custID><xsl:value-of select="@nummer"/></custID>
    </customer>

    <itemlist>
      <xsl:apply-templates select="//artikel"/>
    </itemlist>
  </xsl:template>

  <xsl:template match="artikel">
    <item>
      <itemNo><xsl:value-of select="nummer"/></itemNo>
      <identification><xsl:value-of select="benennung"/></identification>
      <price currency="{/preis/waehrung}">
        <xsl:value-of select="/preis/betrag"/>
      </price>
    </item>
  </xsl:template>
</xsl:transform>
```

XML-Eingangsdatei

XSLT-Prozessor

XSLT-Transformationsheet  
(XML-Datei)

```
<?xml version="1.0" encoding="UTF-8"?>
<order>
  <customer>
    <custID>X-363-23</custID>
  </customer>
  <itemlist>
    <item>
      <itemNO>4711</itemNO>
      <identification>Wusch Superfein</identification>
      <price currency="DEM">195,99</price>
    </item>
  </itemlist>
</order>
```

# Sprachtransformation: XSLT

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" extension-element-prefix="">
  <xsl:output method="xml" encoding="ISO-8859-1" omit-xml-declaration="no" indent="1" />
  <xsl:template match="bestellung">
    <order>
      <xsl:apply-templates select="//kunde"/>
    </order>
  </xsl:template>
  <xsl:template match="kunde">
    <customer>
      <custID><xsl:value-of select="@nummer"/></custID>
    </customer>
    <itemlist>
      <xsl:apply-templates select="//artikel"/>
    </itemlist>
  </xsl:template>

```

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<bestellung>
  <artikel>
    <nummer>4711</nummer>
    <benennung>Wusch Superfein</benennung>
    <preis>
      <betrag>195,99</betrag>
      <waehrung>DEM</waehrung>
    </preis>
    <kunde nummer="X-363-23"/>
  </artikel>
</bestellung>

```

# Sprachtransformation: XSLT

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bestellung>
  <artikel>
    <nummer>4711</nummer>
    <benennung>Wusch Superfein</benennung>
    <preis>
      <betrag>195,99</betrag>
      <waehrung>DEM</waehrung>
    </preis>
    <kunde nummer="X-363-23"/>
  </artikel>
</bestellung>
```

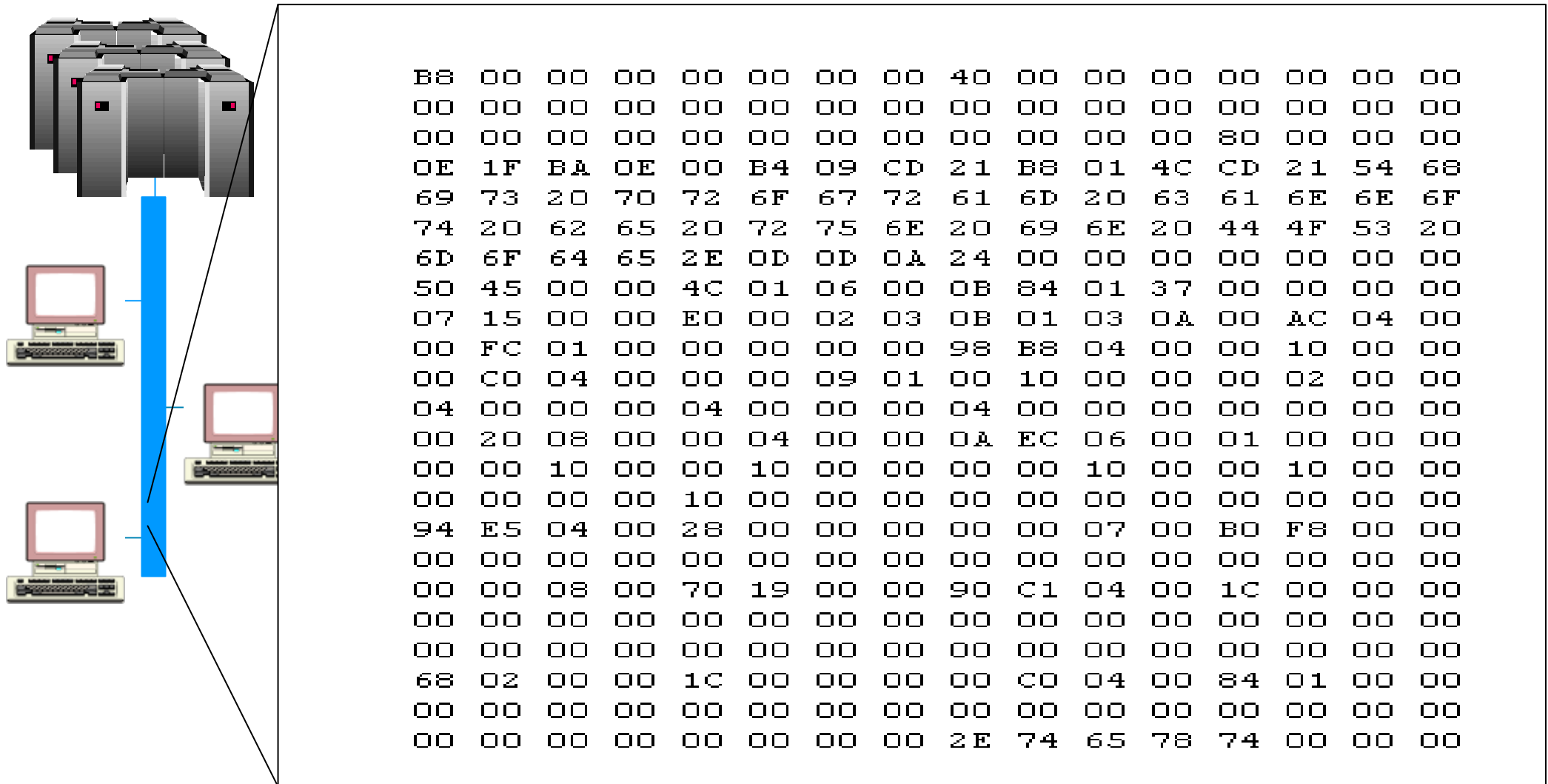
```
<xsl:template match="artikel">
  <item>
    <itemNo><xsl:value-of select="nummer"/></itemNo>
    <identification><xsl:value-of select="benennung"/></identification>
    <price currency="{./preis/waehrung}">
      <xsl:value-of select="./preis/betrag"/>
    </price>
  </item>
</xsl:template>

</xsl:transform>
```

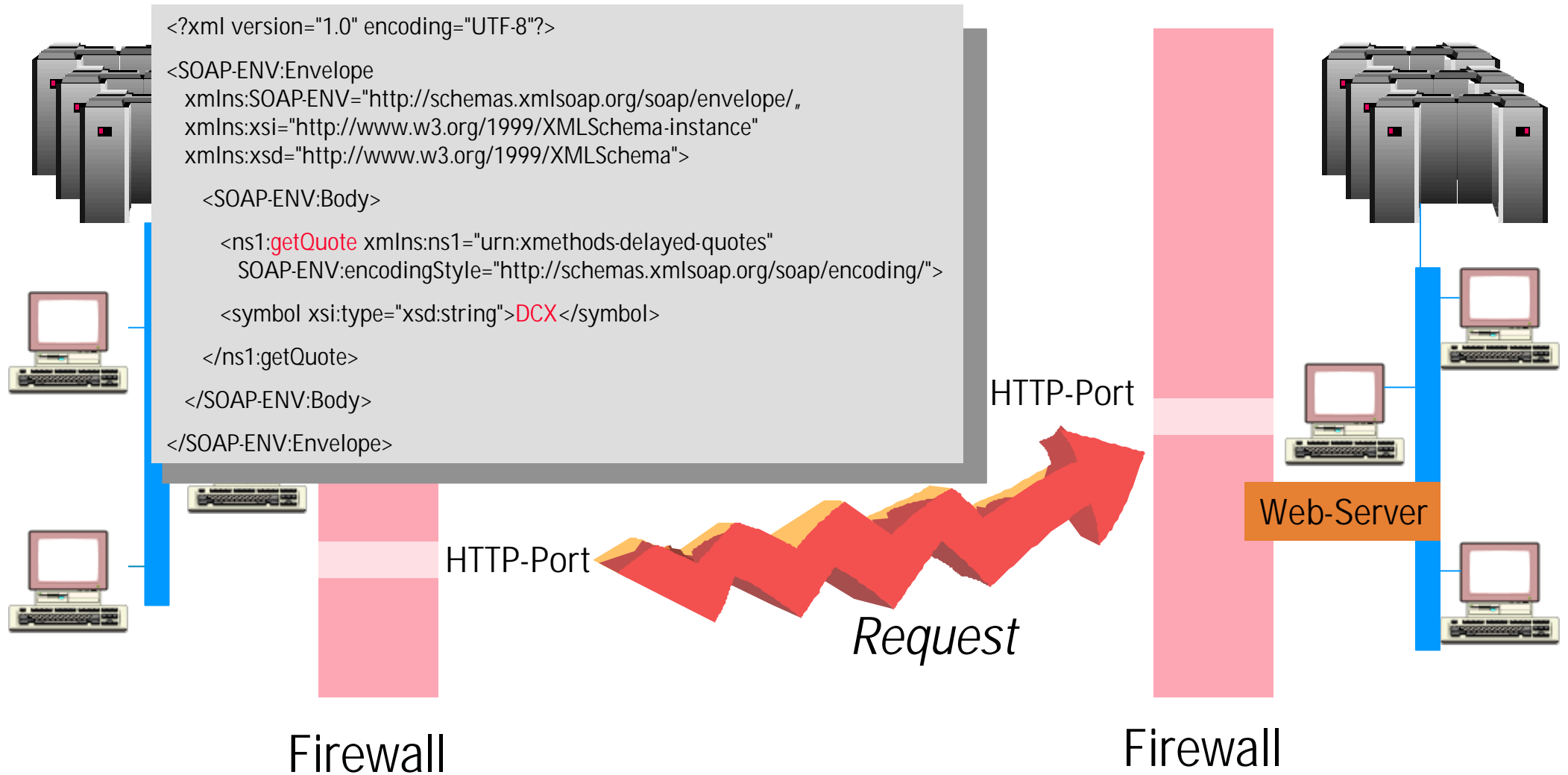
## Systemkommunikation: XML Protocols (XP)

- Einfaches, leichtgewichtiges (*light weight*) Protokoll zum strukturierten und typisierten Informationsaustausch über das Web
- Absolut minimalistischer Ansatz
  - => geringer Implementierungsaufwand
  - => Minimalfunktionalität
- Modular und Erweiterbar
  - Keine Applikations- und Transportsemantik
- Technisch: viergeteilt
  - *Envelope*: Umschlag zur Verpackung der Nutzinformation
  - *Encoding rules*: XML und XML-Schema
  - *RPC-Semantik*: entfernter Funktionsaufruf und Antwort
  - Protokollbindung

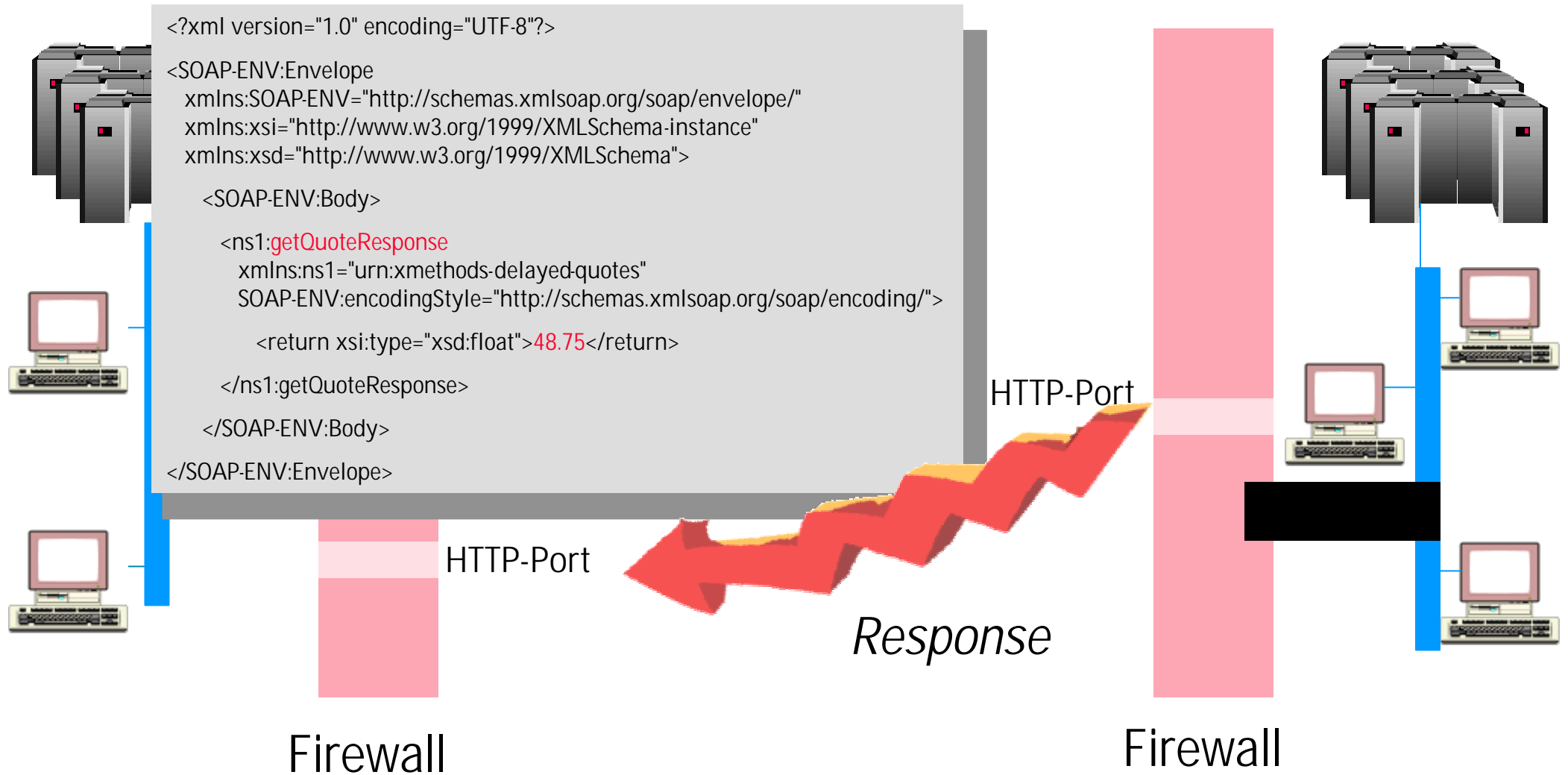
# Systemkommunikation: XP



# Systemkommunikation: XP



# Systemkommunikation: XP



# Systemkommunikation: XP Interoperabilitätsebenen



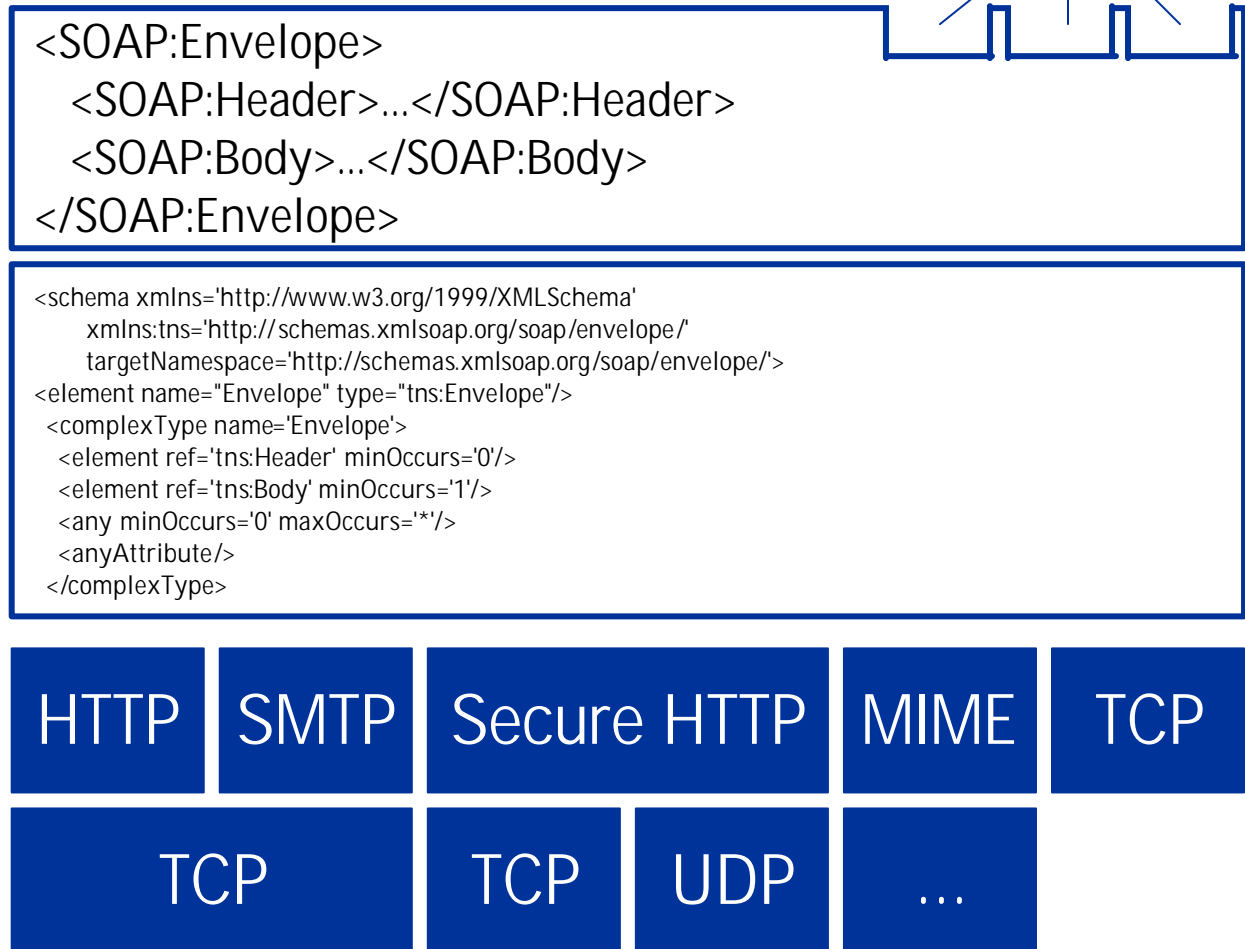
*XP Modules*

Services

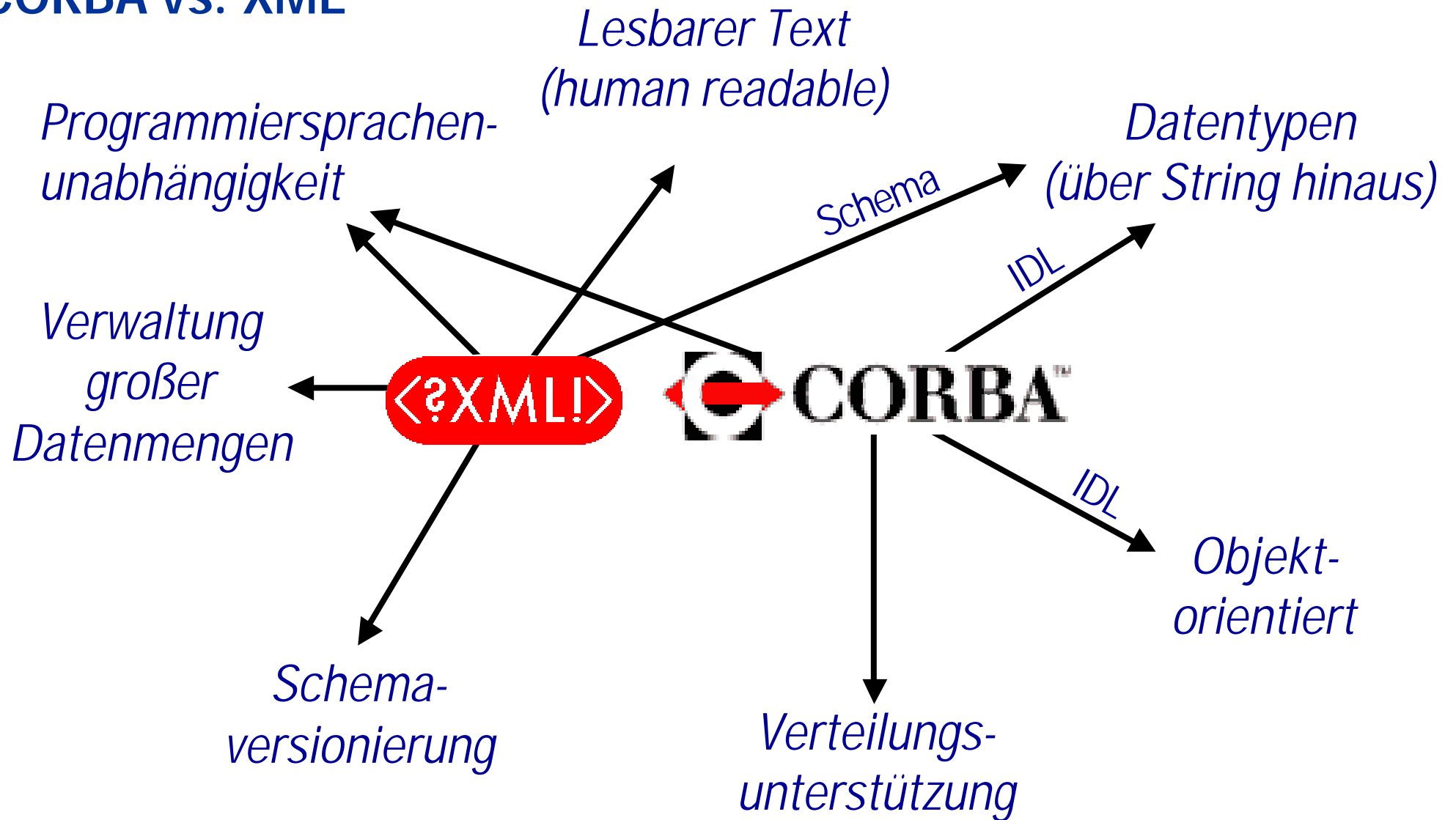
SOAP / XML Protocols

XML  
(v1.0 2<sup>nd</sup> editon  
+ XML Schema)

*Common Internet Protocols*



# CORBA vs. XML



## Referenzen

### XML Einführendes und Vertiefendes

- [www.w3.org](http://www.w3.org)
- [www.xml.com](http://www.xml.com)
- [www.xmlhack.com](http://www.xmlhack.com)
- [www.oasis-open.org/cover](http://www.oasis-open.org/cover)
- [www.mintert.com/xml/trans](http://www.mintert.com/xml/trans) (deutsche Übersetzung der XML Recommendation)
- [www.ibiblio.org/xml/index.html](http://www.ibiblio.org/xml/index.html) (Cafe con Leche)
- [www.simonstl.com](http://www.simonstl.com) (Homepage von Simon St. Laurent)
- [cm.bell-labs.com/cm/cs/who/wadler/xml](http://cm.bell-labs.com/cm/cs/who/wadler/xml) (Homepage von Phil Wadler)
- [www.cetus-links.org/oo\\_xml.html](http://www.cetus-links.org/oo_xml.html)

## W3C's XML-Schema

- [www.w3c.org/TR/xml-schema-0](http://www.w3c.org/TR/xml-schema-0)
- [www.w3c.org/TR/xml-schema-1](http://www.w3c.org/TR/xml-schema-1)
- [www.w3c.org/TR/xml-schema-2](http://www.w3c.org/TR/xml-schema-2)

## XML Metadata Interchange

- [www.omg.org/xml](http://www.omg.org/xml)
- [www.xmiforum.org](http://www.xmiforum.org)
- [www.software.ibm.com/ad/features/xmi.html](http://www.software.ibm.com/ad/features/xmi.html)
- [www.alphaworks.ibm.com/tech/xmitoolkit](http://www.alphaworks.ibm.com/tech/xmitoolkit)

## XSLT

- [www.w3c.org/TR/xslt](http://www.w3c.org/TR/xslt)
- [www.xslt.com](http://www.xslt.com)

## SOAP

- [www.w3.org/TR/SOAP/](http://www.w3.org/TR/SOAP/)
- [www.w3.org/2000/03/29-XML-protocol-matrix](http://www.w3.org/2000/03/29-XML-protocol-matrix)
- [www.w3.org/2000/xp](http://www.w3.org/2000/xp)

Dieser Vortrag und weiterführende Hintergrundinformation  
[www.jeckle.de](http://www.jeckle.de)