

# DAIMLERCHRYSLER

## **XML und Java -- wohin geht die Reise?**

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

### I XML und Programmiersprachen ...

- Was ist die XML heute?
- Aktuelle Entwicklungen des XML-Umfeldes

### II Stand der XML-Technologie ...

- XML-Schema
- SOAP/XMLP
- Ausblick: *Semantic Web*

### III XML und Java ...

- ... eigentlich nichts Neues
- J2SE v1.4 *Merlin*
- *XMLPack*

## XML und Programmiersprachen ...

### Java ist ...

- Programmiersprache
- Laufzeitumgebung
- Enterprise Application Platform (J2EE)



+

### XML ist ...

- Informations- und Datenformat
- Metasprache zur Definition eigener Vokabulare



=

### ... keine:

- Programmiersprache
- Ausführungsumgebung

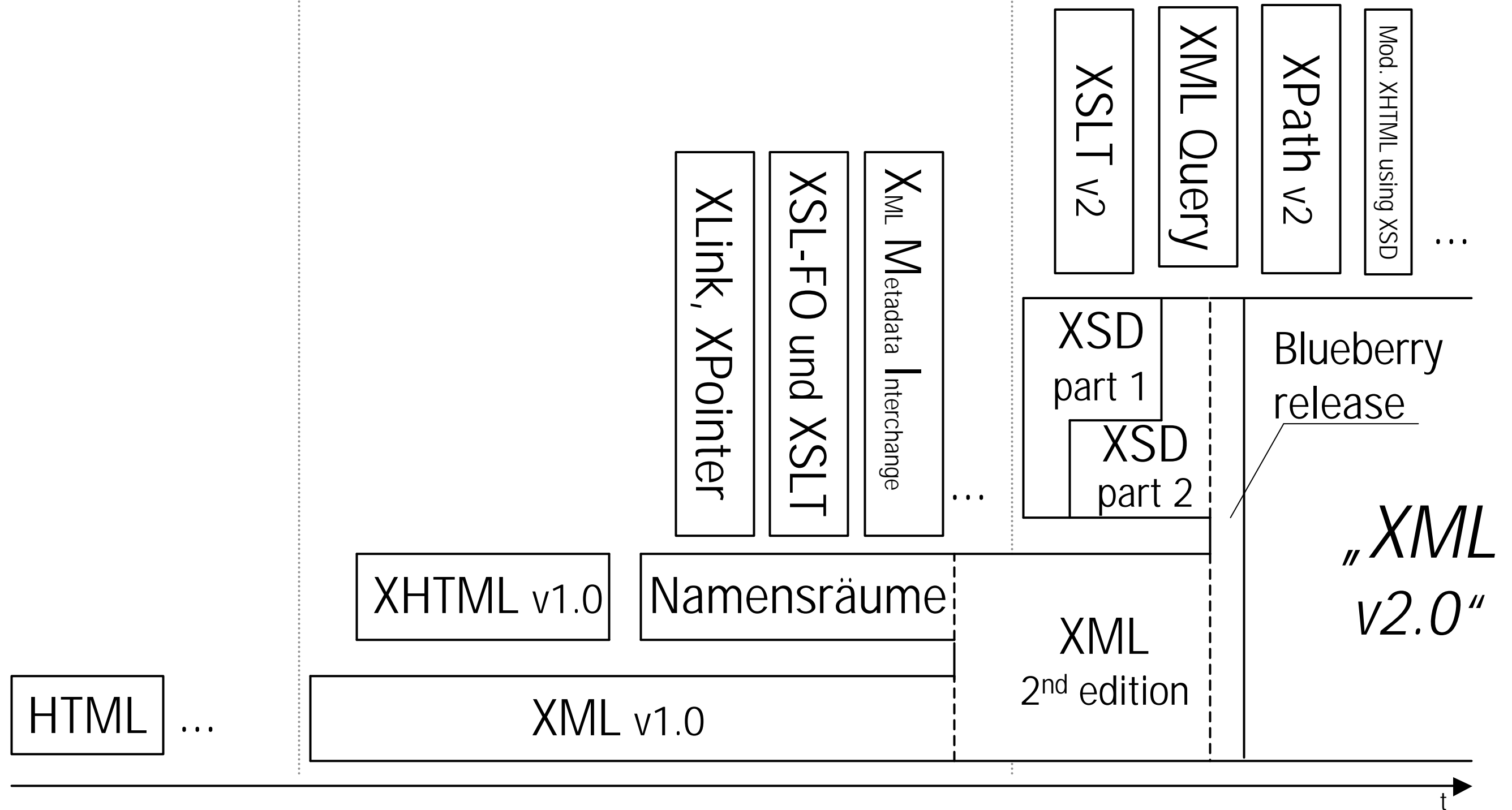


# Stand der XML-Technologie: Web Historie

Web-Prähistorie

1. Generation

2. Generation

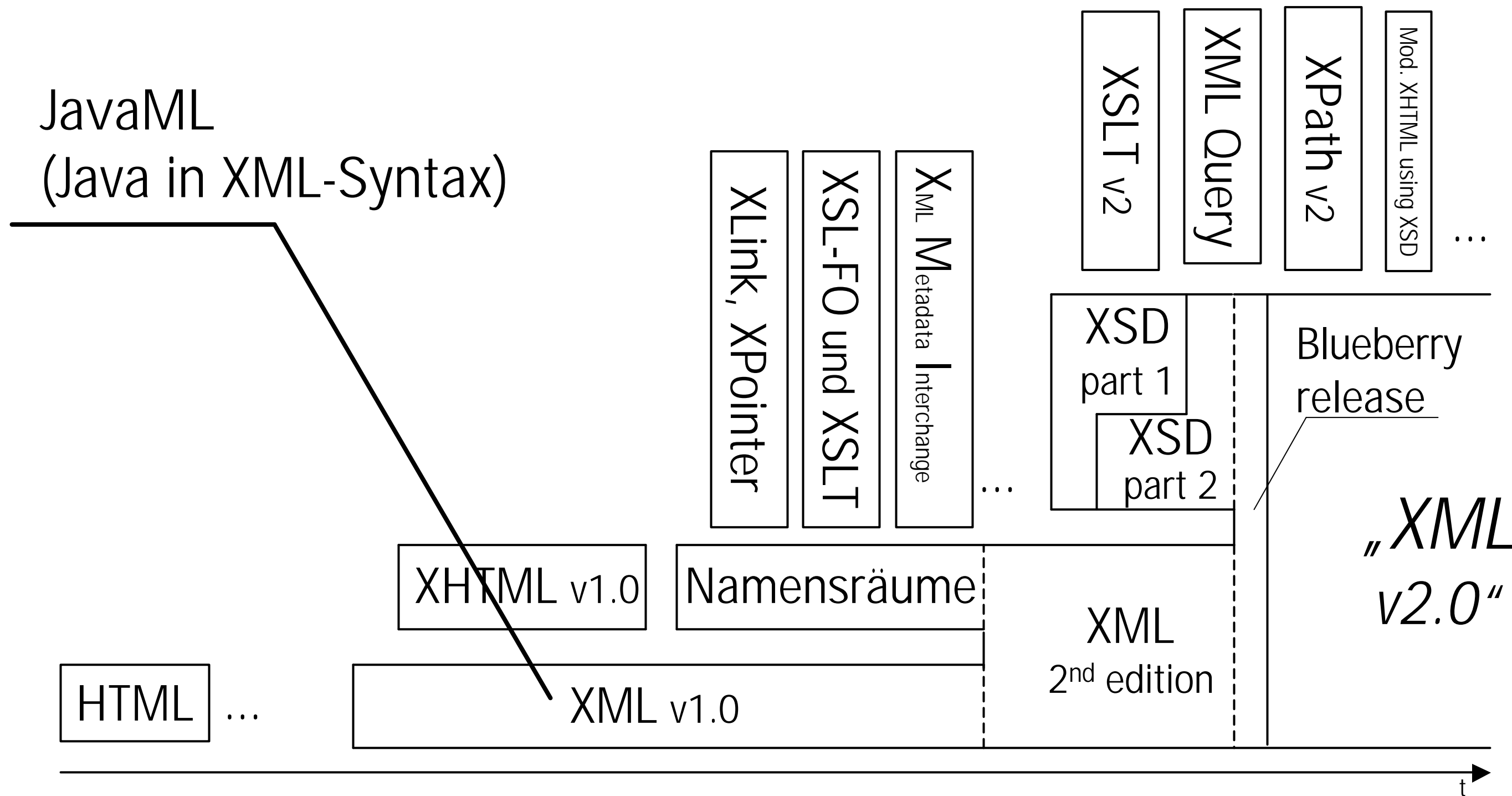


# Stand der XML-Technologie: *Programmiersprachliches an XML*

Web-Prähistorie

1. Generation

2. Generation



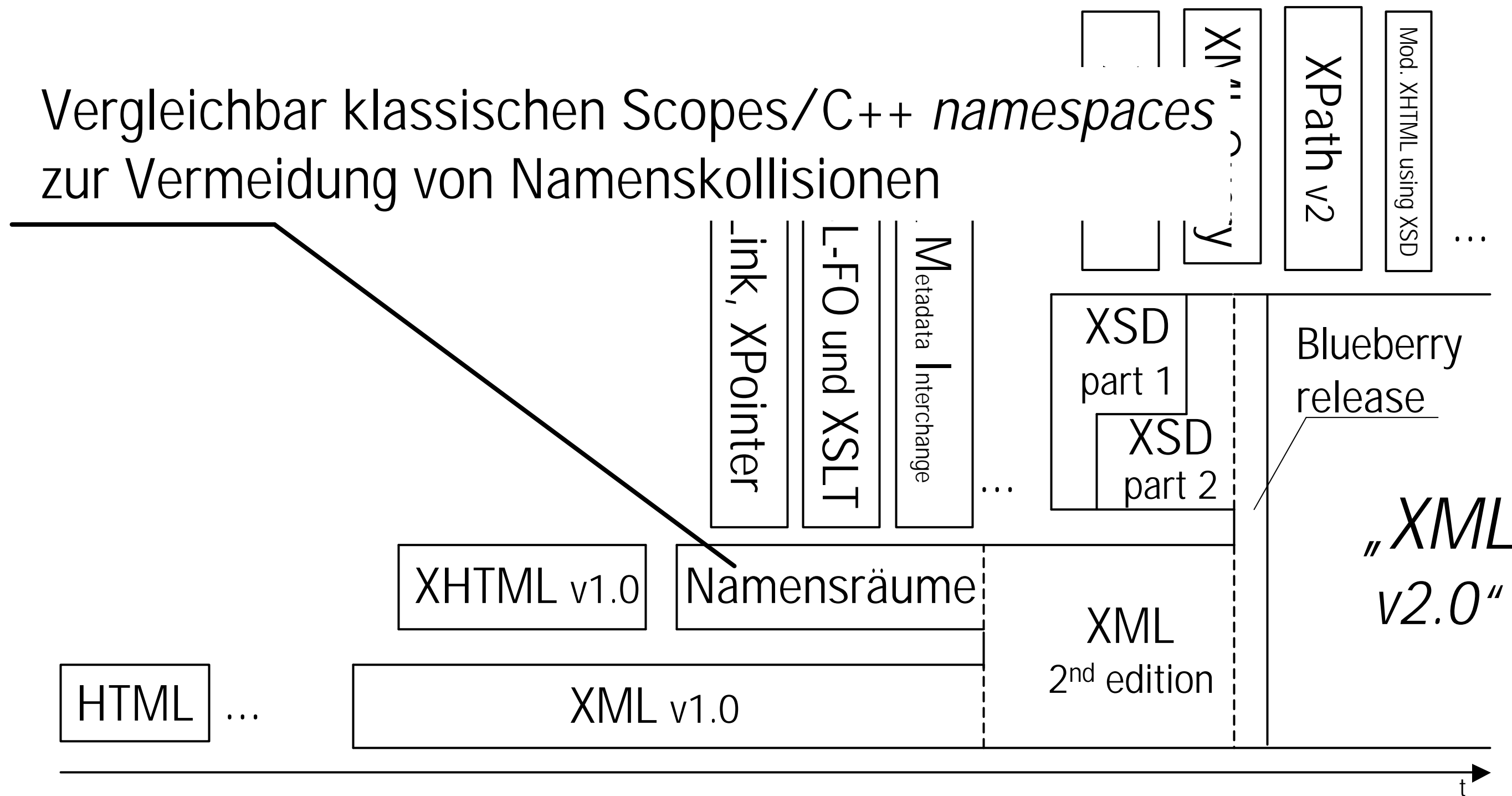
# Stand der XML-Technologie: *Programmiersprachliches an XML*

Web-Prähistorie

1. Generation

2. Generation

Vergleichbar klassischen Scopes/C++ *namespaces* zur Vermeidung von Namenskollisionen



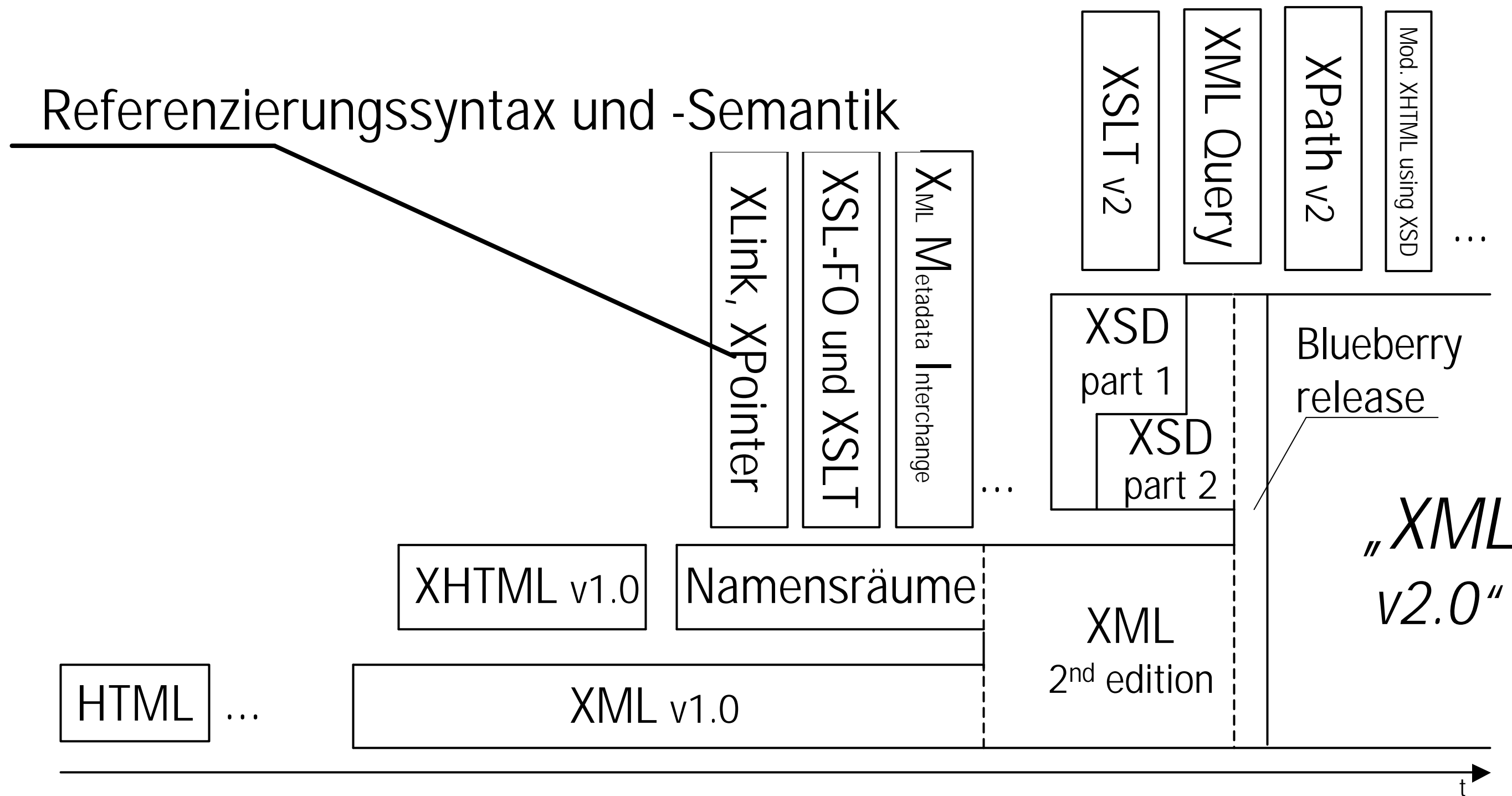
# Stand der XML-Technologie: *Programmiersprachliches an XML*

Web-Prähistorie

1. Generation

2. Generation

## Referenzierungssyntax und -Semantik



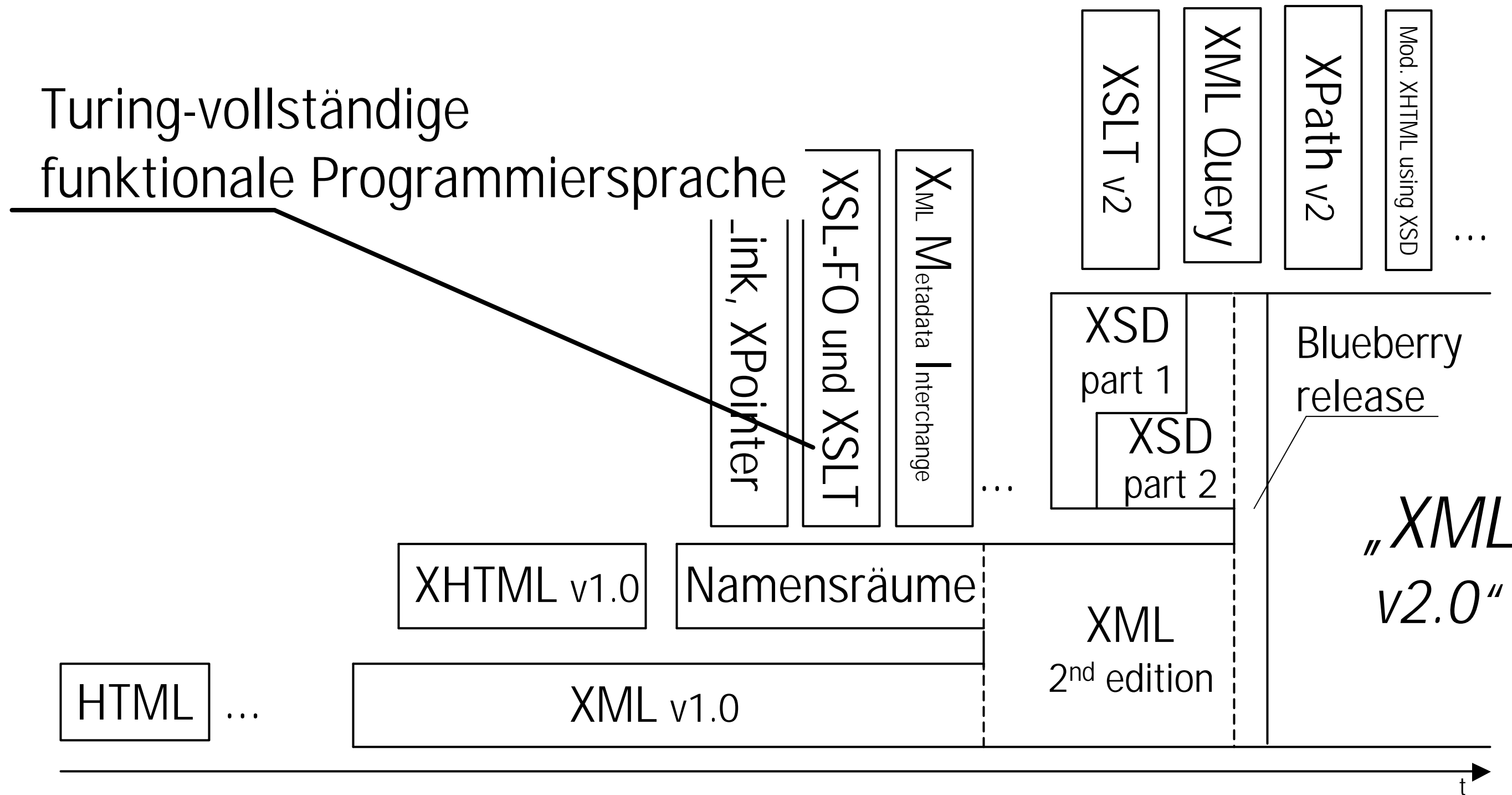
# Stand der XML-Technologie: *Programmiersprachliches an XML*

Web-Prähistorie

1. Generation

2. Generation

Turing-vollständige  
funktionale Programmiersprache



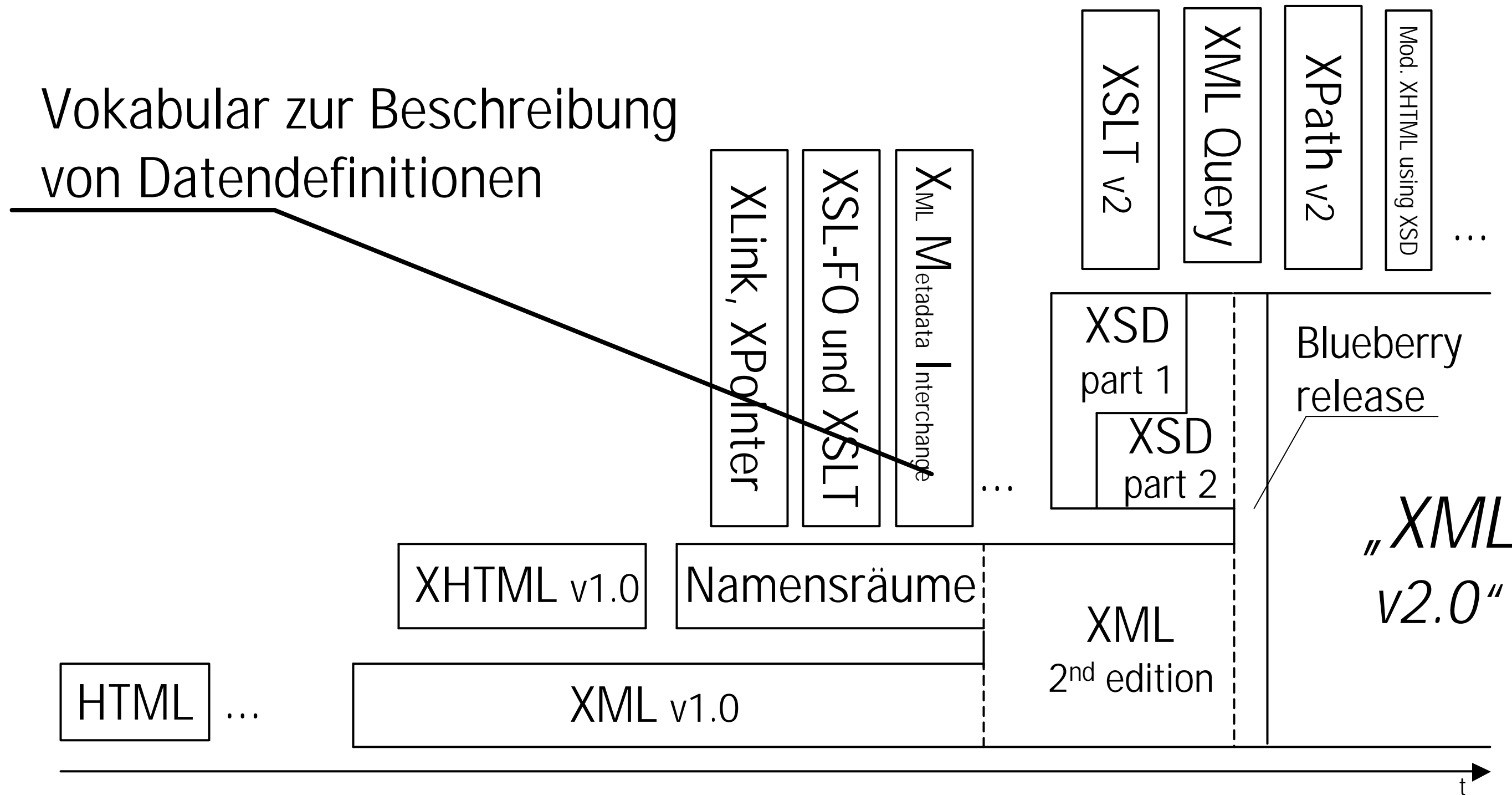
# Stand der XML-Technologie: *Programmiersprachliches an XML*

Web-Prähistorie

1. Generation

2. Generation

Vokabular zur Beschreibung von Datendefinitionen



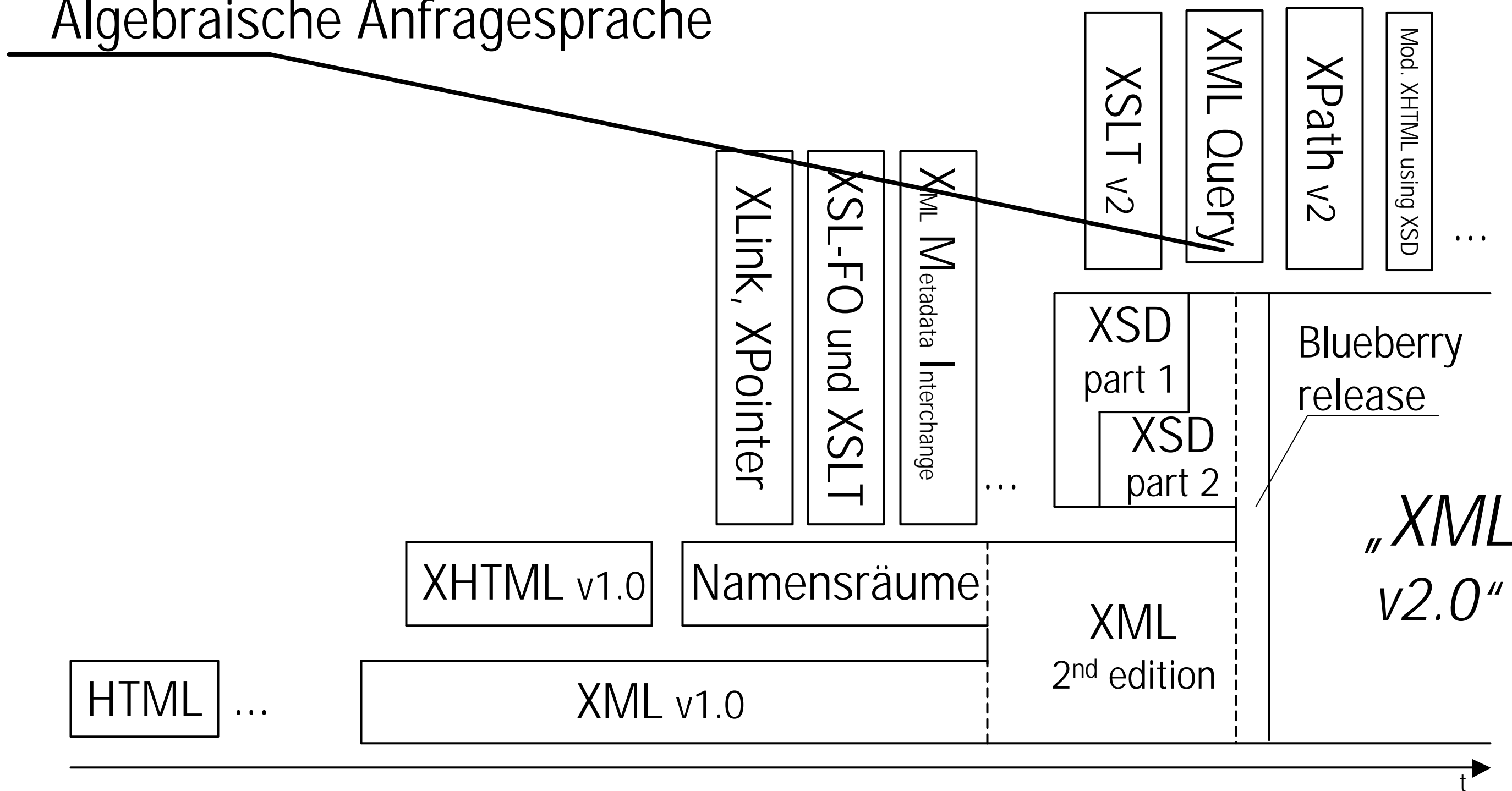
# Stand der XML-Technologie: *Programmiersprachliches an XML*

Web-Prähistoire

1. Generation

2. Generation

## Algebraische Anfragesprache



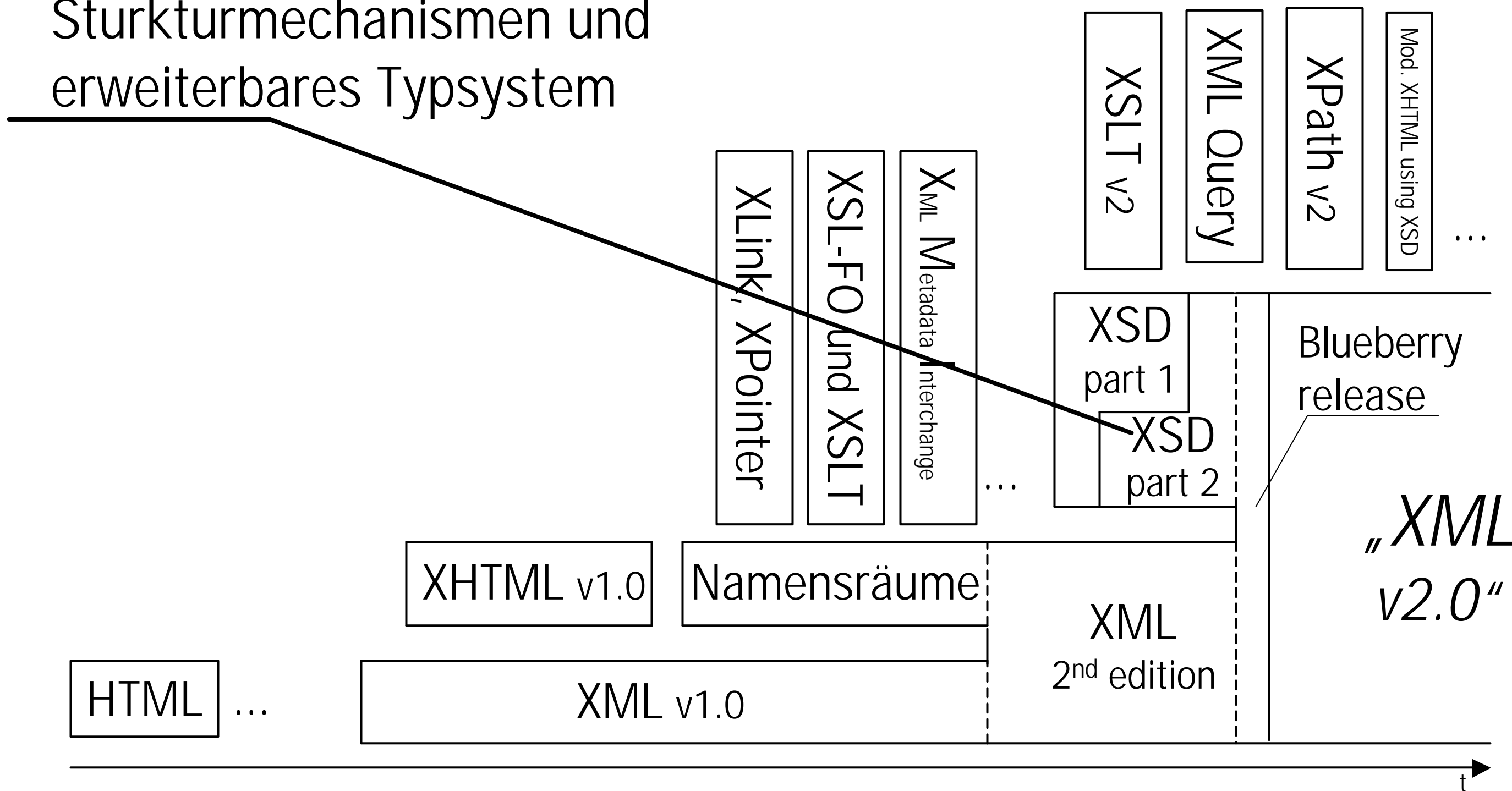
# Stand der XML-Technologie: *Programmiersprachliches an XML*

Web-Prähistoire

1. Generation

2. Generation

Strukturmechanismen und  
erweiterbares Typsystem



## XML-Schema ... in zehn Punkten

- Erlaubt die Erstellung regulärer kontextfreier Grammatiken für XML-Vokabulare
- Part 1 beschreibt Strukturen und Inhaltseinschränkungen
- Part 2 definiert Datentypdefinition für Schema Part 1 und weitere XML-Vokabulare
- Signifikante Erweiterung der DTD-Mächtigkeit, wird diese langfristig ersetzen
- Ist eine XML-Sprache
- Integriert die wichtigsten konkurrierenden Vorgängeransätze
- Seit 2001-05-02 W3C Recommendation
- Basis aller W3C-Standards der zweiten Generation (XPath v2.0, XSLT v2.0, XHTML v2.0, XMLP, ...)
- Werkzeugunterstützung verfügbar
- Erster Schritt der Schema-Bestrebungen, weitere werden folgen ...

## XML-Schema ... ein Beispieldokument

```
<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML und Java – wohin geht die Reise?</Titel>
  <Veranstaltung datum="2001-06-28">
    <Name>Java Forum Stuttgart 2001</Name>
  </Veranstaltung>
  <Referent>
    <Name>Mario Jeckle</Name>
    <Firma>DaimlerChrysler Research and Technology</Firma>
    <URL>http://www.jeckle.de</URL>
    <E-Mail>mario.jeckle@daimlerchrysler.com</E-Mail>
  </Referent>
</Vortrag>
```

## XML-Schema ... eine beschreibende Grammatik

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Vortrag">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Titel" type="xsd:string"/>
        <xsd:element ref="Veranstaltung"/>
        <xsd:element ref="Referent"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Veranstaltung">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Name"/>
      </xsd:sequence>
      <xsd:attribute ref="datum"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:attribute name="datum" type="xsd:date"/>

```

```
<xsd:element name="Name" type="xsd:string"/>
<xsd:element name="Referent">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Name"/>
      <xsd:element name="Firma" type="xsd:string"
        minOccurs="0"/>
      <xsd:element name="URL" type="xsd:anyURI"
        minOccurs="0"/>
      <xsd:element name="E-Mail" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:pattern value="\c{1,}@ \c{1,}.\c{2,3}"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

```
<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML und Java – wohin geht die Reise?</Titel>
  <Veranstaltung datum="2001-06-28">
    <Name>Java Forum Stuttgart 2001</Name>
  </Veranstaltung>
  <Referent>
    <Name>Mario Jeckle</Name>
    <Firma>DaimlerChrysler Research and Technology</Firma>
    <URL>http://www.jeckle.de</URL>
    <E-Mail>mario.jeckle@daimlerchrysler.com</E-Mail>
  </Referent>
</Vortrag>

```

## XML-Schema ... Mächtigkeit

- Strukturell: Attribute und Elemente (wie in DTDs)
- Namespace-Unterstützung
- Atomare Datentypen (int, float, boolean, ...)
- Anwenderdefinierte
  - atomare skalare Datentypen (simpleType)
    - Einschränkung des Wertebereichs (Domänenrestriktion)
  - lexikalische Muster (reguläre Ausdrücke)
  - Aufzählungstypen
  - Mengentypen
  - komplexe Datentypen (complexType)
- Wiederverwendungsunterstützung
  - Vererbung: Restriktion und Erweiterung
  - Schemafragmente: Typen, Attribut- und Elementgruppen
- Substitution
- Erweiterter Schlüsselmechanismus
- NULL-Werte



# XML Schema: Datentypen

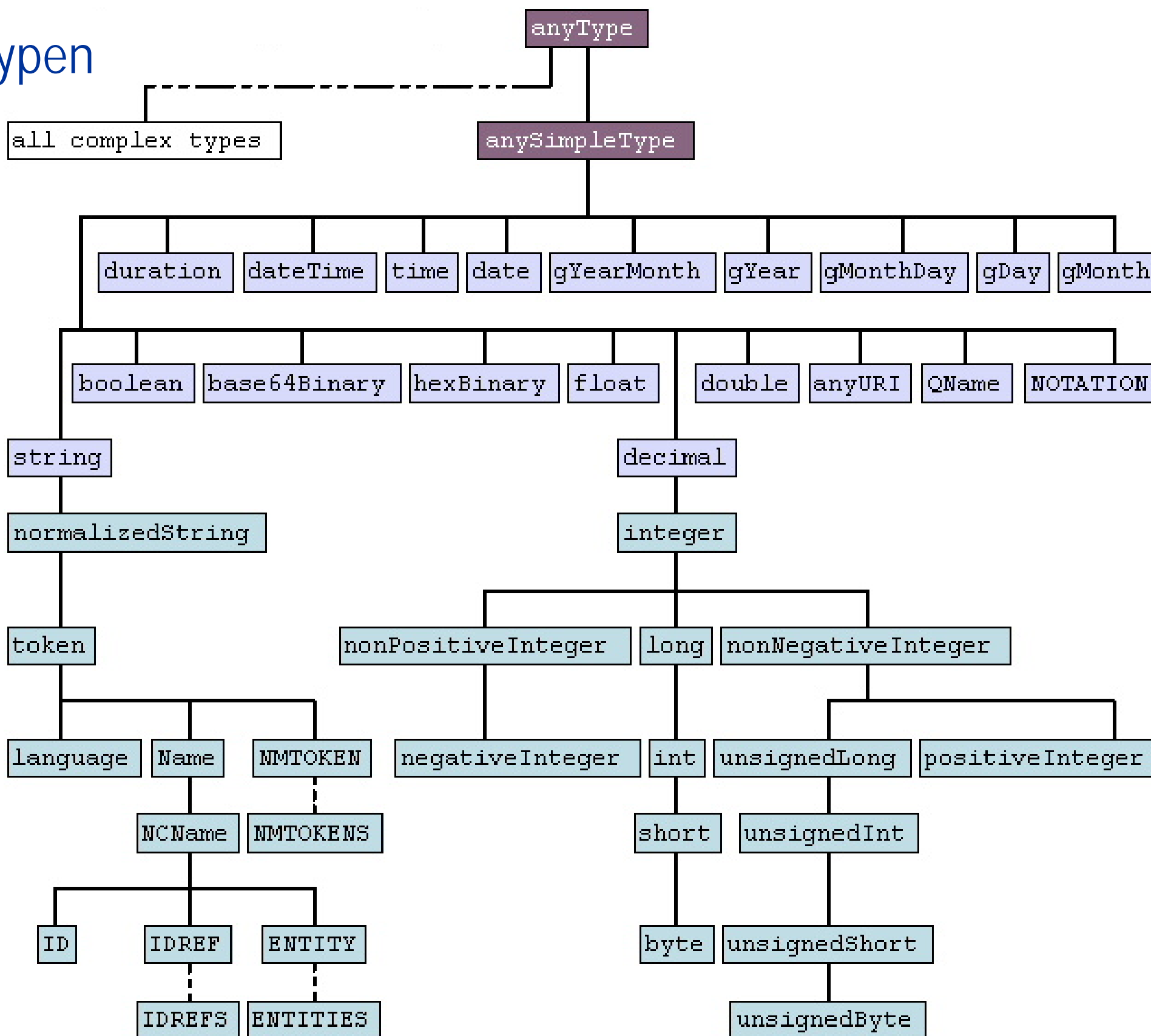
■ Ur-typ

■ Vordefinierter Primitivtyp

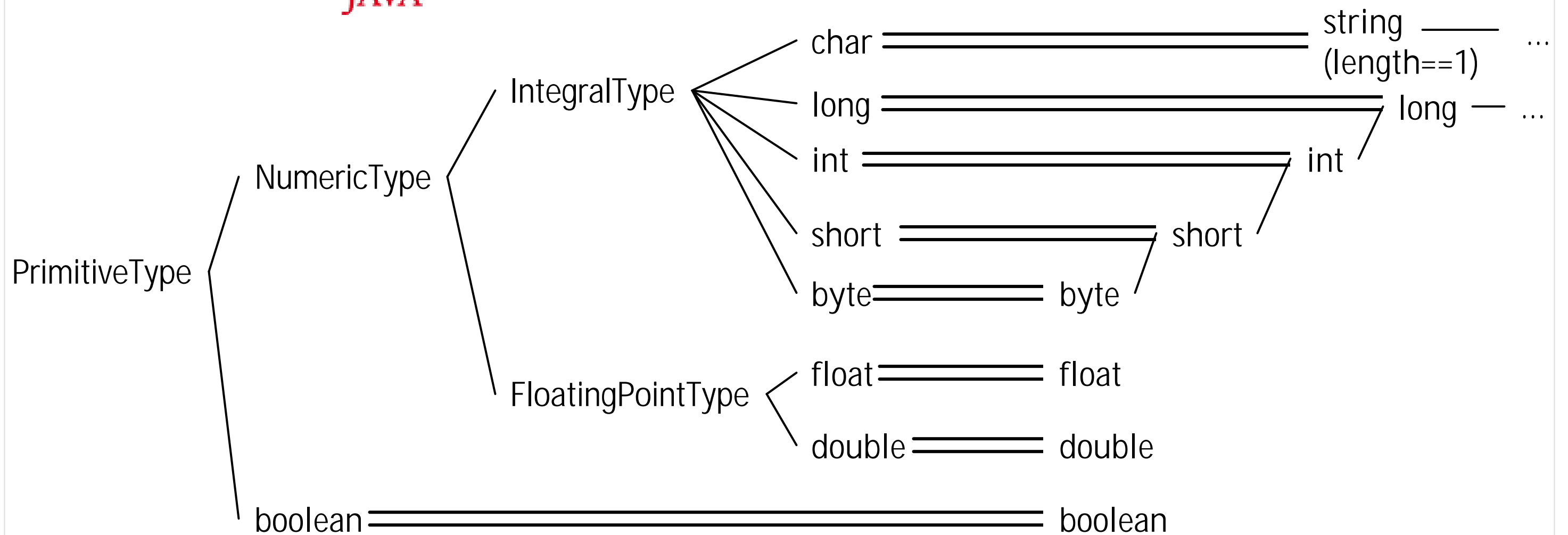
■ Vordefinierter abgeleiteter Typ

— Typeinschränkung

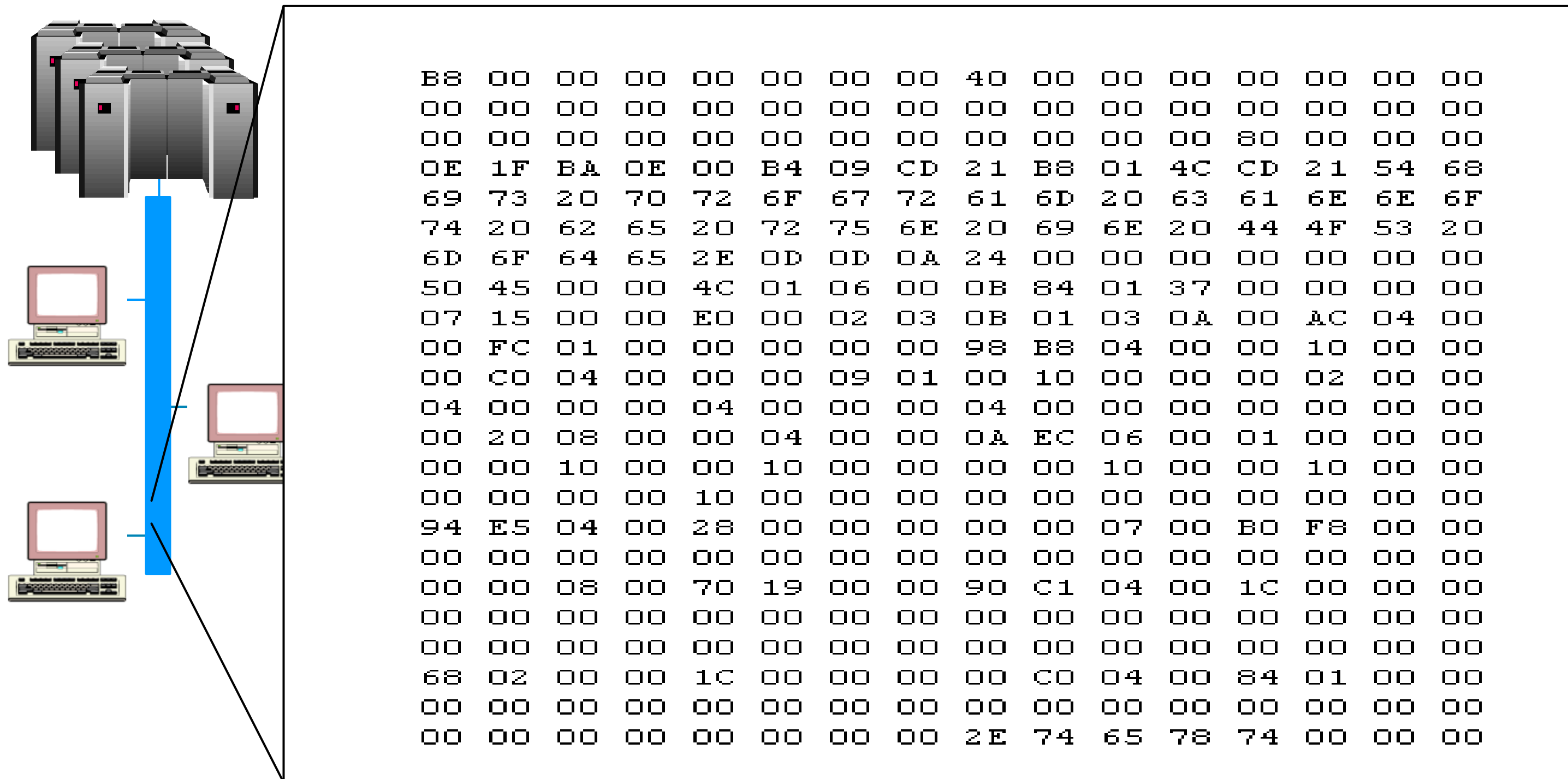
----- Aggregierter Typ



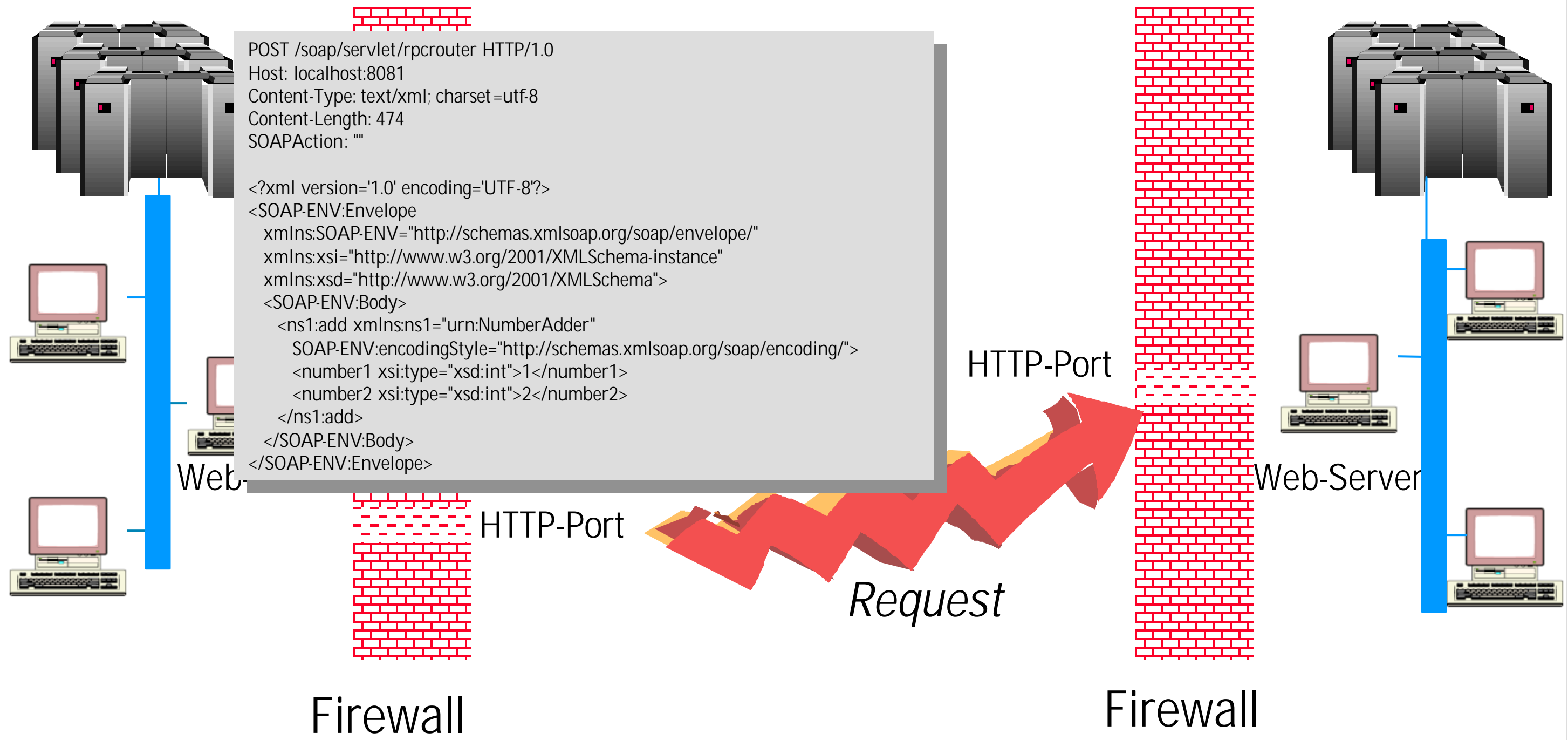
# XML Schema: Datentypen und ihre Abbildung nach Java



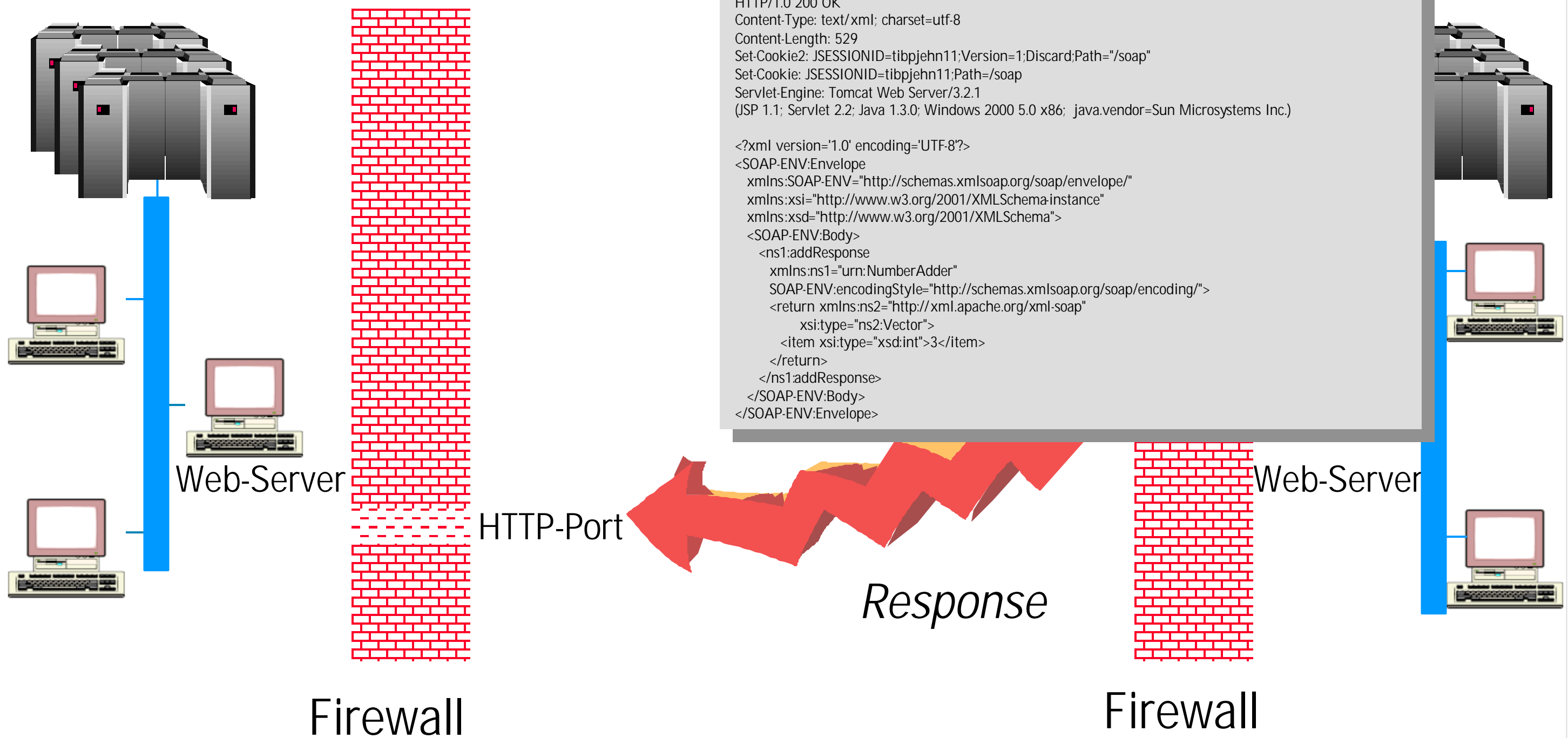
# XML-basierte RPC-Mechanismen: SOAP, ...



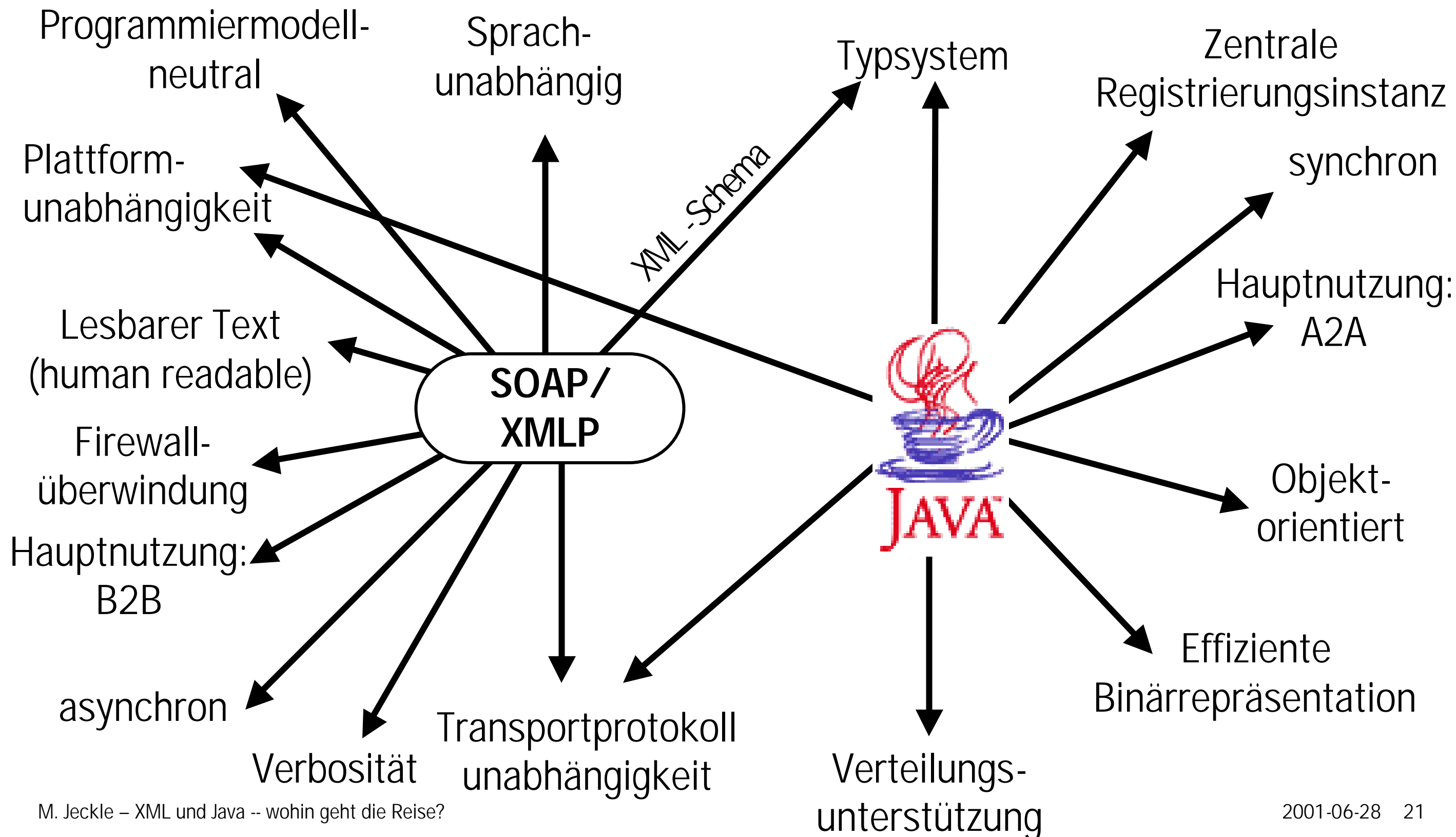
# XML-basierte RPC-Mechanismen: SOAP, ...



# XML-basierte RPC-Mechanismen: SOAP, ...

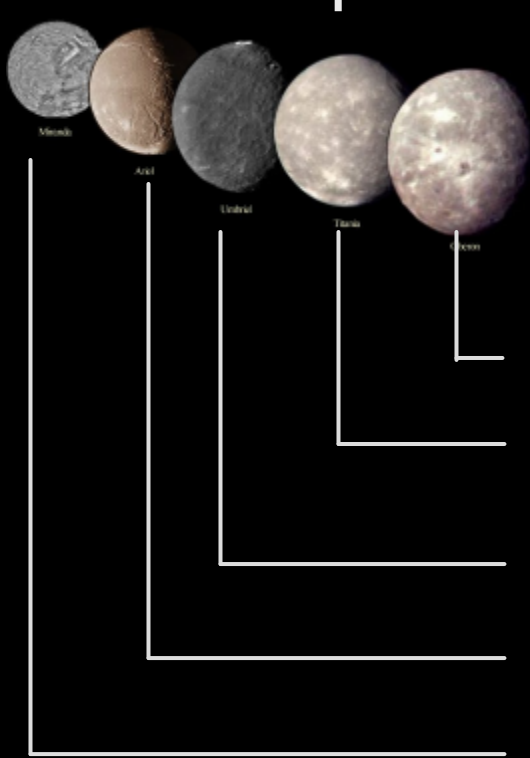


# XML-basierte RPC-Mechanismen: SOAP/XMLP vs. RMI



# Das XML-Universum

## Sprachentwurf



- XML-Schema (XSD)
- DDML
- Schematron
- RELAX NG
- DCD

## Programmierung/ Schnittstellen



SAX

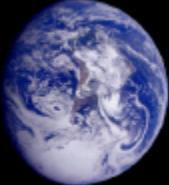
DOM



## Editoren



## Anwendungen



XML-RPC

SOAP/XMLP

Werkzeugunterstützung

Protokolle

Stylesheet-, Linkingprozessoren, ...



## XML und Java ... Simple API for XML (SAX)

- Entwickelt durch Mitglieder der Mailingliste xml-dev auf Initiative von David Megginson
- Zusammenstellung sprachunabhängiger Schnittstellen
- *Light-weight:*
  - einfach (in eigenen Programmen) einzusetzen
  - einfach (in beliebige Programmiersprachen) umzusetzen
- Serieller linearer ereignisbasierter Parsingvorgang
- Definiert und erfordert keine Speicherstrukturen
- SAX ist eine Schnittstelle für Parser, kein Parser!
- Laufzeitverhalten skaliert linear mit Dokumentgröße
- JAXP und J2SE v1.4 enthält (vermutlich) eine Implementierung des aktuellen Standes SAX2

## XML und Java ... Simple API for XML (SAX)

```
import org.xml.sax.helpers.DefaultHandler;  
import javax.xml.parsers.SAXParserFactory;  
import javax.xml.parsers.SAXParser;
```

Implementierung der Standardschnittstelle

```
public class SAXExample extends DefaultHandler
```

```
{
```

```
    public void startDocument()
```

```
    {
```

```
        //...
```

```
    } //startDocument()
```

```
public void startElement(String namespaceURI, String localName, String qName, Attributes atts)
```

```
{
```

```
    // ...
```

```
} //startElement()
```

```
public static void main (String args[]) throws Exception
```

```
{
```

```
    SAXParserFactory spf = SAXParserFactory.newInstance();
```

```
    SAXParser sp = spf.newSAXParser();
```

```
    sp.parse( args[0], new SAXExample() );
```

```
} //main()
```

```
} //class SAXExample
```

## XML und Java ... Simple API for XML (SAX)

```
import org.xml.sax.helpers.DefaultHandler;  
import javax.xml.parsers.SAXParserFactory;  
import javax.xml.parsers.SAXParser;
```

SAX-Parserimplementierung der JAXP

```
public class SAXExample extends DefaultHandler
```

```
{
```

```
    public void startDocument()
```

```
    {
```

```
        //...
```

```
    } //startDocument()
```

```
public void startElement(String namespaceURI, String localName, String qName, Attributes atts)
```

```
{
```

```
    // ...
```

```
} //startElement()
```

```
public static void main (String args[]) throws Exception
```

```
{
```

```
    SAXParserFactory spf = SAXParserFactory.newInstance();
```

```
    SAXParser sp = spf.newSAXParser();
```

```
    sp.parse( args[0], new SAXExample() );
```

```
} //main()
```

```
} //class SAXExample
```

## XML und Java ... Simple API for XML (SAX)

```
import org.xml.sax.helpers.DefaultHandler;
import javax.xml.parsers.SAXParserFactory;
import javax.xml.parsers.SAXParser;

public class SAXExample extends DefaultHandler
{
    public void startDocument()
    {
        //...
    } //startDocument()

    public void startElement(String namespaceURI,
        String localName, String qName, Attributes atts)
    {
        //...
    } //startElement()

    public static void main (String args[]) throws Exception
    {
        SAXParserFactory spf = SAXParserFactory.newInstance();
        SAXParser sp = spf.newSAXParser();
        sp.parse( args[0], new SAXExample() );
    } //main()
} //class SAXExample
```

M. Jeckle - XML und Java -- wohin geht die Reise?

Durch SAX in ihrer Signatur  
vorgegebene  
Callback-Methoden

## XML und Java ... Simple API for XML (SAX)

```
import org.xml.sax.helpers.DefaultHandler;
import javax.xml.parsers.SAXParserFactory;
import javax.xml.parsers.SAXParser;

public class SAXExample extends DefaultHandler
{
    public void startDocument()
    {
        //...
    } //startDocument()

    public void startElement(String namespaceURI,
        String localName, String qName, Attributes atts)
    {
        // ...
    } //startElement()
    public static void main (String args[]) throws Exception
    {
        SAXParserFactory spf = SAXParserFactory.newInstance();
        SAXParser sp = spf.newSAXParser();
        sp.parse( args[0], new SAXExample() );
    } //main()
} //class SAXExample
```

M. Jeckle - XML und Java -- wohin geht die Reise?

Erzeugung eines SAX-basierten  
Parsers und  
prüfen eines XML-Dokuments

## XML und Java ... Document Object Model (DOM)

- W3C standardisierte API für HTML, XML, CSS, etc.  
Stand: Recommendation: DOM Level 1, Level 2; Working Draft: Level 3
- Definiert programmiersprachenunabhängige (beschrieben durch CORBA IDL) baumartige objektorientierte Speicherstruktur für XML mit Operationen zur Navigation und Manipulation  
(z.B. Elemente ändern, löschen, hinzufügen)
- Zumeist auf Basis eines SAX-basierten Parsingvorganges erzeugt
- Speicheraufwand zur Verwaltung des gesamten XML-Dokuments im Hauptspeicher teilweise sehr hoch
- Mit *JDOM* existiert eine für Java optimierte moderne Implementierung  
(d.h. Berücksichtigung von XML-Schema, Nutzung der Collection- und Reflection API, etc.)
- Verfügbar: Diverse Implementierungen; JDOM, DOM4J  
JDOM: JSR

## XML und Java ... Document Object Model (DOM)

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
```

Implementierung der Standardschnittstellen

```
public class DOMExample
{
    public static void main(String[] args) throws Exception
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse( args[0] );

        Element theRootElement = document.getDocumentElement();

        theRootElement.setAttribute("myFirstNewAttribute","01");

        Element aNewElement = document.createElement("myNewElement");

        theRootElement.appendChild( aNewElement );

        System.out.print( theRootElement );
    } //main()
} //class DOMExample
```

M. Jeckle – XML und Java -- wohin geht die Reise?

## XML und Java ... Document Object Model (DOM)

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class DOMExample
{
    public static void main(String[] args) throws Exception
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse( args[0] );

        Element theRootElement = document.getDocumentElement();

        theRootElement.setAttribute("myFirstNewAttribute","01");

        Element aNewElement = document.createElement("myNewElement");

        theRootElement.appendChild( aNewElement );

        System.out.print( theRootElement );
    } //main()
} //class DOMExample
```

M. Jeckle – XML und Java -- wohin geht die Reise?

Erzeugung des  
Parsers und Einlese-  
vorgang

## XML und Java ... Document Object Model (DOM)

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class DOMExample
{
    public static void main(String[] args) throws Exception
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse( args[0] );

        Element theRootElement = document.getDocumentElement();

        theRootElement.setAttribute("myFirstNewAttribute","01");

        Element aNewElement = document.createElement("myNewElement");

        theRootElement.appendChild( aNewElement );

        System.out.print( theRootElement );
    } //main()
} //class DOMExample
```

M. Jeckle – XML und Java -- wohin geht die Reise?

Extraktion eines  
Elements



## XML und Java ... Document Object Model (DOM)

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class DOMExample
{
    public static void main(String[] args) throws Exception
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse( args[0] );

        Element theRootElement = document.getDocumentElement();
        theRootElement.setAttribute("myFirstNewAttribute","01");

        Element aNewElement = document.createElement("myNewElement");

        theRootElement.appendChild( aNewElement );

        System.out.print( theRootElement );
    } //main()
} //class DOMExample
```

M. Jeckle – XML und Java -- wohin geht die Reise?

Hinzufügen eines  
(neuen) Attributs

## XML und Java ... Document Object Model (DOM)

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class DOMExample
{
    public static void main(String[] args) throws Exception
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse( args[0] );

        Element theRootElement = document.getDocumentElement();

        theRootElement.setAttribute("myFirstNewAttribute","01");

        Element aNewElement = document.createElement("myNewElement");

        theRootElement.appendChild( aNewElement );

        System.out.print( theRootElement );
    } //main()
} //class DOMExample
```

M. Jeckle – XML und Java -- wohin geht die Reise?

Erzeugung eines  
neuen Elements



## XML und Java ... Document Object Model (DOM)

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class DOMExample
{
    public static void main(String[] args) throws Exception
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse( args[0] );

        Element theRootElement = document.getDocumentElement();

        theRootElement.setAttribute("myFirstNewAttribute","01");

        Element aNewElement = document.createElement("myNewElement");

        theRootElement.appendChild( aNewElement );

        System.out.print( theRootElement );
    } //main()
} //class DOMExample
```

M. Jeckle – XML und Java -- wohin geht die Reise?

Anhängen des erzeugten  
Elements als zusätzliches Kindelement

## XML und Java ... Document Object Model (DOM)

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.Element;

public class DOMExample
{
    public static void main(String[] args) throws Exception
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse( args[0] );

        Element theRootElement = document.getDocumentElement();

        theRootElement.setAttribute("myFirstNewAttribute","01");

        Element aNewElement = document.createElement("myNewElement");

        theRootElement.appendChild( aNewElement );

        System.out.print( theRootElement );
    } //main()
} //class DOMExample
```

M. Jeckle – XML und Java -- wohin geht die Reise?

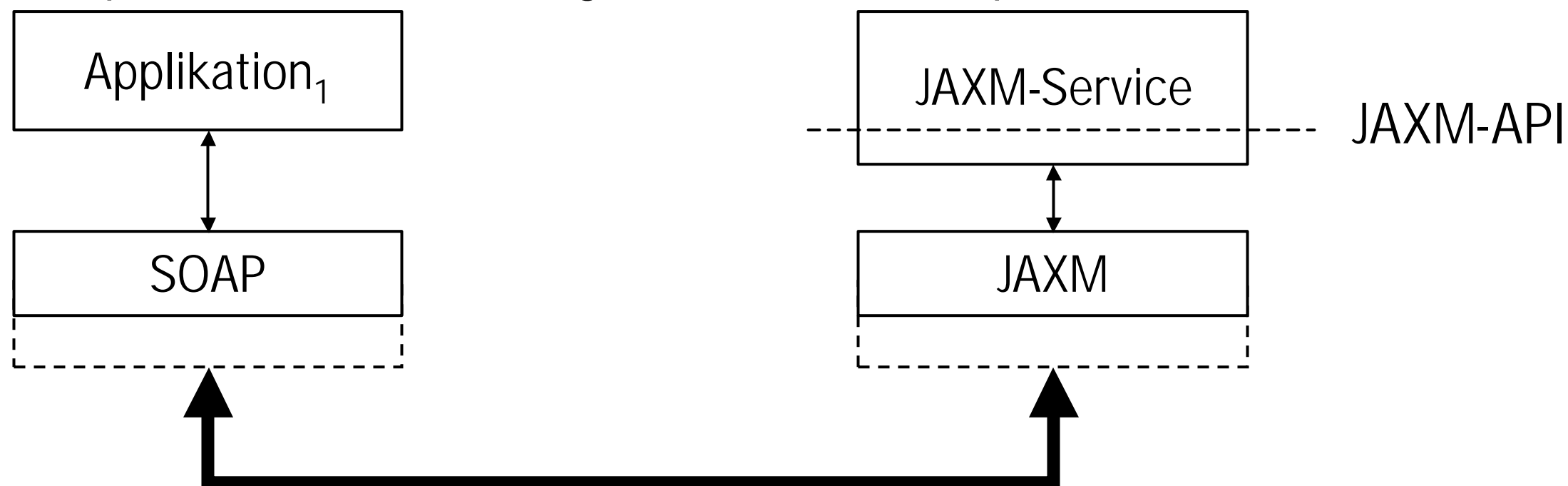
Ausgabe des gesamten Dokuments  
auf Standard-out

## Java und XML ... Java XML API for XML Processing (JAXP)

- Berücksichtigt:
  - XML v1.0 2<sup>nd</sup> edition ;)
  - Namespaces
  - SAX v2.0
  - DOM Level 2
  - XSLT v1.0
- *Plugability layer* realisiert Informationsmöglichkeit hinsichtlich Implementierung der Parser
- JAXP v1.0 ist (vermutlich) Bestandteil von J2SE v1.4
- Verfügbar: JAXP v1.1: JSR, Spezifikation, Implementierung

## Java und XML ... Java APIs for XML Messaging (JAXM)

- Grundidee: Austausch von XML-codierten *Business* Dokumenten (z.B. Bestellungen, Rechnungen oder Auftragsbestätigungen)
- Definiert keine neuen XML-Vokabulare für den Anwendungsbereich
- Entwickelt in Abstimmung mit ähnlichen Aktivitäten des W3C, ebXML (Transport/Packaging & Routing)
- Transport über (Internet-)Standardprotokolle wie HTTP(S) oder SMTP
- Explizite Berücksichtigung der SOAP v1.1-Spezifikation



- Verfügbar: JSR, Spezifikation v0.92 (Juni 2001)

## Java und XML ... Java APIs for XML RPC (JAXRPC)

- Grundidee: Übertragung von entfernten Methodenaufrufen mittels XML
- Überlappung mit SOAP/XMLP; jedoch koordiniert mit W3C-Aktivitäten hierzu
- APIs definieren *stubs* und *skeletons* zur Erzeugung der notwendigen XML-Dokumente auf Seiten des Aufrufers bzw. Aufgerufenen
- API zur Vermittlung der XML-RPCs an Java-Objekte
- Verbindung zu JAXB (XML Serialisierung von Java-Objekten)
- Verfügbar: JSR

## Java und XML ... Java API for XML Registries (JAXR)

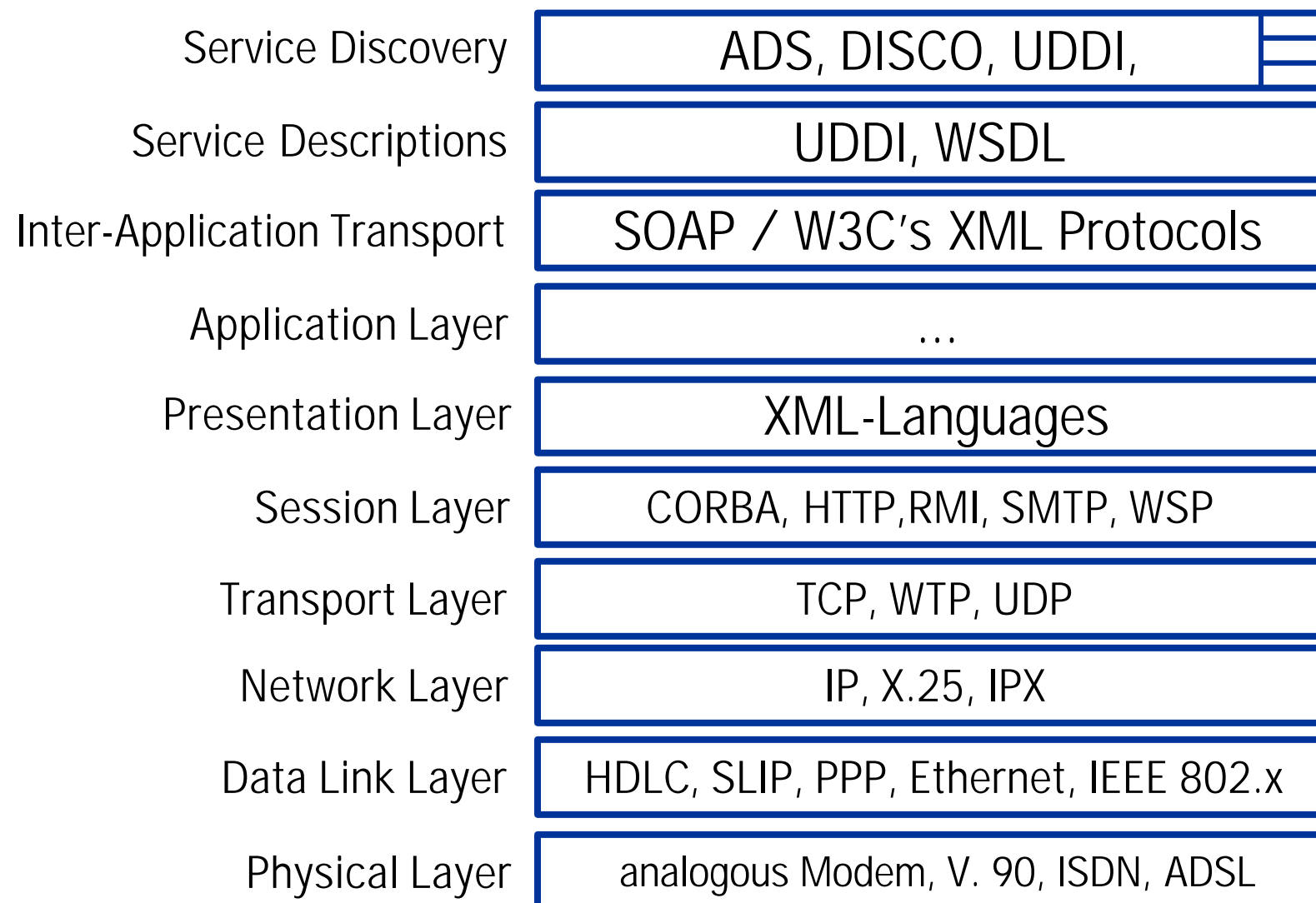
- Grundidee: API zum Zugriff auf Dienstverzeichnisse
- Analog zu *Java Naming and Directory Interface (JNDI)*, jedoch speziell für Bedürfnisse von Web-Services zugeschnitten

- Standard-API zum Zugriff auf: eCo Framework, ebXML, UDDI

- Definiert keinen neuen eigenen Verzeichnisdienst

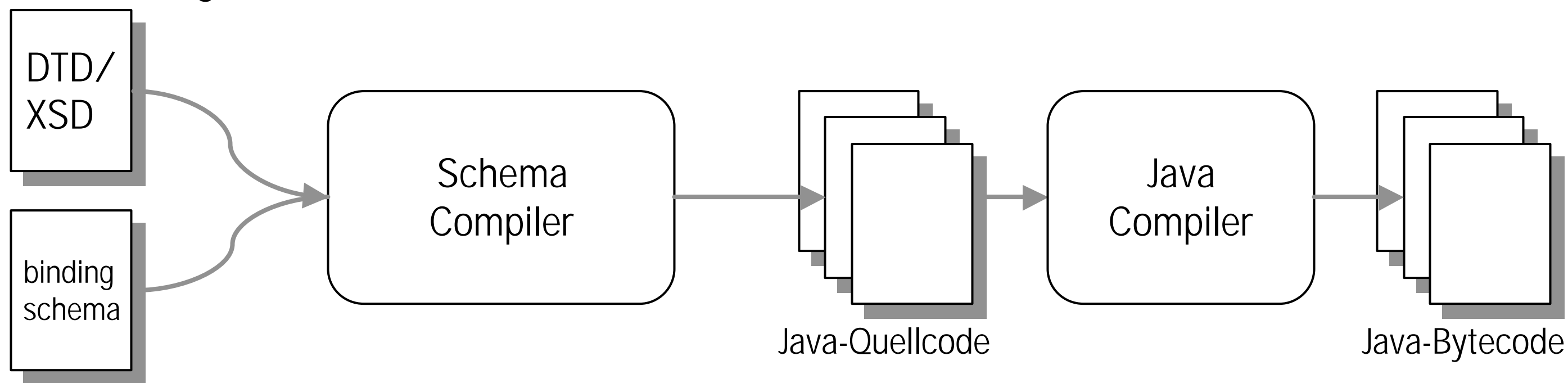
- Bestandteil von J2EE

- Verfügbar: JSR; (public draft für Q2 2001 zu erwarten)



## Java und XML ... Java Architecture for XML Binding (JAXB)

- Grundidee: Verringerung des *Impedance Mismatch* zwischen Java und XML
- Technisch: Generierung von Java-Klassendefinitionen aus Grammatik (Document Type Definitions oder XML Schema) und einer Abbildungsdefinition



- JAXB v1.0 ist (vermutlich) Bestandteil von J2SE v1.4
- Verfügbar: JSR, early acces v1.0 Implementierung (Mai 2001)

## Ressourcen

XML @ W3C:

- <http://www.w3.org/xml>

XML @ SUN:

- <http://java.sun.com/xml>
- <http://www.javasoft.com/xml>

Java XML Projekte @ Apache:

- <http://xml.apache.org>

Diese Vortrag und weiterführende Informationen zum Thema:

- <http://www.jeckle.de>

## Ressourcen

Java Community Process (JCP):

- <http://java.sun.com/aboutJava/communityprocess/index.html>
- (Merlin JSR) <http://jcp.org/jsr/detail/59.jsp>
- Java Architecture for XML Binding (ehemals Projekt *Adelard*) (JAXB)
  - <http://jcp.org/jsr/detail/31.jsp>
  - <http://www.javaworld.com/javaworld/javaone00/j1-00-adelard.html>
- Java API for XML Processing v1.1 (JAXP)
  - Version 1.0: <http://jcp.org/jsr/detail/5.jsp>
  - Version 1.1: <http://jcp.org/jsr/detail/63.jsp>
- Java API for XML Messaging v1.0 (JAXM)
  - <http://jcp.org/jsr/detail/67.jsp>
- Java API for XML Registries v1.0 (JAXR)
  - <http://jcp.org/jsr/detail/93.jsp>
- Java APIs for XML RPC (JAXRPC)
  - <http://jcp.org/jsr/detail/101.jsp>

## Ressourcen

SAX-Parser für Java:

- <http://home.pacbell.net/david-b/xml> (Ælfred)
- <http://java.sun.com/products/xml> (JAXP)
- <http://technet.oracle.com>
- <http://www.alphaworks.ibm.com/formula/xml> (XML4J)
- <http://www.jclark.com/xml/xp/index.html> (XP)
- <http://xml.apache.org/xerces-j/index.html> (Xerces)

DOM-Parser für Java:

- <http://java.sun.com/products/xml> (JAXP)
- <http://www.alphaworks.ibm.com/formula/xml> (XML4J)
- <http://xml.apache.org/xerces-j/index.html> (Xerces)

## Ressourcen

Java Serialization to XML (JSX):

- <http://www.csse.monash.edu.au/~bren/JSX/>

JDOM v1.0:

- <http://jcp.org/jsr/detail/102.jsp>

DOM4J:

- <http://dom4j.org/>

Electric XML:

- <http://www.theminelectric.com/products/xml/xml.html>

Java XML Tutorial @ SUN:

- [http://java.sun.com/xml/tutorial\\_intro.html](http://java.sun.com/xml/tutorial_intro.html)

JavaML:

- <http://www.cs.washington.edu/homes/gjb/JavaML/>