DAIMLERCHRYSLER

Making Machines Talk Together – Systemintegration mit XML-Sprachen

Mario Jeckle
DaimlerChrysler Forschungszentrum Ulm
mario.jeckle@daimlerchrysler.com
mario@jeckle.de
www.jeckle.de

Systemintegration mit XML-Sprachen

I Heterogenität und ihre Ausprägungen

II Problemfelder und Herausforderungen

II Lösungsansatz für die *Enterprise Application Integration*: XML

1 Behebung der syntaktischen Heterogenität

XML-Sprachgewinnung und –design

- XML-Schema (W3C's XSD)
- Sprachgewinnung mit dem *XML Metadata Interchange* (XMI)

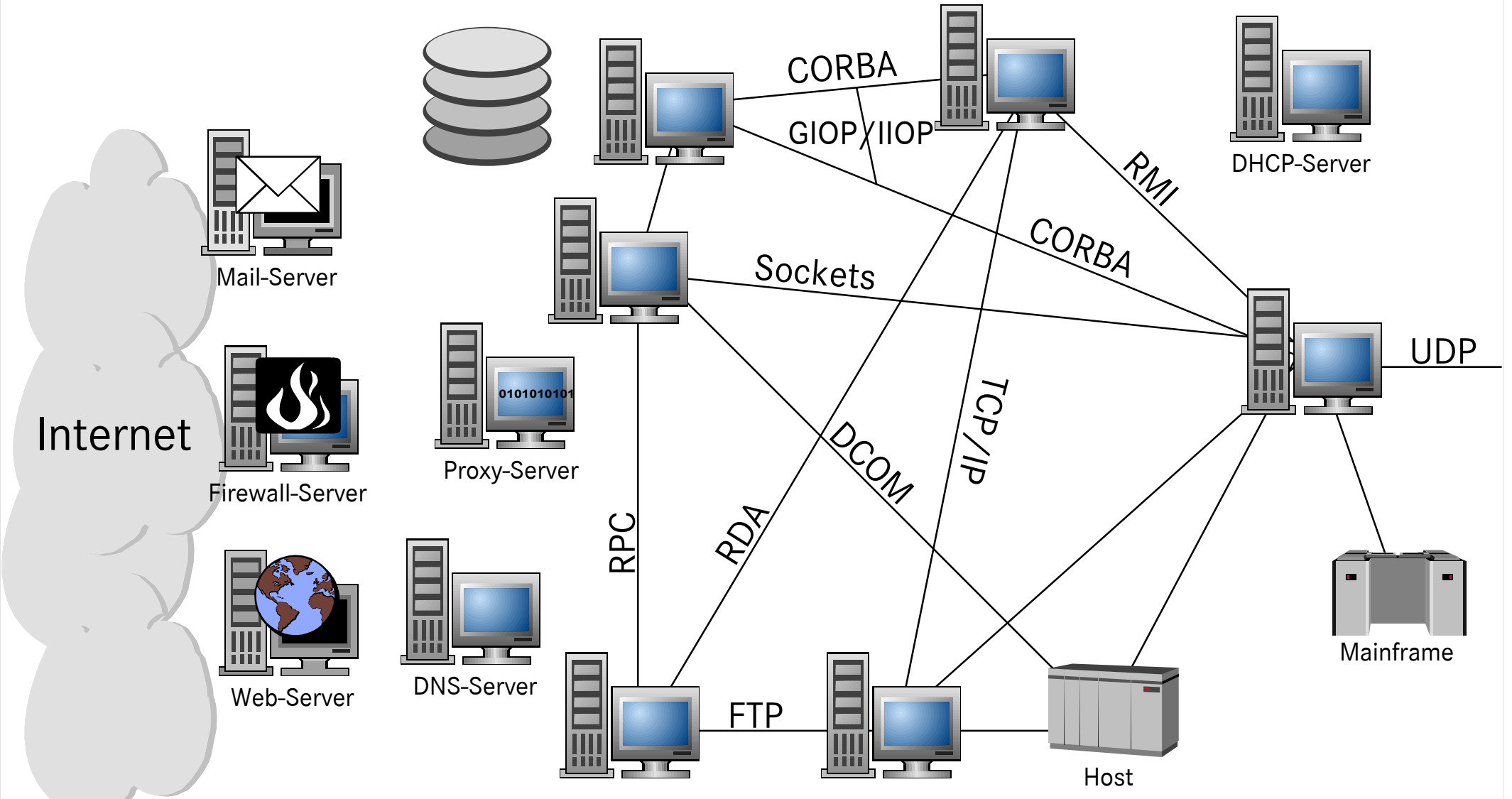
2 Behebung der semantischen Heterogenität

- Sprach- und Schematransformation mit XML Stylesheet Transformations (XSLT)

3 Behebung der Kommunikations-Heterogenität

- Simple Object Access Protocol (SOAP)

Heterogenität

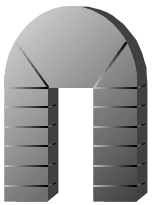


Heterogenitätsdimensionen

Upper Enterprise Application Integration

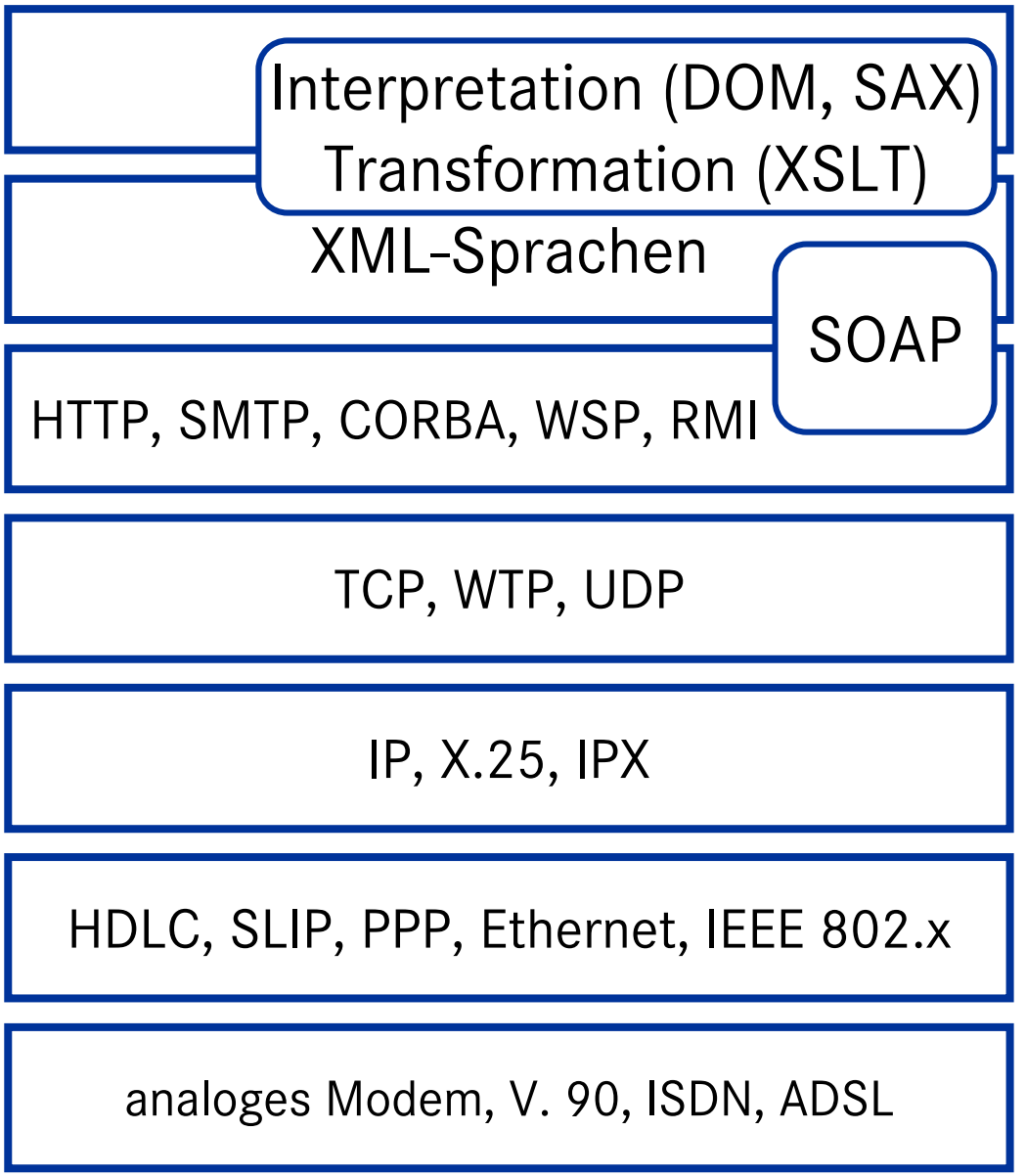
- Applikationsschicht (*Application Layer*)
- Darstellungsschicht (*Presentation Layer*)
- Sitzungsschicht (*Session Layer*)
- Transportschicht (*Transport Layer*)
- Netzschicht (*Network Layer*)
- Sicherungsschicht (*Data Link Layer*)
- Bitübertragungsschicht (*Physical Layer*)

Gateway



Router

Bridge



W3C's XML Schema

Gemeinsamer Wortschatz (Semantik)

- ermöglicht Kommunikation und Interaktion auf Basis einheitlich definierter Begriffe

Formale Beschreibung (Syntax)

- Grundvoraussetzung maschineller Verarbeitung
- Idealerweise (vergleichsweise) einfache Verarbeitung

  (schlanke, eindeutige Definitionen, möglichst kontextfreie oder reguläre Sprache)

Austauschbasis

- Explizite Strukturdefinition ist Grundvoraussetzung des Informationsaustauschs.

W3C's XML Schema – technisch gesehen

- Neuentwicklung durch Arbeitsgruppe des W3C (*candidate recommendation* 2000-10-24*)*
- Mächtigkeit analog des bestehenden DTD-Mechanismus um Dokumentstruktur (Reihenfolge, Auftrittsvielfachheit von Elementen und Attribute) zu beschreiben
  Insbesondere wurden folgende Erweiterungen verwirklicht:
  - Namespace Integration
  - Definition von Einschränkungen für Elementinhalte
  - Integration Strukturschema und primitive Datentypen
  - Vererbung: DTD unterstützt nur *kind-of*-Beziehungen
  - Erweiterter Referenzierungsmechanismus (URI)
- „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
- Einschränkungen an Typen

W3C's XML Schema – Beispiel

```
<?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>
```

W3C's XML Schema – Beispiel

```
<schema xmlns = "http://www.w3.org/2000/08/XMLSchema">
  <element name = "bestellung">
    <complexType mixed= "false">
      <sequence>
        <element ref = "artikel"/>
      </sequence>
    </complexType>
  </element>
  <element name = "nummer" type = "string"/>
  <element name = "benennung" type = "string"/>
```

```
<?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>
```

freie Elementdefinition
(strukturiertes Element)

Elementverwendung

freie Elementdefinition
(einfaches Element)

W3C's XML Schema – Beispiel

```
<element name = "preis">
  <complexType mixed = "false">
    <sequence>
      <element ref = "betrag"/>
      <element ref = "waehrung"/>
    </sequence>
  </complexType>
</element>
```

```
<?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>
```

freie Elementdefinition
(strukturiertes Element)

Elementverwendung

W3C's XML Schema – Beispiel

```
<element name = "artikel">
  <complexType mixed = "false">
    <sequence>
      <element ref = "nummer"/>
      <element ref = "benennung"/>
      <element ref = "preis"/>
      <element ref = "kunde"/>
    </sequence>
  </complexType>
</element>
```

```
<?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>
```

freie Elementdefinition
(strukturiertes Element)

Elementverwendung

W3C's XML Schema – Beispiel

```
<simpleType name="currency">
  <restriction base="decimal">
    <precision value="8"/>
    <scale value="2" fixed="true"/>
    <minExclusive value="0"/>
</simpleType>
<element name = "betrag" type = "currency"  />
<element name = "waehrung" type = "currencyName"/>
<simpleType name="currencyName">
  <restriction base="string">
    <enumeration value="DEM"/>
    <enumeration value="USD"/>
  </restriction>
</simpleType>
```

```
<?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>
```

Anwenderdefinierter Datentyp (abgeleitet von *decimal)*

Typverwendung

Anwenderdefinierter Aufzählungstyp

W3C's XML Schema – Beispiel

```
<simpleType name="kundenNummer">
  <restriction base="string">
    <pattern value="\p{Lu}-\d{3}-\d{2}"/>
  </restriction>
</simpleType>
<element name = "kunde">
  <complexType mixed = "false">
    <attribute name = "nummer" use = "required" type = " kundenNummer "/>
  </complexType>
</element>
</schema>
```

```
<?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>
```

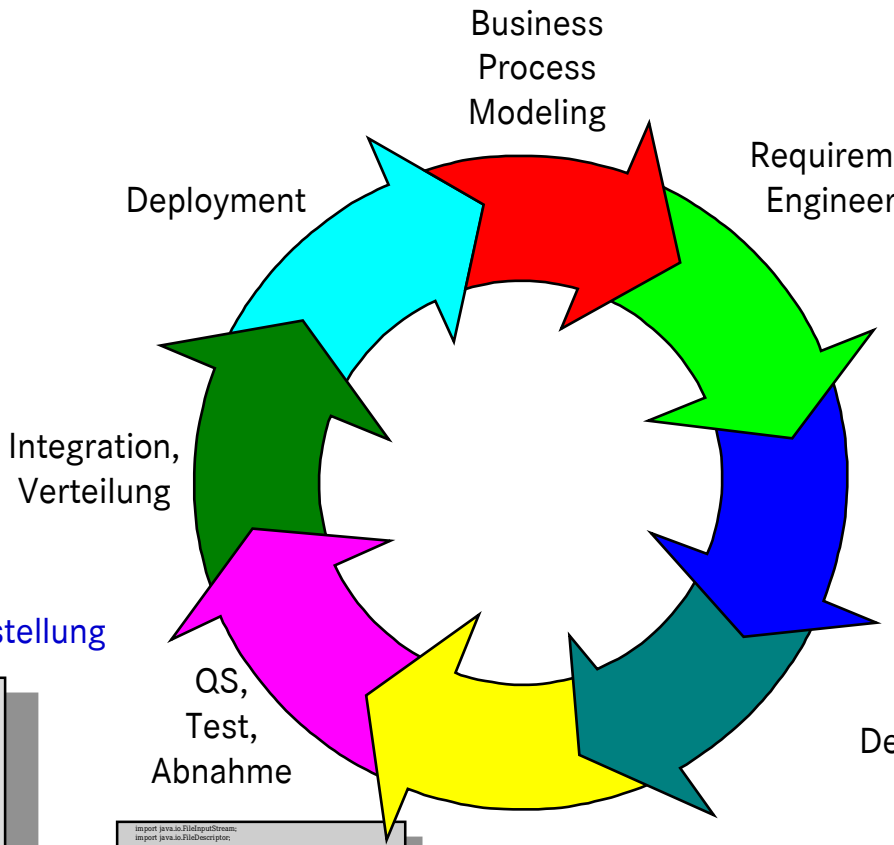
Anwenderdefinierter Datentyp (lexikale Definition)

Typverwendung

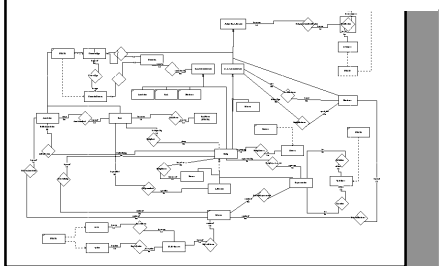
XML-Sprachgewinnung

informale Formulierung

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



Informationsmodell



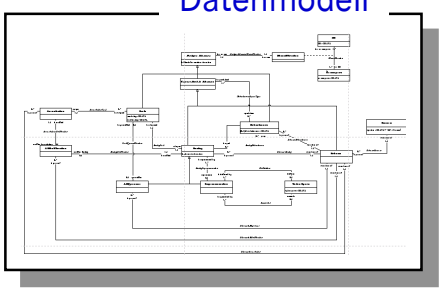
formale Darstellung

```
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

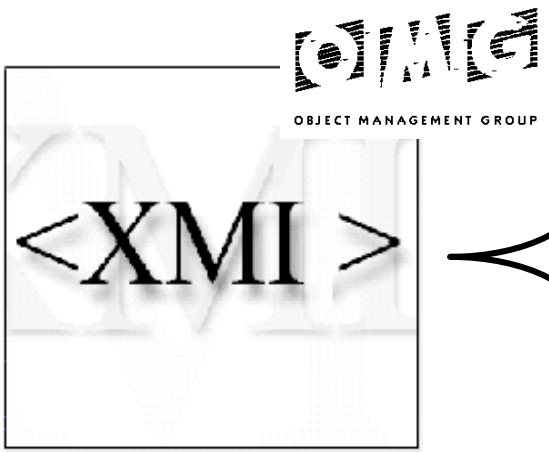
```
import java.io.FileInputStream;
import java.io.FileDescriptor;

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

  private char getCharacterFromStdInput()
  {
    char readCharacter='\0';
    try
    {
      readCharacter = (char) fis.read();

    catch (Exception e)
                                                                    aught");
```

Datenmodell



XML-Sprachgewinnung

Transfer von...



- Meta-Metamodell-Ausprägungen

  (vollständige Modellierunssprachen)

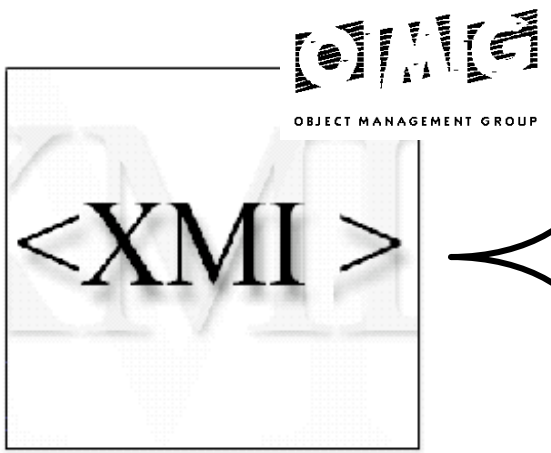
- Metamodell-Ausprägungen

  (vollständige Datenmodelle)

- Modell-Ausprägungen

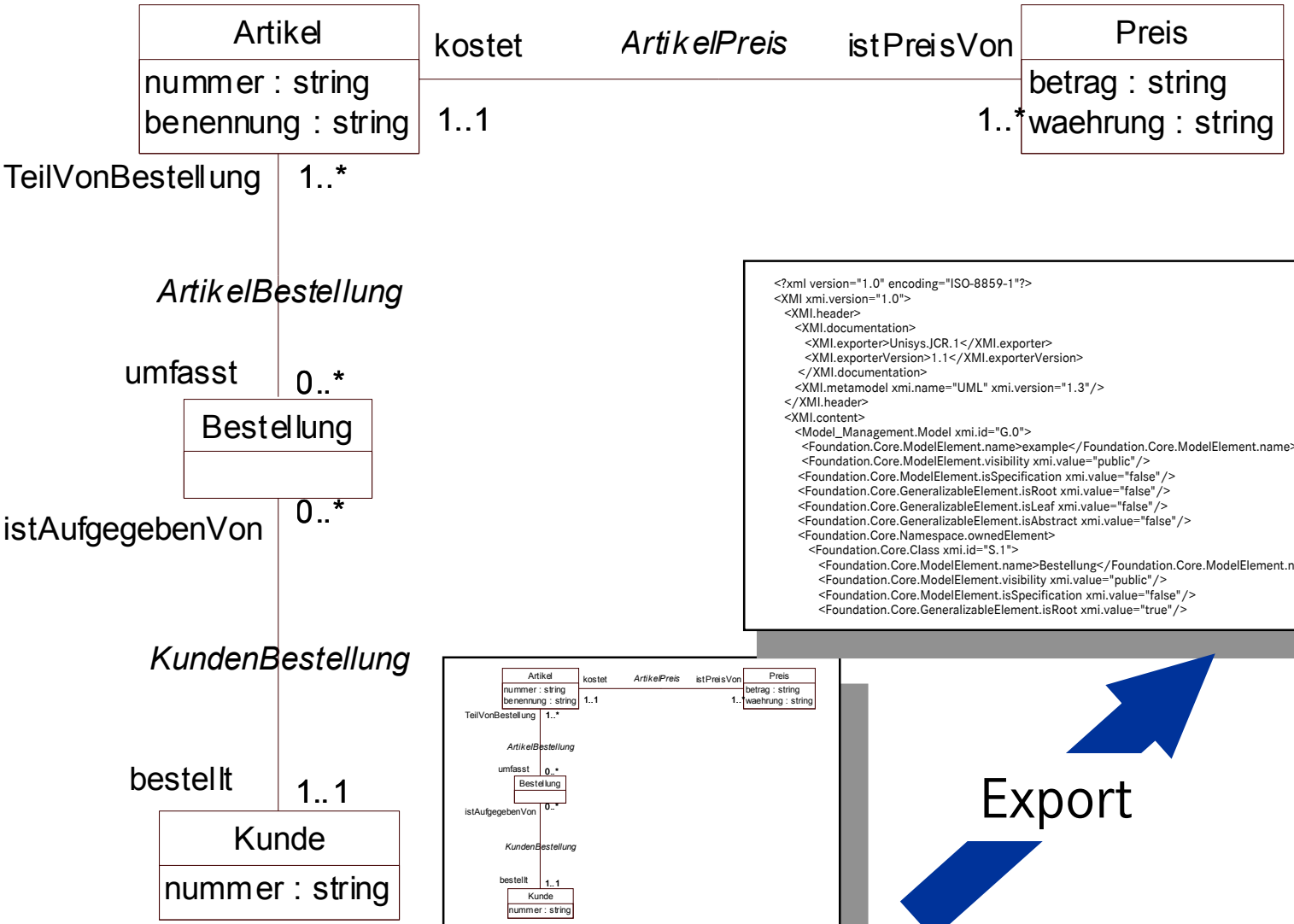
  (vollständige Datenmodell-Ausprägungen/ Instanzen)

XML-Sprachgewinnung



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

XML-Sprachgewinnung – Beispiel




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

Generierung

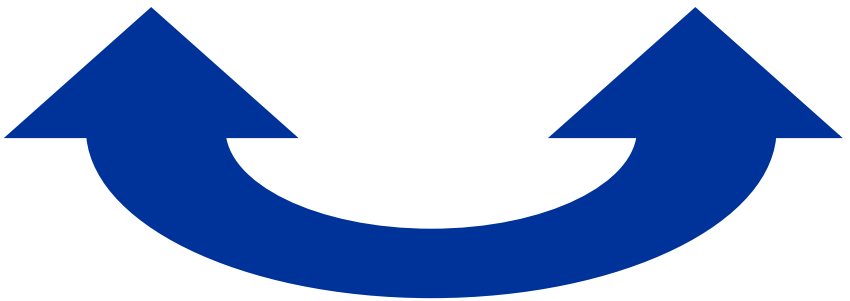
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<XMI xmi.version="1.0">
  <XMI.header>
    <XMI.documentation>
      <XMI.exporter>Unisys.JCR.1</XMI.exporter>
      <XMI.exporterVersion>1.1</XMI.exporterVersion>
    </XMI.documentation>
    <XMI.metamodel xmi.name="UML" xmi.version="1.3"/>
  </XMI.header>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

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>
```

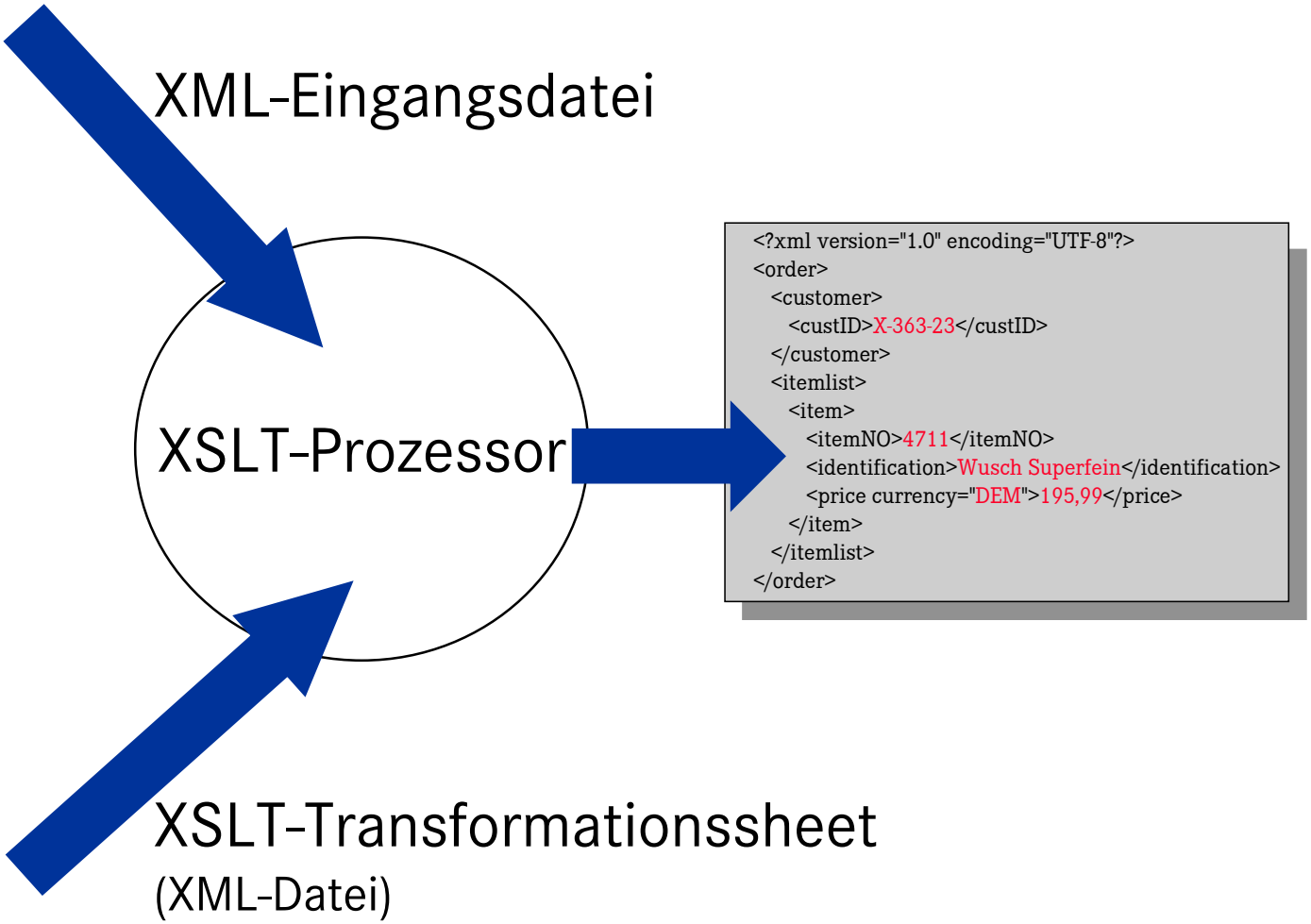


Transformation

Sprachtransformation: XSLT

```
<?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 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"  inde

<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: SOAP

- 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

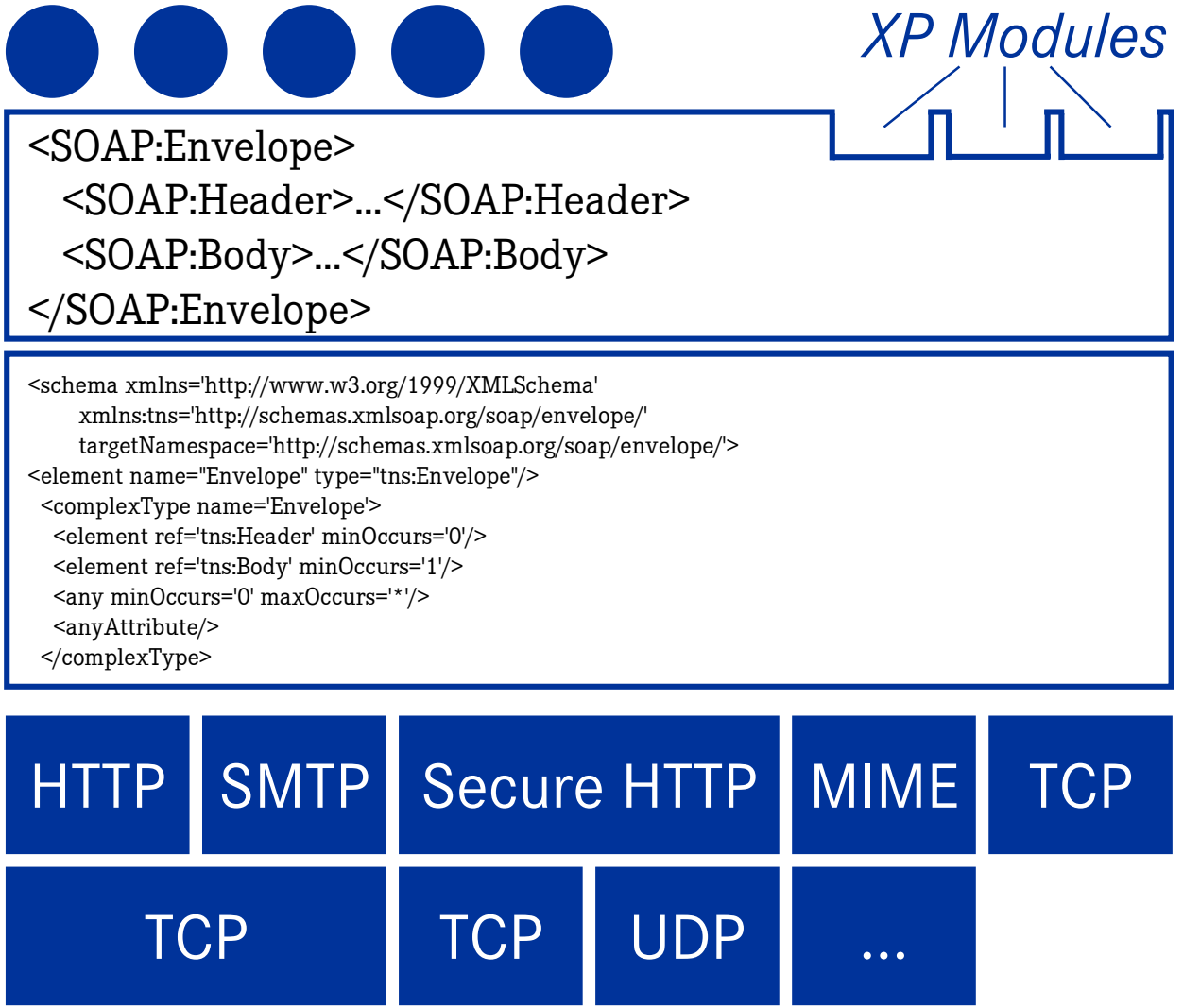
Systemkommunikation: SOAP Interoperabilitätsebenen

Services

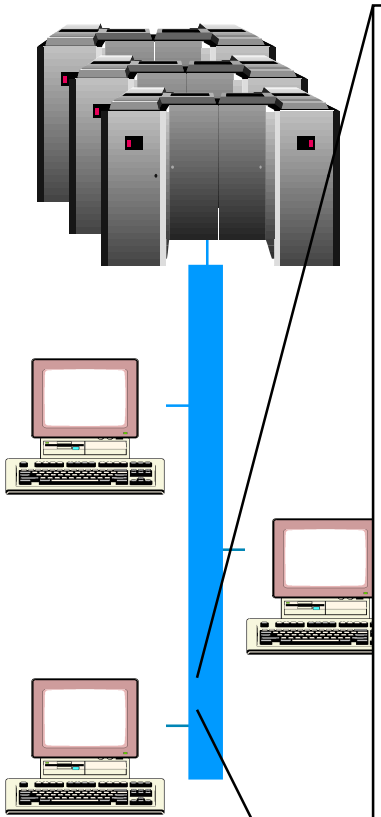
SOAP / XML Protocols

XML
(v1.0 2nd editon
+ XML Schema)

Common Internet Protocols



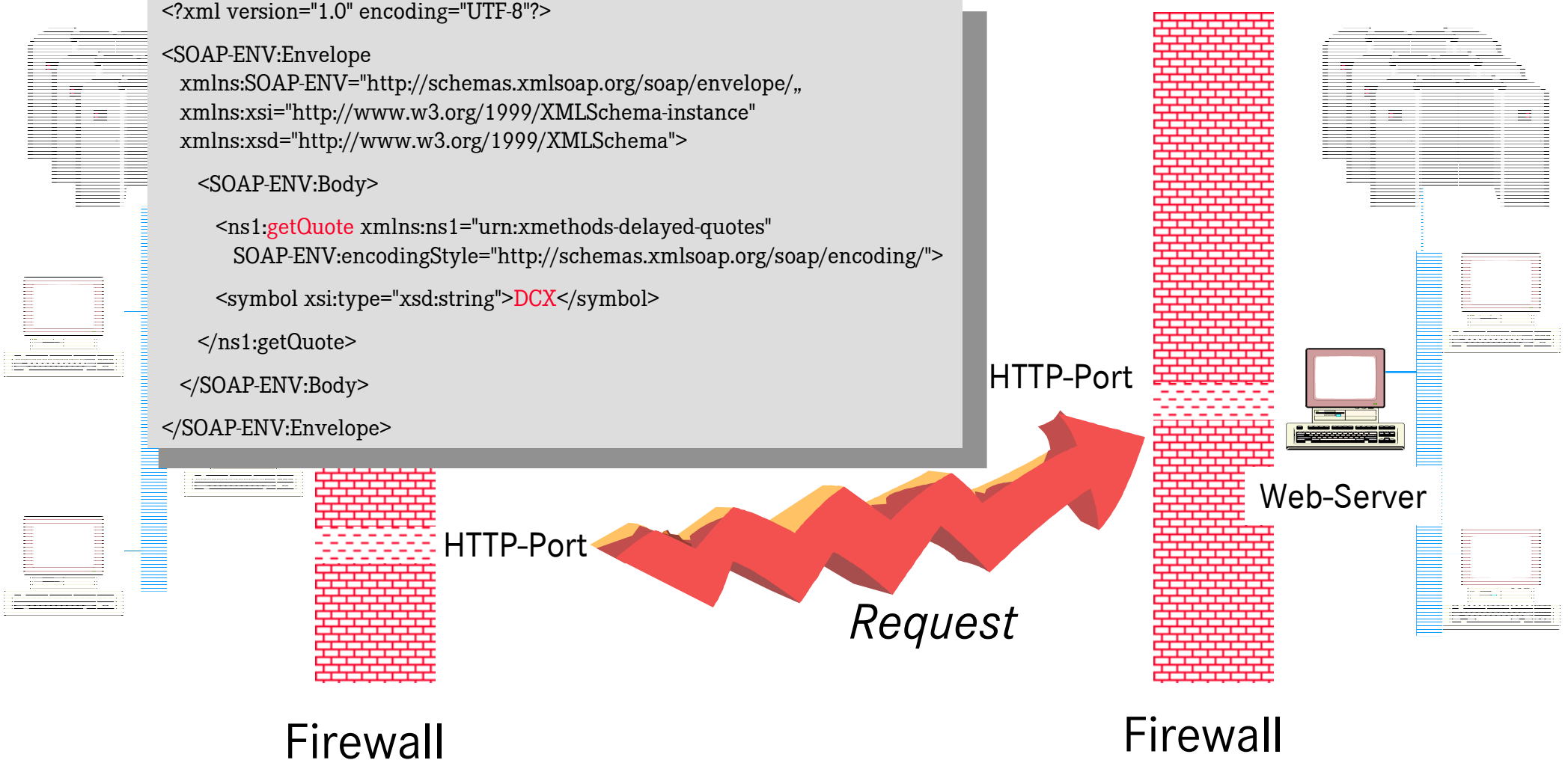
Systemkommunikation: SOAP



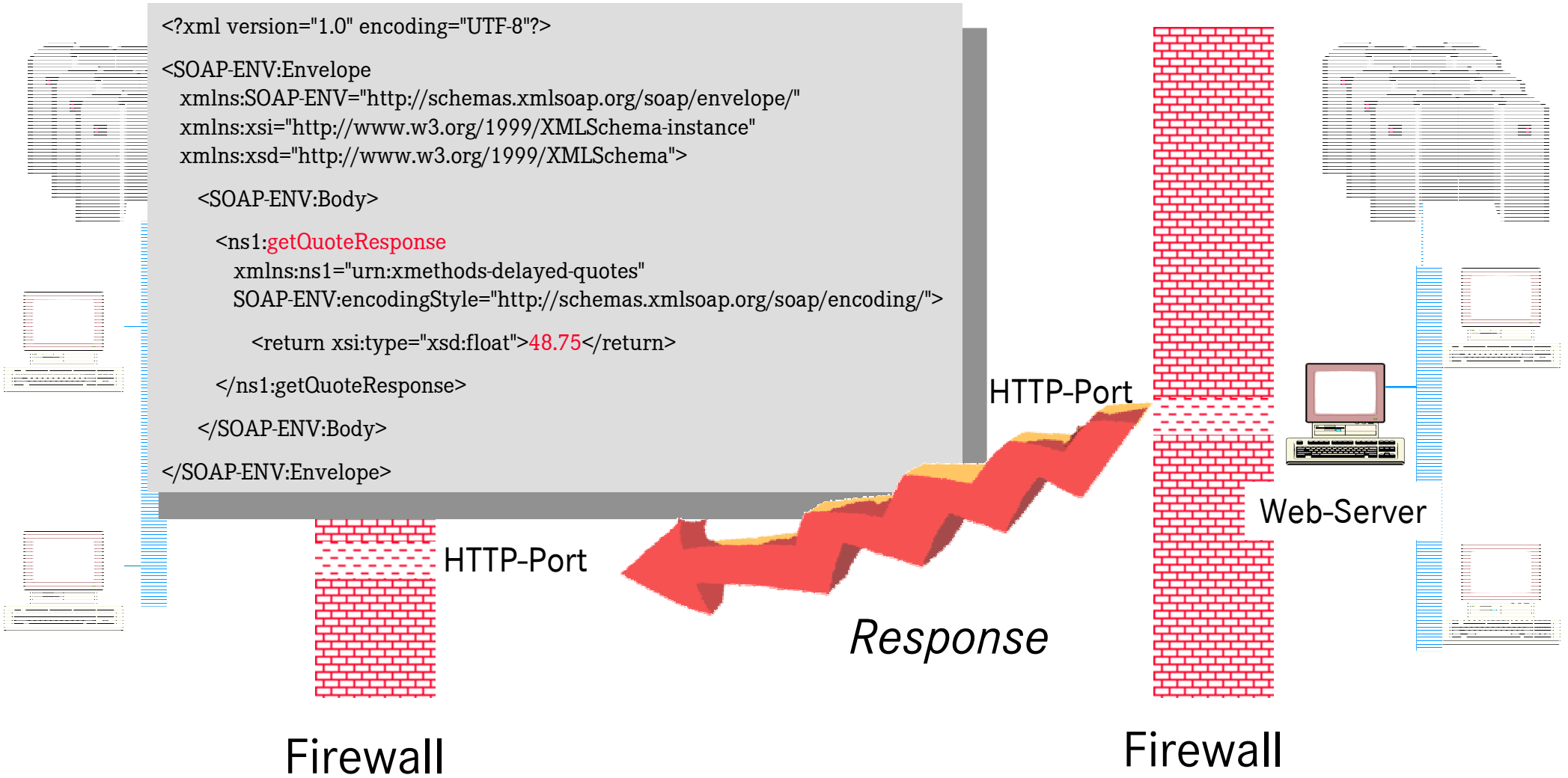
```
B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00
0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68
69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F
74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20
6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00
50 45 00 00 4C 01 06 00 0B 84 01 37 00 00 00 00
07 15 00 00 E0 00 02 03 0B 01 03 0A 00 AC 04 00
00 FC 01 00 00 00 00 00 98 B8 04 00 00 10 00 00
00 C0 04 00 00 00 09 01 00 10 00 00 00 02 00 00
04 00 00 00 04 00 00 00 04 00 00 00 00 00 00 00
00 20 08 00 00 04 00 00 0A EC 06 00 01 00 00 00
00 00 10 00 00 10 00 00 00 00 10 00 00 10 00 00
00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00
94 E5 04 00 28 00 00 00 00 00 07 00 B0 F8 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 08 00 70 19 00 00 90 C1 04 00 1C 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
68 02 00 00 1C 00 00 00 00 C0 04 00 84 01 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00
```

Systemkommunikation: SOAP

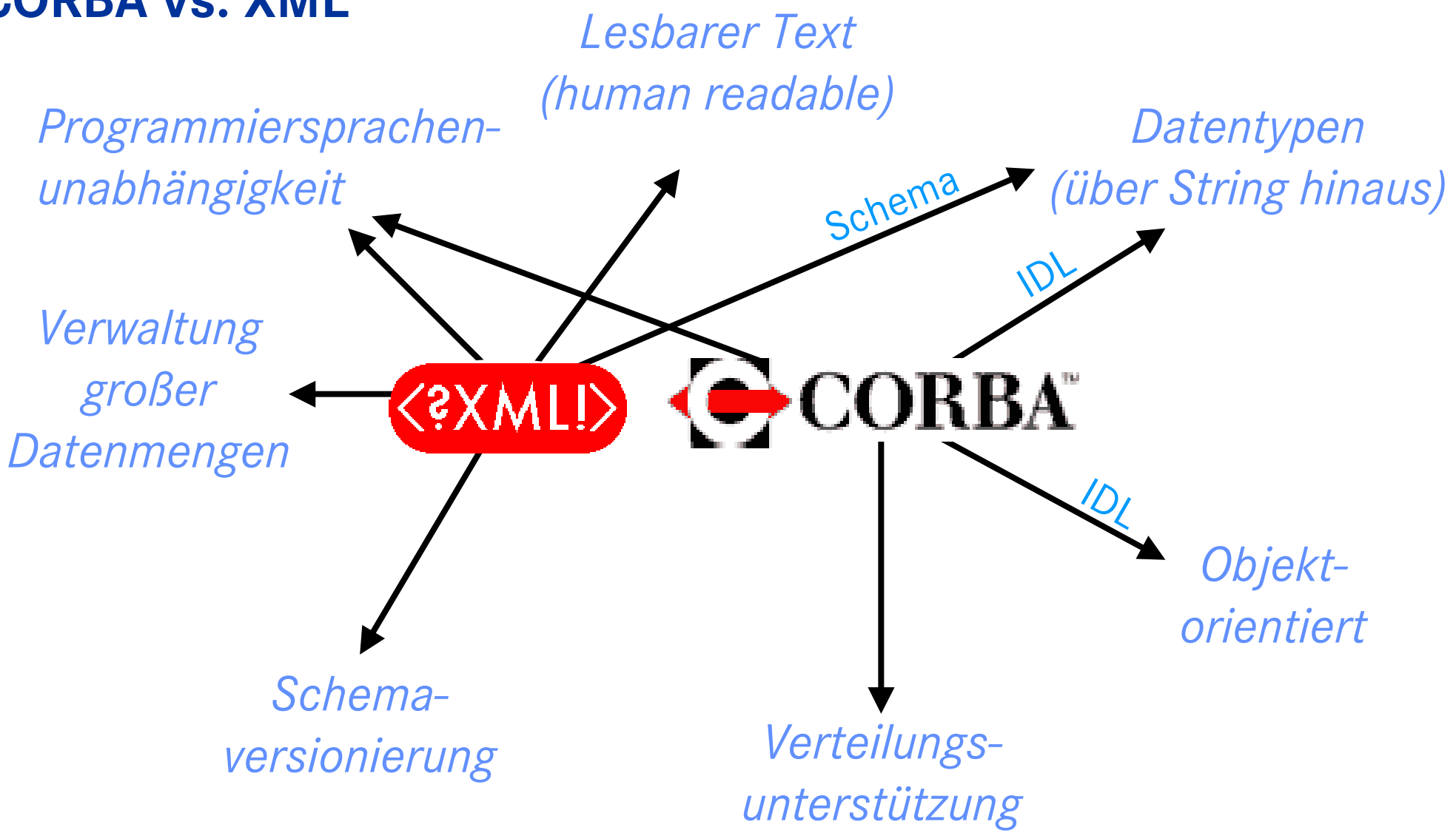
```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/„
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
    <SOAP-ENV:Body>
      <ns1:getQuote xmlns:ns1="urn:xmethods-delayed-quotes"
        SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <symbol xsi:type="xsd:string">DCX</symbol>
    </ns1:getQuote>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



Systemkommunikation: SOAP



CORBA vs. XML


Lesbarer Text
(human readable)
Programmiersprachen-
unabhängigkeit
Datentypen
(über String hinaus)
Schema
IDL
Verwaltung
großer
Datenmengen
<?XML!>
CORBA™
IDL
Objekt-
orientiert
Schema-
versionierung
Verteilungs-
unterstützung

Referenzen

W3C's XML-Schema

- www.w3c.org/TR/xml-schema-0
- www.w3c.org/TR/xml-schema-1
- www.w3c.org/TR/xml-schema-2

XML Metadata Interchange

- www.omg.org/xml
- www.xmiforum.org
- www.software.ibm.com/ad/features/xmi.html
- www.alphaworks.ibm.com/tech/xmitoolkit

XSLT

- www.w3c.org/TR/xslt
- www.xslt.com

SOAP

- www.w3.org/TR/SOAP/
- www.w3.org/2000/03/29-XML-protocol-matrix
- www.w3.org/2000/xp

Dieser Vortrag und weiterführende Hintergrundinformation

www.jeckle.de