



# XML in der Praxis

Ein Referat von der Gruppe 7

Tobias Beckers  
Christian Lohmann  
Adam Mildner  
Marcin Skirzynski



# Gliederung

## (1) XML im allgemeinen Einsatz

## (2) XML im speziellen Einsatz

- XREHA – XML für den Informationsaustausch im Reha-Umfeld
  - XML im B2B-Commerce:  
A framework for sharing product information across enterprises
  - XML im B2B-Commerce:  
WebEDI
  - XML im B2C-Commerce:  
Mobile Agenten zur Angebotsrecherche
- XML als Datenformat (OpenDocument)



## XML im allgemeinen Einsatz

Wie kommt XML in der allgemeinen Praxis zum Einsatz?

- Datenaustausch zwischen Programmen und Datenbanken
  - Konvertieren von XML mit XSLT
  - Beschreibung von Datenstrukturen



# XML im speziellen Einsatz

## XREHA und der Datenaustausch zwischen Beleger und Rehaklinik

Quelle: Alt, B., Prillwitz, J.: XML für den Informationsaustausch im Reha-Umfeld: der XREHA Standard.  
In: Schweizer, B., Großmann, Ch., Meule, M., Gaus, W. (Hg.): Dokumentation - der Schritt ins 3.  
Jahrtausend. 8. DVMD-Tagung in Ulm, 1.-2. April 2004. Ulm 2004, S. 139 - 144. ISBN 3-89559-  
149-1



# XREHA und der Datenaustausch zwischen Beleger und Rehaklinik

XREHA steht für die Spezifikation für die online Reservierung in REHA-Einrichtungen.

Um eine problemlose Kommunikation zwischen einzelnen REHA-Einrichtungen und einem Internetportal zur Reservierung zu ermöglichen wird XREHA, das man auch als 'XML-basierter Datenaustausch im REHA-Umfeld' bezeichnet, angewandt.

- XML wird hierbei zur Strukturierung der Informationen verwendet.
  - Zur Spezifikation werden DTD's verwendet

Reservierungsanfrage im Rehaportal.info - Mozilla

Reservierungsanfrage für die Klinik: (D) Klinik Sommerau, 88436 Eberhardzell

**Freie Kapazitäten**  
In der ausgewählten KW 01 vom 29.12 bis zum 02.01 sind freie Betten vorhanden.

**Zeitpunkt der Rehabilitation**

gewünschter Beginn: 29.12.2003 [TT.MM.JJJJ]  
spätest möglicher Beginn: 02.01.2004 [TT.MM.JJJJ]  
Anzahl Tage: 21

**Ausstattung**  
Die Klinik verfügt über folgende Ausstattung. Wählen Sie bitte die erforderlichen Merkmale. Beachten Sie bitte, dass unter Umständen nicht alle Merkmale miteinander kombinierbar sind.

Zimmer	Bett	Ausstattung
<input type="checkbox"/> Doppelzimmer	<input type="checkbox"/> Doppelbett	<input type="checkbox"/> aufzugnah
<input type="checkbox"/> Einzelzimmer	<input type="checkbox"/> Einzelbett	<input type="checkbox"/> Farbfernseher
	<input type="checkbox"/> Pflegebett	<input type="checkbox"/> Rollstuhlgerecht
	<input type="checkbox"/> AHB	<input type="checkbox"/> Sauerstoffanschluss
	<input type="checkbox"/> ARB	
	<input type="checkbox"/> Reha	



# XREHA und der Datenaustausch zwischen Beleger und Rehaklinik

## XREHA-Spezifikation *durch 6 Kommunikationspaare*

	Anfrage	Antwort
1	Anforderung allgemeiner Belegung <i>(grobe Übersicht verfügbarer Ressourcen)</i>	Antwortdaten allgemeine Belegung
2	Anforderung Ausstattung <i>(Ausstattung der Klinik und weitere relevanten klinikspezifische Informationen: behindertengerecht, aufzugsnah, etc.)</i>	Antwortdaten Ausstattung
3	Differenzierte Anfrage nach freien Ressourcen <i>(Konkrete Anfrage nach einem Termin)</i>	Antwortdaten differenzierte Anfrage
4	Reservierung <i>(nimmt direkt Bezug auf die differenzierte Anfrage und reserviert für diesen Zeitraum)</i>	Reservierungsbestätigung
5	Storno <i>(Reservierungen werden storniert)</i>	Stornobestätigung
6	Ping Anfrage <i>(Systemverfügbarkeit)</i>	Ping Antwort
7		Allgemeine Fehlermeldung



# XREHA und der Datenaustausch zwischen Beleger und Rehaklinik

## DTD der Anforderungsdaten Ausstattung

```
<?xml version="1.0"
  encoding="ISO-8859-1"?>
<!ELEMENT KLINIK_AUSSTATTUNG
  (KLINIK)>
<!ELEMENT KLINIK EMPTY>
<!ATTLIST KLINIK
  KLINIKID CDATA #REQUIRED
  IKNR CDATA #REQUIRED >
```

## XREHA Anforderung Ausstattung

```
<?xml version="1.0"
  encoding="ISO-8859-1"?>
<!--XREHA Anfrage zur
  Ausstattung-Uebersicht-->
<KLINIK_AUSSTATTUNG>
<KLINIK KLINIKID="162"
  IKNR="309197900"/>
</KLINIK_AUSSTATTUNG>
```

Für eine Anforderung zum Element KLINIK\_AUSSTATTUNG braucht, aufgrund des '#REQUIRED', die XML-Datei die 'KLINIK'-Attribute KLINIKID und IKNR als Zeichenkette ('CDATA').



# XREHA und der Datenaustausch zwischen Beleger und Rehaklinik

## DTD der Antwortdaten Ausstattung

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

<!ELEMENT KLINIK_AUSSTATTUNG
  (KLINIK, ZIMMER?, BETT?,
  AUSSTATTUNG?, KUNDE?,
  INDIKATIONEN?, AUFNAHMETAGE?)>
<!ATTLIST KLINIK_AUSSTATTUNG
  VERSION CDATA #IMPLIED>
<!ELEMENT KLINIK EMPTY>
<!ATTLIST KLINIK
  KLINIKID CDATA #REQUIRED
  NAME CDATA #IMPLIED
  IKNR CDATA #REQUIRED >
<!ELEMENT ZIMMER (ART*, LAGE*)>
<!ELEMENT BETT (ART*, LAGE*, TYP*)>
<!ELEMENT ART (#PCDATA)>
<!ATTLIST ART
  TEXT CDATA #REQUIRED
  ID CDATA #REQUIRED>
...
```

Das Fragezeichen gibt an, dass das Attribut („ZIMMER?, BETT?, ...) fehlen darf.

## XREHA Antwortdaten Ausstattung

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--Klinik Ausstattung Übersicht
-28.12.2003-->
<KLINIK_AUSSTATTUNG>
<KLINIK_KLINIKID="162"
  IKNR="309197900"/>
<ZIMMER>
<ART TEXT="Doppelzimmer" ID="DZ"/>
<ART TEXT="Einzelzimmer" ID="EZ"/>
</ZIMMER>
<BETT>
<ART TEXT="Einzelbett" ID="1"/>
<ART TEXT="Doppelbett" ID="2"/>
<ART TEXT="Pflegebett" ID="4"/>
<TYP TEXT="AHB" ID="A"/>
<TYP TEXT="ARB" ID="B"/>
<TYP TEXT="Reha" ID="R"/>
</BETT>
<AUSSTATTUNG>
<A TEXT="Farbfernseher" ID="1"/>
<A TEXT="Sauerstoffanschluss" ID="2"/>
<A TEXT="Rollstuhlgerecht" ID="3"/>
<A TEXT="aufzugnah" ID="4"/>
</AUSSTATTUNG>
<KUNDE>
<ANREDE TEXT="Herr" ID="01"/>
...
```





# XML im speziellen Einsatz

## A framework for sharing product information across enterprises

Quelle: Hyun Kim, Hyung-Sun Kim, Joo-Haeng Lee, Jin-Mi Jung, Jae Yeol Lee, Nam-Chul Do: A  
framework for sharing product information across enterprises.  
Springer-Verlag London Limited 2005



# A framework for sharing product information across enterprises

„Virtual Enterprises“ durch „Collaborative product commerce (CPC)“ - mehrere Unternehmen, die sich in ihren Stärken ergänzen, indem sie ihr „Produktwissen“ untereinander teilen.



# A framework for sharing product information across enterprises

Wichtig dafür!

- ✓ **Product data interoperability**

*Strukturen in der Produktbeschreibung bei vielen Firmen unterschiedlich. Wie können die Informationen trotzdem effektiv geteilt werden?*

- ✓ **System application interoperability**

*Die Systeme der Unternehmen sind untereinander oft heterogen. Eine nahtlose Integration untereinander muss aber sichergestellt werden.*

- ✓ **Process interoperability**

*Private Firmenvorgänge müssen mit den zu veröffentlichenden Vorgängen interagieren.*



# A framework for sharing product information across enterprises

- **Geben ...**
  - *Unternehmen erstellt einen Service, um eigene Produktinformationen zur Verfügung zu stellen*
  - *Registriert den Service öffentlich*
- **... und nehmen nach dem SOA-Prinzip!**
  - *Unternehmen fragt in der 'registry' nach Produkt-Daten*
  - *Bildung eines 'virtuellen Teams', um bestimmte Bedingungen unter den Unternehmen abzuklären*
  - *'business partners' teilen Produktinformationen untereinander*



# A framework for sharing product information across enterprises

## Product data interoperability

*Weil verschiedene Unternehmen ihre Produkt-Informationen in unterschiedlicher Art und Weise benutzen, muss ein Standard-Daten-Schema benutzt werden.*

*→ Benutzt wird das standardisierte ISO STEP PDM Schema, was hier mit Hilfe von XML repräsentiert wird. Insgesamt beinhalten die Produkt Meta-Daten 129 komplexe Type-Elemente.*



# A framework for sharing product information across enterprises

```
ENTITY product;
  id : identifier;
  name : label;
  description : OPTIONAL text;
  frame_of_reference : SET[1:?] OF product_context;
END_ENTITY;
```

(a) EXPRESS

```
<xsd:complexType name="product">
  <xsd:attribute name="id" type="xsd:ID"/>
  <xsd:attribute name="name" type="label"/>
  <xsd:attribute name="description" type="text"
    use="optional"/>
  <xsd:sequence>
    <xsd:element name="frame_of_reference"
      type="product_context"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

(b) XML Schema

- Beispiel eines XML-Schemas für die Produkt Meta-Daten

(a) Produkt Meta-Daten

(b) XML-Schema

*Komplexer Datentyp „product“ mit den Attributen*

*'id' vom Typ ID,*

*'name' vom Typ label,*

*'descripton' (optional) vom Typ text*

*und eine 'sequence' vom Typ*

*product\_context, um darzustellen in*

*welchem Kontext das Produkt zu*

*sehen ist.*

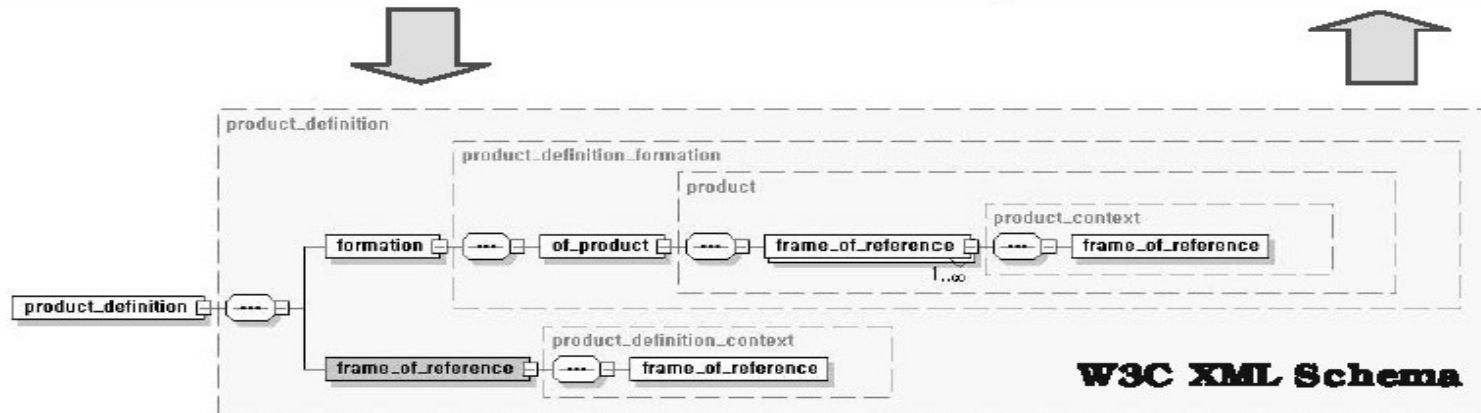
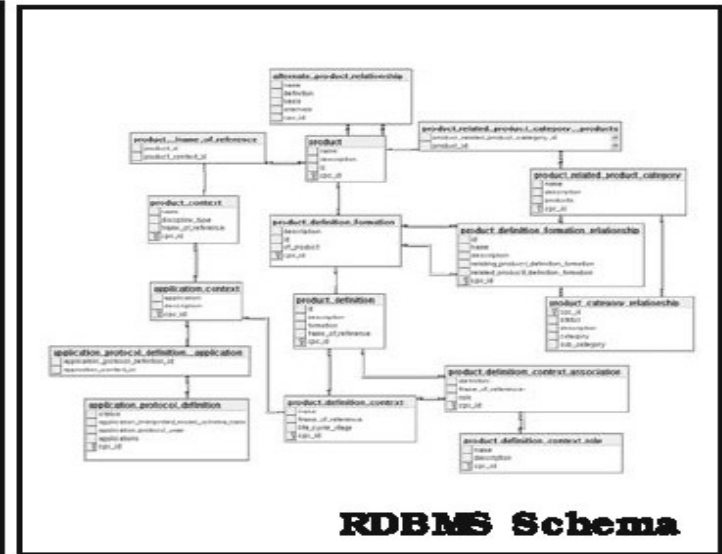
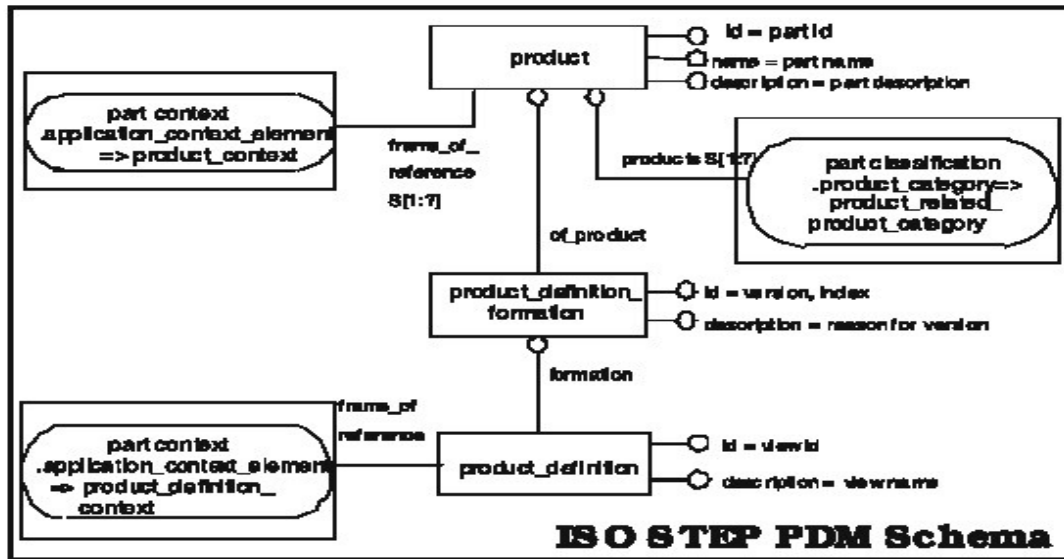


# A framework for sharing product information across enterprises

Um die Produkt Meta-Daten und Informationen effektiv zu verwalten wird noch ein 'repository' konstruiert. Aus diesem Grund wird das XML-Schema nochmal auf eine relationale Datenbank, die mit dem RDBMS Schema arbeitet, abgebildet.



# A framework for sharing product information across enterprises







# A framework for sharing product information across enterprises

## System application interoperability

*Unternehmen setzen Software (CORBA, RMI oder DCOM) ein, die in sich homogen sind, aber untereinander zu Problemen, aufgrund von verschiedenen Programmiersprachen, Plattformen und speziellen Netzwerk-Protokollen, führen können.*

- *Web-Services mit Hilfe des XML-basierenden Protokolls 'Simple Object Access Protocol (SOAP)'*  
*Dabei wird XML zur Repräsentation der Daten benutzt.*



# A framework for sharing product information across enterprises

## • Anfrage

```

<?xml version="1.0" encoding="UTF-8" ?>
- <SOAP-ENV:Envelope xmlns:SOAP-ENV="URN",
    xmlns:SOAP-ENC="URN",
    xmlns:xsi="URN", xmlns:xsd="URN">
- <SOAP-ENV:Body>
  - <GetProduct xmlns="Service URL"
    SOAP-ENV:encodingStyle="URN">
    <id xsi:type="xsd:string">20000</id>
  </GetProduct>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

- Ein SOAP-Envelope (ENV) ist ein Container, der einen optionalen Header enthält und ein Body-Element in dem die eigentlichen Informationen stehen.

## • Antwort

```

<?xml version="1.0" encoding="utf-8" ?>
- <soap:Envelope xmlns:soap="URN",
    xmlns:xsi="URN", xmlns:xsd="URN">
- <soap:Body>
- <GetProductResponse xmlns="Service URL">
- <GetProductResult>
...
...
- <product diffgr:id="product1" msdata:rowOrder="0"
  diffgr:hasChanges="inserted">
  <oid>7</oid>
  <id>20000</id>
  <name>Cab Assembly</name>
  <description>Cab Assembly</description>
</product>
- <product diffgr:id="product2" msdata:rowOrder="1"
  diffgr:hasChanges="inserted">
  <oid>8</oid>
  <id>21000</id>
  <name>Cab Interior</name>
  <description>Cab Interior</description>
</product>
...
...
</GetProductResult>
</GetProductResponse>
</soap:Body>
</soap:Envelope>

```



# XML in der Praxis

## WebEDI

Quelle: Anwendung der Extensible Markup Language (XML): Konzeption und Implementierung einer WebEDI-Lösung ( Peter Buxmann, Frank Ladner, Tim Weitzel )  
A communication architecture for the digital economy ( Tim Weitzel, Peter Buxmann )  
[ <http://www.wiwi.uni-frankfurt.de/~tweitzel/paper/architecture/index.htm> ]



# WebEDI

## Electronic Data Interchange

- Austausch von Geschäftsdaten
- Standardisierte Form
- Beschleunigung von Geschäftsprozessen
- Kosteneinsparung



# WebEDI

Damalige Standards:

- Branchenspezifisch
- Partnerspezifisch
- Landesspezifisch



## WebEDI

### EDI:

- Verbreitung eher gering
- Aufgrund der Vielzahl an Standards
- Inkompatibilität der Standards
- Unflexibel



# WebEDI

## Ziel

- Einfache Verteilung von
  - Rechnungen
  - Bestellungen
  - Produktbeschreibungen
  - uvm.
- Basierend auf offenen Web-Standards
- Weites Spektrum an Clients abdecken
- Einfaches Implementieren



# WebEDI

Beispiel anhand von Lufthansa AirPlus





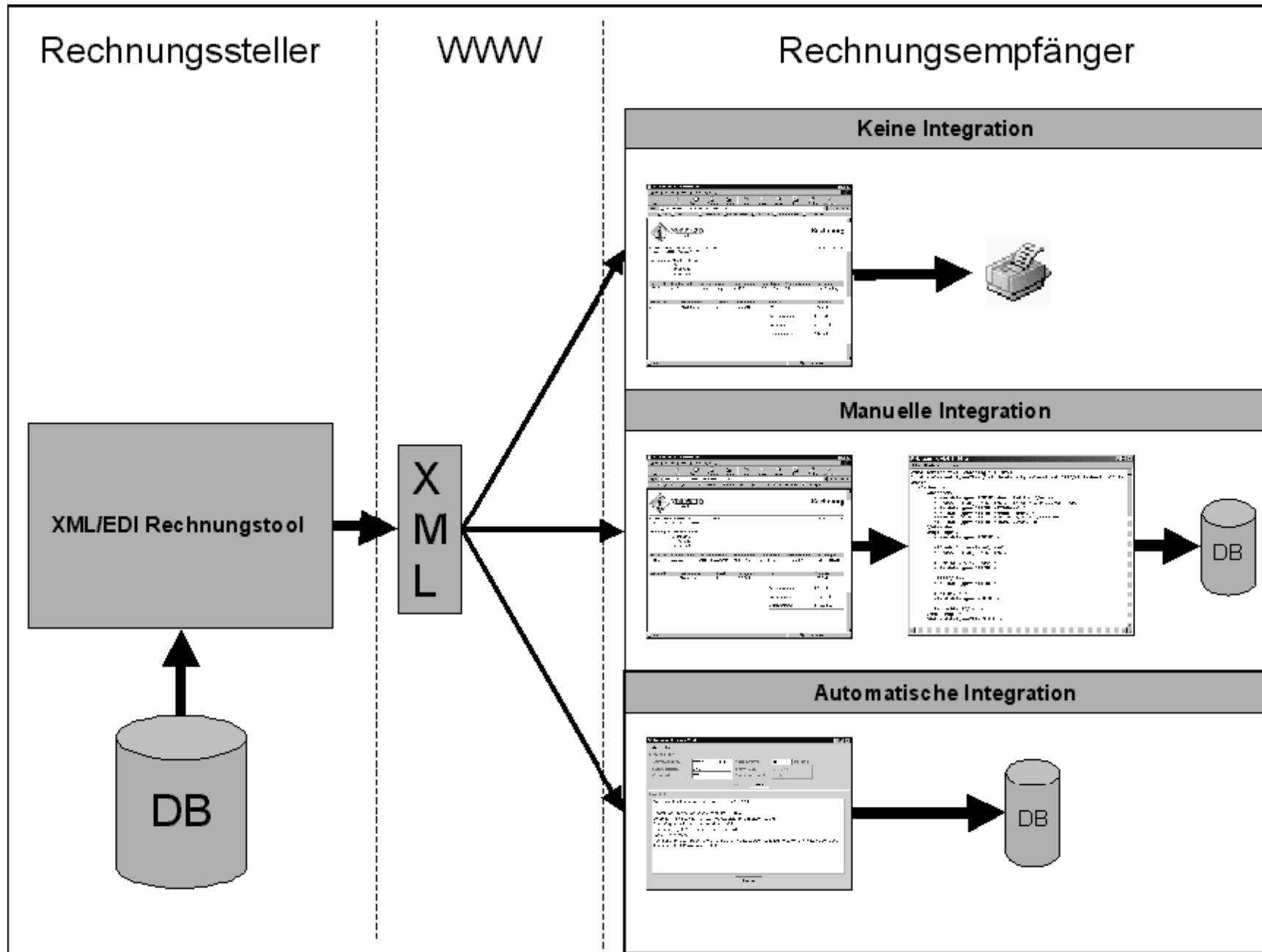


# WebEDI

- Viele Reisen von einzelnen Personen ( z.B. Geschäftsleute)
- Große Anzahl an Rechnungen
- Fehlende Transparenz bei vielen Rechnungen
- Abholen der Rechnungen anstatt schicken



# WebEDI



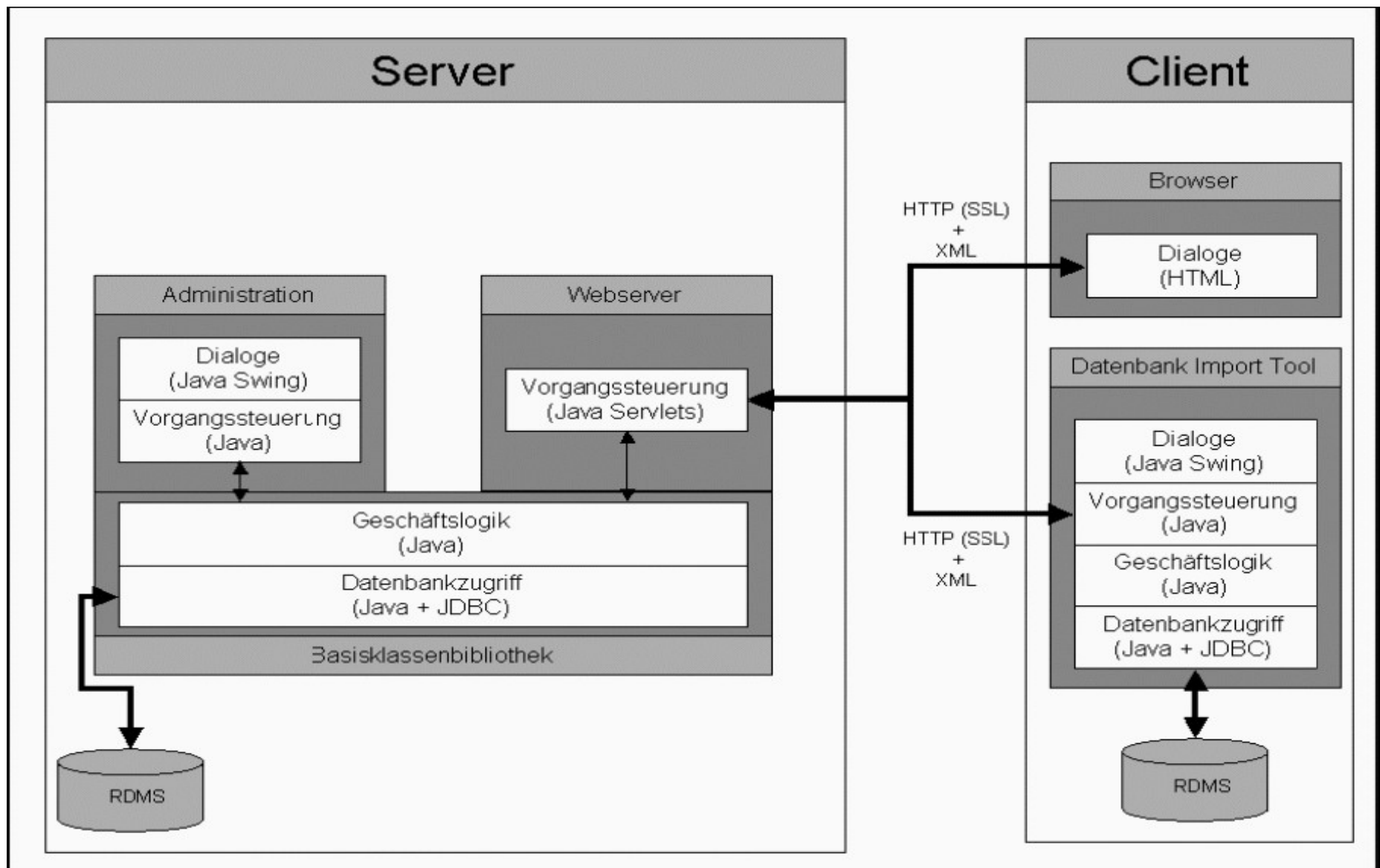


## WebEDI

- Keine Integration: Präsentation und Ausdrucken der XML-Rechnungen über den Webbrowser,
- Manuelle Integration: Herunterladen der XML-Rechnungen über einen Webbrowser und Integration in Inhouse-Systeme sowie
- Automatische Integration: Herunterladen der XML-Rechnungen über eine Java-Applikation mit automatischer Integration in Inhouse-Systeme.



# WebEDI





## WebEDI

### Server Seite

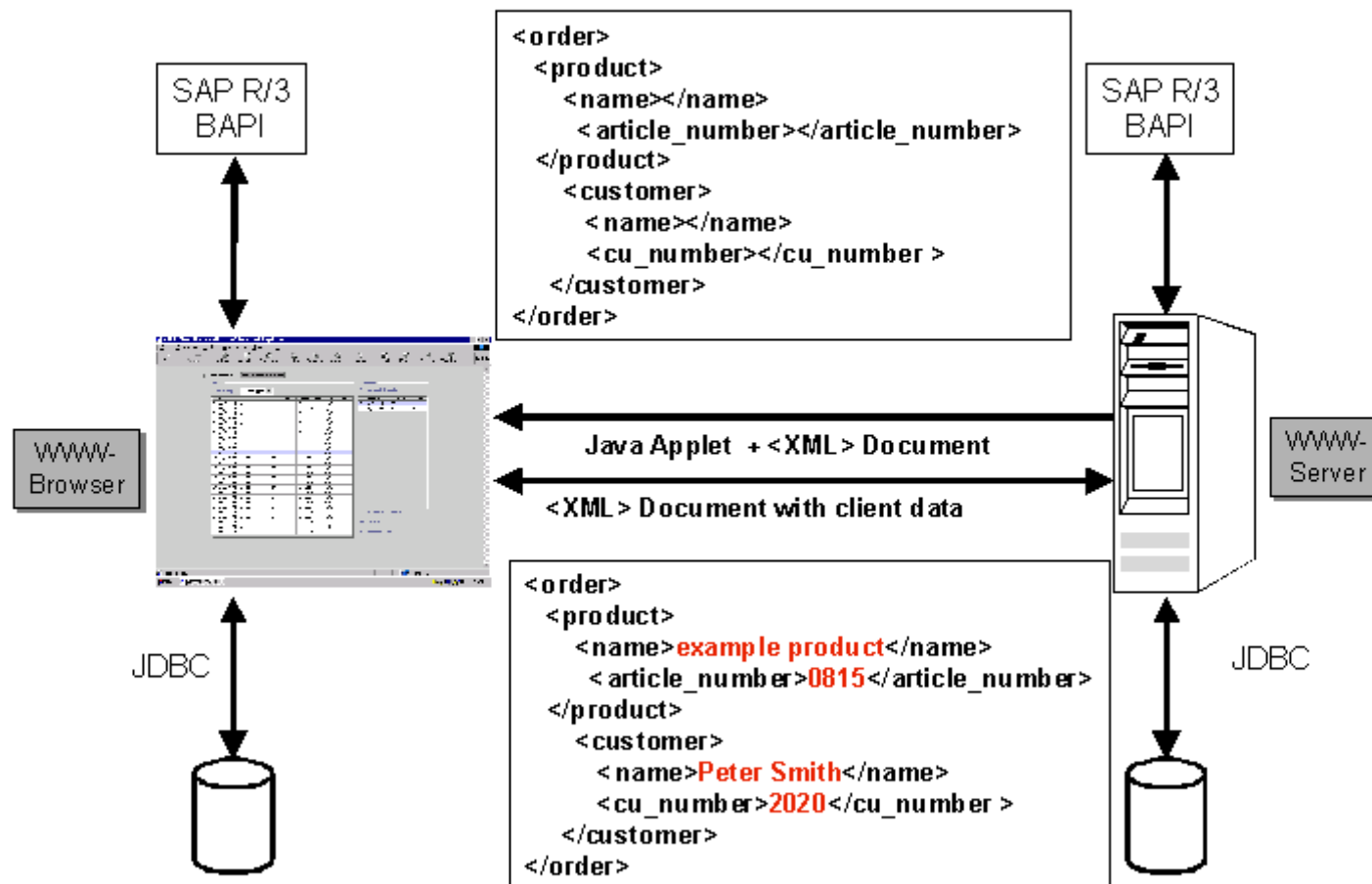
- Benutzerverwaltung
- Konfigurations Editor (Kundenspezifische XML-Struktur)
- Protokollierung
- Datenbank

### Client Seite

- Browser
- Eigenes Abrufprogramm
- Datenbankintegration



# WebEDI





# XML im speziellen Einsatz

## B2C : XML und mobile Agenten zur Angebotsrecherche

Quelle: Frank Siegemund, Clemens H. Cap, Andreas Heuer: Einsatz von mobilen Agenten und XML zur  
Angebotsrecherche im Business-to-Consumer-Commerce  
Verlag Vieweg. Referenz: <http://www.wirtschaftsinformatik.de> 1



## B2C : XML und mobile Agenten zur Angebotsrecherche

Was sind mobile Agenten und wie werden sie zur  
Angebotsrecherche eingesetzt ?

- Intelligente Software
- Migration zwischen Netzknotenpunkten
- Agieren auf Agentenplattformen
- Kommunikation mit anderen Agenten
- „Spionieren“ und Vergleichen im Interesse  
eines Kunden Angebote von Produktanbietern



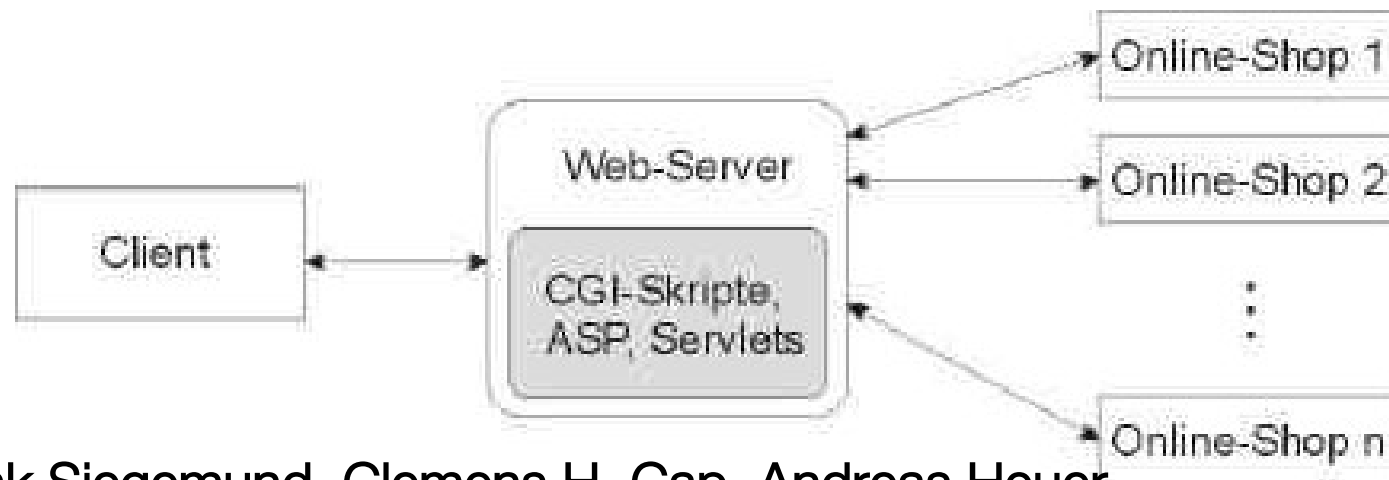


## B2C : XML und mobile Agenten zur Angebotsrecherche

- Verhandlungen mit Anbietern (hier: Online-Shops)
- Bestellung von Artikeln
- Kein Interesse von (großen) Anbietern
- Vorabfilterung von Informationen
- Agieren im Back-End (Front-End: Z.B. HTML Suchmaske)
- Häufig Unterstützung durch Datenbanken



# B2C : XML und mobile Agenten zur Angebotsrecherche

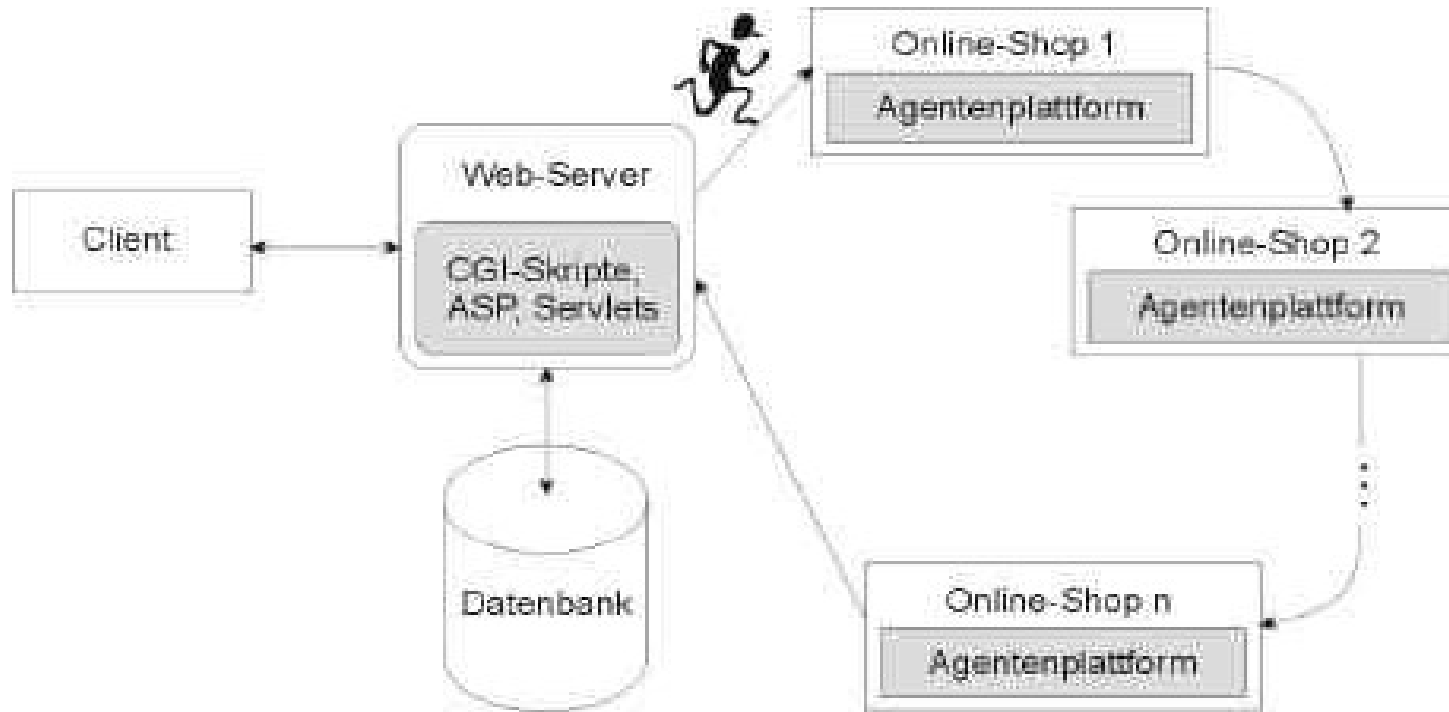


Frank Siegemund, Clemens H. Cap, Andreas Heuer

Recherche ohne mobile Agenten



# B2C : XML und mobile Agenten zur Angebotsrecherche



Frank Siegemund, Clemens H. Cap, Andreas Heuer

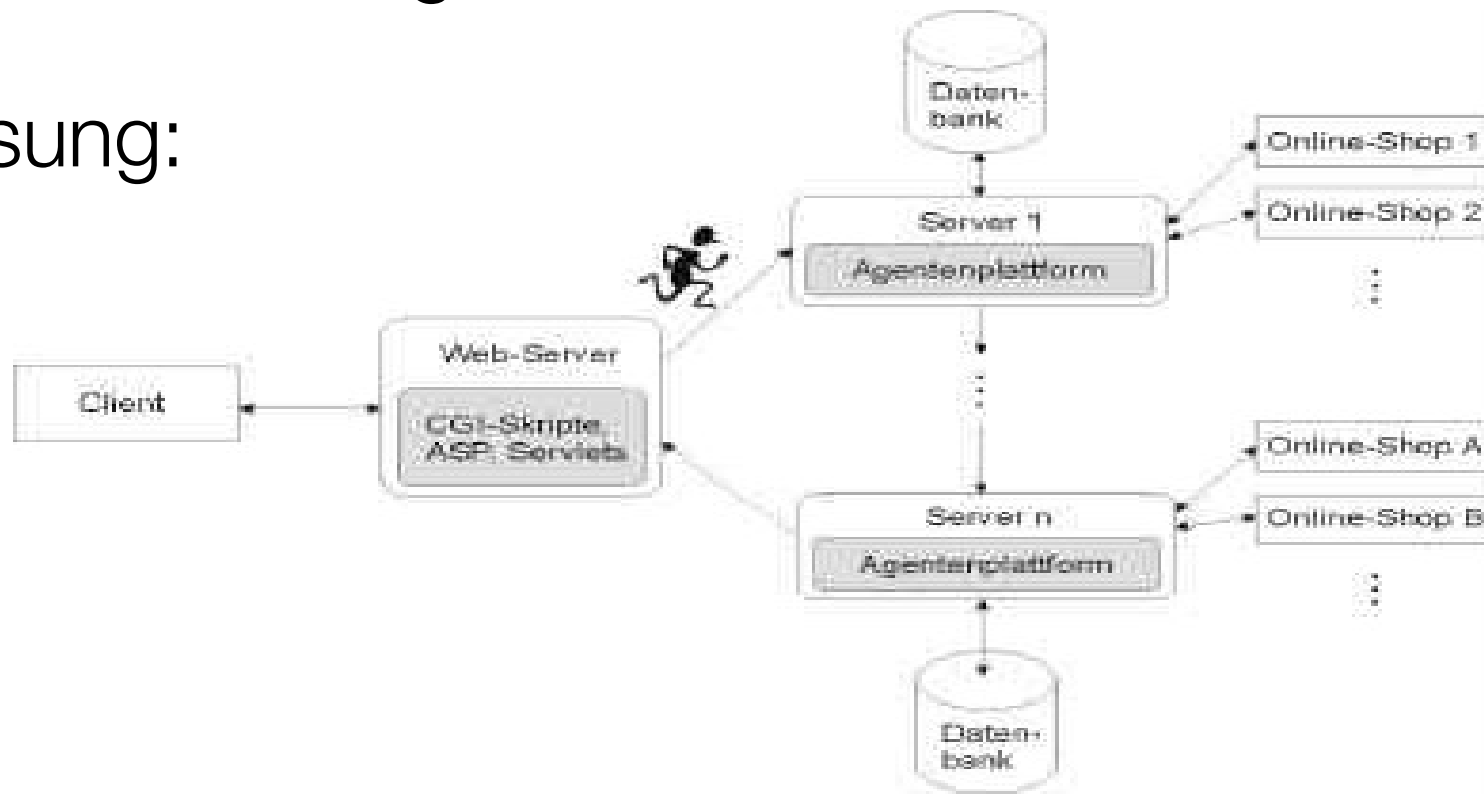
Mit mobilen Agenten und Datenbank

Notwendig: Anbieterunterstützung und  
regelmäßige Aktualisierung unabhängig von  
Anfragen



# B2C : XML und mobile Agenten zur Angebotsrecherche

Lösung:



Frank Siegemund, Clemens H. Cap, Andreas Heuer

Mit mobilen Agenten und Server mit  
Datenbanken



## B2C : XML und mobile Agenten zur Angebotsrecherche

Notwendig auch hier: Regelmäßige Aktualisierung  
unabhängig von Anfragen

Neue Probleme:

- Netzwerk von Recherche-Servern notwendig
- Mobile Agenten besser kein Datenbankzugriff  
auf entfernten Servern, da  
nichtvertrauenswürdig Code.

**Lösung:** Stationäre Agenten auf entf. Servern  
kommunizieren mit mobilen.

Gut: Flexibilität, Erweiterbarkeit



# B2C : XML und mobile Agenten zur Angebotsrecherche

Was hat das mit XML zu tun ?

- Kommunikation Agent-Agent bzw. Agent-Datenbank
- Online-Shops benutzen XML !

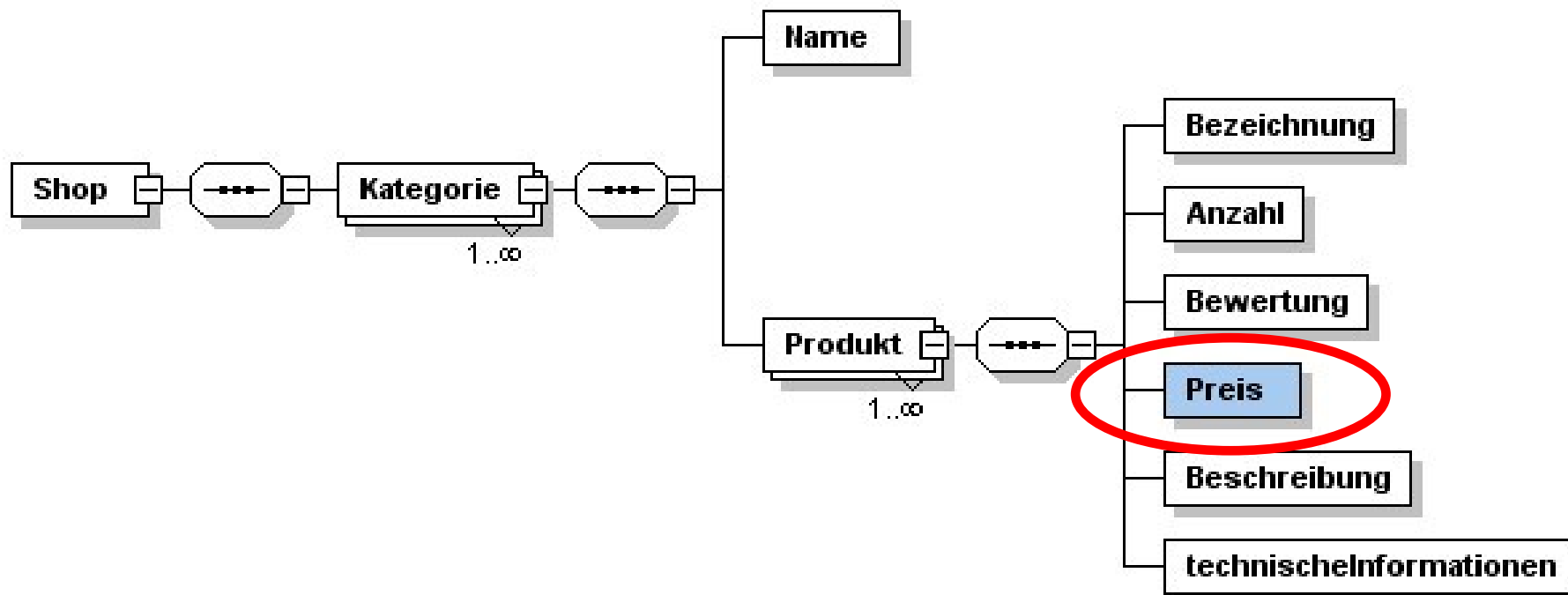
 Datenextraktion vereinfacht



# B2C : XML und mobile Agenten zur Angebotsrecherche

Wie können Daten aus XML Dokumenten sinnvoll extrahiert werden ?

- Einfach : Standardisiertes Schema (wie z.B. Docbook)

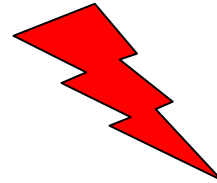




# B2C : XML und mobile Agenten zur Angebotsrecherche

Einfach  , aber nicht Praktikabel 

Grund : Steht im Widerspruch zu Interesse von Online-Shops



**Lösung:** Eigenständiges Finden der relevanten Informationen (hier: Preis zu Artikel)





# XML im speziellen Einsatz

**OASIS**  OpenDocument



## OpenDocument

### **OASIS:**

- nichtkommerzielles Konsortium
- Entwickler des DocBook

### **OpenDocument:**

- Standardisiertes Dateiformat gemäß ISO
- Nachfolger des OpenOffice.org Formates
- einheitliches, nicht-proprietäres Dokumentenformat



# OpenDocument - Aufbau

- JAR-Archiv, jedoch nicht mit \*.jar Endung
- enthält mehrere Dateien:
  - content.xml
  - current.xml
  - manifest.xml
  - mimetype
  - meta.xml
  - settings.xml
  - styles.xml



# OpenDocument - Aufbau

Archive:	testing.odt						
Length	Method	Size	Ratio	Date	Time	CRC-32	Name
-----	-----	-----	-----	-----	-----	-----	-----
39	Stored	39	0%	06-12-06	21:09	0c32c65e	<b>mimetype</b>
0	Stored	0	0%	06-12-06	21:09	00000000	Configurations2/statusbar/
0	Defl:N	2	0%	06-12-06	21:09	00000000	
Configurations2/accelerator/current.xml							
0	Stored	0	0%	06-12-06	21:09	00000000	Configurations2/floater/
0	Stored	0	0%	06-12-06	21:09	00000000	Configurations2/popupmenu/
0	Stored	0	0%	06-12-06	21:09	00000000	Configurations2/progressbar/
0	Stored	0	0%	06-12-06	21:09	00000000	Configurations2/menubar/
0	Stored	0	0%	06-12-06	21:09	00000000	Configurations2/toolbar/
0	Stored	0	0%	06-12-06	21:09	00000000	Configurations2/images/Bitmaps/
2466	Defl:N	660	73%	06-12-06	21:09	1e5e7610	<b>content.xml</b>
8004	Defl:N	1622	80%	06-12-06	21:09	c9d75e99	<b>styles.xml</b>
1047	Stored	1047	0%	06-12-06	21:09	0db9c8d8	<b>meta.xml</b>
650	Defl:N	341	48%	06-12-06	21:09	e2989818	Thumbnails/thumbnail.png
6926	Defl:N	1109	84%	06-12-06	21:09	f95b0b9f	<b>settings.xml</b>
1866	Defl:N	318	83%	06-12-06	21:09	39d76235	META-INF/ <b>manifest.xml</b>
-----	-----	-----	-----	-----	-----	-----	-----
20998		5138	76%				15 files



# OpenDocument - mimetype

- besteht aus nur einer einzigen Zeile
- legt den MIME-Type für das Dokument fest
- ist **nicht komprimiert**
- dadurch Dateityp **ohne** Dekomprimierungsprogramm ermittelbar



# OpenDocument - mimetype

Document Type	MIME Type	Endung
Text document	application/vnd.oasis.opendocument.text	odt
Graphics document (Drawing)	application/vnd.oasis.opendocument.graphics	odg
Presentation document	application/vnd.oasis.opendocument.presentation	odp
Spreadsheet document	application/vnd.oasis.opendocument.spreadsheet	ods
Chart document	application/vnd.oasis.opendocument.chart	odc
Image document	application/vnd.oasis.opendocument.image	odi
Formula document	application/vnd.oasis.opendocument.formula	odf
Global Text document	application/vnd.oasis.opendocument.text-master	odm

**Quelle:** OASIS OpenDocument Essentials



# OpenDocument - manifest.xml

- speichert die Struktur des Dateiarchivs
- legt Wurzel und nochmals MIME Type fest

## Dokumentenkopf:

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest:manifest
xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0">
  <manifest:file-entry manifest:media-
type="application/vnd.oasis.opendocument.text" manifest:full-path="/" />
```

## Auszug:

```
<manifest:file-entry manifest:media-type="text/xml" manifest:full-
path="content.xml" />
```



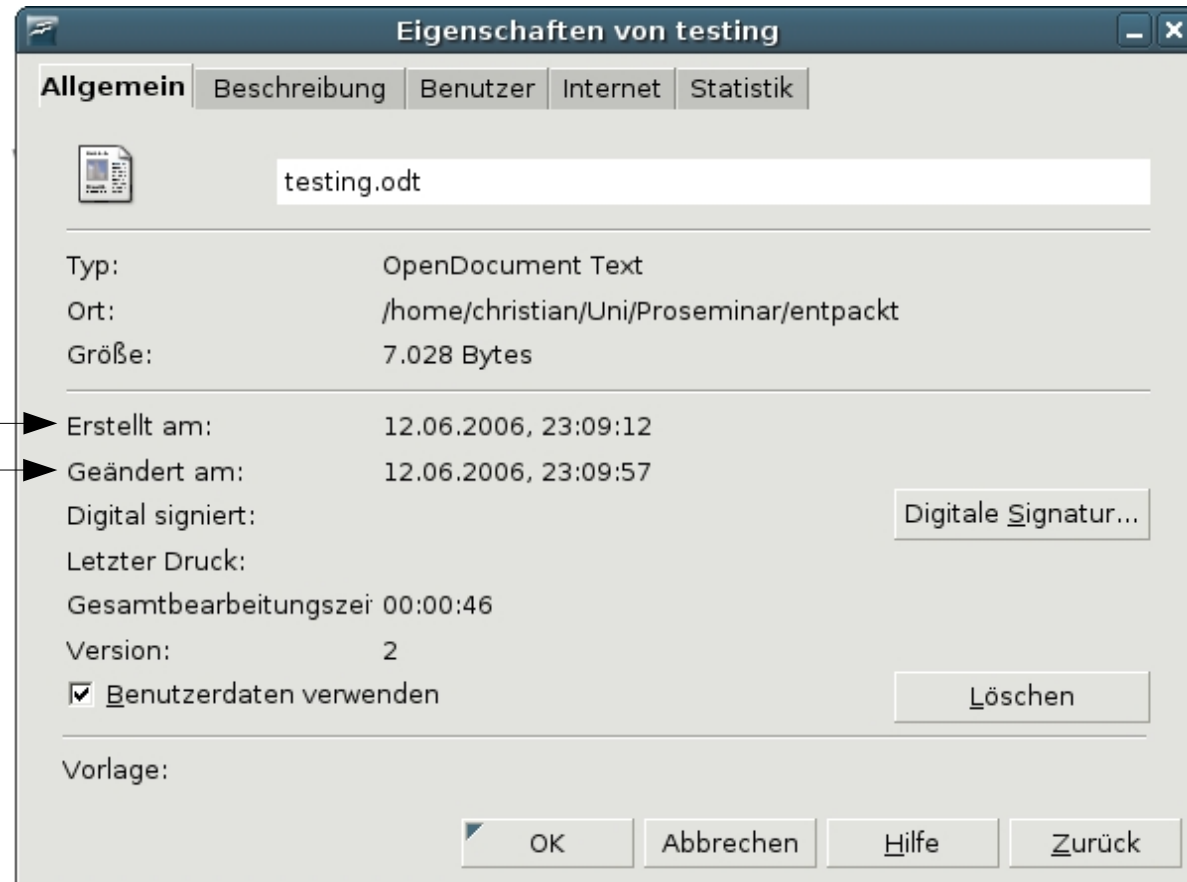
# OpenDocument - meta.xml

- enthält Daten über das Dokument selbst (Autor etc.)
- enthält unter anderem „Dublin Core“ Elemente





# OpenDocument - meta.xml



```
<meta:creation-date>2006-06-12T23:09:12</meta:creation-date>
```

```
<dc:date>2006-06-12T23:09:57</dc:date>
```

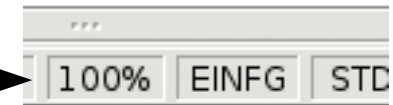


# OpenDocument - settings.xml

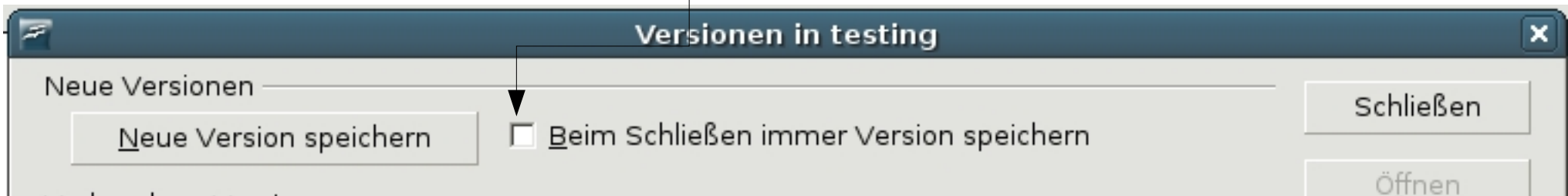
- speichert Einstellungen der Anwendung, mit der die Datei bearbeitet wurde

Beispiele:

```
<config:config-item config:name="ZoomFactor"  
config:type="short">100</config:config-item>
```



```
<config:config-item config:name="SaveVersionOnClose"  
config:type="boolean">false</config:config-item>
```

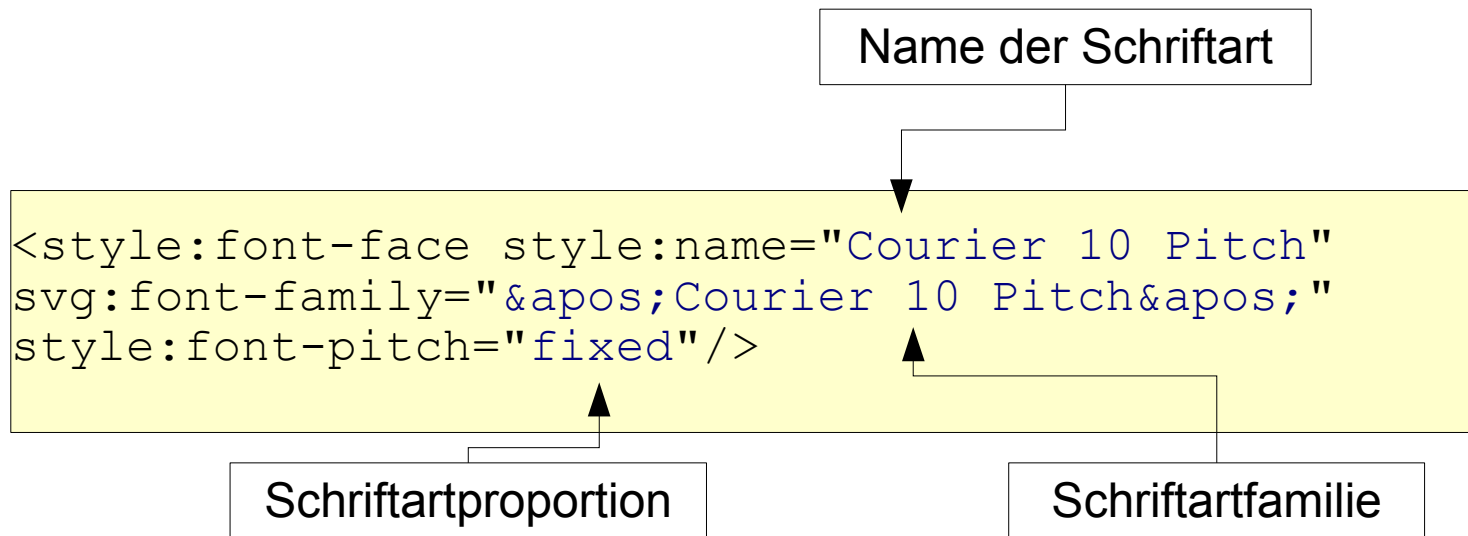




# OpenDocument - styles.xml

- enthält sämtliche Informationen über Formatierungen, die im Dokument benutzt werden und wurden

Beispiel:





## OpenDocument - content.xml

### **enthält:**

- eigentlichen Inhalt des Dokuments
- Stilvorlagen
- Infos über Dokumenttyp



# Aufbau - content.xml

## Kopf

- Namespace-deklarationen
- Version

## Script-Bereich

- optional, trotzdem immer enthalten
- enthält ggf. Makros

## Styles-Bereich

- enthält Informationen über die Stilvorlagen

## Body

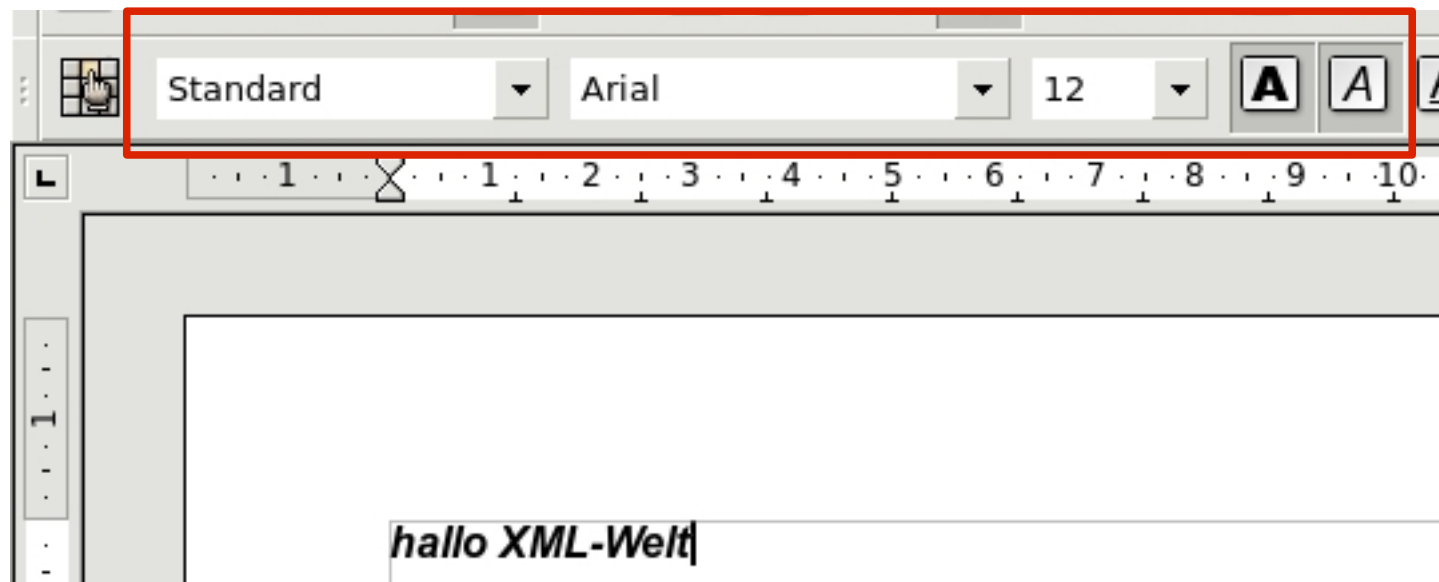
- Info über Dokumenttyp
- Dokumenteninformation



# OpenDocument - content.xml

Beispiel:

```
<text:p text:style-name="P1">hallo XML-Welt</text:p>
```



```
<style:style style:name="P1" style:family="paragraph" style:parent-style-name="Standard">  
  <style:text-properties style:font-name="Arial" fo:font-style="italic" fo:font-weight="bold" style:font-style-asian="italic" style:font-weight-asian="bold" style:font-style-complex="italic" style:font-weight-complex="bold" />  
</style:style>
```



# OpenDocument - Quellen

- OASIS OpenDocument Essentials:  
2005 - J. David Eisenberg – O'Reilly
- [www.wikipedia.org](http://www.wikipedia.org)
- [www.openoffice.org](http://www.openoffice.org)



## XML in der Praxis



Vielen Dank fürs Zuhören und viel Spaß beim  
Spiel!

