

Quo vadis Web- Anwendungsentwicklung? XML based Webdevelopment

Thomas Bayer
Steffen Schluff
Tobias Kieninger

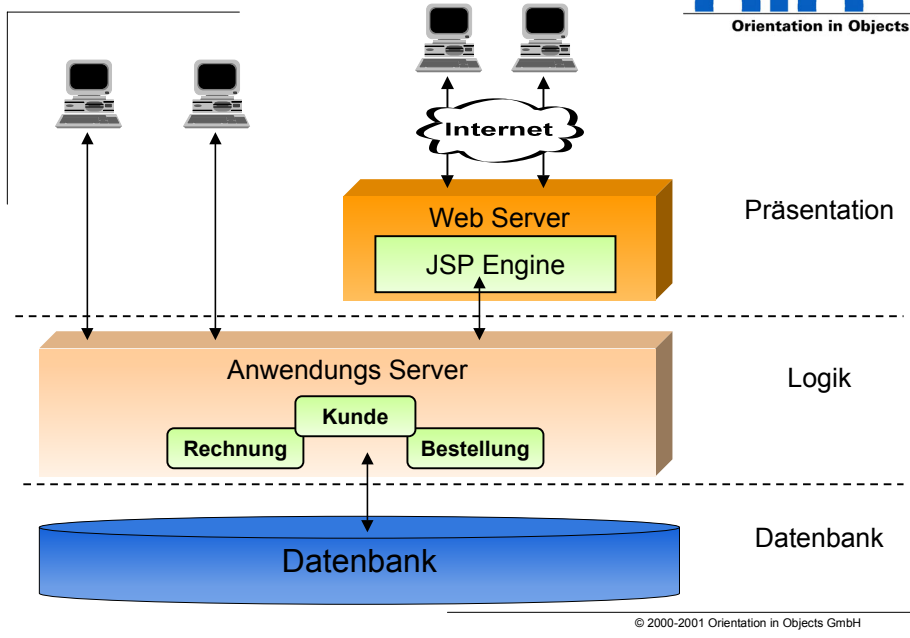
bayer@oio.de

Orientation in Objects GmbH
Weinheimer Str. 68
68309 Mannheim
www.oio.de

Inhalt

- **3 Schichten Architektur**
- XML based Webpublishing
- Web Publishing Frameworks
- Pipeline Model
- Dynamischer Inhalt
- XML Anwendungsarchitekturen
- Integration einer XSP-Anwendung in ein B2B-System

3 Schichten Architektur



3 Schichten Architektur: Vorteile

- Trennung von Daten, Logik und Präsentation
- Über Anwendungen hinaus verwertbare Geschäftsobjekte
- Flexibel für Änderungen

3 Schichten Architektur: Wirklichkeit



- Standardisierte Infrastruktur erfordert sehr viel Einarbeitung
- Im Client steht sehr viel Code (oft mehr als 60% der Anwendung)
- Trennung von Aufgaben im Team nur schwer möglich
- Rapid Prototyping nur eingeschränkt möglich

Business Logik



- Pflege
- Datenabfrage
- Berechtigungen
- Kalkulationen
- Statusänderungen
- Prüfungen, Randbedingungen
- Prüfen von Zeiträumen und -Punkten
- ...

Client Logik



- Validieren der Eingaben
- Zeigen und Erklären der Eingabefehler
- Aufbereitung der Fehlermeldungen
- Sortieren, Gruppieren, Summieren und Aggregieren
- Formatieren
- Kommunikation mit Server
- Steuerung Anwendungsfluß

- Webclient (zusätzlich)
 - Verwaltung von Sessions
 - Pageing

Inhalt



- 3 Schichten Architektur
- XML based Webpublishing
- Web Publishing Frameworks
- Pipeline Model
- Dynamischer Inhalt
- XML Anwendungsarchitekturen
- Integration einer XSP-Anwendung in ein B2B-System

Probleme

Webpublishing

- Starres Layout
- Großer Änderungsaufwand
- Aufwand im Contentmanagement

Webentwicklung

- Unübersichtliche ServerPage Anwendungen
- Problematische Trennung der Kompetenzen und Aufgaben
- Starres Layout
- Sehr viel Aufwand für „Client Logik“

XML based Webpublishing

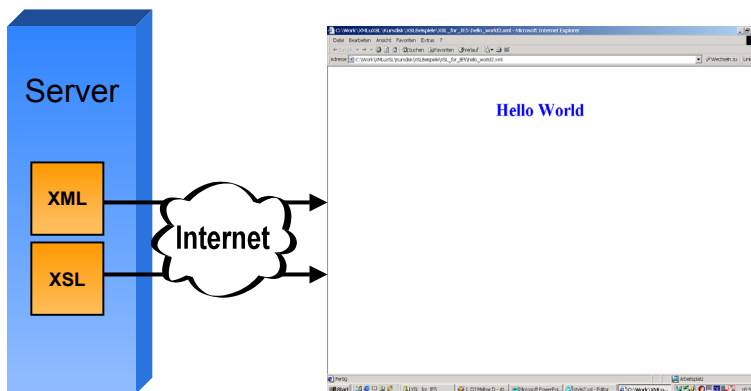
„if your troubles are site management, if your graphics people are always in the way, if your HTML authors always mess up your page logic, if your managers see no results in hiring new people to work on the site - go on and make your life easier!“

Zitat aus der Cocoon Dokumentation

XML based Webpublishing

- Seiten werden aus XML Quellen erzeugt
- Daten sind strukturiert
- Inhalt ist von der Präsentation getrennt
- Crossbrowser fähig

XSL im Browser



XSL im Browser

Vorteile:

- Trennung von Content und Style
- Suchmaschinen könnten XML Daten auswerten
- XML Daten können von Fremdsystemen eingelesen und verwertet werden

Nachteile:

- Menge der übertragenen Informationen
- Browser muss XSL unterstützen

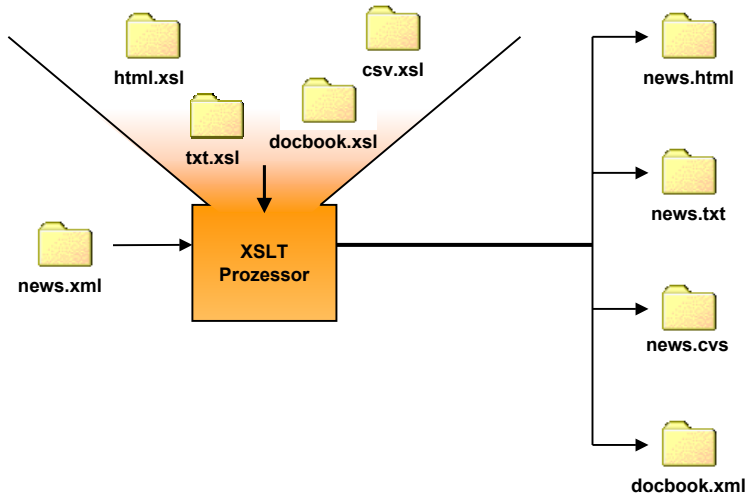
Serverside XML

- Server wandelt XML in HTML, WML,... um
- Aufbereitung für verschiedene Clients ist möglich
 - IE, Netscape, WAP-Client, ...
- Keine speziellen Anforderungen an den Browser
 - Java, Neue Version, ...

XSLT Prozessor



Orientation in Objects



© 2000-2001 Orientation in Objects GmbH

XML generierte Webpage



Orientation in Objects

Orientation in Objects GmbH - Microsoft Internet Explorer

Adresse <http://www.oio.de/>

Orientation in Objects

Home
[Schulung](#)
[Beratung](#)
[Projekte](#)
[Jobs](#)
[Bookmarks](#)
[Story](#)
[Sitemap](#)
[Kontakt](#)

info@oio.de

Wir sind Dienstleister im Bereich der objektorientierten Softwareentwicklung.

Zu unserem Angebot gehören:

Schulungen

Machen Sie sich fit in den Bereichen [Java](#), [Objektorientierung](#), [XML](#), und [Enterprise Application Integration](#)

Beratung

Wir beraten im Bereich der Softwareentwicklung von der Analyse bis zum Refactoring. [mehr...](#)

Projekte

Mit Java, CORBA und XML führen wir Projekte für unsere Kunden durch oder unterstützen diese bei der Realisierung. [mehr...](#)

News

Katalog auf der CeBIT

Die Firma Outpark präsentiert auf der CeBIT (Halle 5 / A.56) eine c-procurement-Software mit SAP-Anbindung, in die der OIO-Katalog integriert wurde. [mehr...](#)

Beiträge auf der JAX2001

Auf der JAX 2001 in Frankfurt halten wir Vorträge und Workshops zu den Themen XML, XSLT und Cocoon. [mehr...](#)

CeBIT HANNOVER

jax2001 Konferenz für Java, Apache, XML

© 2000-2001 Orientation in Objects GmbH

Generierung im Batch

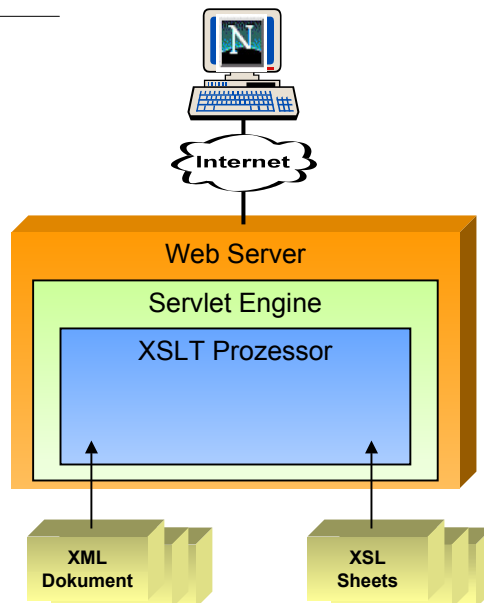
Vorteile:

- Schnelle Bearbeitung von Abrufen
- Konventioneller Web Server ist ausreichend
- Robuster im Betrieb

Nachteile:

- Keine dynamische Seitengenerierung

Idee: XSLT im Servlet



Inhalt

- 3 Schichten Architektur
- XML based Webpublishing
- **Web Publishing Frameworks**
- Pipeline Model
- Dynamischer Inhalt
- XML Anwendungsarchitekturen
- Integration einer XSP-Anwendung in ein B2B-System

Cocoon

- Apache Cocoon is a 100% pure Java publishing framework that relies on new W3C technologies (such as XML, XSL, SVG, etc..) to provide web content.

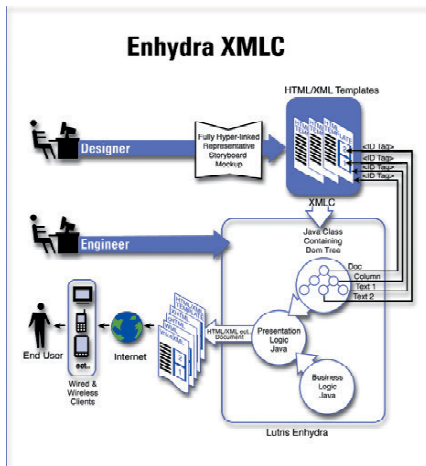
Quelle: Cocoon 2 README

A small revolution

Stefano Mazzocchi (Autor und Architekt von Cocoon)



Enhydra XMLC



(Quelle: www.lutris.com)

- HTML / XML Template mit speziellen Tags erzeugen
- XMLC compiliert Template in Java Klasse
- Methoden werden erzeugt, um die speziellen Tags zu verändern
- Java kann dynamische Inhalte einfügen
- Ansatz quasi Servlet-Templates, anders als Cocoon oder AxKit
- Download: www.enhydra.org

Enhydra XML - Beispiel

```
<HTML>
  <HEAD>
    <TITLE id="title">XMLC Beispiel</TITLE>
  </HEAD>

  <BODY>
    <H1>Hello World</H1>
    <SPAN id="para1">Innerhalb von SPAN-Tags kann
      geändert werden</SPAN>
```

Text außerhalb von SPAN-Tags oder außerhalb von Tags mit einem ID-Attribut kann nicht verändert werden.

```
</BODY>
```

- Modul für Apache Webserver
- Möglichkeiten zur XML-Transformation
 - XML wird mit XSLT über Pipelines transformiert
 - Perl ähnliche Skriptsprache XPathScript (XPS)
 - XSP für Perl
- Perl-DBI basierte SQL-Taglib
- Hohe Effizienz
- Festlegen von Pipelines entweder über `<?xml-stylesheet?>` Anweisungen oder Einträge in der Apache-Konfiguration
- Download: www.axkit.org

AxKit - Beispiel mit XPS

Auszug aus XML-Datei:

```
<buch>
  <autor>
    <name>James Joyce</name>
    ...
```

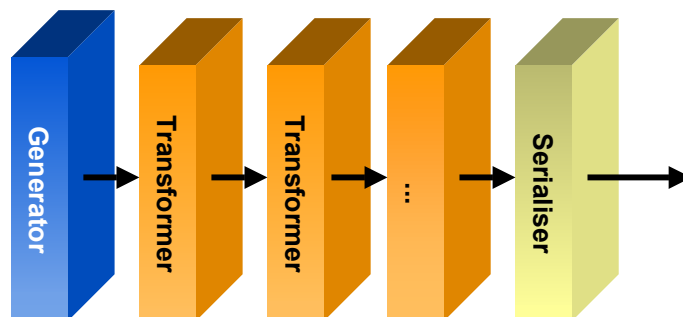
XPS-Datei:

```
<html>
  <head><title>XPS-Beispiel</title></head>
  <body>
    Der Autor ist:
    <%=findvalue("/buch/autor/name/text()") %>
  </body>
</html>
```

Inhalt

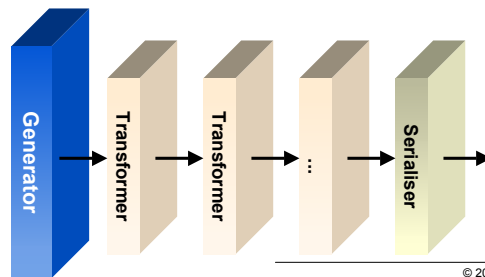
- 3 Schichten Architektur
- XML based Webpublishing
- Web Publishing Frameworks
- **Pipeline Model**
- Dynamischer Inhalt
- XML Anwendungsarchitekturen
- Integration einer XSP-Anwendung in ein B2B-System

Pipeline Model



Standard Generatoren

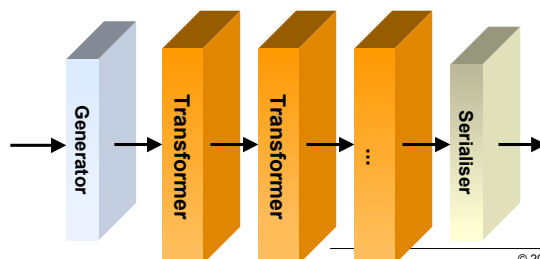
Generator	Beschreibung
DirectoryGenerator	Erzeugt XML Verzeichnisliste
FileGenerator	Liebt XML File ein und parst es
HTMLGenerator	Liebt HTML File ein und generiert daraus SAX Ereignisse im XHTML Format
ImageDirectoryGenerator	Um Bilder erweiterter DirectoryGenerator
PhpGenerator	Liebt von PHP ein über PHP Servlet Schnittstelle
ServerPagesGenerator	Liebt XML File und interpretiert Inhalt als eXtensible Server Page
StatusGenerator	



© 2000-2001 Orientation in Objects GmbH

Standard Transformer

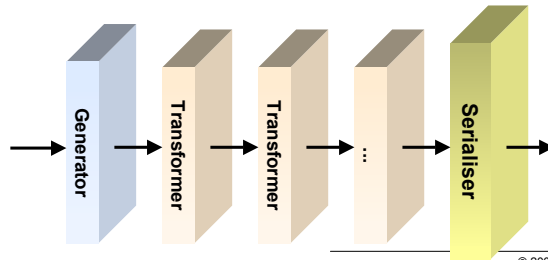
Transformer	Zweck
LogTransformer	Kann in eine Pipeline eingeklinkt werden. Protokolliert die SAX Ereignisse
SQLTransformer	Für SQL Abfragen
XalanTransformer	Für XSLT Transformationen
XTTransformer	Für XSLT Transformationen (XT)
XIncludeTransformer	Einbetten von XML Dokumenten



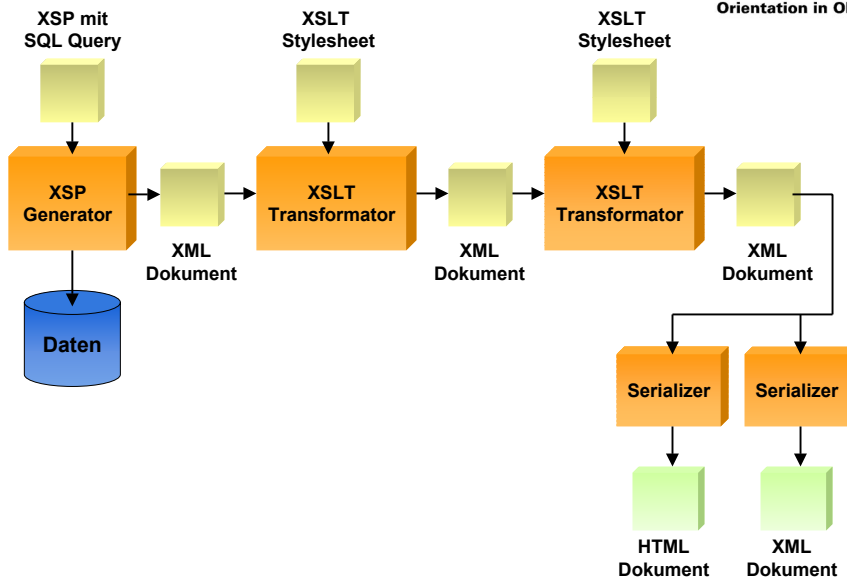
© 2000-2001 Orientation in Objects GmbH

Standard Serializer

Serializer	Erzeugt
HTMLSerializer	HTML Dokumente
XMLSerializer	XML Dokumente
FOPSerializer	PDF Dokumente
LinkSerializer	Zeigt die Ziele der Links
SVGSerializer	SVG Grafik
TextSerializer	Textfile



Ablauf



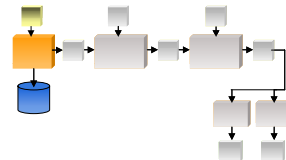
Sitemap (Cocoon 2)

Definiert die Pipeline, die zu einem Seitenaufruf gehört

```
<map:match pattern="test/*.xsp">
  <map:generate type="serverpages"
    src="docs/samples/test/{1}.xsp"/>
  <map:transform src="stylesheets/test/semtable-html.xsl"/>
  <map:transform src="stylesheets/dynamic-page2html.xsl"/>
  <map:serialize type="html"/>
</map:match>
```

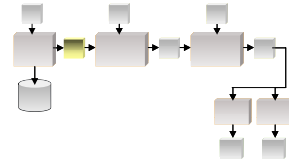
XSP mit SQL Query

```
...
<page><content>
<seminare>
  <esql:connection>
  <esql:pool>seminar</esql:pool>
  <esql:execute-query>
    <esql:query>select id, name from seminar</esql:query>
    <esql:results>
      <esql:row-results>
        <seminar>
          <id><esql:get-string column="id"/></id>
          <name><esql:get-string column="name"/></name>
        </seminar>
      </esql:row-results>
    </esql:results>
  </esql:execute-query>
</esql:connection>
</seminare>
</content></page>
...
```



XML Output der XSP

```
<?xml version="1.0" encoding="UTF-8" ?>
<page>
  <content>
    <seminare>
      <seminar>
        <id>X.EINF</id>
        <name>XML Einführung</name>
      </seminar>
      <seminar>
        <id>X.JAVA</id>
        <name>XML mit Java</name>
      </seminar>
    </seminare>
  </content>
</page>
```



XSLT umwandeln in HTML Tabelle

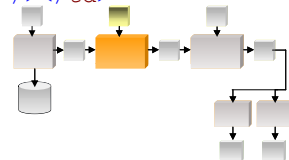
```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="seminare">
    <table>
      <xsl:apply-templates/>
    </table>
  </xsl:template>

  <xsl:template match="seminar">
    <tr>
      <td><xsl:value-of select="name"/></td>
    </tr>
  </xsl:template>

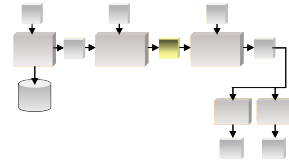
  ...

</xsl:stylesheet>
```



XML Output mit Tabelle

```
<page>
  <content>
    <table>
      <tr>
        <td>XML Einführung</td>
      </tr>
      <tr>
        <td>XML mit Java</td>
      </tr>
    </table>
  </content>
</page>
```



XSLT zur HTML Generierung

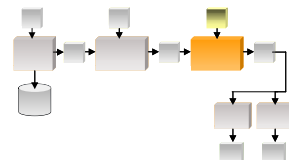
```
<?xml version="1.0"?>

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="page">
    <html>
      <body>
        <xsl:apply-templates select="content"/>
        <br/>
        generated by Cocoon
      </body>
    </html>
  </xsl:template>

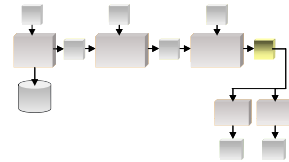
  ...

</xsl:stylesheet>
```



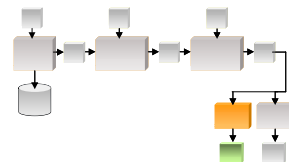
XML Output

```
<html>
  <body>
    <content>
      <table>
        <tr>
          <td>XML Einf&uuml;hrung</td>
        </tr>
        <tr>
          <td>XML mit Java</td>
        </tr>
      </table>
    </content>
    <br/>
    generated by Cocoon
  </body>
</html>
```

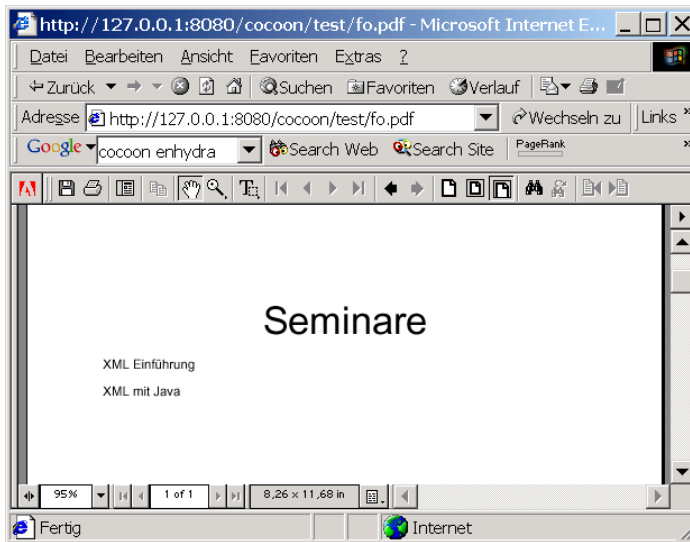


HTML Output

```
<html>
  <body>
    <content>
      <table>
        <tr>
          <td>XML Einf&uuml;hrung</td>
        </tr>
        <tr>
          <td>XML mit Java</td>
        </tr>
      </table>
    </content>
    <br/>
    generated by Cocoon
  </body>
</html>
```



XSL:FO



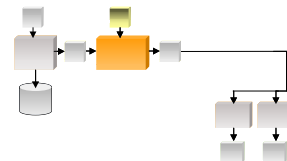
Stylesheet für XSL:FO

```
<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <!-- Hier Seitenformat festlegen (fehlt!) -->

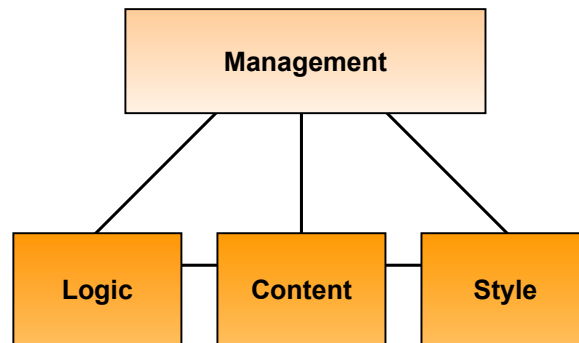
    <fo:page-sequence master-name="all">
      <fo:static-content flow-name="xsl-region-after"/>
      <fo:flow flow-name="xsl-region-body">
        <fo:block font-size="36pt" space-before.optimum="24pt"
          text-align="center">Seminare</fo:block>
        <xsl:apply-templates/>
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>

<xsl:template match="seminar">
  <fo:block font-size="12pt" space-before.optimum="12pt" text-align="left">
    <xsl:value-of select="name"/></fo:block>
</xsl:template>
```

...



Pyramid Model of Web Contracts



Quelle: Cocoon Dokumentation

Inhalt

- 3 Schichten Architektur
- XML based Webpublishing
- Web Publishing Frameworks
- Pipeline Model
- **Dynamischer Inhalt**
- XML Anwendungsarchitekturen
- Integration einer XSP-Anwendung in ein B2B-System

Server Pages

```
<jsp:useBean id="kontakt" class="Kontakt"/>

<html>
  <body>
    <h1>Bestätigung!</h1>
    <p>Sehr geehrter
      <%=kontakt.getTitel()%> <%=kontakt.getName()%>,
      wir werden Ihnen in den nächsten Tagen Ihre
      Bestellung liefern.
    </p>
  </body>
</html>
```

eXtensible Server Pages

- XSP wird mit XSLT in Source Code umgewandelt
- Der Source Code wird kompiliert
- Der neu erzeugte Code wird geladen
- Der neu erzeugte Code wird ausgeführt
- Cocoon, AxKit

```
<help:useBean id="kontakt" class="beanClass" />
<page>
  <content>
    <title>Bestätigung!</title>
    <para>Sehr geehrter
      <xsp:expr>kontakt.getTitel()</xsp:expr>
      <xsp:expr>kontakt.getName()</xsp:expr>,
      wir werden Ihnen in den nächsten Tagen Ihre
      Bestellung liefern.</para>
    </content>
  </page>
```

eXtensible Server Pages: Taglibs



- ESQL
- MAIL
- REQUEST
- RESPONSE
- LOG
- SESSION
- XSP
- UTIL

eXtensible Server Pages: Taglibs



3rd Party Taglibs

Name	Zweck
DB Prism	Emuliert Oracle Web Application Server (für Cartridges)
EJB	
LDAP	
Auth	Authentifizierung und Passwortabfrage
SOAP	
IMAP	Mailversand über IMAP Protokoll

XML based Webdevelopment



Orientation in Objects

Vorteile

- Trennung von Daten, Logik und Style
- Multichannelfähigkeit
- Gute Aufteilung der Arbeiten im Team
- Kurze Entwicklungszeiten
- Flexibilität

Nachteile

- Overhead
- Mehr Komponenten
 - Verwaltung, Management
- Geschäftslogik ist nicht in Geschäftsobjekten gekapselt

© 2000-2001 Orientation in Objects GmbH

Inhalt

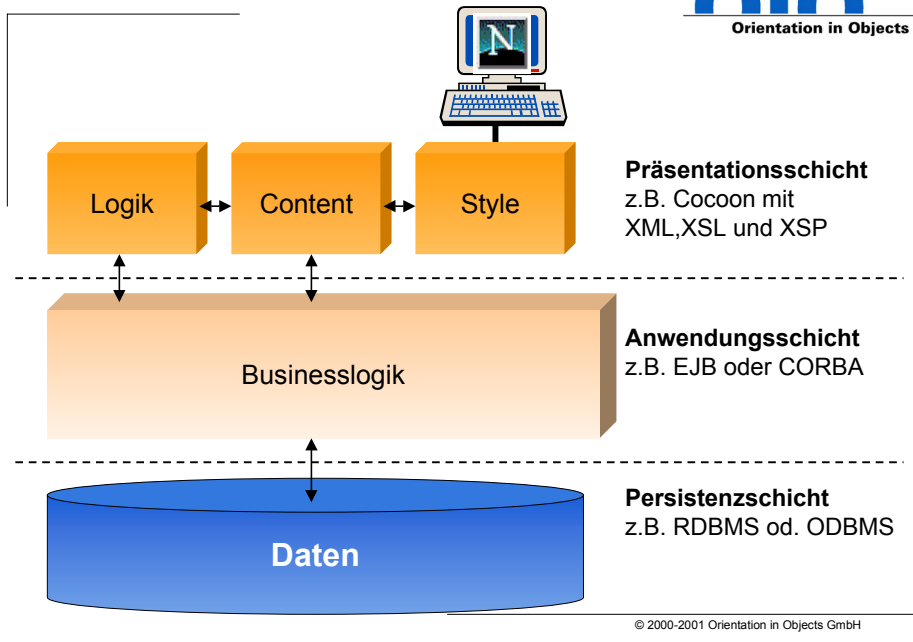


Orientation in Objects

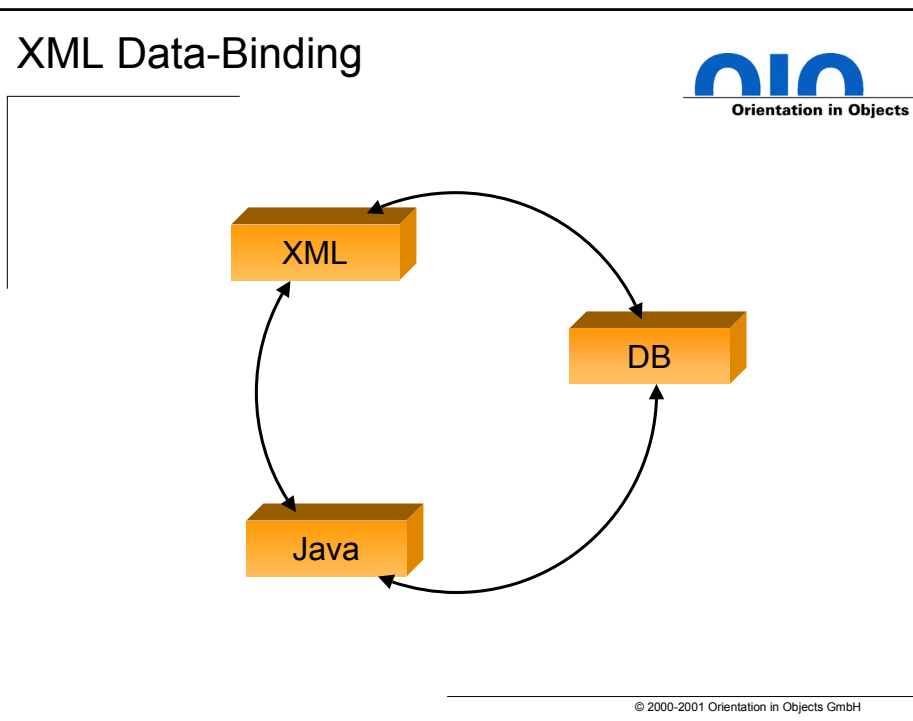
- 3 Schichten Architektur
- XML based Webpublishing
- Web Publishing Frameworks
- Pipeline Model
- Dynamischer Inhalt
- XML Anwendungsarchitekturen
- Integration einer XSP-Anwendung in ein B2B-System

© 2000-2001 Orientation in Objects GmbH

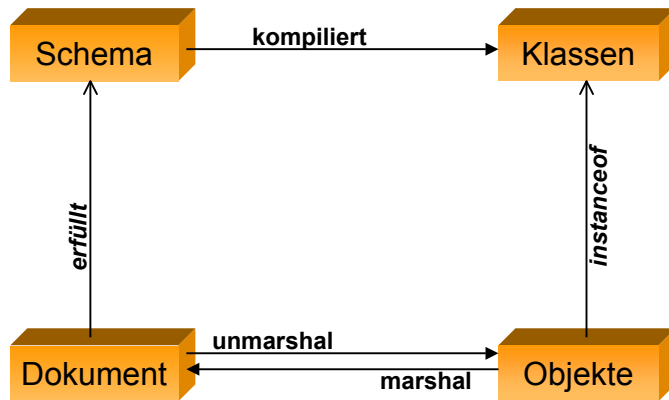
Erweiterte Drei-Schichten-Architektur



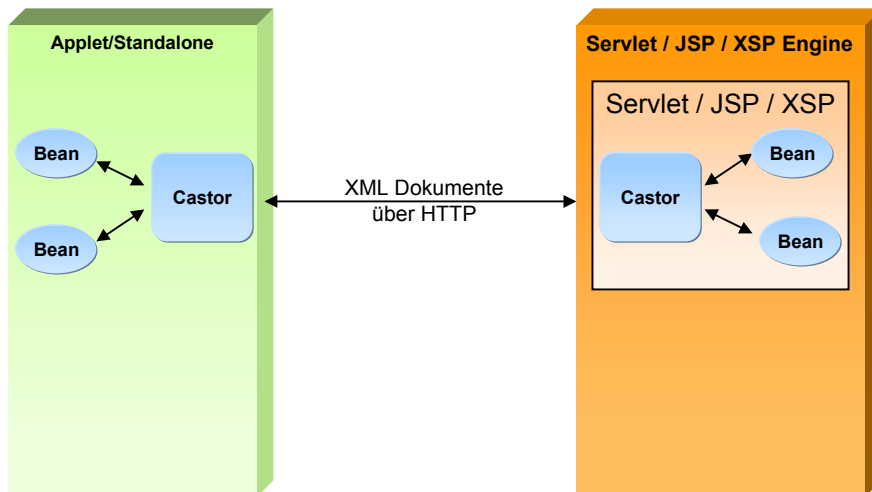
XML Data-Binding



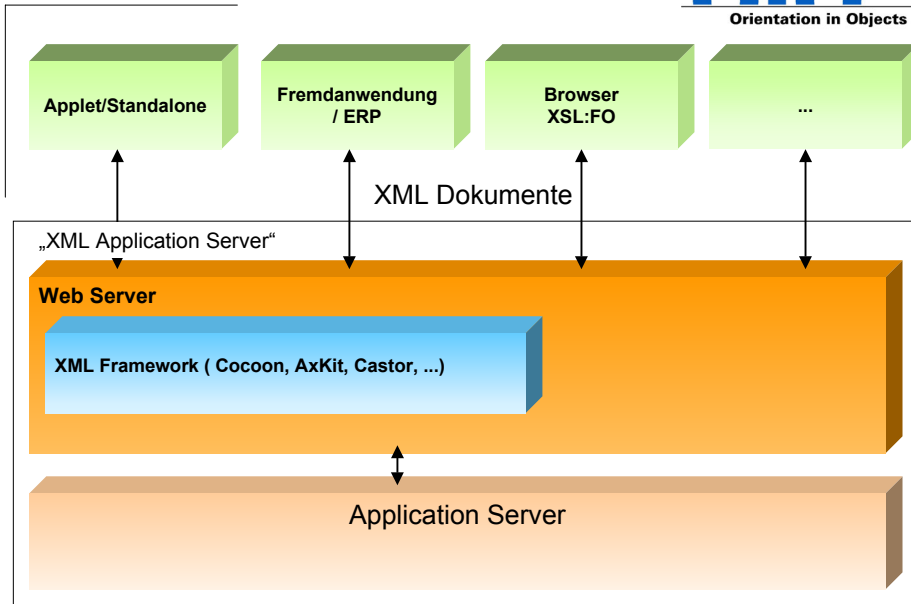
XML Data-Binding



XML Data-Binding

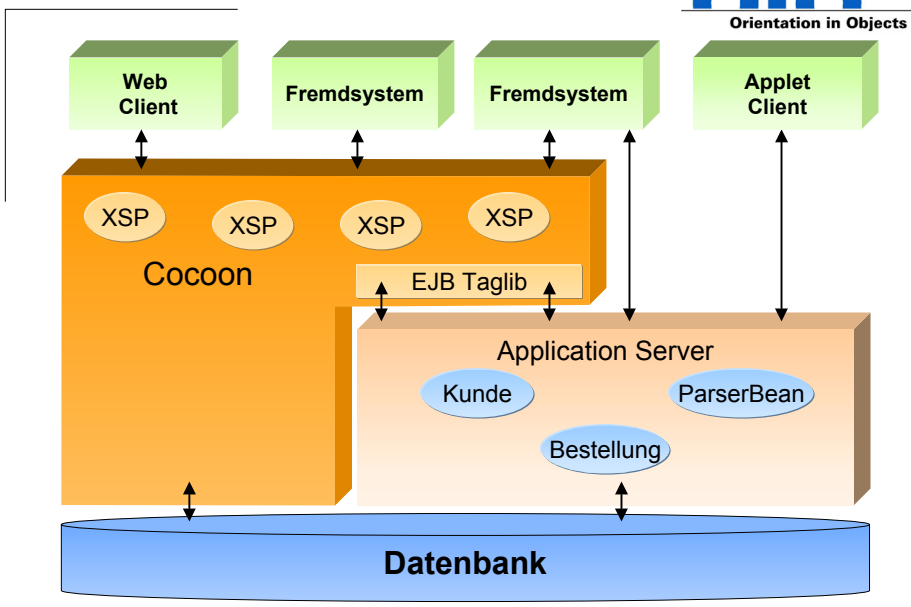


EAI mit „XML Application Server“



© 2000-2001 Orientation in Objects GmbH

Kombinierte Architektur



© 2000-2001 Orientation in Objects GmbH

Aufgabenverteilung



Enterprise JavaBeans

- Geschäftsobjekte
- Transaktionen
- Umwandlung zwischen XML und Objekten

Cocoon

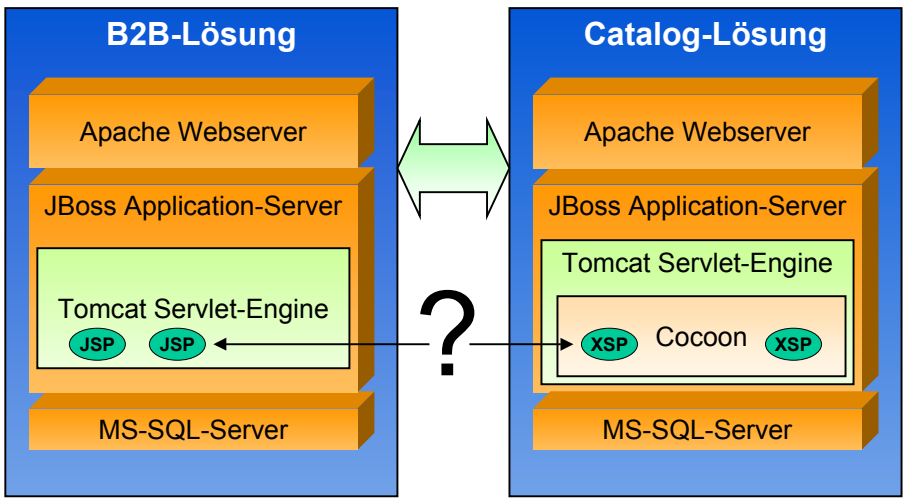
- Präsentation
- Reine Abfragen
- Abfrage von Massendaten (mit SQL)
- XML Dokumente erzeugen (z.B. BMECat Katalog)

Inhalt

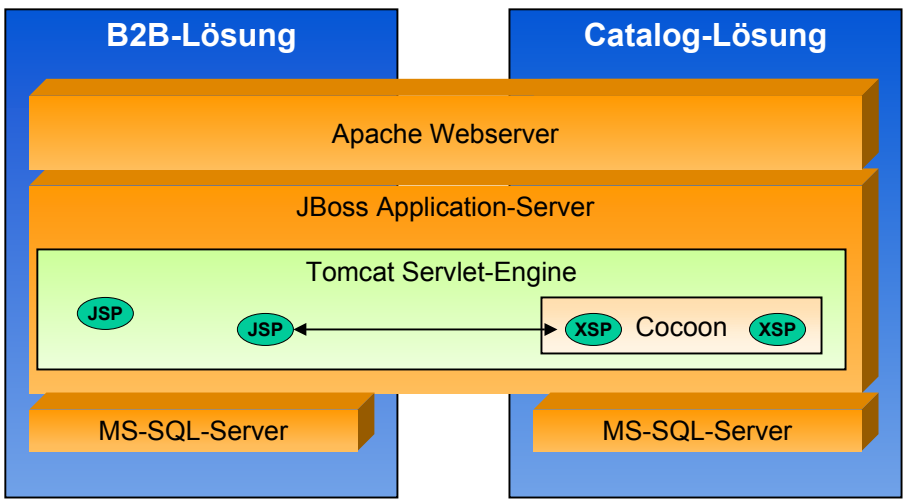


- 3 Schichten Architektur
- XML based Webpublishing
- Web Publishing Frameworks
- Pipeline Model
- Dynamischer Inhalt
- XML Anwendungsarchitekturen
- **Integration einer XSP-Anwendung in ein B2B-System**

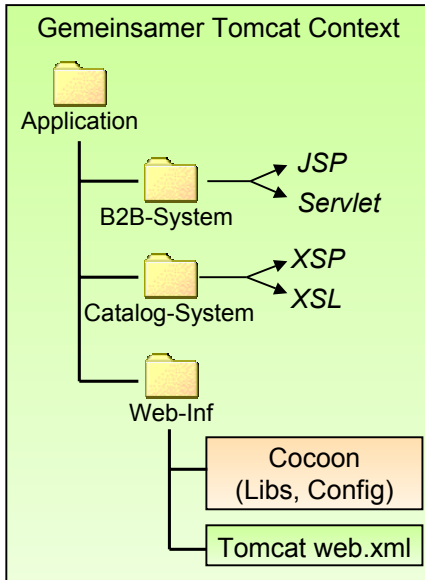
Integration JSP / XSP: Problem



Integration JSP / XSP: Ziel



Integration JSP / XSP: Lösung

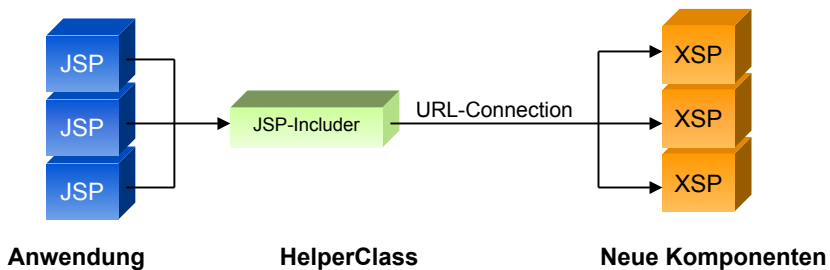


- Gemeinsames Verzeichnis erstellen
- Cocoon in Web-Inf installieren
- web.xml in Web-Inf anpassen
- server.xml von Tomcat anpassen

Migration JSP / XSP: Lösung

- Inhalt von XSP mit HelperClass direkt in JSP einlesen
- Relative Links lassen sich durch gemeinsamen Context leicht übersetzen

```
<% JspIncluder.includeAll("/cat/suche.xsp",out); %>
```



Literatur und Quellen

- Specification JSP 1.1
<http://java.sun.com/products/jsp/download.html>
- Specification XSL 1.0 (candidate recommendation)
<http://www.w3.org/TR/xsl/>
- Specification XSLT 1.0 (recommendation)
<http://www.w3.org/TR/xslt.html>
- Specification XSLT 1.1 (working draft)
<http://www.w3.org/TR/xslt11/>

Literatur und Quellen

- Apache Software Foundation
<http://xml.apache.org>
- Adding XML Capabilities with Cocoon (Stefano Mazzocchi)
<http://xml.apache.org/cocoon/>
- Lutris open source enterprise
<http://www.lutris.com>
- Enhydra Open Source for E-Business
<http://www.enhydra.org>
- AxKit XML Application Server
<http://www.axkit.org>



**Vielen Dank für die
Aufmerksamkeit!**

Thomas Bayer
bayer@oio.de